

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL
DE HUAMANGA**

FACULTAD DE CIENCIAS AGRARIAS

ESCUELA PROFESIONAL DE INGENIERÍA AGRÍCOLA



TESIS:

**Desarrollo de un software para el diseño hidráulico de
canales abiertos mediante Matlab R2025b - Huamanga -
Ayacucho - 2025**

**Para optar el título profesional de:
INGENIERA AGRÍCOLA**

**PRESENTADO POR:
Bach. Liliana Katerin VILCA HUARANCCA**

**ASESOR:
Ing. John Samuel CAZORLA ORIHUELA**

AYACUCHO - PERÚ

2026

DEDICATORIA

Dedico este logro a mi familia, por su amor incondicional y su apoyo constante; a mis amigos, por acompañarme en cada paso de este camino; y a mis docentes, por su guía y sabiduría, que han sido fundamentales en mi formación. A todos ustedes, gracias por creer en mí y ayudarme a alcanzar este objetivo.

AGRADECIMIENTO

Agradezco a Dios por su amor incondicional, por iluminar mi camino y por proporcionarme la sabiduría y fortaleza necesarias para lograr mis metas.

A mi alma mater, la Universidad Nacional de San Cristóbal de Huamanga, por brindarme la oportunidad de formar parte de esta institución.

A mi Escuela Profesional de Ingeniería Agrícola y a todos mis estimados docentes por sus valiosos consejos y enseñanzas durante mi formación académica.

Mi sincero agradecimiento a mi asesor de tesis, Ingeniero Agrícola John Samuel Cazorla Orihuela, por su apoyo constante e incondicional en el desarrollo de esta investigación.

A los miembros del jurado de tesis, al Ing. Vance Giorgio Fernández Huamán, y M.Sc. Eleazar Chuchón Angulo, así como al presidente de la Comisión el Ing. Efraín Chuchón Prado, por sus aportaciones y sugerencias que han enriquecido esta investigación.

A mi familia, por su constante apoyo incondicional.

ÍNDICE GENERAL

	Pág.
DEDICATORIA.....	ii
AGRADECIMIENTO	iii
ÍNDICE GENERAL	iv
ÍNDICE TABLAS.....	vii
ÍNDICE FIGURAS.....	viii
RESUMEN	x
ABSTRACT.....	xi
INTRODUCCIÓN	1
CAPÍTULO I	4
MARCO TEÓRICO.....	4
1.1 Antecedentes de la investigación.....	4
1.1.1 Nivel internacional.....	4
1.1.2 Nivel nacional	5
1.1.3 Nivel local	5
1.2 Bases teóricas.....	6
1.2.1 Fundamentos de hidráulica de canales abiertos	6
1.2.2 Clasificación de los canales abiertos	6
1.2.3 Elementos geométricos del flujo	6
1.2.4 Caracterización matemática de las secciones	7
1.2.5 Tipos de flujo en canales abiertos.....	8
1.2.6 Flujo según su variación en el tiempo	8
1.2.7 Flujo según su variación en el espacio.....	9
1.2.8 Flujo según el régimen dinámico (número de Reynolds)	10
1.2.9 Flujo según el régimen hidráulico (número de Froude).....	11
1.2.10 Caracterización geométrica de las secciones.....	12
1.2.11 Fundamentos de flujo uniforme	17
1.2.12 Fundamentos de flujo crítico	18
1.2.13 Métodos numéricos aplicados.....	19
1.2.14 Herramientas computacionales	20
1.3 Definición de términos	21
1.4 Marco conceptual.....	21
CAPÍTULO II.....	22

METODOLOGÍA	22
2.1 Tipo de investigación.....	22
2.2 Nivel de investigación	22
2.3 Diseño de la investigación	22
2.4 Población y muestra.....	22
2.5 Técnicas e instrumentos de recolección de datos	23
2.6 Técnicas de procesamiento y análisis de datos.....	23
2.7 Metodología.....	23
2.7.1 <i>Secuencia para el desarrollo del software Canaly en la versión Matlab R2025b</i>	23
2.7.2 <i>Implementación de algoritmos numéricos</i>	24
2.7.3 <i>Modelado trascendente para la sección circular</i>	28
2.7.4 <i>Geometría diferencial en la sección parabólica</i>	30
2.7.5 <i>Resolución para el tirante crítico (y_c)</i>	32
2.7.6 <i>Métodos y tolerancias computacionales</i>	38
2.7.7 <i>Interfaz del software Canaly con App Designer</i>	38
CAPÍTULO III.....	46
RESULTADOS Y DISCUSIÓN	46
3.1 Resultados.....	46
3.1.1 <i>Sistematizar los modelos matemáticos y ecuaciones</i>	46
3.1.1.1 Sección circular	46
3.1.1.2 Sección rectangular	47
3.1.1.3 Sección trapezoidal	47
3.1.1.4 Sección triangular.....	48
3.1.1.5 Sección parabólica	48
3.1.1.6 Aspectos de la dinámica de fluidos.....	48
3.1.1.7 Energía específica (E)	49
3.1.1.8 Flujo crítico	49
3.1.2 <i>Objetivo 2: Implementación de algoritmos numéricos</i>	50
3.1.2.1 Métodos y tolerancias computacionales	50
3.1.3 <i>Objetivo 3: Diseño de la interfaz gráfica con App Designer</i>	52
3.1.3.1 Ventanas del software.....	53
3.1.4 <i>Objetivo 4: Validación y análisis comparativo frente a HCanales</i>	57
3.1.4.1 Comparación de resultados en cada uno de los procesos.....	58

3.2 Discusiones	70
CONCLUSIONES	73
RECOMENDACIONES.....	75
REFERENCIAS BIBLIOGRÁFICAS	76
ANEXOS	78

ÍNDICE TABLAS

	Pág.
Tabla 1 <i>Resumen de elementos geométricos en canales abiertos</i>	7
Tabla 2 <i>Resumen de métodos y tolerancias computacionales</i>	38
Tabla 3 <i>Resumen de componentes técnicos implementados</i>	45
Tabla 4 <i>Métodos y tolerancias computacionales aplicadas en el software Canaly</i>	51
Tabla 5 <i>Consolidado de comparación decimal entre Canaly (Matlab) con HCanales</i>	70

ÍNDICE FIGURAS

	Pág.
Figura 1 <i>Flujo permanente vs. flujo no permanente</i>	9
Figura 2 <i>Diagramas comparativos de la línea de agua y el fondo del canal</i>	10
Figura 3 <i>Diagramas de visualización de líneas de corriente</i>	11
Figura 4 <i>Diagramas de la curva de energía específica y propagación de ondas</i>	12
Figura 5 <i>Sección Trapezoidal</i>	13
Figura 6 <i>Sección Rectangular</i>	14
Figura 7 <i>Sección Triangular</i>	15
Figura 8 <i>Sección Circular</i>	16
Figura 9 <i>Sección Parabólico</i>	17
Figura 10 <i>Entorno de programación Matlab R2025b</i>	20
Figura 11 <i>Figuras representadas en el programa Canaly</i>	24
Figura 12 <i>Resumen de fórmulas utilizadas en el software Canaly</i>	24
Figura 13 <i>Algoritmo de Brent (fzero) en Matlab</i>	25
Figura 14 <i>Diagrama de flujo de Fzero en las secciones prismáticas</i>	26
Figura 15 <i>Código Matlab del cálculo de la sección Rectangular, trapezoidal, triangular</i>	27
Figura 16 <i>Código Matlab del cálculo de Caudal Rectangular, trapezoidal, triangular</i>	28
Figura 17 <i>Código Matlab del cálculo de sección circular</i>	29
Figura 18 <i>Código Matlab del cálculo de sección circular</i>	30
Figura 19 <i>Código Matlab del cálculo de sección Parabólica</i>	31
Figura 20 <i>Código Matlab del cálculo de sección Parabólico</i>	32
Figura 21 <i>Código Matlab del cálculo de sección Rectangular, trapezoidal, triangular</i> <i>(critico)</i>	33
Figura 22 <i>Código Matlab del cálculo de sección rectangular, trapezoidal y triangular</i> <i>(critico)</i>	34
Figura 23 <i>Código Matlab del cálculo de sección circular (critico)</i>	35
Figura 24 <i>Código Matlab del cálculo de sección parabólico (critico)</i>	35
Figura 25 <i>Código Matlab del cálculo de sección circular (critico)</i>	36
Figura 26 <i>Código Matlab del cálculo de sección Parabólica (critico)</i>	37
Figura 27 <i>Presentación de la interfaz de las opciones principales del programa Canaly</i>	39
Figura 28 <i>Estructura de clase única en Matlab</i>	40

Figura 29	<i>Propiedades Privadas y Públicas en Matlab (App Designer)</i>	40
Figura 30	<i>Interfaz de cálculo de tirante normal del programa Canaly</i>	41
Figura 31	<i>Grafica generada en Canaly</i>	43
Figura 32	<i>Exportación a Excel en Canaly</i>	44
Figura 33	<i>Generación de una tabla de resultados en formato Excel</i>	44
Figura 34	<i>Interfaz gráfica del software Canaly</i>	52
Figura 35	<i>Ventana para el cálculo de parámetros hidráulicos flujo normal</i>	53
Figura 36	<i>Ventana para el cálculo de parámetros hidráulicos flujo crítico</i>	54
Figura 37	<i>Modelo de inserción de datos incorrectos y visualización de advertencia..</i>	55
Figura 38	<i>Modelo del cálculo en el software Canaly y visualización de gráfico</i>	55
Figura 39	<i>Visualización de la ventana conceptos</i>	56
Figura 40	<i>Visualización de la ventana ayuda</i>	56
Figura 41	<i>Conversión de la aplicación a un archivo ejecutable .exe</i>	57
Figura 42	<i>Interfaz del menú principal del software HCanales v3.1</i>	58
Figura 43	<i>Interfaz del menú principal del software Canaly</i>	59
Figura 44	<i>Cálculo de tirante normal de la sección trapezoidal en Canaly</i>	59
Figura 45	<i>Cálculo de tirante normal de la sección trapezoidal en HCanales v3.1</i>	60
Figura 46	<i>Cálculo de tirante normal de la sección Rectangular en Canaly</i>	60
Figura 47	<i>Cálculo de tirante normal de la sección Rectangular en HCanales v3.1</i>	61
Figura 48	<i>Cálculo de tirante normal de la sección Triangular en Canaly</i>	61
Figura 49	<i>Cálculo de tirante normal de la sección triangular en HCanales v3.1</i>	62
Figura 50	<i>Cálculo de tirante normal de la sección parabólico en Canaly</i>	62
Figura 51	<i>Cálculo de tirante normal de la sección Parabólico en HCanales v3.1</i>	63
Figura 52	<i>Cálculo de tirante normal de la sección Circular en Canaly</i>	63
Figura 53	<i>Cálculo de tirante normal de la sección circular en HCanales v3.1</i>	64
Figura 54	<i>Cálculo de tirante crítico de la sección trapezoidal en Canaly</i>	65
Figura 55	<i>Cálculo de tirante critico de la sección trapezoidal en HCanales v3.1</i>	65
Figura 56	<i>Cálculo de tirante critico de la sección Rectangular en Canaly</i>	66
Figura 57	<i>Cálculo de tirante critico de la sección trapezoidal en HCanales v3.1</i>	66
Figura 58	<i>Cálculo de tirante critico de la sección triangular en Canaly</i>	67
Figura 59	<i>Cálculo de tirante critico de la sección triangular en HCanales v3.1</i>	67
Figura 60	<i>Cálculo de tirante critico de la sección parabólico en Canaly</i>	68
Figura 61	<i>Cálculo de tirante critico de la sección parabólico en HCanales v3.1</i>	68
Figura 62	<i>Cálculo de tirante critico de la sección Circular Canaly</i>	69
Figura 63	<i>Cálculo de tirante critico de la sección circular en HCanales v3.1</i>	69

RESUMEN

La investigación llevada a cabo, se planteó como objetivo principal desarrollar un software especializado en el entorno Matlab R2025b para automatizar el cálculo de los parámetros hidráulicos y geométricos en canales abiertos de sección prismática. El diseño hidráulico de canales que, comúnmente, se ha realizado mediante métodos manuales e iterativos que invierten una mayor cantidad de tiempo y son propensos a cometer errores de precisión. Ante esta problemática en busca de una solución, se desarrolló el software denominado Canaly utilizando el entorno App Designer incorporada en Matlab, utilizando algoritmos numéricos basados en métodos iterativos abiertos avanzados como, la Secante y el Algoritmo de Brent, el procedimiento metodológico adoptado en esta investigación se caracteriza por ser aplicada y tecnológica, con un enfoque cuantitativo y una investigación no experimental, orientado a la implementación de algoritmos numéricos para la solución de problemas prácticos. El software calcula el tirante normal, tirante crítico, área, perímetro mojado, radio hidráulico y energía específica para cinco secciones fundamentales: rectangular, trapezoidal, triangular, parabólica y circular. Con el fin de corroborar la confiabilidad del software desarrollado, se aplicaron pruebas de validación mediante la inserción de casos, comparando los resultados obtenidos por el software Canaly frente al software HCanales v3.1. Los resultados de la validación demostraron una fiabilidad del 99.9%, con errores relativos despreciables (inferiores al 0.001), confirmando que la programación mediante Matlab no solo reduce el tiempo de cálculo, sino que posee una precisión técnica confiable, moderna para los profesionales y estudiantes de ingeniería en la región de Ayacucho.

Palabras clave: Matlab, Diseño hidráulico, Canales, Algoritmos numéricos, HCanales, Canaly.

ABSTRACT

The main objective of the research carried out was to develop specialized software in the Matlab R2025b environment to automate the calculation of hydraulic and geometric parameters in open channels of prismatic section. The hydraulic design of the channels has usually been carried out using manual and iterative methods that invest more time and are prone to precision errors. Faced with this problem in search of a solution, the software called Canaly was developed using the App Designer environment incorporated in Matlab, using numerical algorithms based on advanced open iterative methods such as the Secante and the Brent Algorithm; The methodological procedure adopted in this research is characterized by being applied and technological, with a quantitative approach and non-experimental research. oriented to the implementation of numerical algorithms for the resolution of practical problems. The software calculates the normal staybar, the critical direction bar, the area, the wet perimeter, the hydraulic radius, and the specific energy for five fundamental sections: rectangular, trapezoidal, triangular, parabolic, and circular. To corroborate the reliability of the developed software, validation tests were applied by inserting cases, comparing the results obtained by the Canaly software with the HCanales v3.1 software. The validation results demonstrated a reliability of 99.9%, with negligible relative errors (less than 0.001), confirming that programming with Matlab not only reduces calculation time, but also has modern and reliable technical accuracy for engineering professionals and students in the Ayacucho region.

Keywords: Matlab, Hydraulic Design, Canals, Numerical Algorithms, HCanales, Canaly.

INTRODUCCIÓN

En el campo de la ingeniería, la planificación y administración de obras hidráulicas representan pilares fundamentales para el desarrollo rural y la seguridad hídrica. Dentro de este contexto, los canales abiertos son los elementos de conducción más utilizados para la conducción de agua para riego, drenaje y abastecimiento. La determinación precisa de sus parámetros hidráulicos como el tirante normal, crítico y la energía específica, el tipo de régimen de flujo es una tarea que constituye un aspecto fundamental, ya que determina no sólo la eficiencia operativa de la obra, sino también su estabilidad estructural y la optimización de los costos de inversión.

Históricamente, el cálculo de estas variables se ha realizado mediante métodos manuales que dependen de la aplicación de fórmulas empíricas, como la ecuación de Manning, y el uso de herramientas analógicas como tablas y nomogramas. Si bien estos métodos han sido la base de la hidráulica clásica, presentan limitaciones críticas en la práctica profesional contemporánea. El análisis de las ecuaciones que describen al uniforme y el flujo crítico a menudo requiere procesos iterativos complejos, dado que el tirante se encuentra implícito en la geometría de la sección. Esto conlleva un riesgo elevado de error humano, imprecisiones por redondeo y una inversión de tiempo considerable que ralentiza el flujo de trabajo en las etapas de diseño.

La presente investigación, titulada "Desarrollo de un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b – Huamanga – Ayacucho - 2025", surge como una solución tecnológica ante esta problemática. El núcleo del trabajo consiste en el desarrollo del software Canaly, una herramienta integral programada en el entorno App Designer de Matlab, este programa ha sido diseñado para automatizar y sistematizar el cálculo hidráulico, permitiendo al proyectista obtener resultados de alta fidelidad mediante la implementación de algoritmos numéricos de vanguardia.

A nivel regional, en Ayacucho, la topografía accidentada y la necesidad de modernizar la agricultura exigen que los ingenieros cuenten con herramientas precisas y adaptadas a la realidad técnica. Sin embargo, todavía existe una marcada dependencia de métodos convencionales o de software generalista que no siempre cumple con los requerimientos específicos del diseño agrícola local. Ante este escenario, el problema de investigación se define en torno a la necesidad de evaluar cómo la automatización de modelos matemáticos en un entorno de desarrollo robusto puede elevar los estándares de precisión y eficiencia en comparación con el cálculo manual tradicional.

La investigación se justifica desde una perspectiva técnica y científica por la implementación de métodos de resolución de raíces, tales como los algoritmos de la Secante y el método de Brent. Estos métodos permiten alcanzar precisiones computacionales con una tolerancia de hasta 1.0×10^{-8} metros, garantizando la convergencia incluso en secciones transversales complejas como las parabólicas o circulares. Desde un punto de vista profesional, el software Canaly proporciona una interfaz gráfica intuitiva que facilita la visualización dinámica de la sección transversal, permitiendo al usuario realizar análisis de sensibilidad de manera inmediata al modificar variables como el caudal, rugosidad del material y la pendiente.

La hipótesis central del estudio sostiene que el desarrollo de un software especializado en Matlab optimiza significativamente la precisión técnica y reduce los tiempos de procesamiento de datos, eliminando la incertidumbre propia del cálculo manual. Para demostrarlo, se establecieron objetivos claros: sistematizar los modelos hidráulicos de flujo para secciones prismáticas (rectangular, trapezoidal, triangular, parabólica y circular), diseñar una interfaz de usuario funcional y moderna, y validar los resultados del programa mediante una comparación estadística con el software patrón HCanales v3.1.

Finalmente, la metodología adoptada en esta investigación fue de carácter aplicado y tecnológico, siguiendo un enfoque cuantitativo. El proceso abarcó desde la sistematización de las bases teóricas de la hidráulica de canales hasta la codificación de la lógica de control y la fase de validación rigurosa. Con este aporte, se busca entregar a la comunidad de la Universidad Nacional de San Cristóbal de Huamanga y a los

profesionales del sector una herramienta potente y confiable que contribuya al avance de la infraestructura hidráulica en el Perú.

Objetivo general

Desarrollar un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b, que permita optimizar los procesos de cálculo y mejorar la eficiencia y precisión en el diseño hidráulico en Huamanga - Ayacucho - 2025.

Objetivos específicos

1. Sistematizar los modelos matemáticos de flujo uniforme (Manning) y flujo crítico para secciones prismáticas (rectangular, trapezoidal, triangular, parabólico y circulares), definiendo las ecuaciones constitutivas del software.
2. Implementar algoritmos numéricos de raíces en el lenguaje Matlab R2025b, estableciendo criterios de convergencia y tolerancia para la resolución de ecuaciones no lineales implícitas.
3. Diseñar una interfaz gráfica de usuario (GUI) moderna mediante App Designer, diferenciando de la arquitectura obsoleta GUIDE, para garantizar la interactividad y visualización dinámica de la sección transversal.
4. Validar la fiabilidad del software desarrollado mediante un análisis comparativo de resultados (tirante normal y crítico) frente al software patrón HCanales, evaluando el error relativo porcentual.

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes de la investigación

La revisión de la literatura especializada evidencia una tendencia global hacia la migración de cálculos hidráulicos manuales a entornos de desarrollo automatizados, debido a la necesidad de minimizar el error humano y optimizar tiempos en proyectos de ingeniería agrícola y civil.

1.1.1 Nivel internacional

En el ámbito latinoamericano, la automatización del diseño hidráulico ha sido objeto de diversos estudios recientes. (Castro y Monta, 2024), en la Universidad Politécnica Salesiana (Ecuador), realizaron su investigación donde desarrollaron un aplicativo en Matlab con interfaz gráfica (GUI) con el propósito de calcular tirantes y diseño de canales. En su estudio, titulado "Desarrollo de una aplicación en Matlab para diseño hidráulico", las autoras validaron la herramienta resolviendo ejercicios de textos clásicos de hidráulica, demostrando que la automatización reduce el tiempo de resolución en un 80% comparado con el cálculo manual y elimina los errores de redondeo en las iteraciones de flujo uniforme. Esta investigación es un antecedente directo para la presente tesis, pues confirma la viabilidad de Matlab como entorno de desarrollo para herramientas didácticas y profesionales.

Por otro lado, Vera (2023), presentó su tesis en la Universidad de Manabí sobre el "Desarrollo de un software de hidráulica para calcular canales abiertos mediante Python". El autor realizó una validación cruzada comparando los resultados de su algoritmo "Pydraulic" frente al software comercial HCanales v3.1. Los resultados mostraron una discrepancia menor al 0.01% en el cálculo de tirantes normales y críticos para secciones prismáticas, concluyendo que el desarrollo de software propio permite una mayor

flexibilidad para adaptar los coeficientes de rugosidad a condiciones locales, algo que los programas cerrados no siempre permiten con facilidad.

Asimismo, Ladino et al. (2021), publicaron en la revista Tecnura el artículo “Desarrollo de una aplicación móvil para el aprendizaje interactivo en problemas de ingeniería civil: aplicación a la hidráulica de canales abiertos”, los investigadores implementaron el método numérico de Newton-Raphson para resolver la ecuación de Manning en dispositivos móviles. Su conclusión principal fue que las herramientas digitales interactivas no sólo mejoran la precisión del cálculo, sino que facilitan la comprensión del comportamiento del flujo (subcrítico vs. supercrítico) al permitir visualizar cambios en tiempo real, un enfoque que este proyecto busca replicar mediante el uso de Matlab R2025b.

1.1.2 Nivel nacional

En el contexto peruano, la ingeniería hidráulica también ha incursionado en el desarrollo de software para optimizar diseños en zonas altoandinas. Davila (2023), en su investigación de grado en la Universidad Tecnológica de los Andes (Apurímac), que lleva por título “Diseño de canal de riego con máxima eficiencia hidráulica mediante software Hec- Ras...”, realizó un análisis comparativo entre métodos tradicionales y modelación computacional en el sector Mariño. Aunque su enfoque se centró en HEC-RAS, sus antecedentes resaltan que la programación de scripts personalizados para el cálculo de "Máxima Eficiencia Hidráulica" es superior al uso de tablas estáticas, especialmente en topografías accidentadas similares a las de Ayacucho.

1.1.3 Nivel local

Vílchez (2015), sentó las bases del uso de este lenguaje en su estudio realizado en la Universidad Nacional de San Cristóbal de Huamanga, sobre la “Modelación hidrodinámica de sistemas de drenaje pluvial con el método de elementos finitos y Matlab”, este estudio fue enfocado en el drenaje urbano en la selva cusqueña (Kimbiri), demostró la robustez de Matlab para manejar grandes volúmenes de datos hidrológicos y su capacidad para generar reportes técnicos precisos, justificando así la elección de esta plataforma para el presente proyecto de tesis.

1.2 Bases teóricas

1.2.1 Fundamentos de hidráulica de canales abiertos

La hidráulica de canales abiertos constituye una rama fundamental en la mecánica de fluidos encargada del análisis de los movimientos y las características del agua en conductos donde la superficie del flujo permanece en contacto con la atmósfera. A diferencia de los flujos en tuberías presurizadas, el movimiento del agua en canales abiertos está influenciado principalmente por la gravedad, la geometría del canal y la resistencia al flujo.

Según Chow (1959), un canal abierto se define como un medio de transporte por donde circula el líquido con una superficie libre, siendo el peso del agua el principal agente impulsor del movimiento y se encuentra presente comúnmente en drenaje, canales de riego y ríos, desempeñando un papel crucial en la ingeniería agrícola.

Asimismo, Robert (1985), señala que el estudio de estos sistemas requiere considerar simultáneamente las variables como la velocidad, pendiente, tirante, y la rugosidad del material, debido a su influencia directa en la capacidad de transporte del flujo.

En este contexto, el análisis hidráulico en canales nos permite diseñar infraestructuras eficientes para el transporte de agua, optimizando el uso del recurso hídrico y reduciendo pérdidas por fricción o desbordamiento.

1.2.2 Clasificación de los canales abiertos

Según Chow (1959), los canales abiertos se clasifican en prismáticos y no prismáticos. Los canales prismáticos mantienen secciones transversales uniforme y pendiente constante. La dinámica de flujo en canales se caracteriza por poseer la superficie expuesta a la presión atmosférica, diferenciándose de los flujos en conductos cerrados por la ausencia de presión hidrostática completa en la superficie superior.

1.2.3 Elementos geométricos del flujo

Chapra y Canale (2015), destacan la importancia de los parámetros en los elementos geométricos ya que son funciones de la forma de secciones transversales y del

tirante de agua en la modelación numérica para ingeniería. Las variables fundamentales se describen en la siguiente Tabla.

Tabla 1

Resumen de elementos geométricos en canales abiertos

Elemento	Símbolo	Descripción	Fórmula general
Tirante	y	Profundidad vertical del flujo desde la solera.	Variable
Área Hidráulica	A	Superficie transversal ocupada por el fluido.	$f(y)$
Perímetro Mojado	P	Longitud de contacto entre el agua y las paredes.	$f(y)$
Radio Hidráulico	R	Relación entre el área y el perímetro mojado.	A/P
Espejo de Agua	T	Ancho de la sección en la superficie libre.	dA/dy
Profundidad Hidráulica	D	Relación entre el área y el espejo de agua.	A/T

Nota. Adaptado de *Hidráulica de canales abiertos*, por Ven Te Chow, 1994, McGraw – Hill.

1.2.4 Caracterización matemática de las secciones

Las ecuaciones geométricas son la base del código fuente. Sotelo (2002), enfatiza su aplicación en canales artificiales para garantizar la eficiencia del transporte hídrico.

- Tirante de Agua (y): Es la profundidad máxima del fluido en la sección analizada, definida como la distancia vertical medida desde el punto más bajo de la plantilla (solera) hasta la superficie libre del agua. Es la variable independiente fundamental en los algoritmos de resolución del software.
- Profundidad de Flujo (d): Es la profundidad del flujo medida en dirección perpendicular a la solera del canal. Si el fondo tiene una inclinación θ , la relación es:

$$d = y \cdot \cos \theta \dots\dots\dots (Eq. 1)$$

En canales de baja pendiente ($S < 0.10$), se asume $y \approx d$.

- Área Mojada (A): Superficie transversal ocupada efectivamente por el fluido. Es el parámetro base para la ecuación de continuidad:

$$Q = V \cdot A \dots\dots\dots (Eq. 2)$$

En el software, el área se calcula integrando la forma geométrica respecto al tirante y.

- **Perímetro Mojado (P):** Longitud de la línea de contacto entre el fluido y las paredes sólidas del canal (no incluye el espejo de agua). Es la zona de fricción donde se disipa la energía por rugosidad.
- **Radio Hidráulico (R):** Cociente:

$$R = A / P \dots\dots\dots(Eq. 3)$$

Representa la eficiencia del canal; físicamente, indica la relación entre la fuerza impulsora (peso del agua) y la resistencia al flujo (fricción). Un canal óptimo minimiza P para maximizar R.

- **Espejo de Agua (T):** Ancho horizontal de la sección en la superficie libre. Es un factor determinante para calcular la celeridad de las ondas y el Número de Froude (Fr).
- **Profundidad Hidráulica (D):** Definida como:

$$D = A / T \dots\dots\dots(Eq. 4)$$

Representa el tirante medio de la sección y se utiliza en el análisis de estabilidad de flujo y cálculo de energía específica.

- **Ancho de Solera (b):** Dimensión horizontal del fondo de la sección (en canales trapezoidales y rectangulares). Es un dato de entrada crítico en el software.
- **Talud (z):** Relación de la proyección horizontal a la vertical de las paredes laterales (z:1). Su valor es dictado por la estabilidad del suelo para evitar el colapso de las paredes.

1.2.5 Tipos de flujo en canales abiertos

El flujo en canales abiertos presenta diferentes comportamientos que pueden clasificarse en función de su variación temporal, espacial y dinámica. Esta clasificación es esencial para el análisis hidráulico y el diseño de obras, ya que permite predecir el comportamiento del agua bajo distintas condiciones.

1.2.6 Flujo según su variación en el tiempo

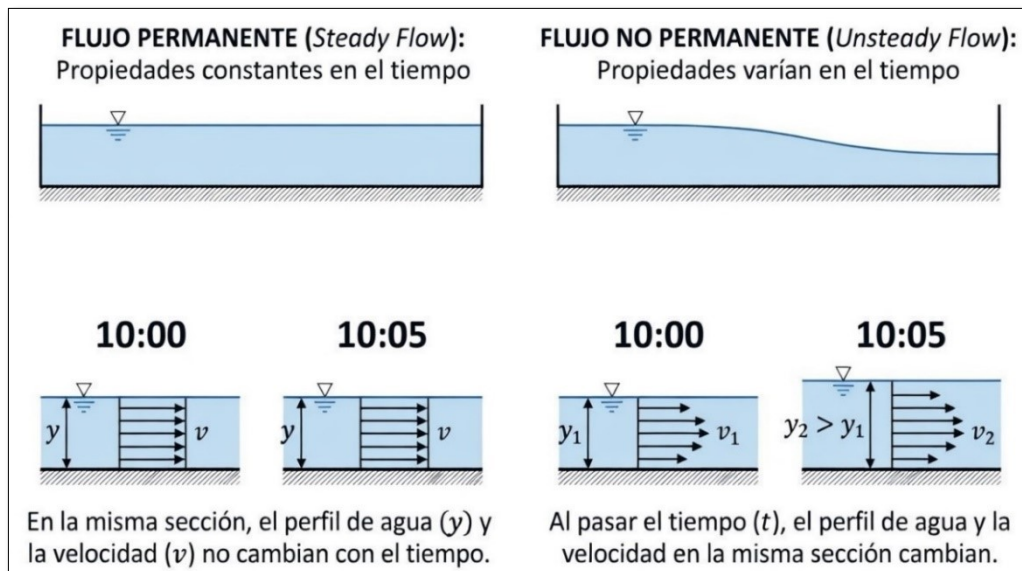
El flujo puede clasificarse como permanente o no permanente dependiendo de si sus propiedades varían con el tiempo.

El flujo permanente se caracteriza porque variables como el caudal, la velocidad y el tirante permanecen constantes en el tiempo. Este tipo de flujo es común en canales de riego operando bajo condiciones controladas.

Por otro lado, el flujo no permanente ocurre cuando estas variables cambian con el tiempo, como sucede durante eventos de crecida o descarga variable. Según Ven Te Chow (1959), este tipo de flujo requiere análisis más complejos debido a su naturaleza transitoria.

Figura 1

Flujo permanente vs. flujo no permanente



1.2.7 Flujo según su variación en el espacio

Las condiciones en un flujo pueden experimentar cambios a lo largo del recorrido del canal.

Un flujo uniforme se presenta cuando el tirante, la velocidad y la sección transversal permanecen constantes en toda la longitud del canal, a este tipo de flujo se le considera como una idealización utilizada en el diseño hidráulico y es la base de la ecuación de Manning (Chow, 1959).

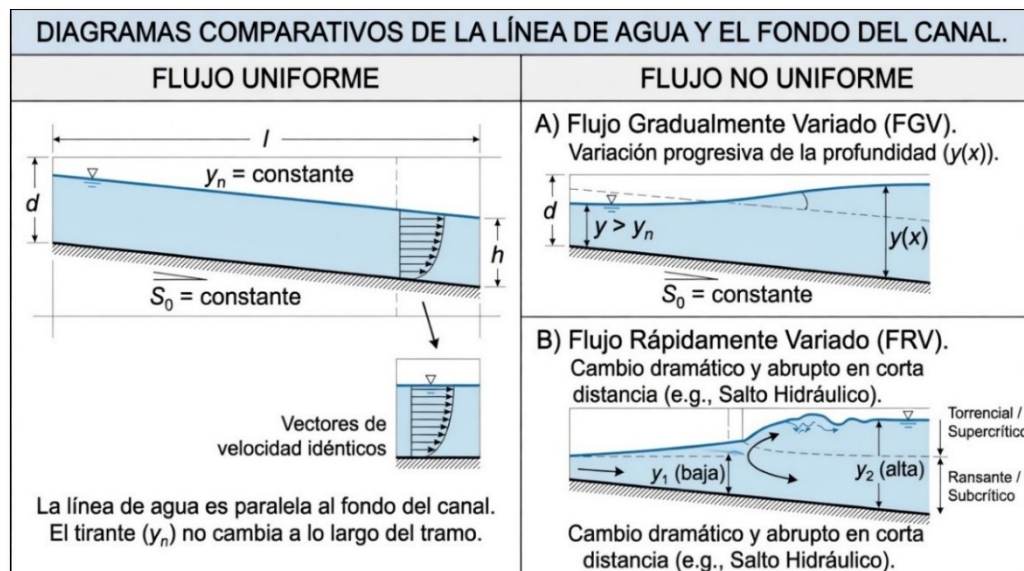
En contraste, el flujo no uniforme ocurre cuando estas variables cambian a lo largo del canal. Este puede subdividirse en:

- Flujo gradualmente variado (FGV): donde los cambios son suaves y progresivos.
- Flujo rápidamente variado (FRV): donde los cambios son bruscos, como en el salto hidráulico.

Según Chin (2013), el análisis del flujo no uniforme es fundamental para comprender fenómenos como remansos y transiciones de energía en estructuras hidráulicas.

Figura 2

Diagramas comparativos de la línea de agua y el fondo del canal



1.2.8 Flujo según el régimen dinámico (número de Reynolds)

El Número de Reynolds clasifica a un flujo. La clasificación de un flujo puede clasificarse mediante la proporción de la fuerza inercial y la viscosa (White, 2011).

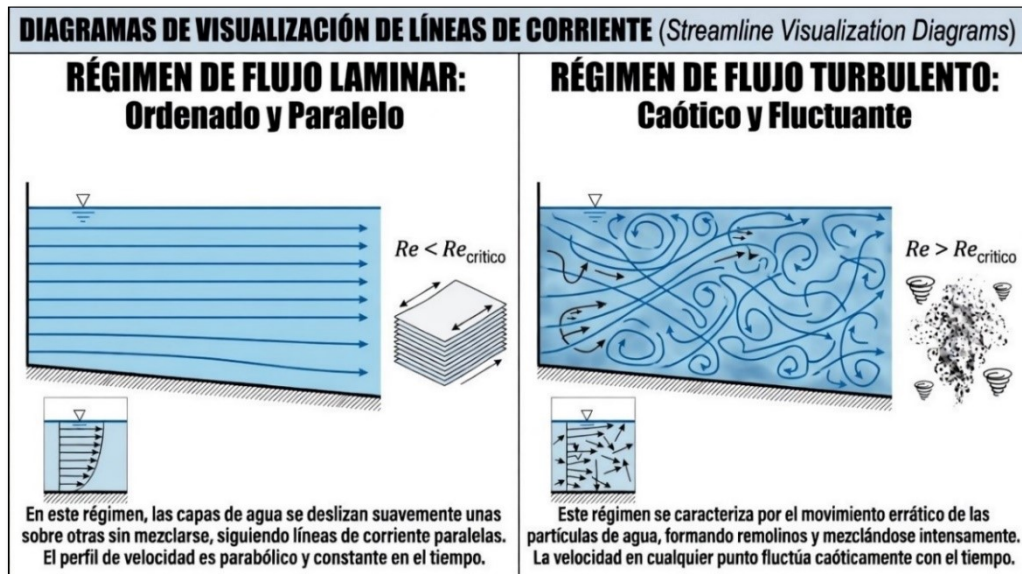
De acuerdo con White (2011):

- Un flujo laminar se caracteriza por tener un movimiento suave y uniforme, en donde sus partículas se desplazan en trayectorias ordenadas y paralelas.
- Un flujo turbulento tiene movimiento irregular y desordenada, donde predominan las fuerzas inerciales sobre las viscosas.

En canales abiertos, el flujo es predominantemente turbulento debido a las altas velocidades y dimensiones involucradas.

Figura 3

Diagramas de visualización de líneas de corriente



1.2.9 Flujo según el régimen hidráulico (número de Froude)

El número de Froude es considerado como un parámetro adimensional y relaciona fuerza inercial con la gravitacional y permite clasificar el flujo según su energía (Chow, 1959).

$$Fr = \frac{V}{\sqrt{g \cdot D_H}} \dots \dots \dots (Eq. 5)$$

Donde:

V: Velocidad del flujo.

g: Aceleración de la gravedad (9.81 m/s²).

D_H: Profundidad hidráulica.

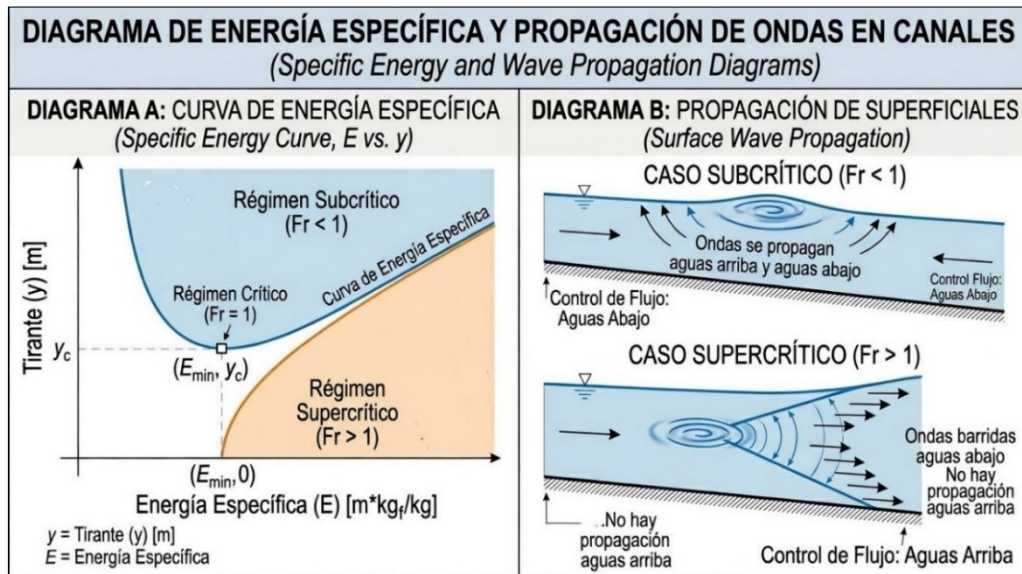
Según Chow (1959), este parámetro define tres regímenes:

- Flujo subcrítico ($Fr < 1$): caracterizado por bajas velocidades y control en dirección del flujo.
- Flujo crítico: representa el estado de mínima energía específica.
- Flujo supercrítico ($Fr > 1$): caracterizado por altas velocidades y control aguas arriba.

Esta clasificación es esencial para el diseño hidráulico, especialmente en la determinación de estructuras de disipación de energía.

Figura 4

Diagramas de la curva de energía específica y propagación de ondas



1.2.10 Caracterización geométrica de las secciones

Según Cuevas (2016), Matlab destaca como un entorno de cálculo matemático caracterizado por su accesibilidad y capacidad para el desarrollo de interfaces gráficas multidimensionales. Su versatilidad lo posiciona como un recurso fundamental tanto en la industria como en la academia, facilitando desde el aprendizaje del álgebra matricial básica hasta la resolución de problemas complejos en áreas como la geofísica, la inteligencia artificial, el procesamiento de señales y la modelación de sistemas dinámicos.

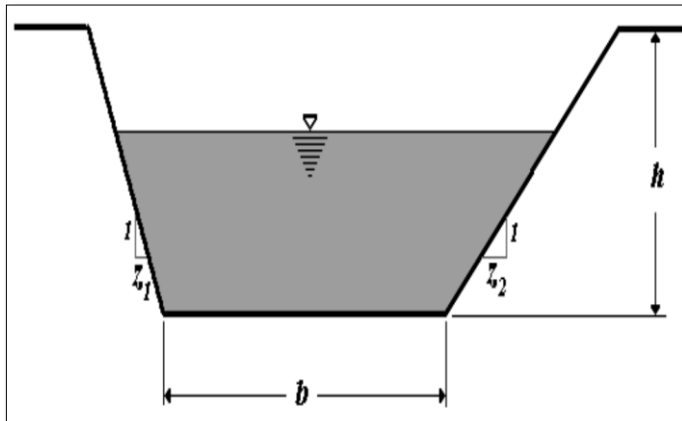
El programa implementa cinco modelos geométricos.

a. Sección trapezoidal

La sección rectangular es una opción práctica debido a su firmeza y a la posibilidad de funcionar sin recubrimientos bajo ciertas condiciones. Sin embargo, para optar por una sección trapezoidal, es indispensable realizar una evaluación económica previa que permita comparar ambas alternativas y seleccionar la más eficiente en términos de inversión y ejecución, (Cadavid, 2006, pág. 38).

Figura 5

Sección Trapezoidal



Nota. Adaptado de *Hidráulica de canales fundamentales*, por Juan Cadavid R., 2006, Fondo Editorial Universidad EAFIT.

Según Cadavid (2006), los canales trapezoidales poseen los mismos parámetros básicos que una sección rectangular, no obstante, se distinguen por una geometría en la que los taludes dependen del parámetro z y la profundidad h , dichos taludes se diseñan con el propósito de asegurar la estabilidad geotécnica de la sección transversal, y además cuando la pendiente del canal pueda originar velocidades elevadas, es recomendable proteger las paredes mediante algún tipo de revestimiento.

- Área (A):

$$A = (b + zy)y \dots \dots \dots (Eq. 6)$$

- Perímetro (P):

$$P = b + 2y\sqrt{1 + z^2} \dots \dots \dots (Eq. 7)$$

- Espejo de Agua (T):

$$T = b + 2zy \dots \dots \dots (Eq. 8)$$

- Radio Hidráulico (R):

$$R = \frac{(b+zy)y}{b+2y\sqrt{1+z^2}} \dots \dots \dots (Eq. 9)$$

Variables: b = ancho de base; z = valor del talud.

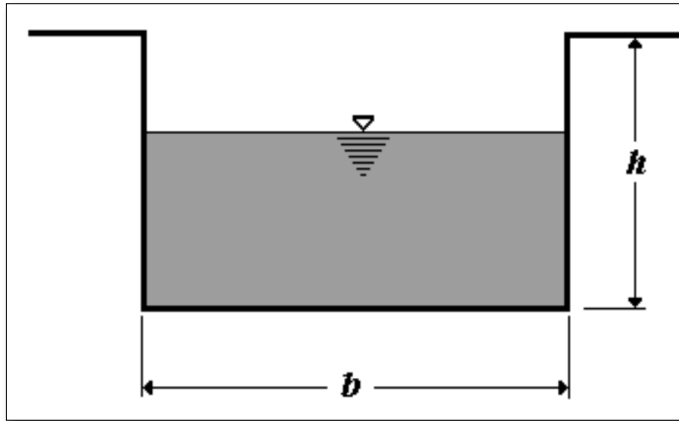
b. Sección rectangular

Como señala Cadavid (2006), la sección de forma rectangular es una de las opciones más elementales y prácticas para el transporte de fluidos, definida por su base b y su altura h . Su aplicación frecuente en la hidráulica se debe a esta simplicidad, aunque,

desde una perspectiva geotécnica, su ejecución por excavación suele demandar el empleo de revestimientos de concreto para asegurar la estabilidad de sus paredes verticales.

Figura 6

Sección Rectangular



Nota. Adaptado de *Hidráulica de canales fundamentales*, por Juan Cadavid R., 2006, Fondo Editorial Universidad EAFIT.

Como señala Cadavid (2006), la morfología rectangular también se aplica a excavaciones naturales o artesanales sin revestir, tales como las acequias rurales. Su estabilidad está garantizada tanto por la baja profundidad de la obra como por la calidad del terreno, especialmente cuando se trata de excavaciones en roca que soportan cortes verticales estables.

- Área (A):

$$A = by \dots\dots\dots(Eq. 10)$$
- Perímetro (P):

$$P = b + 2y \dots\dots\dots(Eq. 11)$$
- Espejo de Agua (T):

$$T = b \dots\dots\dots(Eq. 12)$$
- Radio Hidráulico (R):

$$R = \frac{by}{b+2y} \dots\dots\dots(Eq. 13)$$

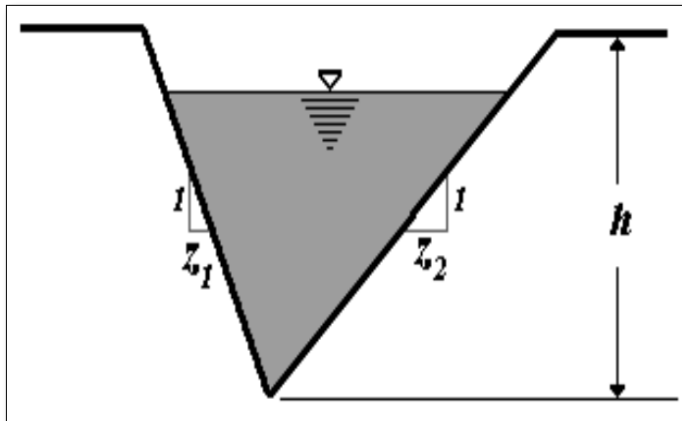
c. Sección triangular

De acuerdo con Cadavid (2006), los canales triangulares son estructuras diseñadas principalmente para el control de escorrentía y drenaje. Se caracterizan por una

configuración donde los taludes están en función del parámetro z y la profundidad h . Es frecuente encontrarlos integrados a la infraestructura vial o en la zona de descarga de presas, aprovechando la misma pendiente del terreno o del terraplén para conformar uno de sus lados.

Figura 7

Sección Triangular



Nota. Adaptado de *Hidráulica de canales fundamentos*, por Juan Cadavid R., 2006, Fondo Editorial Universidad EAFIT.

- Área (A):

$$A = zy^2 \dots\dots\dots (Eq. 14)$$
- Perímetro (P):

$$P = 2y\sqrt{1 + z^2} \dots\dots\dots (Eq. 15)$$
- Espejo de Agua (T):

$$T = 2zy \dots\dots\dots (Eq. 16)$$
- Radio Hidráulico (R):

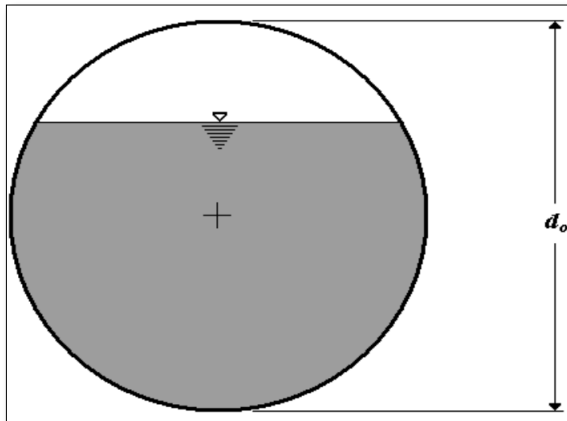
$$R = \frac{zy}{2\sqrt{1+z^2}} \dots\dots\dots (Eq. 17)$$

d. Sección circular (flujo a sección parcial)

De acuerdo con Cadavid (2006), el parámetro distintivo de los canales circulares es su diámetro. Esta sección está diseñada principalmente para trabajar de manera cerrada, cumpliendo funciones de colector en redes de saneamiento y drenaje pluvial. Aunque es inusual verla operar como canal a cielo abierto, posee utilidad técnica en esta modalidad cuando se requiere realizar el aforo de una corriente.

Figura 8

Sección Circular



Nota. Adaptado de Hidráulica de canales fundamentales, por Juan Cadavid R., 2006, Fondo Editorial Universidad EAFIT.

- Ángulo Central (θ):

$$\theta = 2 \arccos \left(1 - \frac{2y}{d_o} \right) \text{ (en radianes)} \dots\dots\dots (Eq. 18)$$
- Área (A):

$$A = \frac{1}{8} (\theta - \sin \theta) d_o^2 \dots\dots\dots (Eq. 19)$$
- Perímetro (P):

$$P = \frac{1}{2} \theta d_o \dots\dots\dots (Eq. 20)$$
- Espejo de Agua (T):

$$T = d_o \sin(\theta/2) \dots\dots\dots (Eq. 21)$$
- Radio Hidráulico (R):

$$R = \frac{1}{4} d_o \left(1 - \frac{\sin \theta}{\theta} \right) \dots\dots\dots (Eq. 22)$$

Donde: d_o es el diámetro interno de la tubería.

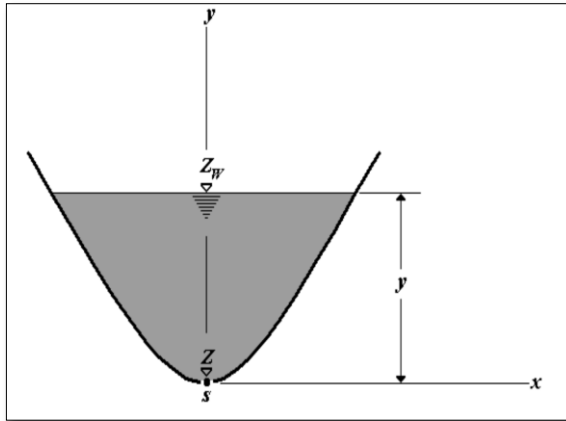
e. Sección parabólica

De acuerdo con Cadavid (2006), para la representación espacial de un canal, se emplea un sistema de coordenadas cuyo punto de origen se localiza en el fondo de la sección transversal. Según este esquema, el plano de la sección está delimitado por las variables x e y , dispuestas de forma ortogonal entre sí. La dimensión longitudinal se integra mediante la coordenada s , la cual se orienta paralelamente a la dirección del flujo

sobre el fondo, permitiendo así una descripción tridimensional del comportamiento hidráulico.

Figura 9

Sección Parabólico



Nota. Adaptado de *Hidráulica de canales fundamentales*, por Juan Cadavid R., 2006, Fondo Editorial Universidad EAFIT.

- Área (A):

$$A = \frac{2}{3}Ty \dots\dots\dots(Eq. 23)$$

- Perímetro (P):

$$P = T + \frac{8y^2}{3T} \text{ (Aproximación para canales anchos)} \dots\dots\dots(Eq. 24)$$

$$P = \frac{T}{2} \sqrt{1 + \left(\frac{4y}{T}\right)^2} + \frac{T^2}{8y} \ln \left(\frac{4y}{T} + \sqrt{1 + \left(\frac{4y}{T}\right)^2} \right) \dots\dots\dots(Eq. 25)$$

Para el cálculo del perímetro se utilizó la Eq. 25 (Villón, 2007).

- Espejo de Agua (T):

$$T = \frac{3A}{2y} \dots\dots\dots(Eq. 26)$$

o definido por la geometría $y=kx^2$.

1.2.11 Fundamentos de flujo uniforme

El flujo uniforme constituye el estado hidráulico idealizado en el que la profundidad del agua, conocida como tirante normal (y_n), permanece constante a lo largo de todo el desarrollo del conducto. En esta condición, el espejo de agua es exactamente paralela al fondo del conducto, donde la pendiente de la línea de energía (S_f) con la pendiente geométrica del fondo del canal (S_0). Según la teoría clásica de Chow (1959),

este régimen de equilibrio dinámico se alcanza cuando las fuerzas gravitatorias componentes del peso del fluido, que impulsan el flujo hacia aguas abajo, se equilibran con las fuerzas de resistencia por fricción generadas en la interfaz entre el fluido y el perímetro mojado. La ecuación que gobierna este equilibrio es la fórmula de Manning la cual es una expresión empírica fundamental en la ingeniería hidráulica:

$$Q = \frac{1}{n} A R^{2/3} S_0^{1/2} \dots\dots\dots (Eq. 27)$$

Donde:

Q: representa el caudal volumétrico ($\frac{m^3}{s}$).

n: coeficiente de rugosidad de Manning, varía según el material del canal.

A: es el área de sección transversal (m^2).

$R_h = \frac{A}{P}$: radio hidráulico (m), definido como la división entre el área y el perímetro mojado.

S_0 : pendiente longitudinal del fondo (adimensional).

La determinación de la profundidad normal (y_n) se obtiene mediante la resolución implícita de la ecuación anterior para la variable y, considerando que tanto el área (A) como el perímetro mojado (P) son funciones geométricas no lineales del tirante. Debido a que este cálculo suele ser trascendente en secciones complejas como las circulares o parabólicas, el software requiere el empleo de métodos numéricos avanzados para el cálculo de raíces.

1.2.12 Fundamentos de flujo crítico

El flujo crítico representa la condición hidráulica de transición entre el régimen subcrítico (flujo tranquilo con velocidades bajas) y el régimen supercrítico (flujo rápido con velocidades altas). Se define formalmente como la condición de flujo donde la energía específica (E) alcanza su valor mínimo absoluto para un caudal constante dado. La energía específica, referida al fondo del canal como plano de referencia, se expresa mediante la ecuación de Bernoulli simplificada (Chow, 1959):

$$E = y + \frac{Q^2}{2g A^2} \dots\dots\dots (Eq. 28)$$

Donde g es la aceleración gravitatoria estándar ($9.81 \frac{m}{s^2}$). La condición matemática para hallar el mínimo de esta función se obtiene derivando la energía con respecto al tirante ($\frac{dE}{dy}$) e igualando el resultado a cero:

$$\frac{dE}{dy} = 1 - \frac{Q^2 T}{g A^3} = 0 \dots\dots\dots (Eq. 29)$$

Esta deducción conduce directamente a la definición del número de Froude (Fr), el cual debe ser igual a la unidad en condiciones críticas:

$$Fr = \frac{V}{\sqrt{g \frac{A}{T}}} = 1 \dots\dots\dots (Eq. 30)$$

En esta expresión, $T = \frac{dA}{dy}$ representa el ancho del espejo de agua en la superficie libre y $V = \frac{Q}{A}$ se considera como velocidad media del flujo. Mientras que en secciones rectangulares el tirante crítico puede despejarse algebraicamente ($y_c = \sqrt[3]{Q^2/(gb^2)}$), en secciones de geometría variable como trapezoidales o circulares, el software debe recurrir a la resolución numérica de la ecuación de energía mínima para garantizar precisión (Chow, 1959).

1.2.13 Métodos numéricos aplicados

Debido a la naturaleza no lineal y a menudo trascendente de las ecuaciones de flujo uniforme y crítico, el software Canaly integra una arquitectura de cálculo basada en algoritmos numéricos de alta robustez y convergencia garantizada:

- **Algoritmo de Brent (fzero):** Es una combinación híbrida que utiliza métodos como la bisección, secante e interpolación cuadrática inversa. Es seleccionado para el cálculo del tirante normal debido a su fiabilidad absoluta; no requiere el cálculo de derivadas y garantiza la convergencia siempre que exista un cambio de signo en el intervalo inicial definido. Se opera con una tolerancia de error de 1.0×10^{-6} m.
- **Método de la secante:** Se aplica específicamente en el análisis de secciones parabólicas para resolver la integral de línea del perímetro mojado y en escenarios donde la función de área no es monótona. Utiliza dos semillas iniciales para aproximar la raíz sin la necesidad de evaluar la derivada de la función.

Búsqueda con restricción geométrica: Para evitar soluciones físicamente imposibles o zonas de inestabilidad matemática (como el "lleno relativo" en tuberías circulares), el software limita los intervalos de búsqueda entre el 0.1% y el 95% de la capacidad máxima de la sección.

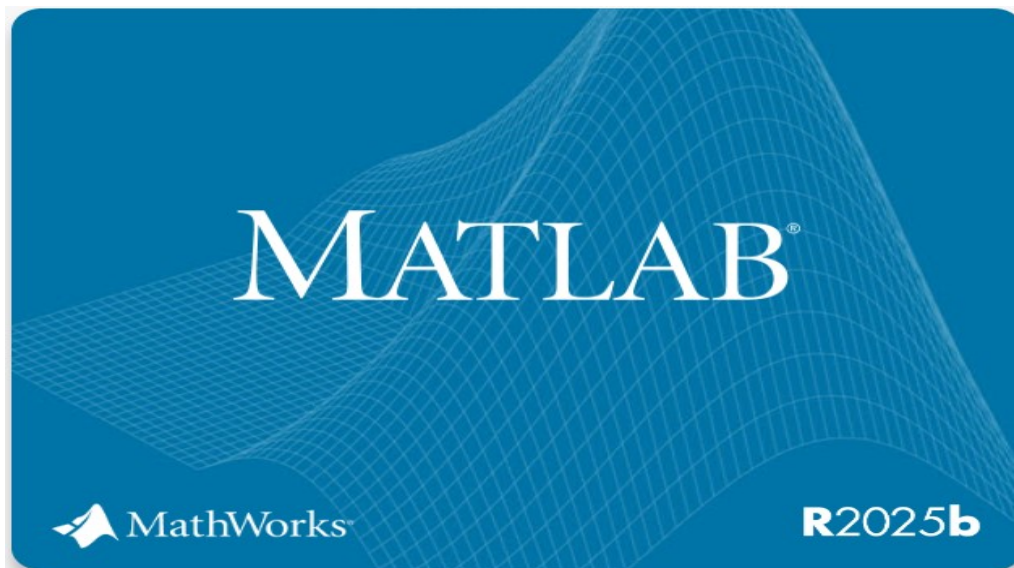
1.2.14 Herramientas computacionales

El ecosistema tecnológico de Canaly se fundamenta en Matlab R2025b, una plataforma líder en computación técnica. La implementación aprovecha las siguientes capacidades avanzadas:

- **App Designer:** Se utilizó este entorno de desarrollo orientado a objetos para diseñar una interfaz gráfica (GUI) profesional. Esto permite encapsular la lógica del negocio hidráulico dentro de clases y métodos que responden a eventos del usuario en tiempo real.
- **Bibliotecas de visualización:** Se emplean funciones de bajo nivel como `patch` para el dibujo de las secciones transversales, `text` para el acotado dinámico y `axis equal` para asegurar que las proporciones geométricas mostradas en pantalla correspondan a la realidad técnica (estilo CAD).
- **Validación de datos y Robustez:** El código incorpora bloques de manejo de excepciones `try-catch` y componentes `NumericEditField` que validan rangos de entrada (ej. pendientes negativas o rugosidades fuera de límites físicos), evitando errores de ejecución y garantizando la integridad de los resultados.
- **Interoperabilidad:** El motor incluye capacidades de exportación vía COM para generar reportes automatizados en Microsoft Excel, facilitando la documentación de proyectos de ingeniería.

Figura 10

Entorno de programación Matlab R2025b



Nota. Adaptado de MathWorks, 2025.

1.3 Definición de términos

Canal abierto: Conducto o cauce en el que el fluido fluye con una superficie libre en contacto directo con la atmósfera, manteniendo una presión superficial igual a la presión atmosférica local.

- Tirante (y): Parámetro que indica la profundidad vertical del agua, medida desde la base hasta la superficie libre (m).
- Caudal (Q): Magnitud física que expresa el volumen de agua que atraviesa en un tiempo determinado (m^3/s).
- Rugosidad de Manning (n): Coeficiente empírico que cuantifica la resistencia al flujo debida a la irregularidad de las paredes y el lecho del canal.
- Flujo uniforme: Régimen donde el tirante, el área y la velocidad permanecen invariantes a lo largo del eje longitudinal del canal ($S_0=S_f$).
- Número de Froude (Fr): Parámetro adimensional, razón de las fuerzas inerciales con las gravitacionales. Es el indicador definitivo del régimen de flujo (supercrítico, crítico y subcrítico).

1.4 Marco conceptual

El núcleo de Canaly reside en la transferencia sistemática del conocimiento teórico clásico hacia un entorno de ingeniería digital de alta fidelidad. Esta integración permite mejorar la precisión respecto a los métodos tradicionales de cálculo manual o tablas, los cuales suelen ser propensos a errores de interpolación. La relación bidireccional entre la hidráulica y el software permite:

- Automatización Compleja: Resolución instantánea de ecuaciones implícitas que de otro modo requerirían múltiples iteraciones manuales.
- Validación Gráfica: Proporcionar al ingeniero una representación visual inmediata de la sección diseñada, mejorando la comprensión del comportamiento del flujo.
- Fiabilidad Técnica: La aplicación de estándares modernos de desarrollo de software garantiza que los resultados sean consistentes, auditables y listos para su aplicación en expedientes técnicos profesionales.

CAPÍTULO II

METODOLOGÍA

2.1 Tipo de investigación

La investigación es de tipo Aplicada y Tecnológica. Es aplicada porque busca utilizar las teorías de la mecánica de fluidos (ecuaciones de Manning y Froude) para resolver problemas prácticos en el diseño de infraestructura hídrica. Es tecnológica debido a que el resultado final es un artefacto funcional (software desarrollado en Matlab R2025b) diseñado para optimizar procesos de cálculo de ingeniería.

2.2 Nivel de investigación

El nivel es Descriptivo-Correlacional. Es descriptivo porque detalla y caracteriza las propiedades geométricas y cinemáticas de los fluidos en diversas secciones. Asimismo, posee un componente correlacional/comparativo al evaluar la precisión de los resultados del software frente a soluciones analíticas y herramientas validadas.

2.3 Diseño de la investigación

Se emplea un diseño No experimental, descriptivo y transversal.

- No experimental: No existe manipulación deliberada de variables físicas; se trabaja con modelos matemáticos y simulaciones numéricas.
- Transversal: La recolección de datos y la validación de algoritmos que se realizan en un único periodo de tiempo definido.

2.4 Población y muestra

- Población: Está constituida por el universo de secciones prismáticas (canales abiertos y conductos cerrados) que operan bajo condiciones de flujo libre y están regidos por las leyes de la hidráulica de canales.

- Muestra: Se utiliza muestreo no probabilístico intencional, integrada por 5 geometrías fundamentales (rectangular, trapezoidal, triangular, parabólica y circular).

-

2.5 Técnicas e instrumentos de recolección de datos

- Técnicas: Se aplicará el Análisis Documental para la recopilación de fundamentos teóricos y la Observación Sistemática para el registro de los datos procesados por el software.
- Instrumentos: Se utilizarán fichas de registro de datos hidráulicos, protocolos de prueba de software (debugging) y matrices de tabulación para organizar los resultados obtenidos en Matlab.

2.6 Técnicas de procesamiento y análisis de datos

El procesamiento se realizará mediante el Método Comparativo y el análisis estadístico de errores.

- Validación Estadística: Se compararán los resultados del software desarrollado frente a la herramienta HCanales v3.1, evaluando los márgenes de error absoluto y relativo.
- Análisis Numérico: Se empleará los métodos numéricos para garantizar la convergencia de los resultados en el cálculo de tirantes críticos y normales, asegurando la confiabilidad técnica de la herramienta.

2.7 Metodología

2.7.1 Secuencia para el desarrollo del software Canaly en la versión Matlab R2025b

Se realizó la estructuración de la matemática de flujo uniforme y crítico, no solo consistió en recopilar fórmulas, sino en establecer una jerarquía de variables que permitiera al software resolver la geometría de forma precisa para las diferentes secciones como la rectangular, trapezoidal, triangular, circular y parabólica.

Figura 11

Figuras representadas en el programa Canaly

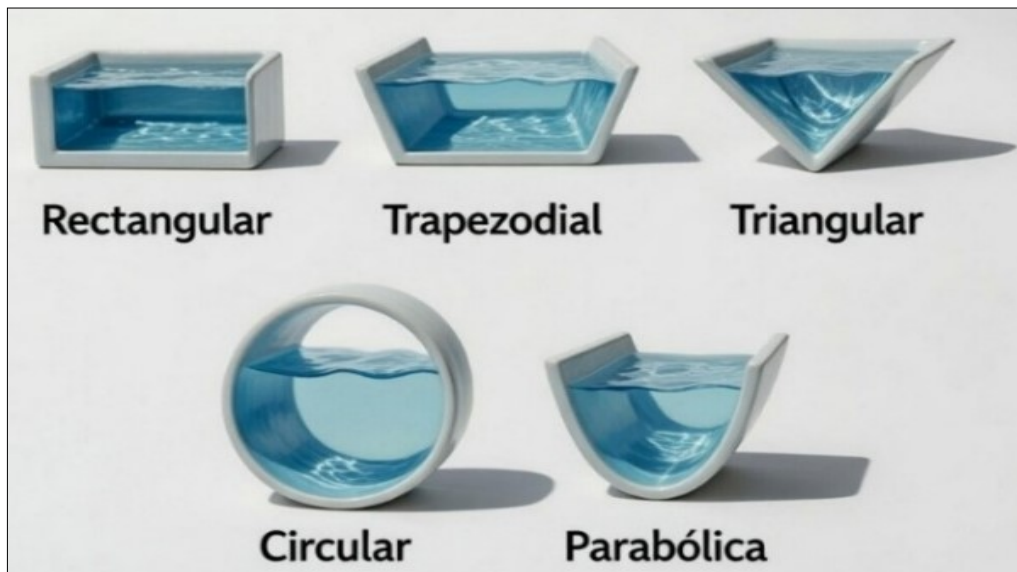


Figura 12

Resumen de fórmulas utilizadas en el software Canaly

Sección	Area hidráulica (A)	Perímetro mojado (P)	Radio hidráulico (R)	Espejo de agua (T)
 Rectangular	by	$b+2y$	$\frac{by}{b+2y}$	b
 Trapezoidal	$(b+zy)y$	$b+2y\sqrt{1+z^2}$	$\frac{(b+zy)y}{b+2y\sqrt{1+z^2}}$	$b+2zy$
 Triangular	zy^2	$2y\sqrt{1+z^2}$	$\frac{zy}{2\sqrt{1+z^2}}$	$2zy$
 Circular	$\frac{(\theta - \text{sen}\theta)D^2}{8}$	$\frac{\theta D}{2}$	$\left(1 - \frac{\text{sen}\theta}{\theta}\right) \frac{D}{4}$	$\left(\sin \frac{\theta}{2}\right) D$ or $2\sqrt{y(D-y)}$
 Parabólica	$2/3Ty$	$T + \frac{8y^2}{3T}$	$\frac{2T^2y}{3T^2 + 8y^2}$	$\frac{3A}{2y}$

Se tomaron en cuenta los aspectos y ecuaciones de la dinámica de fluidos.

2.7.2 Implementación de algoritmos numéricos

La arquitectura computacional del sistema ha sido diseñada bajo un paradigma de alta fidelidad numérica para la resolución de las ecuaciones fundamentales de la

hidráulica de canales abiertos. Este objetivo se centra en la transposición de modelos matemáticos implícitos en la ejecución de algoritmos, garantizando estabilidad, precisión y eficiencia en el procesamiento de datos.

El cálculo del tirante normal constituye el núcleo del diseño de canales en régimen uniforme. Se basa en la ecuación de Manning, que establece un equilibrio dinámico entre la componente del peso del fluido y las fuerzas de resistencia al flujo generadas por la rugosidad del contorno.

Para las secciones trapezoidales, rectangulares y triangulares, se ha seleccionado el Método de Brent. Este algoritmo emplea lo métodos de la bisección y los métodos abiertos como es la secante y la interpolación inversa cuadrática; el método calcula las raíces de una función, que no requiere el cálculo de la derivada analítica de la función (evitando la complejidad de derivar el Radio Hidráulico respecto al tirante), mediante un proceso, el software evalúa la función de error $f(y)=\Phi(y)-Q_{\text{objetivo}}$, donde $\Phi(y)$ es la capacidad de transporte. El algoritmo detecta el cambio de signo en el intervalo $[0.0001,100]$ y converge con una precisión de hasta 10^{-6} metros.

Figura 13

Algoritmo de Brent (fzero) en Matlab

```

function CalcularButtonPushed(app, event)
% --- 1. LLENADO AUTOMÁTICO DE EDITOR (Si los campos están vacíos) ---
if app.CaudalEditField.Value == 0
    app.CaudalEditField.Value = 2.50;
    app.AnchodeSolerEditField.Value = 1.00;
    app.TaludEditField.Value = 1.40;
    app.RugosidadEditField.Value = 0.015;
    app.PendientesEditField.Value = 0.001;
end

% --- 2. CAPTURA DE VARIABLES ---
Q = app.CaudalEditField.Value;
b = app.AnchodeSolerEditField.Value;
z = app.TaludEditField.Value;
n = app.RugosidadEditField.Value;
S = app.PendientesEditField.Value;

% --- 3. RESOLUCIÓN DE LA ECUACIÓN DE MANNING ---
% PASO A: Definición de la Función Objetivo (Ecuación Transcendente)
% Aquí se crea una función anónima f(y). Como no podemos despejar "y" algebraicamente,
% definimos f(y) = 0 para encontrar el valor donde la igualdad de Manning se cumple.
f_manning = @(y) (((b + z*y)^5/3) / ((b + 2*y*sqrt(1 + z^2))^(2/3)) - (Q * n / sqrt(S));

try
% PASO B: APLICACIÓN DEL MÉTODO NUMÉRICO (fzero)
% Aquí está el método numérico propiamente dicho.
% MATLAB utiliza el algoritmo de Brent (una combinación de bisección, secante e interpolación).
% Se le da un intervalo [0.0001, 100] donde se busca el cambio de signo de la función.
yn = fzero(f_manning, [0.0001, 100]);

if yn > 0

% --- 4. CÁLCULOS HIDRÁULICOS ---
g = 9.81;
A = (b + z*yn) * yn; % Área
P = b + 2 * yn * sqrt(1 + z^2); % Perímetro
R = A / P; % Radio Hidráulico
T = b + 2 * z * yn; % Espesor de agua
v = Q / A; % Velocidad
D = A / T; % Profundidad Hidráulica
Fr = v / sqrt(g * D); % Froude
E = yn + (v^2 / (2 * g)); % Energía

% --- 5. ASIGNACIÓN A COMPONENTES ---
% Se han asignado las variables a los nombres de los componentes de UI
app.TiranteNormalEditField.Value = round(yn, 4); % Tirante Normal (yn)
app.AreaHidraulicaEditField.Value = round(A, 4); % Área Hidráulica (A)

```

Figura 14

Diagrama de flujo de Fzero en las secciones prismáticas

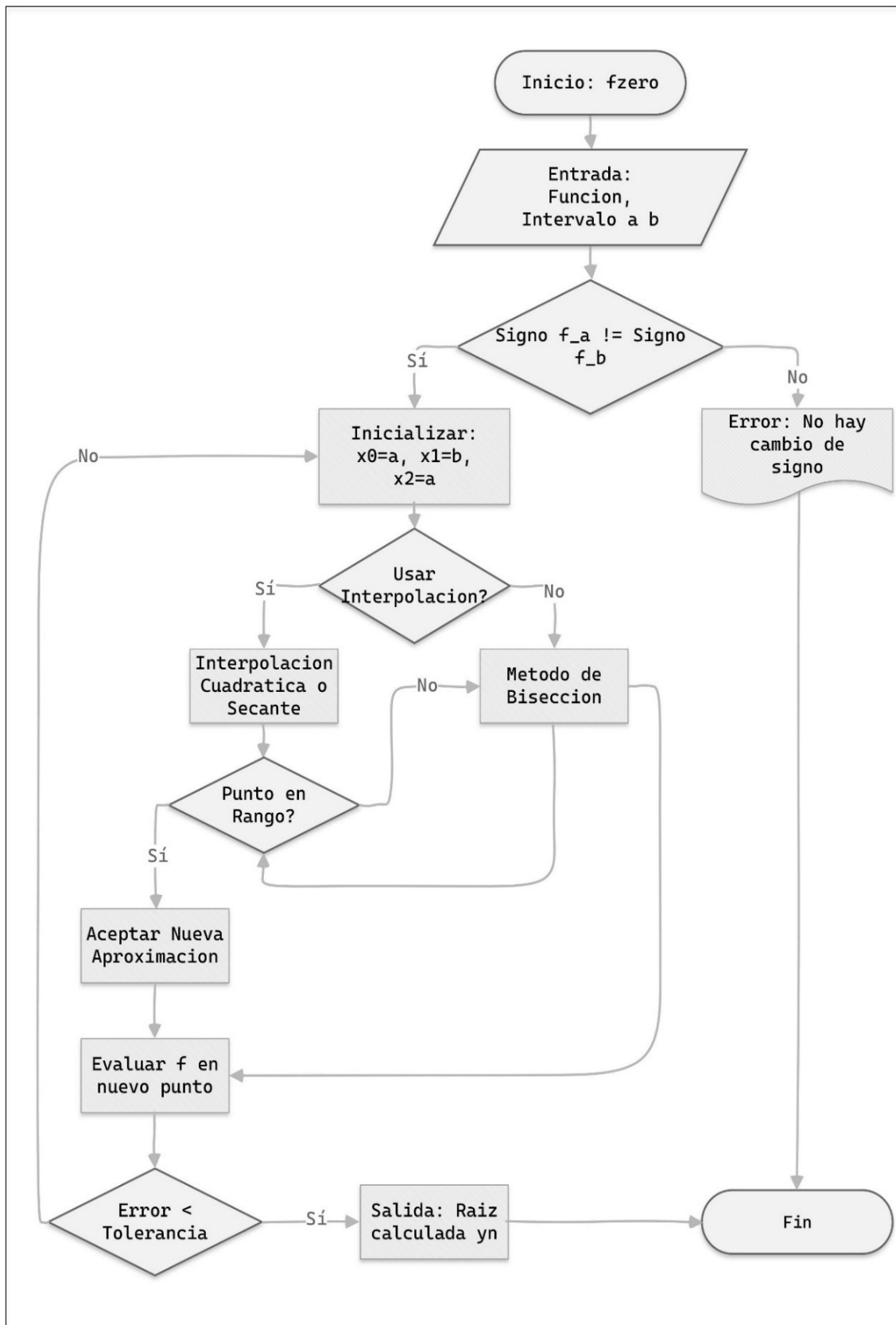


Figura 15

Código Matlab del cálculo de la sección Rectangular, trapezoidal, triangular

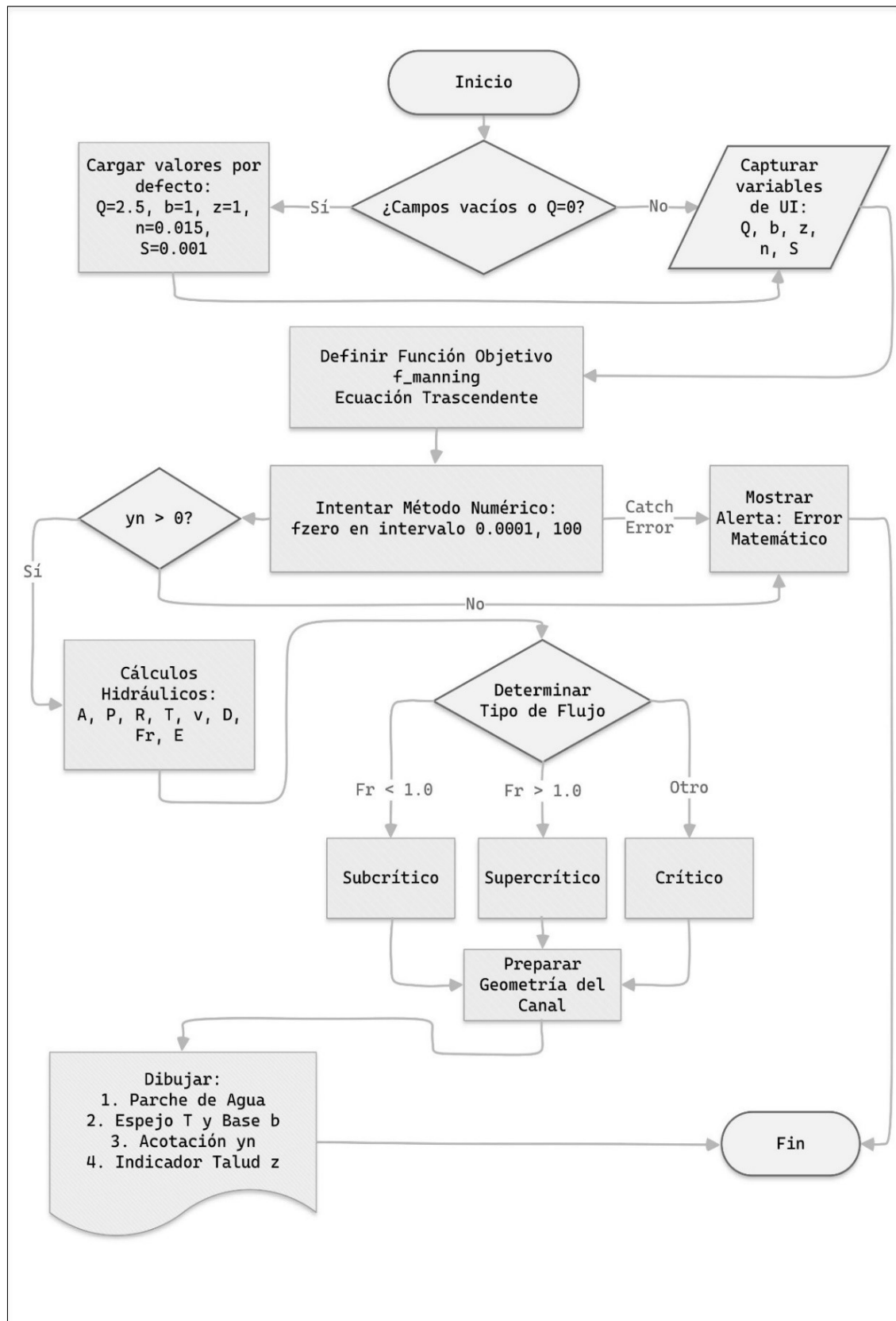
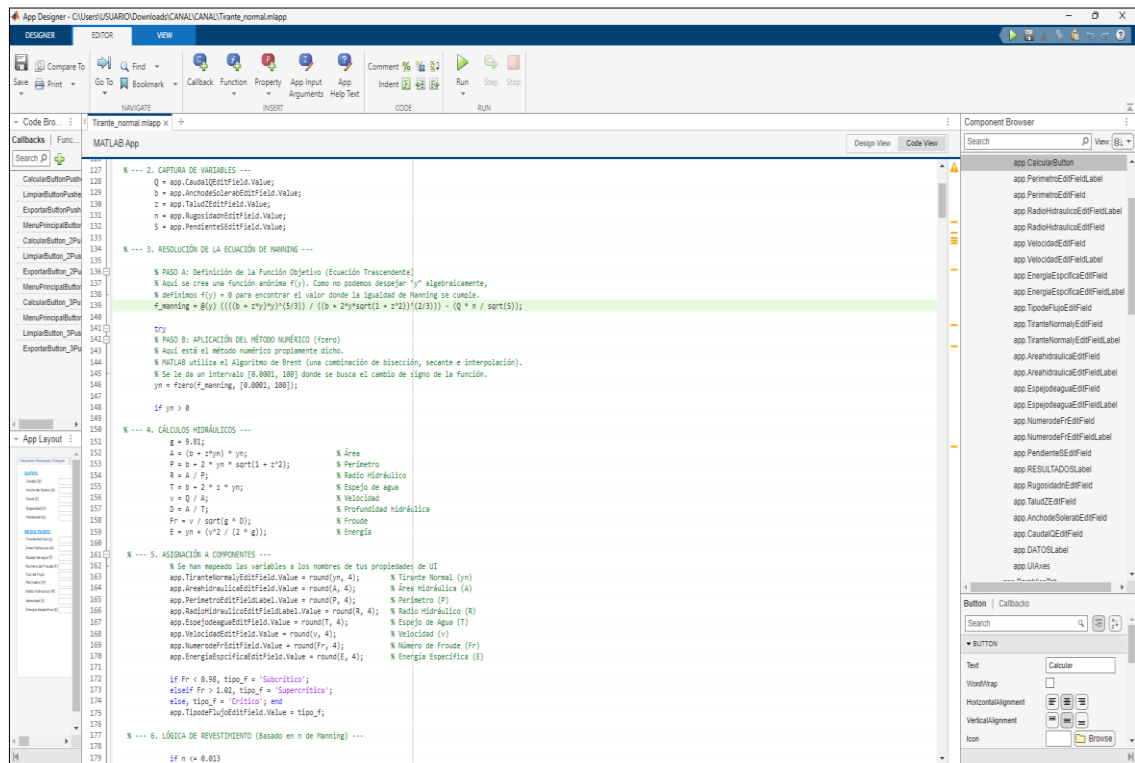


Figura 16

Código Matlab del cálculo de Caudal Rectangular, trapezoidal, triangular



```

% --- 2. CAPTURA DE VARIABLES ---
127 Q = app.CaudalTextField.Value;
128 b = app.AnchoDeCanalTextField.Value;
129 z = app.TaludTextField.Value;
130 n = app.RugosidadTextField.Value;
131 S = app.PendienteTextField.Value;
132
133 % --- 3. RESOLUCIÓN DE LA ECUACIÓN DE MANNING ---
134
135 % PASO A: Definición de la Función Objetivo (Ecuación Trascendente)
136 % Aquí se crea una función anónima f(y). Como no podemos despejar "y" algebraicamente,
137 % derivamos f(y) = 0 para encontrar el valor donde la igualdad de Manning se cumple.
138 f_manning = @(y) (((b + z*y)/y)^(5/3)) / ((b + z*y)*sqrt(1 + z^2)) - (Q*n) / sqrt(S);
139
140 try
141     % PASO B: APLICACIÓN DEL MÉTODO NUMÉRICO (fzero)
142     % Aquí está el método numérico propiamente dicho.
143     % MATLAB utiliza el Algoritmo de Brent (una combinación de bisección, secante e interpolación).
144     % Se le da un intervalo [0.0001, 100] donde se busca el cambio de signo de la función.
145     ym = fzero(f_manning, [0.0001, 100]);
146
147     if ym > 0
148
149 % --- 4. CÁLCULOS HIDRÁULICOS ---
150
151 g = 9.81; % Área
152 A = (b + z*ym) * ym; % Área
153 P = b + 2 * ym * sqrt(1 + z^2); % Perímetro
154 R = A / P; % Radio Hidráulico
155 T = b + 2 * z * ym; % Espejo de agua
156 v = Q / A; % Velocidad
157 D = A / T; % Profundidad hidráulica
158 Fr = v / sqrt(g * D); % Froude
159 E = ym + (v^2 / (2 * g)); % Energía
160
161 % --- 5. ASIGNACIÓN A COMPONENTES ---
162 % Se han mapeado las variables a los nombres de los componentes de UI
163 app.TiranteNormalTextField.Value = round(ym, 4); % Tirante Normal (ym)
164 app.AreaHidraulicaTextField.Value = round(A, 4); % Área Hidráulica (A)
165 app.PerimetroTextField.Value = round(P, 4); % Perímetro (P)
166 app.RadioHidraulicoTextField.Value = round(R, 4); % Radio Hidráulico (R)
167 app.EspejoAguaTextField.Value = round(T, 4); % Espejo de Agua (T)
168 app.VelocidadTextField.Value = round(v, 4); % Velocidad (v)
169 app.NumeroFroudeTextField.Value = round(Fr, 4); % Número de Froude (Fr)
170 app.EnergiaEspecificTextField.Value = round(E, 4); % Energía Específica (E)
171
172 if Fr < 0.98, tipo_f = 'subcrítico';
173 elseif Fr > 1.2, tipo_f = 'superocrítico';
174 else, tipo_f = 'crítico'; end
175 app.TipoDeflujoTextField.Value = tipo_f;
176
177 % --- 6. LÓGICA DE REVESTIMIENTO (Basado en n de Manning) ---
178
179 if n <= 0.013

```

2.7.3 Modelado trascendente para la sección circular

En conductos cerrados operando como canales (alcantarillado), las variables dependen de la función trascendente del ángulo central θ , el algoritmo resuelve $y \rightarrow \theta \rightarrow \{A, P, R_h\} \rightarrow Q$.

Se ha programado una lógica especial para manejar el fenómeno de "lleno relativo". Dado que el caudal máximo ocurre aproximadamente a $y/D \approx 0.938$ y no a tubo lleno ($y/D = 1$), el algoritmo incluye un buscador de máximos locales para evitar que el método de búsqueda de raíces falle por la no-monotonía de la curva de Manning en tuberías.

Figura 17

Código Matlab del cálculo de sección circular

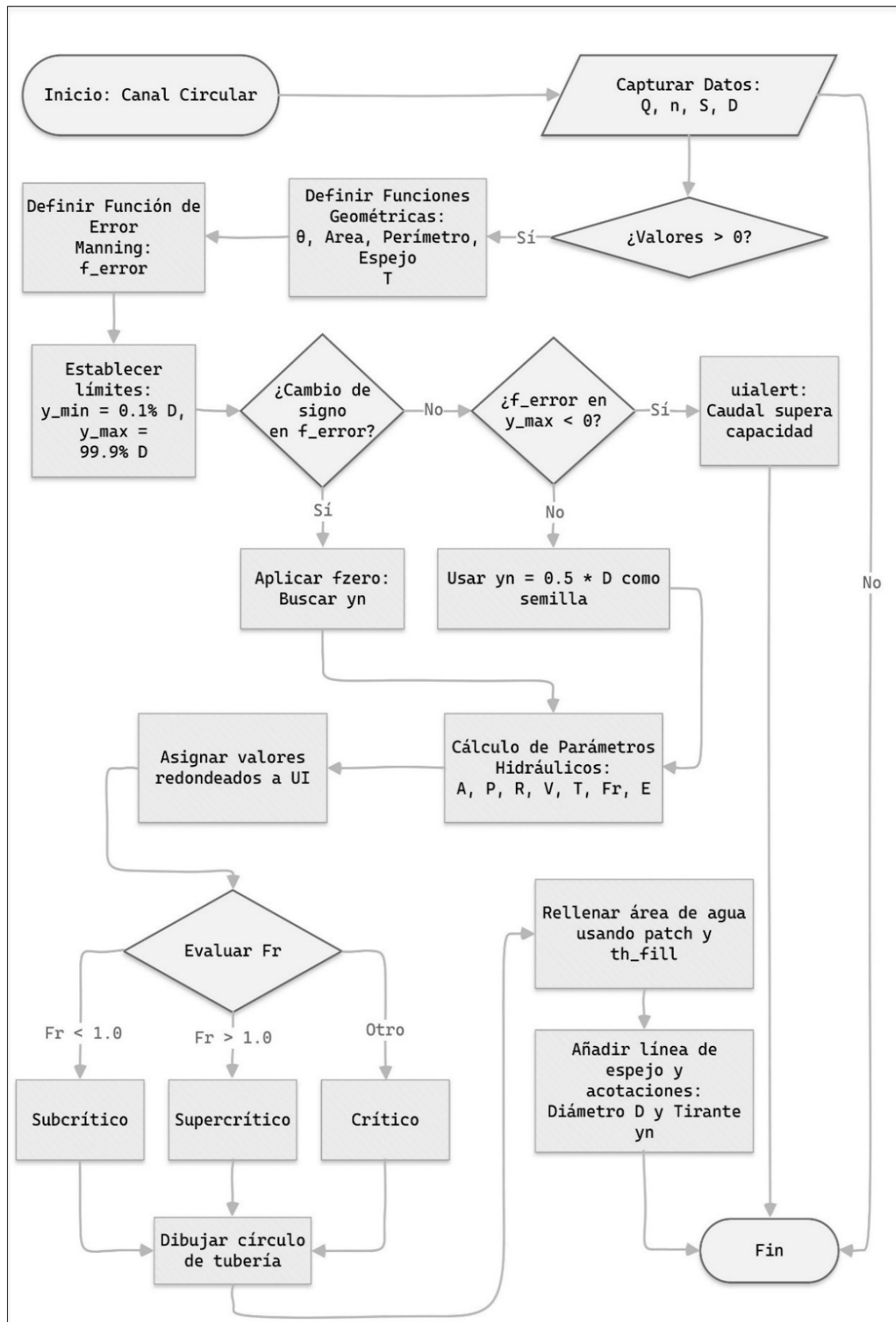
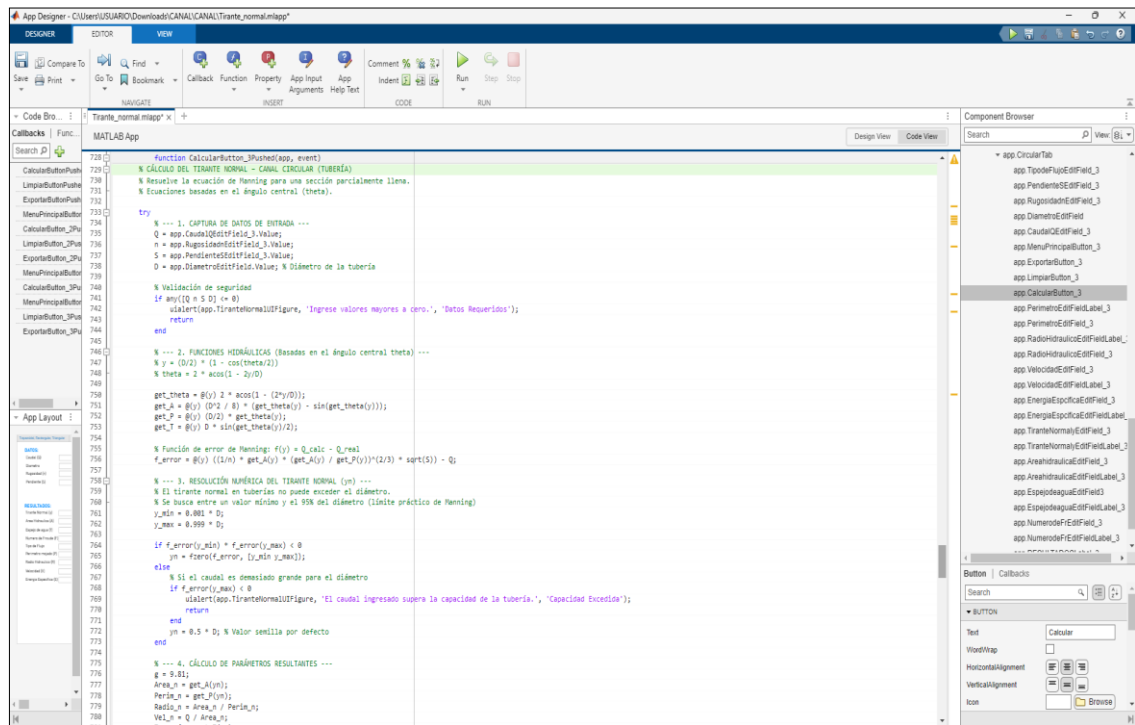


Figura 18

Código Matlab del cálculo de sección circular



2.7.4 Geometría diferencial en la sección parabólica

La sección parabólica representa un desafío mayor debido a que su perímetro mojado no es una función algebraica simple, para el cálculo del perímetro (P) se implementó la resolución exacta de la integral de línea para la curva $y=Cx^2$, se utilizó el método de la Secante debido a la naturaleza logarítmica del perímetro, se utiliza un método de la secante con dos semillas iniciales. Esto permite una convergencia robusta sin enfrentar las oscilaciones que a veces presenta el método de Newton-Raphson en funciones con curvatura variable.

Figura 19

Código Matlab del cálculo de sección Parabólica

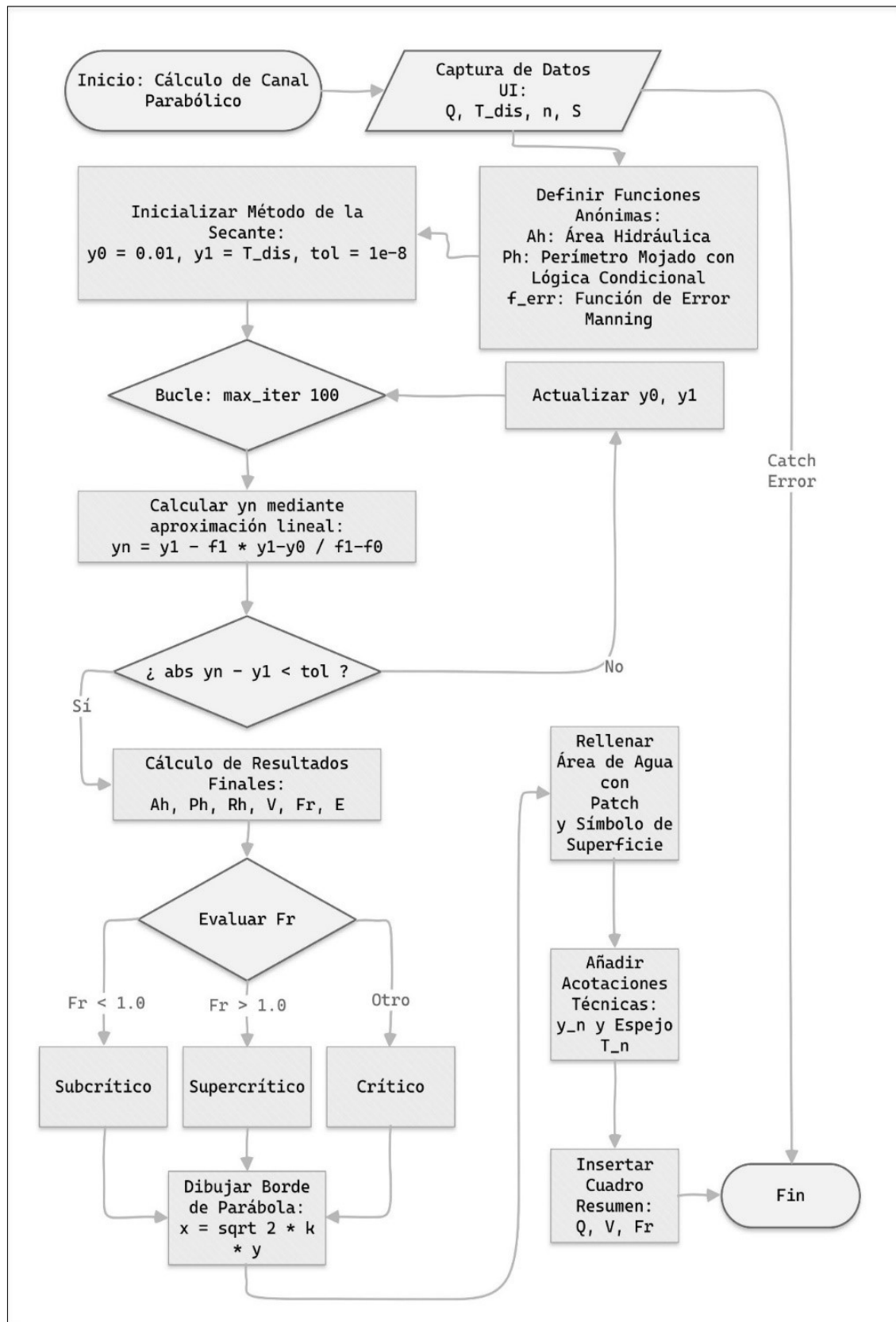
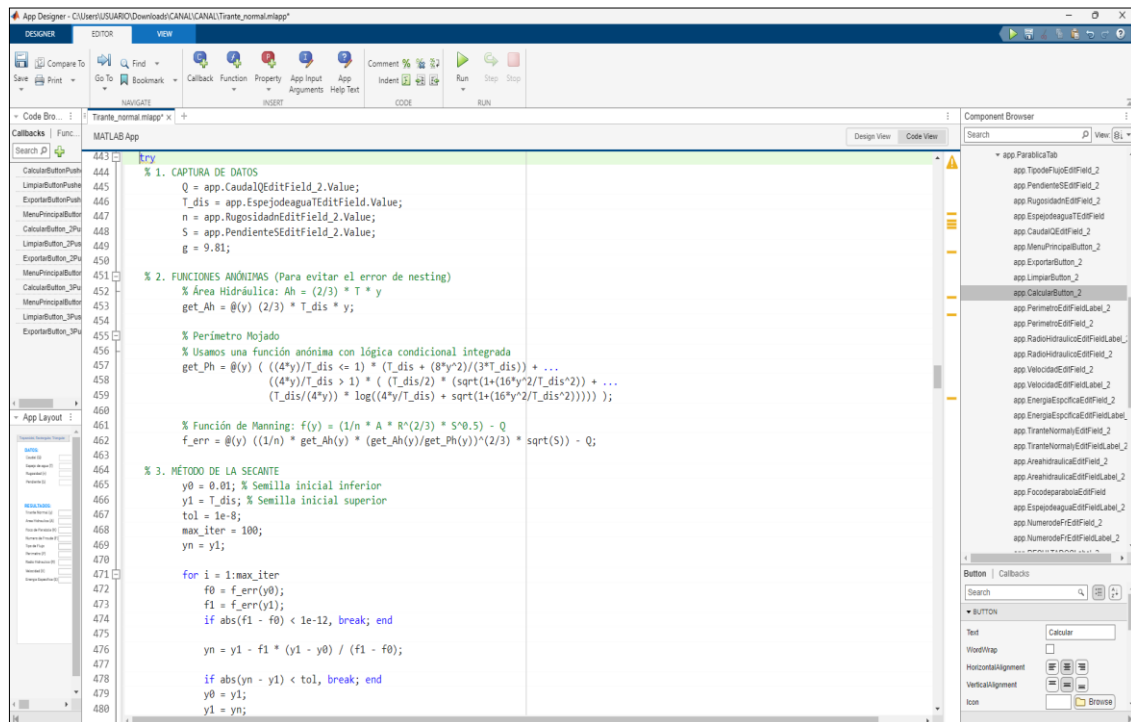


Figura 20

Código Matlab del cálculo de sección Parabólico



2.7.5 Resolución para el tirante crítico (y_c)

El tirante crítico define el estado de energía específica mínima. La relevancia de este cálculo radica en la identificación del régimen de flujo (subcrítico, crítico o supercrítico).

Para una sección trapezoidal, se maneja la variabilidad del talud (z) y la base (b). El algoritmo es capaz de colapsar a una sección rectangular si $z=0$ o a una triangular si $b=0$ sin perder estabilidad numérica.

Figura 21

Código Matlab del cálculo de sección Rectangular, trapezoidal, triangular (critico)

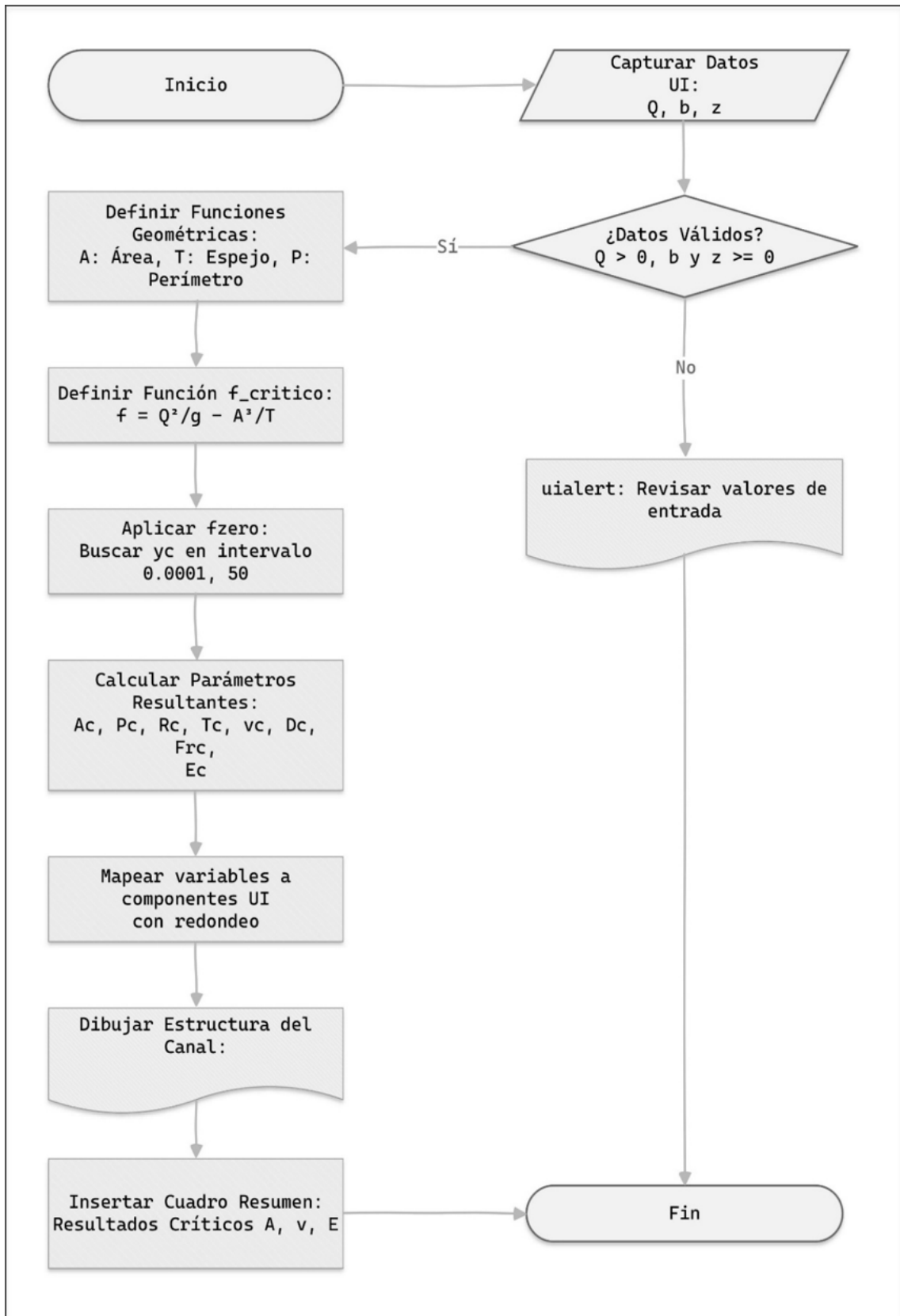
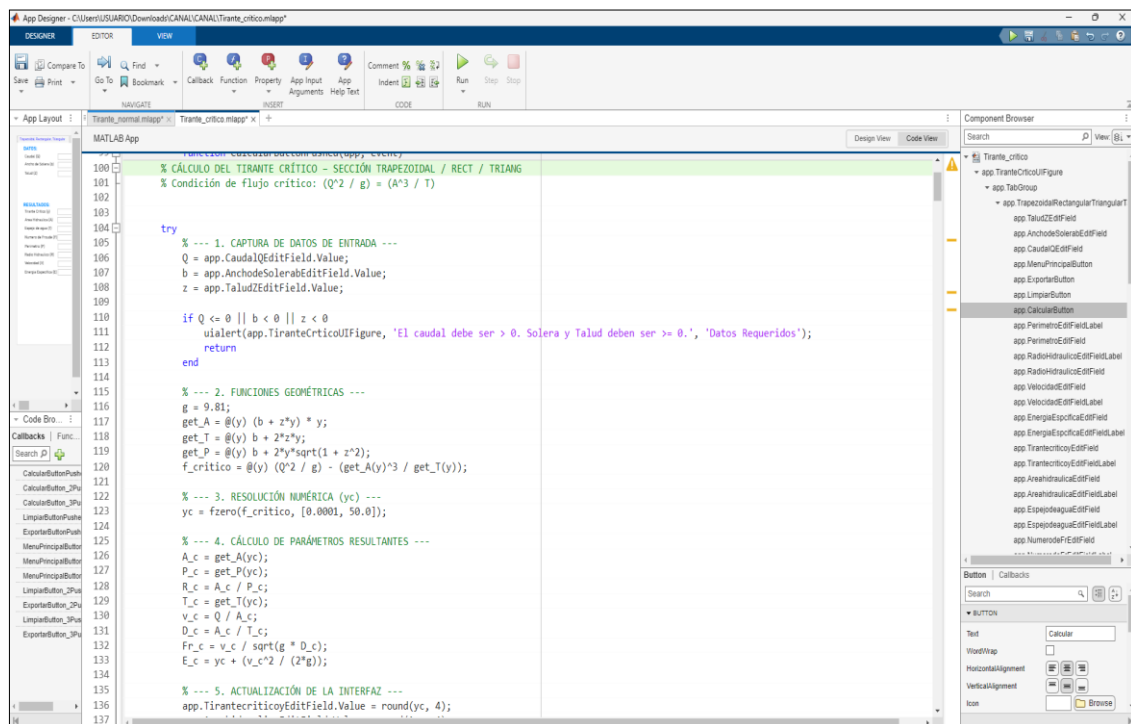


Figura 22

Código Matlab del cálculo de sección rectangular, trapezoidal y triangular (crítico)



Para una sección circular, el código implementa un motor geométrico que define el área (A) y el espejo de agua (T) en función del tirante. Al utilizar la función fzero en un intervalo de 0.1% a 95% del diámetro, el programa garantiza estabilidad numérica y evita soluciones físicamente imposibles. El software localiza el tirante crítico (y_c) con una precisión de 4 decimales mediante la convergencia de la energía mínima.

Para una sección parabólica, se incorporó una validación dinámica mediante uialert para prevenir errores por datos inconsistentes en el caudal o ancho. Para el cálculo del perímetro mojado (P_h), el código emplea una integración exacta con funciones logarítmicas, asegurando una determinación precisa del radio hidráulico (R_h). Este nivel de detalle es fundamental para la fiabilidad de futuros cálculos de rozamiento y la aplicación de la fórmula de Manning.

Figura 23

Código Matlab del cálculo de sección circular (critico)

```

337 % =====
338 % CÁLCULO DEL TIRANTE CRÍTICO - SECCIÓN CIRCULAR
339 % =====
340 try
341     Q = app.CaudalEditField_3.Value;
342     D = app.DiametroEditField.Value;
343     if Q <= 0 || D <= 0, return; end
344
345     g = 9.81;
346     get_theta = @(y) 2 * acos(1 - (2*y/D));
347     get_A = @(y) (D^2 / 8) * (get_theta(y) - sin(get_theta(y)));
348     get_T = @(y) D * sin(get_theta(y)/2);
349     get_P = @(y) (D/2) * get_theta(y);
350     f_obj = @(y) (Q^2 / g) - (get_A(y)^3 / get_T(y));
351     yc = fzero(f_obj, [0.001*D, 0.95*D]);
352
353     A_c = get_A(yc); T_c = get_T(yc); v_c = Q / A_c; E_c = yc + (v_c^2 / (2*g));
354     P_c = get_P(yc); R_c = A_c / P_c; Fr_c = v_c / sqrt(g * (A_c/T_c));
355
356     app.TiranteCriticoEditField_3.Value = round(yc, 4);
357     app.AreaHidraulicaEditField_3.Value = round(A_c, 4);
358     app.EspejodeaguaEditField_3.Value = round(T_c, 4);
359     app.EnergiaEspecificadEditField_3.Value = round(E_c, 4);
360     app.PerimetroEditFieldLabel_3.Value = round(P_c, 4);
361     app.RadioHidraulicoEditFieldLabel_3.Value = round(R_c, 4);
362     app.VelocidadEditField_3.Value = round(v_c, 4);
363     app.NumeroFroudeEditField_3.Value = round(Fr_c, 2);
364
365 % GRÁFICA PROFESIONAL
366 cla(app.UIAxes_3); hold(app.UIAxes_3, 'on');
367 c_agua = [0.15 0.55 0.85];
368 esp = D * 0.2;
369
370 % Dibujo Tubería
371 ang = linspace(0, 2*pi, 100);
372 plot(app.UIAxes_3, (D/2)*cos(ang), (D/2) + (D/2)*sin(ang), 'k', 'LineWidth', 2.5);
373
374 % Relleno Agua
  
```

Figura 24

Código Matlab del cálculo de sección parabólico (critico)

```

210 % --- 1. CAPTURA DE VARIABLES (SOLO Q Y T) ---
211 Q = app.CaudalEditField_2.Value;
212 T_max = app.EspejodeaguaEditField.Value;
213 g = 9.81;
214
215 % Obtener la figura contenedora de forma dinámica para evitar errores
216 fig = ancestor(app.UIAxes_2, 'figure');
217
218 % Validación inicial
219 if Q <= 0 || T_max <= 0
220     uialet(fig, 'Ingrese Caudal (Q) y Espejo de agua (T) válidos.', 'Atención');
221     return;
222 end
223
224 % --- 2. CÁLCULO DEL TIRANTE CRÍTICO (yc) ---
225 % Condición crítica: Q^2/g = A^3 / T | Parábola: A = (2/3)*T*y
226 f_critico = @(y) ( ((2/3) * T_max * y)^3 / T_max ) - (Q^2 / g);
227
228 try
229     % Resolución numérica para yc
230     yc = fzero(f_critico, [0.0001, 100]);
231
232 % --- 3. PARÁMETROS HIDRÁULICOS DERIVADOS ---
233 Ah = (2/3) * T_max * yc;
234 v = Q / Ah;
235 Dh = Ah / T_max;
236 Fr = v / sqrt(g * Dh); % Confirmación (debe dar ~1.0)
237 E = yc + (v^2 / (2 * g));
238 k_parabola = T_max^2 / (16 * yc); % y = x^2 / (4k)
239
240 % Perimetro Mojado (Fórmula exacta para precisión)
241 t_param = (4 * yc) / T_max;
242 Ph = (T_max / 2) * (sqrt(1 + t_param^2) + (1 / t_param) * log(t_param + sqrt(1 + t_param^2)));
243 Rh = Ah / Ph;
244
245 % --- 4. ASIGNACIÓN A COMPONENTES DE LA INTERFAZ ---
246 app.TiranteCriticoEditField_2.Value = round(yc, 4);
247 app.AreaHidraulicaEditField_2.Value = round(Ah, 4);
  
```

Figura 25

Código Matlab del cálculo de sección circular (critico)

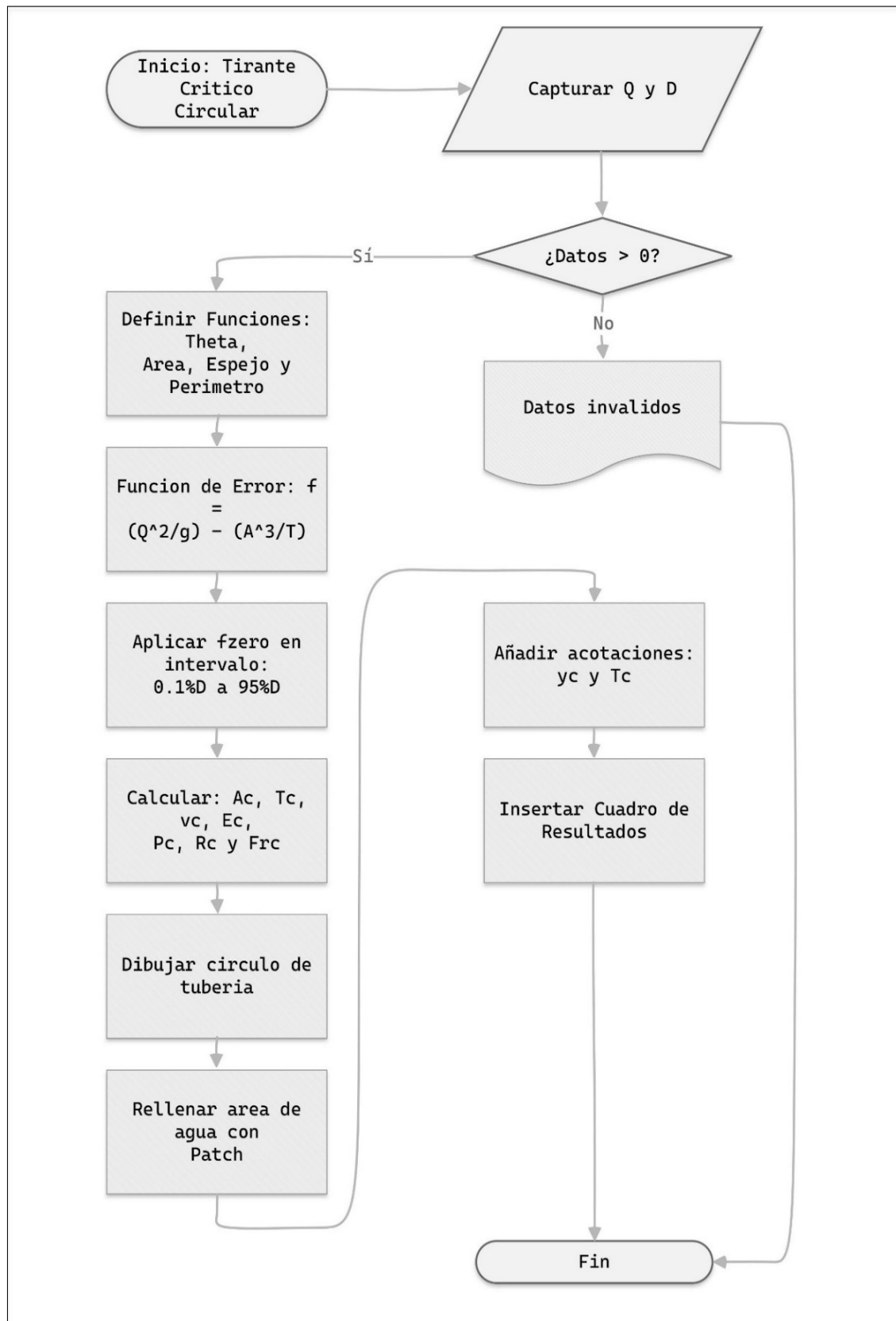
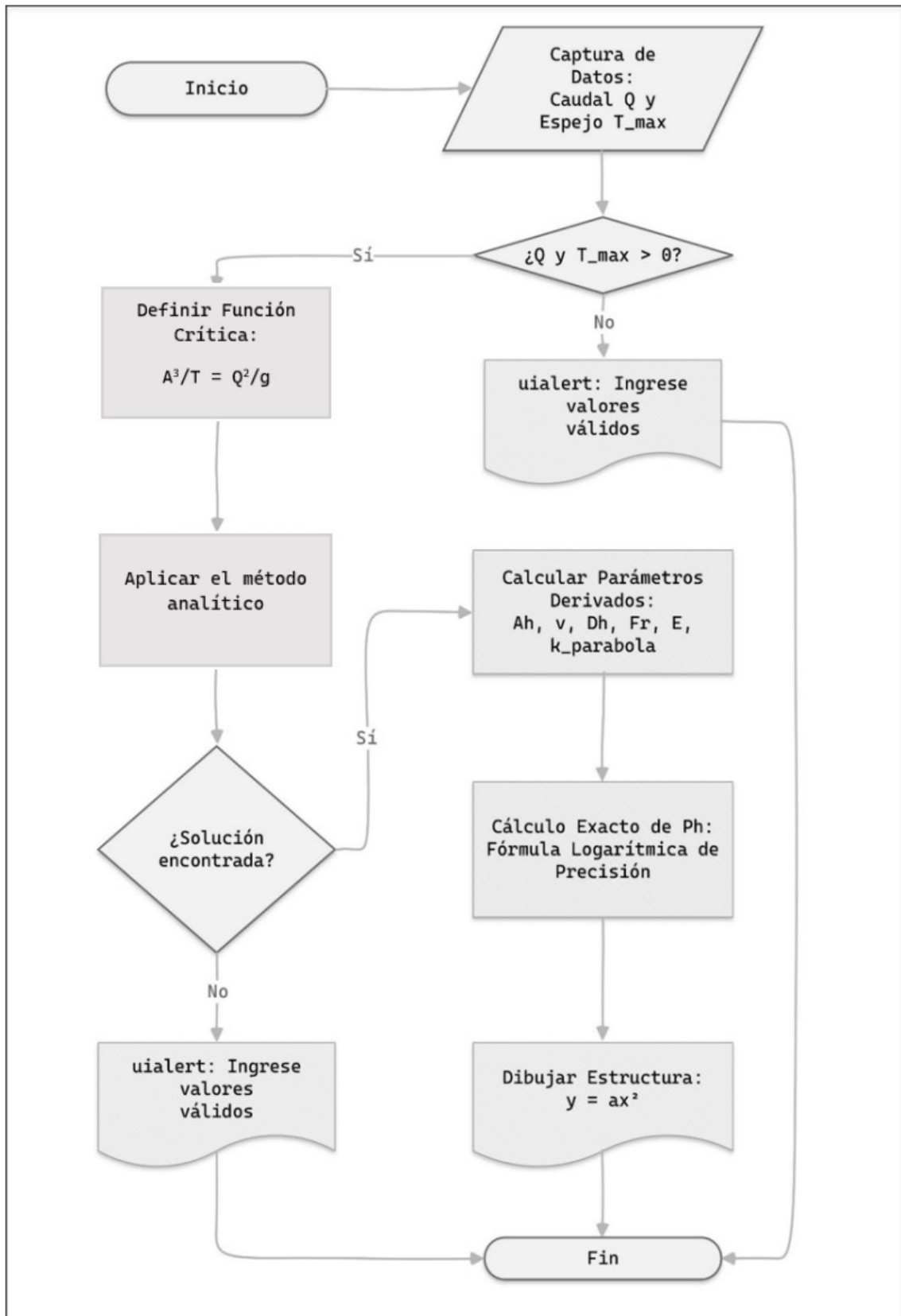


Figura 26

Código Matlab del cálculo de sección Parabólica (critico)



2.7.6 Métodos y tolerancias computacionales

Para garantizar la transparencia del proceso algorítmico, se presenta la siguiente tabla con las especificaciones técnicas de los motores de cálculo:

Tabla 2

Resumen de métodos y tolerancias computacionales

Parámetro Hidráulico	Tipo de Sección	Método Numérico	Tolerancia (m)	Lógica de Programación
Tirante Normal (y)	Prismática	Brent (fzero)	1.0×10^{-6}	Búsqueda en intervalo físicamente posible.
Tirante Normal (y)	Circular	Brent (fzero)	1.0×10^{-6}	Control de capacidad máxima
Tirante Normal (y)	Parabólica	Secante	1.0×10^{-8}	Resolución de integral de línea logarítmica.
Tirante Crítico (y_c)	Prismática	Brent (fzero)	1.0×10^{-6}	Optimización de la curva de Energía.
Tirante Crítico (y_c)	Circular	Brent (fzero)	1.0×10^{-6}	Formulas ecuación de flujo crítico
Tirante Crítico (y_c)	Parabólica	Analítico	-	Resolución de la relación $A^3/T = Q^2/g$.

2.7.7 Interfaz del software Canaly con App Designer

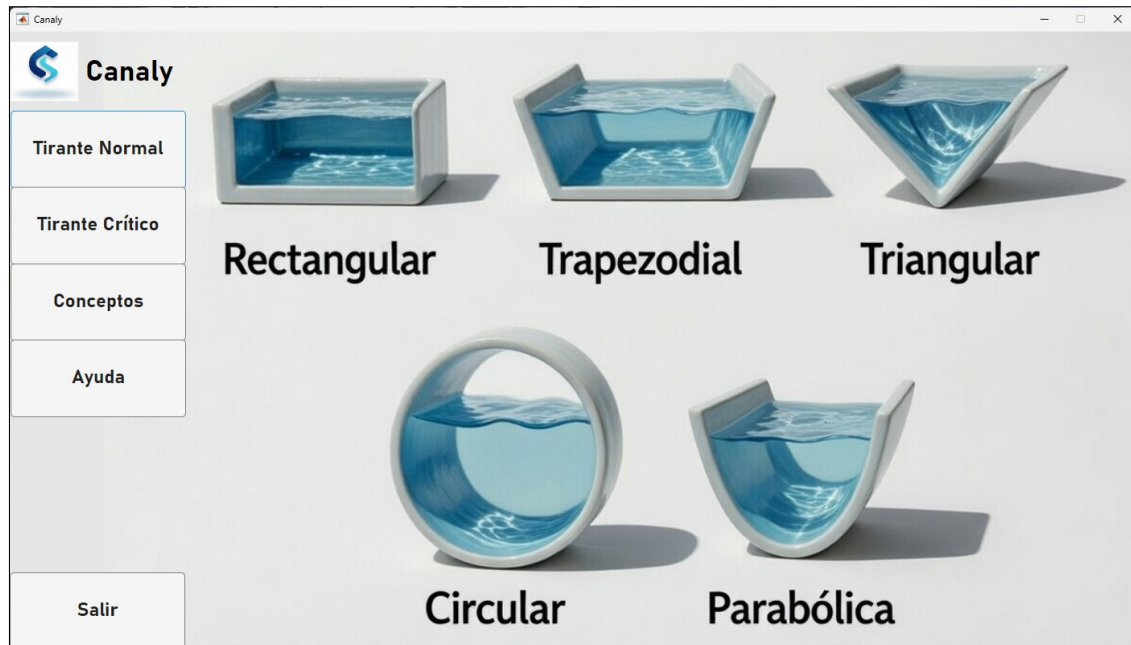
El desarrollo del software se realizó en la versión de Matlab R2025b, donde no se limita solo al cálculo, sino que también incluye un motor de visualización que actúa como un postprocesador de ingeniería.

El sistema desarrollado incorpora una representación gráfica profesional mediante el uso de áreas hidráulicas con transparencia dinámica y líneas de contorno diferenciadas para identificar claramente la estructura del canal y la superficie libre del agua. Asimismo, emplea una escala 1:1 para evitar distorsiones visuales en los taludes y facilitar la inspección geométrica, también cuenta con un sistema automático de acotado que posiciona las etiquetas de manera óptima, evitando superposiciones y manteniendo una presentación similar a un plano, además integra mecanismos de validación estática y dinámica para detectar errores en los datos de entrada o fallas de convergencia en los métodos numéricos, notificando oportunamente al usuario. Finalmente, los resultados

obtenidos pueden exportarse a Excel en tablas normalizadas, siguiendo estándares de presentación utilizados en ingeniería.

Figura 27

Presentación de la interfaz de las opciones principales del programa Canaly

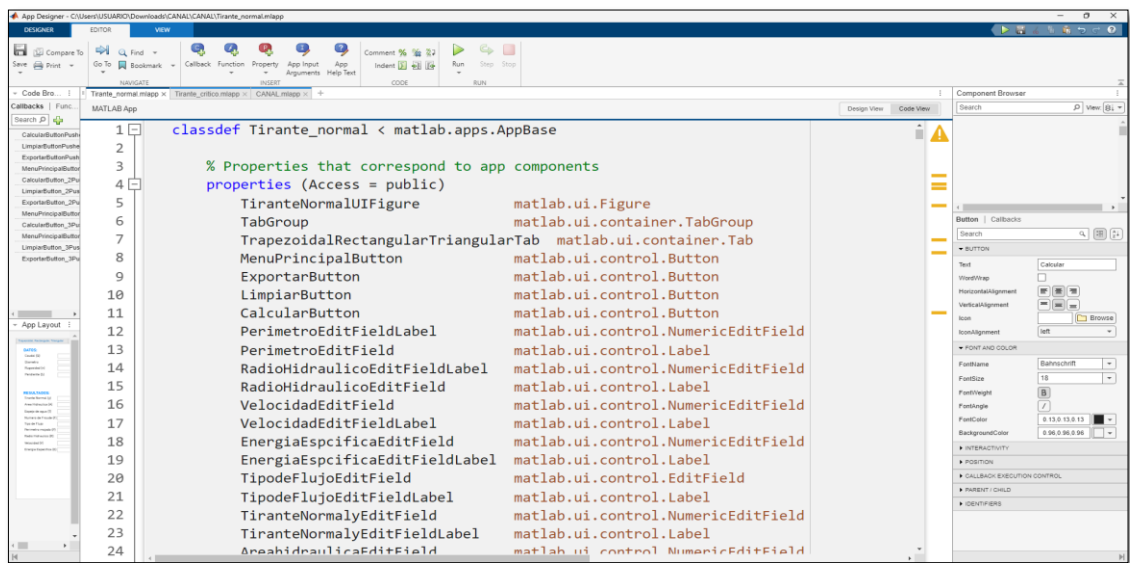


El diseño de la Interfaz Gráfica de Usuario (GUI) no se concibió únicamente como un contenedor de datos, sino como un Entorno de Ingeniería Integrado (IEE). Utilizando el framework de App Designer, se desarrolló una aplicación orientada a objetos (OOP) que permite una interacción fluida entre el usuario y los motores de cálculo hidráulico.

La aplicación está construida bajo una estructura de clase única en Matlab (`classdef AppName < Matlab.apps.AppBase`), lo que permite que todas las variables, propiedades y funciones residan en un entorno encapsulado.

Figura 28

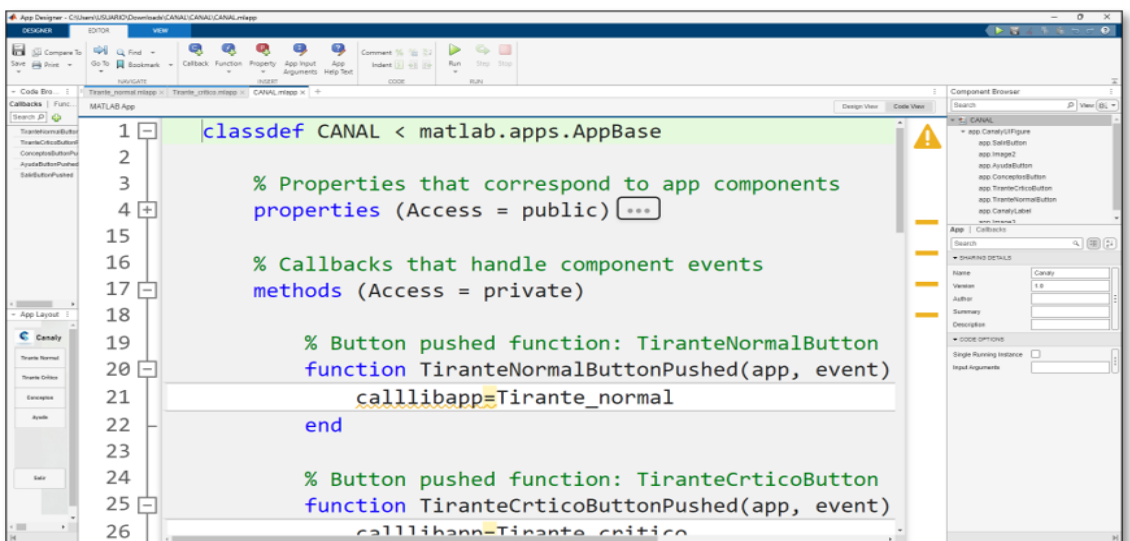
Estructura de clase única en Matlab



Se definieron propiedades internas para almacenar los resultados de los cálculos (tirantes, áreas, radios hidráulicos) de forma global dentro de la app. Esto permite que la pestaña de "Gráficos" o "Reportes" pueda acceder a los datos generados en la pestaña de "Cálculo" sin necesidad de recalcular. En lugar de saturar los Callbacks de los botones, se crearon Funciones de Soporte (StartupFcn y funciones privadas). Por ejemplo, existe una función única de Actualización de Ejes que es llamada por múltiples componentes, reduciendo la redundancia de código y facilitando el mantenimiento.

Figura 29

Propiedades Privadas y Públicas en Matlab (App Designer)

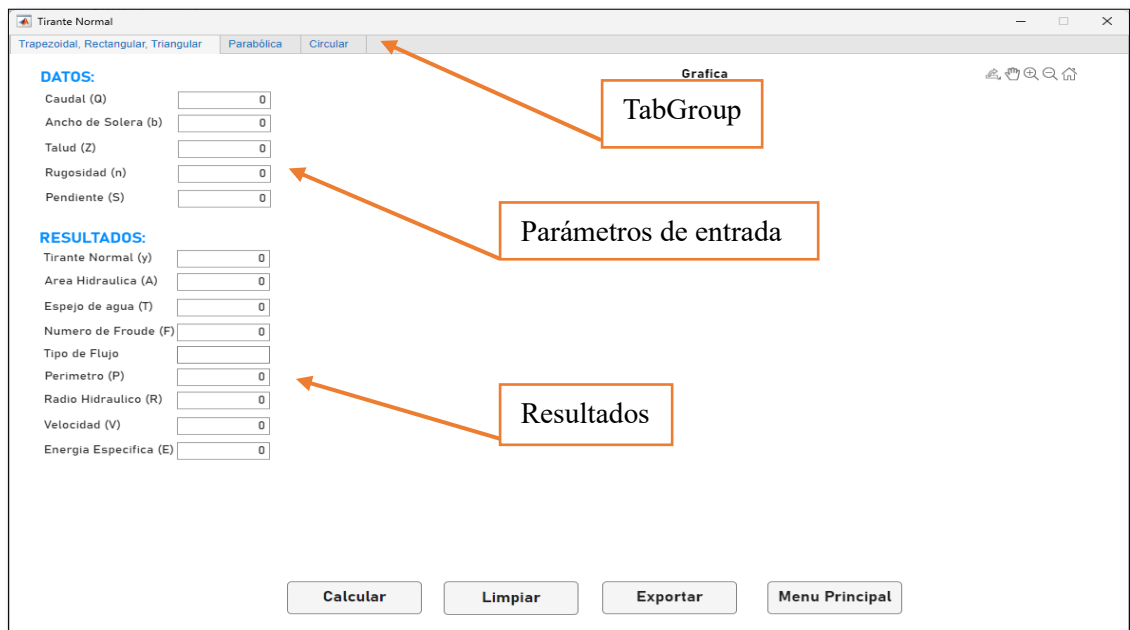


1. Organización Mediante Tab Groups y Panels

- Contenedores de Primer Nivel: Se implementó un TabGroup principal que segmenta la aplicación por tipo de sección (rectangular, trapezoidal, triangular, circular y parabólica). Esto evita el desorden visual y permite que cada sección tenga sus propios parámetros específicos (como el diámetro en círculos o el talud en trapecios).
- Paneles de Control Estáticos: Dentro de cada pestaña, se utilizaron Panels con títulos descriptivos y bordes definidos. El panel "Parámetros de Entrada" utiliza un color de fondo sutilmente diferenciado para destacar sobre el panel de "Resultados", guiando intuitivamente al usuario sobre dónde debe interactuar primero.

Figura 30

Interfaz de cálculo de tirante normal del programa Canaly



2. Componentes de Entrada y Validación en Tiempo Real

- Numeric Edit Fields con Límites: A diferencia de los campos de texto, estos componentes permiten configurar límites (Limits). Si un usuario intenta ingresar una rugosidad de Manning de 0.00 o un valor negativo, el componente puede resaltar en rojo o revertir automáticamente al último valor válido, actuando como un filtro preventivo de errores.

- Etiquetas Dinámicas: Las etiquetas de las unidades (m , m^3/s , m/m) están fijadas al lado de cada campo, eliminando ambigüedades en el sistema de medidas (Sistema Internacional).

Un software de ingeniería debe comunicar sus procesos de manera transparente. Se implementaron múltiples mecanismos de retroalimentación:

- Indicadores de Régimen (Gauges y Lamps):

Se integró un Medidor de Froude tipo Gauge con escalas de colores: Verde ($Fr < 1$, Subcrítico), Amarillo ($Fr \approx 1$, Crítico) y Rojo ($Fr > 1$, Supercrítico).

Este feedback visual inmediato permite al ingeniero realizar análisis de sensibilidad de forma rápida: al variar la pendiente, observa instantáneamente cómo la aguja se desplaza hacia zonas de flujo inestable.

- Diálogos de Interrupción (Modals):

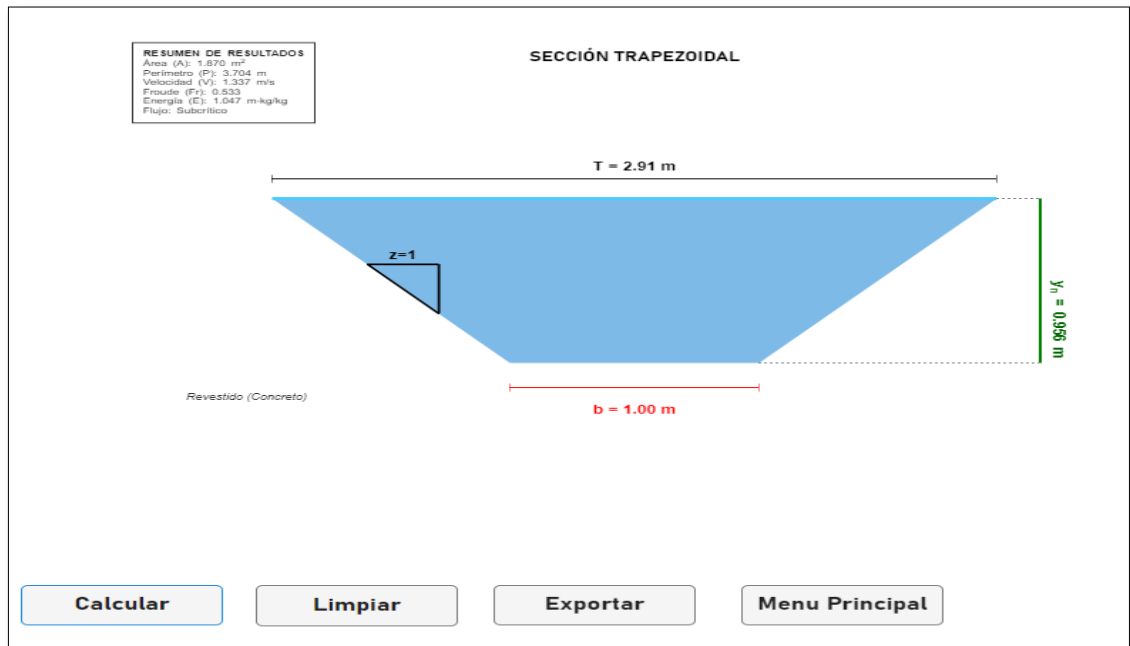
Para errores críticos (como la falta de convergencia en un tirante circular debido a un caudal excesivo), se programaron uialert personalizados. Estos cuadros de diálogo incluyen iconos de advertencia y mensajes técnicos que sugieren la acción correctiva (ej. "Aumente el diámetro de la tubería o reduzca el caudal").

El componente UIAxes es la pieza central de la visualización. No es una gráfica estática, sino un esquema técnico dinámico controlado por el algoritmo de dibujo.

- Geometría Diferenciada: Se programó el uso de diferentes estilos de línea (Sólida para el contorno del canal, Punteada para niveles de referencia y Gruesa para la superficie libre del agua).
- Inyección de Texto en Ejes: Utilizando la función `text()`, el software inyecta etiquetas con los valores calculados (y_n , y_c , T) directamente en el gráfico. Estas etiquetas tienen una lógica de "auto-reposicionamiento" para que siempre aparezcan centradas respecto a las cotas, independientemente del tamaño del canal.
- Interactividad: Los ejes permiten el uso de herramientas de inspección de datos (Data Tips), de modo que el usuario puede hacer clic en cualquier punto de la sección transversal y obtener las coordenadas (x,y) exactas.

Figura 31

Grafica generada en Canaly



3. Conectividad y Salida de Datos (Export Engine)

El diseño de la interfaz culmina en la capacidad de extraer la información para su uso en informes de consultoría.

- Botones de Acción Global: Situados generalmente en la parte inferior o en una pestaña dedicada, estos botones activan el servidor COM de Windows para interactuar con Microsoft Excel.
- Barra de Progreso (uiprogressdlg): Para operaciones que toman más de un segundo (como la exportación de reportes detallados), se diseñó una barra de progreso animada que informa al usuario: "Generando reporte en Excel... Espere por favor". Esto mejora la percepción de rendimiento de la app.

Figura 32

Exportación a Excel en Canaly

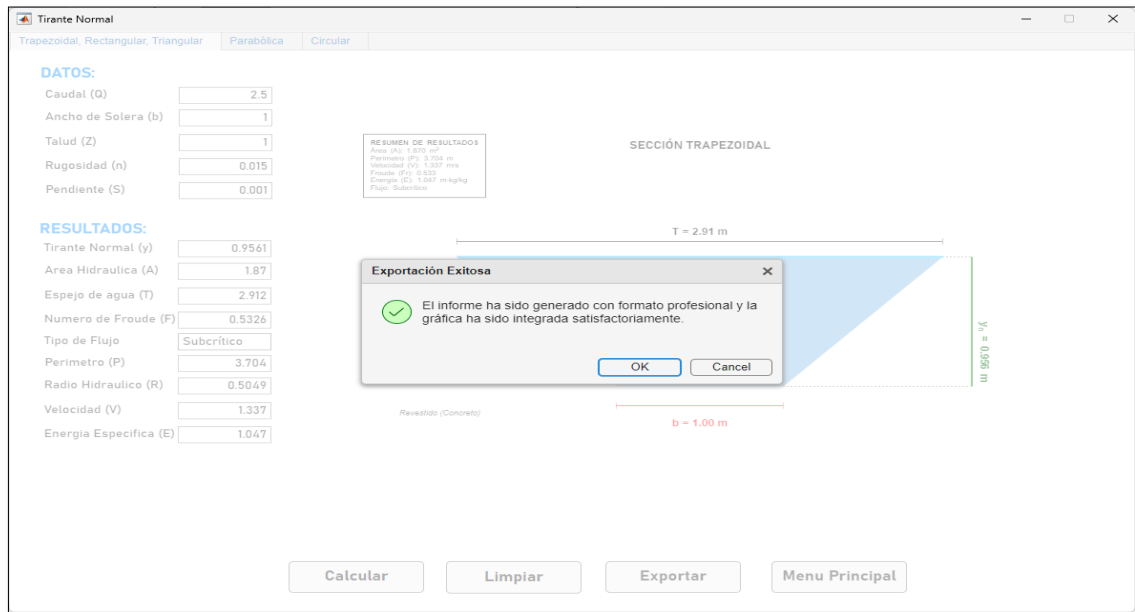
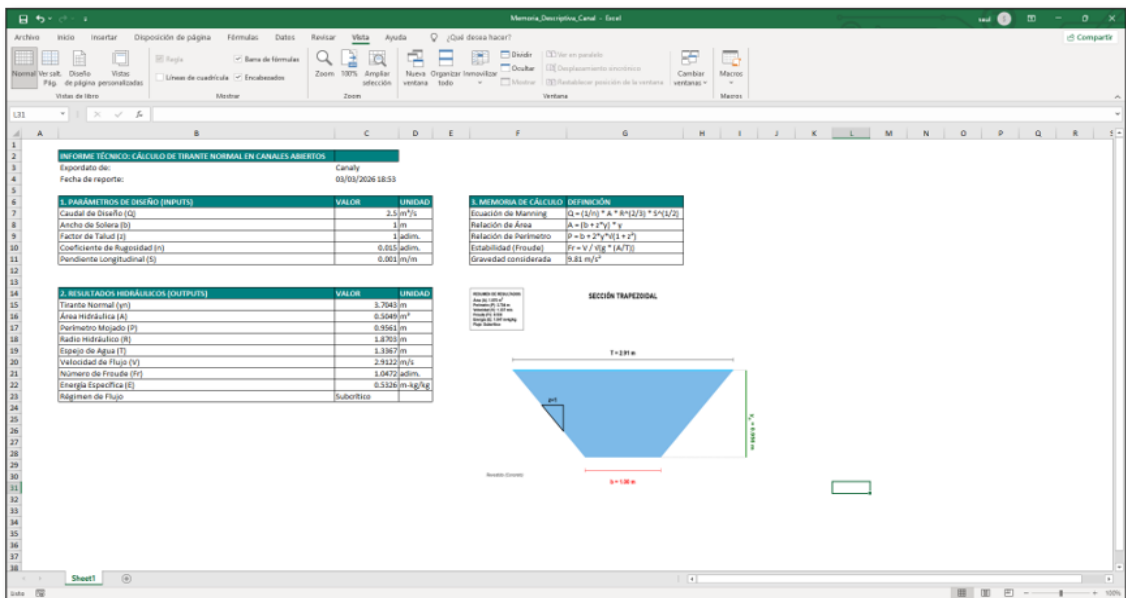


Figura 33

Generación de una tabla de resultados en formato Excel



El software para ser utilizado en diferentes equipos que no tengan instalados el Matlab, se procedió a realizar la compilación en ejecutable .exe, mediante el uso del Standalone Applications, mediante el uso del paquete Application Compiler, lo cual fue descargado a Matlab para proceder a la conversión a ejecutable.

Tabla 3*Resumen de componentes técnicos implementados*

Componente	Implementación específica	Impacto en el diseño
Tab Group	4 pestañas (Geometrías + Reportes)	Organización lógica y modular.
NumericEditField	Máscara de entrada con 4 decimales	Precisión y control de tipos de datos.
Gauge / Lamp	Lógica de colores según el Régimen	Identificación instantánea de problemas hidráulicos.
UIAxes	Renderizado con patch y axis equal	Representación física sin distorsiones (CAD-Style).
Button Callbacks	Algoritmos de optimización (fzero)	Respuesta rápida y cálculos en milisegundos.
HTML UI (Optional)	Visualización de documentación técnica	Ayuda integrada dentro de la misma aplicación.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1 Resultados

La revisión bibliográfica realizada permitió identificar y seleccionar las ecuaciones fundamentales empleadas en el desarrollo del software. Estas ecuaciones fueron organizadas según las diferentes secciones geométricas de los canales, considerando las características particulares de cada tipo. Para cada sección se presentan los procedimientos de cálculo correspondientes, basados en los principios expuestos en el libro Hidráulica de Canales Abiertos del profesor Wen Te Chow, Villón, Cadavid.

3.1.1 Sistematizar los modelos matemáticos y ecuaciones

La información recopilada de flujo uniforme y crítico sirvió como referencia para la ejecución de los cálculos manuales iniciales, los cuales constituyeron la base para las etapas posteriores del proyecto, incluyendo el análisis, diseño, programación, pruebas y validación del software. Este conjunto de actividades forma parte del desarrollo del software. A continuación, se describen detalladamente los cálculos asociados a cada sección geométrica de canal abierto considerada en la investigación.

3.1.1.1 Sección circular

Esta geometría se caracteriza por el diámetro D , su utilización como canal abierto es poco frecuente, generalmente se utiliza como una sección de conducción.

- Ángulo θ :

$$\theta = 2 \cdot \arccos\left(1 - \frac{2y}{D}\right)$$

- Área normal (A):

$$A = \frac{(\theta - \sin \theta) \cdot D^2}{8}$$

- Perímetro mojado normal (P):

$$P = \frac{\theta \cdot D}{2}$$

- Radio hidráulico (R_h):

$$R_h = \left(1 - \frac{\sin \theta}{\theta}\right) \cdot \frac{D}{4}$$

- Espejo de agua (T):

$$T = 2 \cdot \sqrt{y \cdot (D - y)}$$

3.1.1.2 Sección rectangular

Se utiliza el ancho de base b y altura y . Es la sección más simple entre todas las usadas en las aplicaciones prácticas de la hidráulica de canales.

- Área normal (A):

$$A = b \cdot y$$

- Perímetro mojado normal (P):

$$P = b + 2y$$

- Radio hidráulico (R_h):

$$R_h = \frac{b \cdot y}{b + 2y}$$

- Espejo de agua (T):

$$T = b$$

3.1.1.3 Sección trapezoidal

Determinada por la base b , el tirante y , y los taludes laterales z . Es la solución más recomendada para excavaciones, ya que los taludes brindan estabilidad geotécnica.

- Área normal (A):

$$A = (b + z \cdot y) \cdot y$$

- Perímetro mojado normal (P):

$$P = b + 2y \cdot \sqrt{1 + z^2}$$

- Radio hidráulico (R_h):

$$R_h = \frac{(b + z \cdot y) \cdot y}{b + 2y\sqrt{1 + z^2}}$$

- Espejo de agua (T):

$$T = b + 2 \cdot z \cdot y$$

3.1.1.4 Sección triangular

Caracterizada por el talud z y el tirante y . Se utiliza primordialmente en canales destinados al drenaje.

- Área normal (A):

$$A = z \cdot y^2$$

- Perímetro mojado (P):

$$P = 2y \cdot \sqrt{1 + z^2}$$

- Radio hidráulico (R_h):

$$R_h = \frac{z \cdot y}{2\sqrt{1 + z^2}}$$

- Espejo de agua (T):

$$T = 2 \cdot z \cdot y$$

3.1.1.5 Sección parabólica

Se determina a través del espejo de agua T y la profundidad y . Se emplea en canales revestidos y es la forma común en canales naturales.

- Área normal (A):

$$A = \frac{2}{3} \cdot T \cdot y$$

- Perímetro mojado (P):

$$P = T + \frac{8y^2}{3T}$$

$$P = \frac{T}{2} \sqrt{1 + \left(\frac{4y}{T}\right)^2} + \frac{T^2}{8y} \ln \left(\frac{4y}{T} + \sqrt{1 + \left(\frac{4y}{T}\right)^2} \right)$$

- Radio hidráulico (R_h):

$$R_h = \frac{2T^2y}{3T^2 + 8y^2}$$

- Espejo de agua (T):

$$T = \frac{3A}{2y}$$

3.1.1.6 Aspectos de la dinámica de fluidos

- Velocidad media (v):

$$v = \frac{1}{n} \cdot R_h^{2/3} \cdot S^{1/2}$$

- Ecuación de continuidad:

$$Q = A \cdot v$$

- Carga piezométrica (H_p):

$$H_p = Z + \frac{p}{\gamma}$$

- Carga hidráulica (H):

$$H = Z + \frac{p}{\gamma} + \alpha \frac{V^2}{2g}$$

- Número de Froude (F_r):

$$F_r = \frac{V}{\sqrt{g \cdot A/T}}$$

- Número de Reynolds (R_e):

$$R_e = \frac{d \cdot V \cdot \rho}{\mu}$$

3.1.1.7 Energía específica (E)

- Rectangular:

$$E = y + \frac{\alpha Q^2}{2gb^2y^2}$$

- Triangular:

$$E = y + \frac{\alpha Q^2}{2gz^2y^4}$$

- Trapezoidal:

$$E = y + \frac{\alpha Q^2}{2gy^2(b + zy)^2}$$

3.1.1.8 Flujo crítico

- Canal Rectangular:

$$y_c = \sqrt[3]{\frac{\alpha q^2}{g}}$$

- Canal Triangular:

$$y_c = \sqrt[5]{\frac{2\alpha Q^2}{gz^2}}$$

- Energía crítica (rectangular):

$$E_c = \frac{3}{2} y_c$$

3.1.2 *Objetivo 2: Implementación de algoritmos numéricos*

La arquitectura computacional del sistema ha sido diseñada bajo un paradigma de alta fidelidad numérica para la resolución de las ecuaciones fundamentales de la hidráulica de canales abiertos. Este objetivo se centra en la transposición de modelos matemáticos implícitos a algoritmos ejecutables, garantizando estabilidad, precisión y eficiencia en el procesamiento de datos.

3.1.2.1 **Métodos y tolerancias computacionales**

Se establecieron criterios de convergencia para los algoritmos iterativos que consistieron en verificar el error relativo aproximado entre las iteraciones sucesivas calculadas mediante los diferentes métodos numéricos utilizados como el algoritmo de Brent (fzero) y la secante, en las cuales los diferentes métodos utilizaron iterativas sucesivas con una tolerancia de 1.0×10^{-6} y 1.0×10^{-8} en el caso de tirante normal para sección parabólica hasta satisfacer un criterio de convergencia (Chapra, 2018). Adicionalmente se definió un número máximo de iteraciones como mecanismo de control para evitar procesos iterativos indefinidos en caso de alcanzarse la convergencia. Estos criterios permitieron garantizar la estabilidad numérica y la precisión de las soluciones obtenidas para las ecuaciones establecidas para el cálculo de los tirantes resueltas en Matlab R2025b. Para garantizar la transparencia del proceso algorítmico, se presenta la siguiente tabla con las especificaciones técnicas de los motores de cálculo:

- **Implementación del Algoritmo de Brent (fzero)**

Para las secciones trapezoidales, rectangulares y triangulares, se ha seleccionado el Método de Brent, basado en las técnicas de secante, bisección e interpolación cuadrática inversa.

El software evalúa la función de error $f(y) = \Phi(y) - Q_{objetivo}$, donde $\Phi(y)$ es la capacidad de transporte. El algoritmo detecta el cambio de signo en el intervalo $[0.0001, 100]$ y converge con una precisión de hasta 10^{-6} metros.

En secciones circulares, en conductos cerrados operando como canales (alcantarillado), las variables dependen de la función trascendente del ángulo central θ .

El algoritmo resuelve $y \rightarrow \theta \rightarrow \{A, P, R_h\} \rightarrow Q$. Se ha programado una lógica especial para manejar el fenómeno de "lleno relativo". Dado que el caudal máximo ocurre aproximadamente a $y/D \approx 0.938$ y no a tubo lleno ($y/D = 1$), el algoritmo incluye un buscador de máximos locales para evitar que el método de búsqueda de raíces falle por la no-monotonía de la curva de Manning en tuberías.

- **Método de la Secante**

Debido a la naturaleza logarítmica del perímetro, se utiliza un método de la secante con dos semillas iniciales. Esto permitió una convergencia robusta sin enfrentar las oscilaciones que a veces presenta el método de Newton-Raphson en funciones con curvatura variable.

- **Método Analítico**

Este método permitió resolver de manera eficiente las ecuaciones implícitas asociadas a las secciones parabólicas, mediante la inserción de datos de caudal y espejo de agua, aplicando la ecuación directamente para el cálculo del tirante crítico que corresponde a la incógnita.

El procedimiento consistió en definir un intervalo de búsqueda que contuviera la raíz y la evaluación sucesivamente la función hasta localizar el valor del tirante que satisface la condición de flujo crítico, que garantizó una elevada precisión en los resultados obtenidos, este enfoque permitió resolver satisfactoriamente las ecuaciones implícitas asociadas a la geometría parabólica, donde es posible obtener una solución analítica directa.

Tabla 4

Métodos y tolerancias computacionales aplicadas en el software Canaly

Parámetro Hidráulico	Tipo de Sección	Método Numérico	Tolerancia (m)
Tirante Normal (y)	Prismática	Brent (fzero)	1.0×10^{-6}
Tirante Normal (y)	Circular	Brent (fzero)	1.0×10^{-6}
Tirante Normal (y)	Parabólica	Secante	1.0×10^{-8}
Tirante Crítico (y_c)	Prismática	Brent (fzero)	1.0×10^{-6}
Tirante Crítico (y_c)	Circular	Brent (fzero)	1.0×10^{-6}
Tirante Crítico (y_c)	Parabólica	Analítico	-

Para el desarrollo del software propuesto se empleó la metodología UML (Lenguaje Unificado de Modelado), la cual permitió estructurar y planificar el diseño lógico del sistema antes de iniciar la etapa de codificación, los procedimientos y cálculos desarrollados de forma manual fueron representados gráficamente, facilitando la descripción, especificación, visualización, construcción y documentación de los diferentes componentes del proyecto.

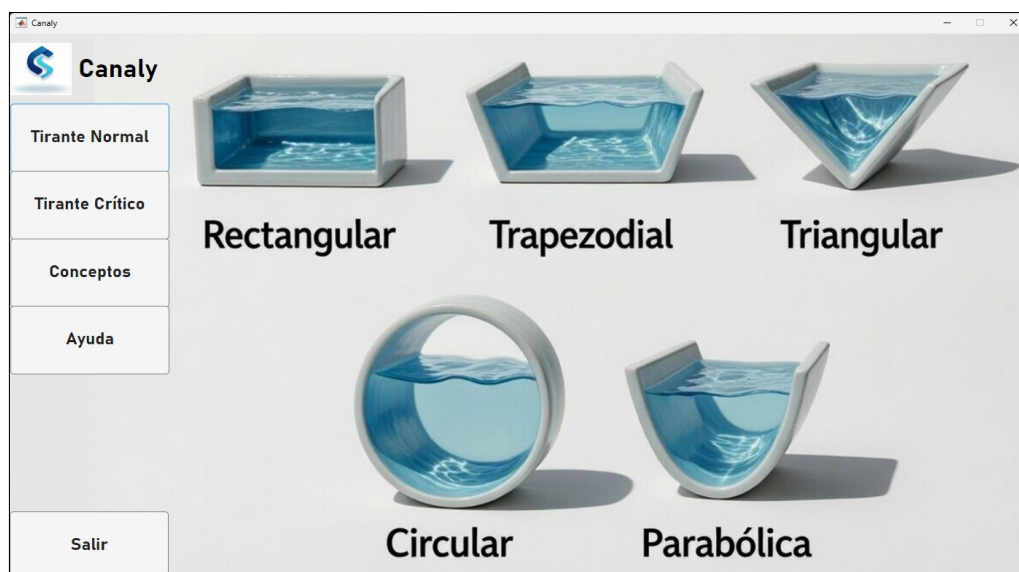
Se utilizaron diagramas de actividades para representar de manera secuencial el flujo de procesos, decisiones y operaciones involucradas en el funcionamiento de la aplicación. Estos diagramas, similares a los diagramas de flujo empleados en el desarrollo de software, permitieron definir con claridad la lógica de programación y la interacción entre las distintas etapas del cálculo hidráulico. En las figuras siguientes se presentan los diagramas UML elaborados para cada tipo de canal considerado en la investigación, los cuales sirvieron como base para la implementación de los algoritmos en la versión Matlab R2025b.

3.1.3 *Objetivo 3: Diseño de la interfaz gráfica con App Designer*

El diseño de la Interfaz Gráfica de Usuario (GUI) no se concibió únicamente como un contenedor de datos, sino como un Entorno de Ingeniería Integrado. Utilizando el framework de App Designer, se desarrolló una aplicación orientada a objetos (OOP) que permite una interacción fluida entre el usuario y los motores de cálculo hidráulico.

Figura 34

Interfaz gráfica del software Canaly



3.1.3.1 Ventanas del software

La aplicación fue construida bajo una estructura de clase única en Matlab, lo que permite que todas las variables, propiedades y funciones residan en un solo entorno.

Para garantizar un funcionamiento eficiente de la aplicación, se definieron propiedades internas destinadas al almacenamiento de los resultados obtenidos durante los cálculos hidráulicos, tales como el tirante, el área hidráulica y el radio hidráulico. Esta estrategia permite conservar la información generada y utilizarla en distintos módulos de la aplicación sin necesidad de ejecutar nuevamente los cálculos, mejorando así la eficiencia del procesamiento. Gracias a ello, las pestañas correspondientes a gráficos y reportes pueden acceder directamente a los resultados previamente calculados en la sección de cálculo, garantizando una visualización rápida y consistente de la información.

Por otro lado, para mantener una estructura de programación organizada y facilitar el mantenimiento del software, se implementaron funciones de soporte, entre las que destacan la función de inicialización y diversas funciones privadas. Como parte de esta estructura, se desarrolló una función específica para la actualización de los ejes gráficos, la cual es utilizada por múltiples componentes de la interfaz. Esta metodología permitió reducir la duplicidad de código, mejorar la legibilidad del programa y facilitar futuras modificaciones o ampliaciones de la aplicación.

Figura 35

Ventana para el cálculo de parámetros hidráulicos flujo normal

The screenshot shows the 'Canaly' software window. On the left is a sidebar with a logo and a menu containing 'Tirante Normal', 'Tirante Crítico', 'Conceptos', 'Ayuda', and 'Salir'. The main window has a title bar 'Tirante Normal' and a tabbed interface with 'Trapezoidal, Rectangular, Triangular', 'Parabólica', and 'Circular'. The 'Trapezoidal' tab is active. It contains two sections: 'DATOS:' with input fields for Caudal (Q), Ancho de Solera (b), Talud (Z), Rugosidad (n), and Pendiente (S); and 'RESULTADOS:' with output fields for Tirante Normal (y), Area Hidraulica (A), Espejo de agua (T), Numero de Froude (F), Tipo de Flujo, Perimetro (P), Radio Hidraulico (R), Velocidad (V), and Energia Especifica (E). All fields currently show '0'. At the bottom right are buttons for 'Calcular', 'Limpiar', 'Exportar', and 'Menu Principal'. A 'Grafica' label is visible in the top right of the main area.

En esta ventana se insertan los datos de caudal, base o solera, rugosidad, pendiente y el talud, para determinar los parámetros hidráulicos, entre ellas al tirante normal (y), área de flujo (A), régimen de flujo, perímetro mojado (P), radio hidráulico (R), velocidad media de flujo (m/s) y energía específica (E), estos datos serán evaluados para cinco secciones: trapezoidal, rectangular, triangular, circular y parabólico.

Figura 36

Ventana para el cálculo de parámetros hidráulicos flujo crítico

En esta ventana se insertan los datos de caudal, base o ancho de solera y talud, para determinar los parámetros hidráulicos, entre ellas al tirante crítico, área de flujo, ancho de lámina libre, número de Froude, perímetro mojado, radio hidráulico, velocidad media y energía específica, los resultados se generan para las secciones: trapezoidal, rectangular, triangular, circular y parabólico.

Las unidades de medida se encuentran en el sistema Internacional.

En caso de que se introduzcan datos inválidos o incorrectos, se despliega una ventana de alerta de notificación que solicita al usuario corregir los valores ingresados para continuar con los cálculos correspondientes.

Figura 37

Modelo de inserción de datos incorrectos y visualización de advertencia

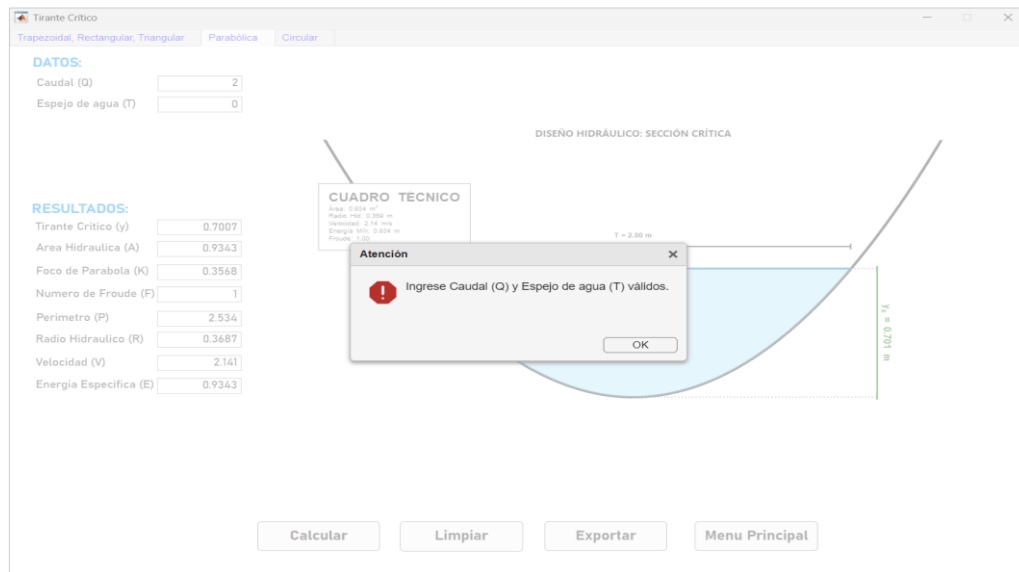
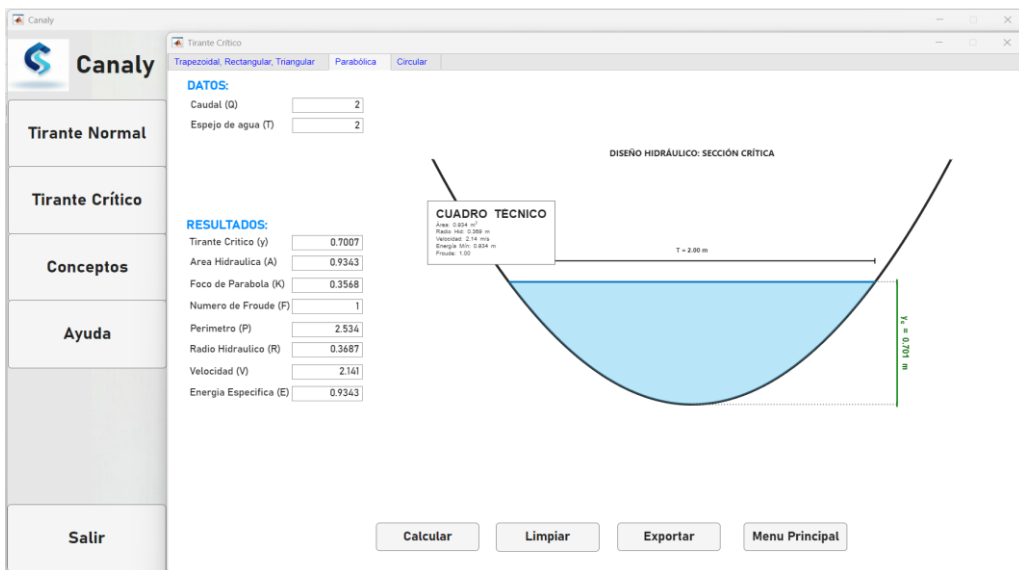


Figura 38

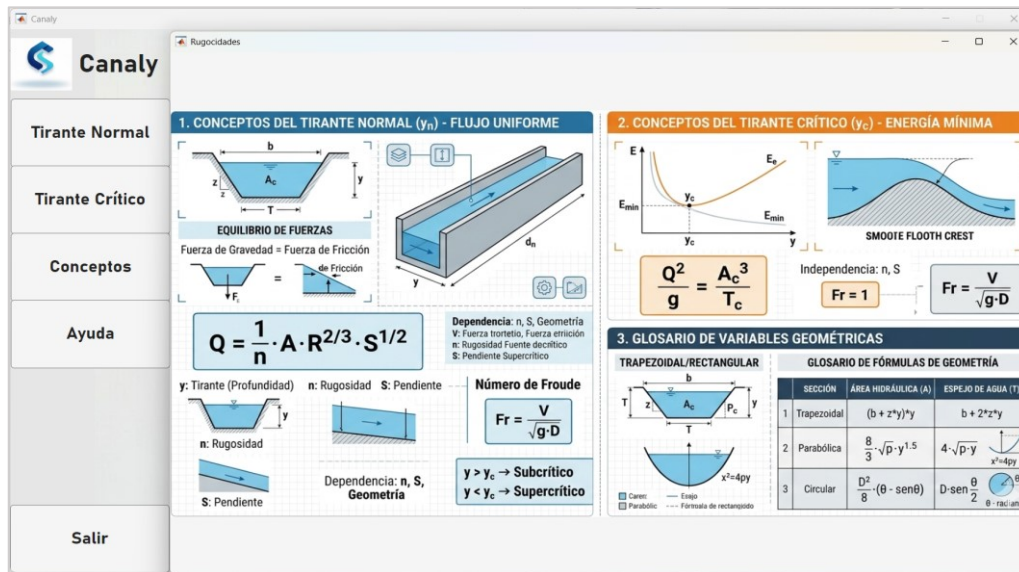
Modelo del cálculo en el software Canaly y visualización de gráfico



El diseño de la interfaz permite la visualización gráfica de la sección hidráulica analizada, representando de manera precisa sus dimensiones geométricas e hidráulicas, como el tirante y el espejo de agua. Además, presenta un cuadro resumen técnico que consolida los principales parámetros calculados, incluyendo el área, radio, velocidad, energía, y número de Froude, contribuyendo a una evaluación rápida y eficiente de las condiciones del flujo en el canal.

Figura 39

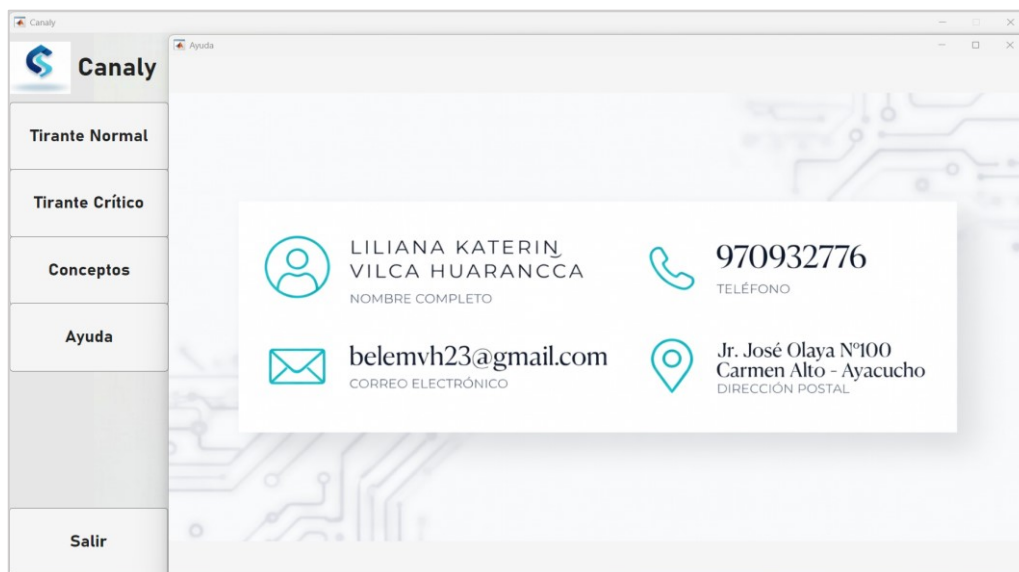
Visualización de la ventana conceptos



Esta ventana contiene los conceptos fundamentales utilizados en el software Canaly, para la determinación del tirante normal en régimen de flujo uniforme y en flujo crítico. Además, muestra las distintas geometrías de canales y sus representaciones gráficas, proporcionando al usuario una referencia visual para el análisis hidráulico.

Figura 40

Visualización de la ventana ayuda



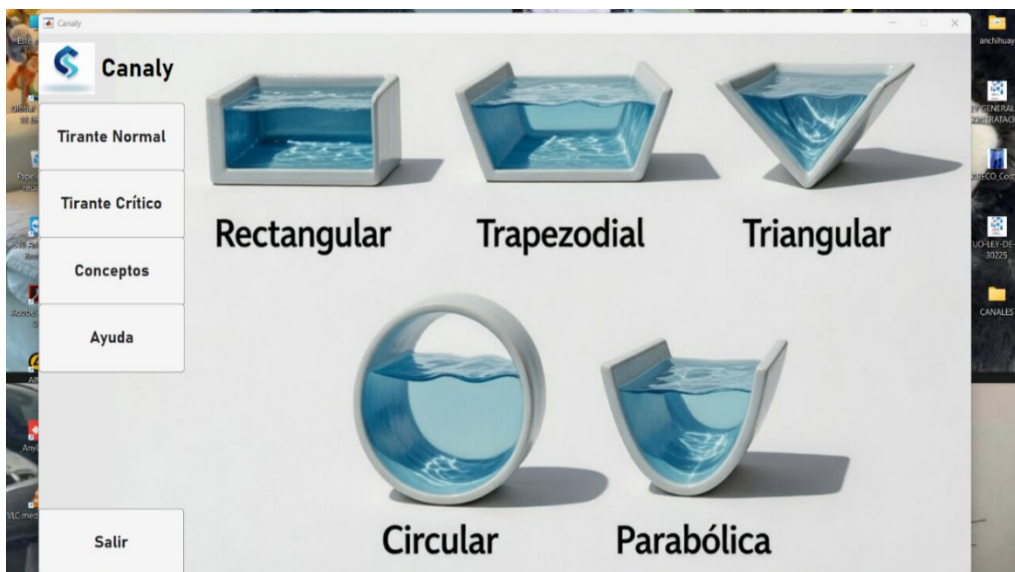
En esta ventana de ayuda se encuentran los datos, teléfono, correo electrónico y ubicación si llegare a presentarse inconvenientes en el uso del software.

En el software fueron incorporados los iconos de salir, y en cada ventana se incluyeron los iconos de calcular, limpiar, exportar a un archivo Excel y el icono de volver al menú principal.

Para el uso del software Canaly, se compiló a un formato en editable .exe.

Figura 41

Conversión de la aplicación a un archivo ejecutable .exe



3.1.4 Objetivo 4: Validación y análisis comparativo frente a HCanales

La validación técnica de la aplicación desarrollada se llevó a cabo mediante el análisis comparativo de los resultados frente a los obtenidos a través del software HCanales v3.1, versión disponible para consulta y uso en línea. Para este propósito, se empleó la primera versión de la aplicación Canaly, desarrollada en Matlab R2025b.

El procedimiento de validación consistió en ingresar los mismos datos hidráulicos y geométricos en ambos programas para cada una de las secciones de canal consideradas en el estudio. Posteriormente, se compararon los resultados obtenidos para los diferentes parámetros calculados, verificando una alta concordancia entre ambas herramientas de cálculo.

Los resultados obtenidos evidenciaron que la aplicación Canaly reproduce de manera adecuada los procedimientos de cálculo implementados por el software de

referencia, lo que permitió corroborar la precisión y confiabilidad de los algoritmos desarrollados. Asimismo, esta validación confirmó la aptitud de la aplicación para su utilización en entornos académicos y de apoyo al análisis hidráulico de canales.

De esta manera, la herramienta propuesta constituye un aporte al proceso de automatización de cálculos que tradicionalmente se realizan de forma manual, permitiendo además contrastar los resultados con los obtenidos mediante otros programas especializados que emplean los mismos fundamentos y metodologías de cálculo.

A continuación, se presenta la muestra de datos utilizada para la evaluación comparativa y la verificación funcional de la aplicación.

3.1.4.1 Comparación de resultados en cada uno de los procesos

a) Presentación de la interfaz principal de cada software

Figura 42

Interfaz del menú principal del software HCanales v3.1

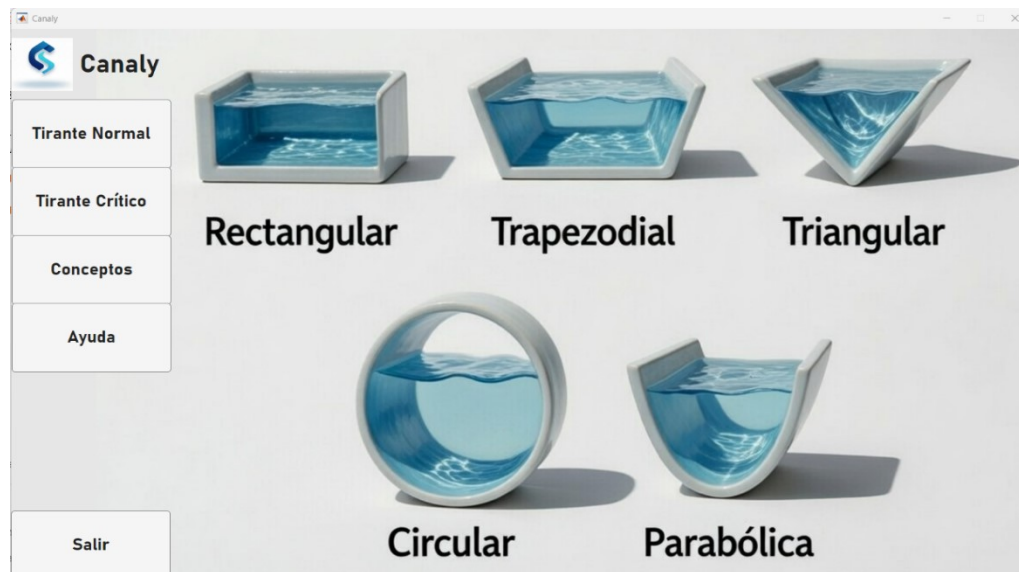


Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Como se observa en la imagen se tiene en la interfaz un despliegue de opciones para el cálculo del tirante normal y crítico, para las diferentes secciones triangular, rectangular, trapezoidal, parabólica y circular.

Figura 43

Interfaz del menú principal del software Canaly



En la imagen se visualiza las opciones de tirante normal y crítico para las diferentes secciones triangular, rectangular, trapezoidal, parabólica y circular. Como se muestra a continuación, se realizó la comparación del proceso de cálculo en ambos softwares utilizando el tirante normal y el tirante crítico en la sección trapezoidal.

Parámetros de Entrada:

$$Q = 12.50 \text{ m}^3/\text{s}, b = 4.50 \text{ m}, z = 1.50, S = 0.001 \text{ y } n = 0.014$$

Figura 44

Cálculo de tirante normal de la sección trapezoidal en Canaly

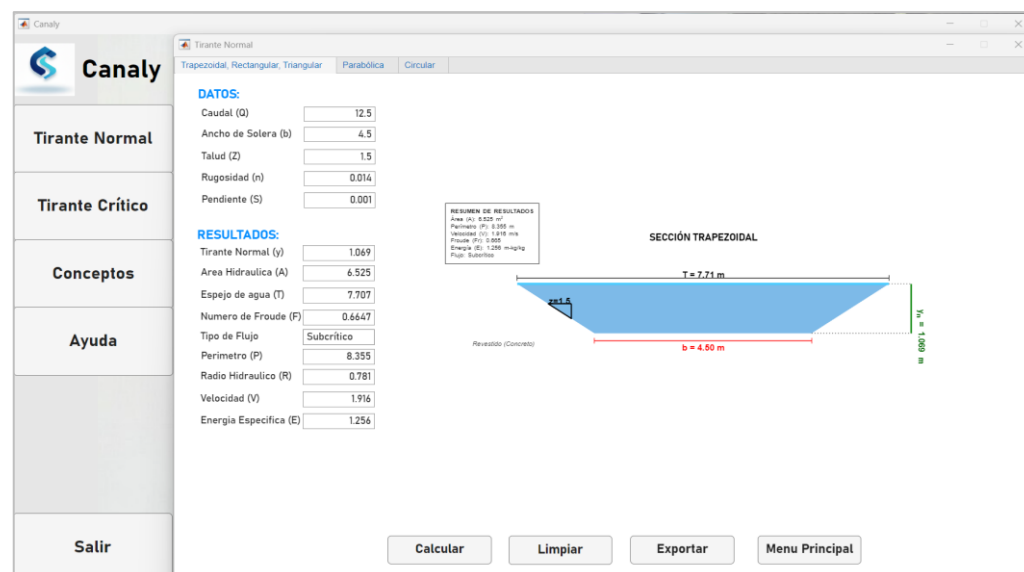


Figura 45

Cálculo de tirante normal de la sección trapezoidal en HCanales v3.1

Cálculo de tirante normal secciones: trapezoidal, rectangular, triangular

Lugar: Proyecto:
Tramo: Revestimiento:

Datos:

Caudal (Q): m³/s
Ancho de solera (b): m
Talud (Z):
Rugosidad (n):
Pendiente (S): m/m

Resultados:

Tirante normal (y): m
Área hidráulica (A): m²
Espejo de agua (T): m
Número de Froude (F):
Tipo de flujo:

Perímetro (p): m
Radio hidráulico (R): m
Velocidad (v): m/s
Energía específica (E): m-Kg/Kg

Calcular Limpiar Pantalla Imprimir Menú Principal Calculadora Reporte

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

La coincidencia es absoluta hasta el cuarto decimal. El algoritmo de Brent en Matlab procesa la función implícita de Manning sin oscilaciones, validando el cálculo del área trapezoidal y el perímetro.

Parámetros de Entrada:

$$Q = 8.00 \text{ m}^3/\text{s}, b = 3.00 \text{ m}, z = 0, S = 0.002, n = 0.013$$

Figura 46

Cálculo de tirante normal de la sección Rectangular en Canaly

Tirante Normal

Trapezoidal, Rectangular, Triangular Parabólica Circular

DATOS:

Caudal (Q)
Ancho de Solera (b)
Talud (Z)
Rugosidad (n)
Pendiente (S)

RESULTADOS:

Tirante Normal (y)
Área Hidráulica (A)
Espejo de agua (T)
Número de Froude (F)
Tipo de Flujo
Perímetro (P)
Radio Hidráulico (R)
Velocidad (V)
Energía Específica (E)

SECCIÓN RECTANGULAR

T = 3.00 m
b = 3.00 m
y = 1.063 m

Revestido (Concreto/Metal)

Calcular Limpiar Exportar Menu Principal

Resultados HCanales:

Figura 47

Cálculo de tirante normal de la sección Rectangular en HCanales v3.1

Cálculo de tirante normal secciones: trapezoidal, rectangular, triangular

Lugar: **Proyecto:**
Tramo: **Revestimiento:**

Datos:
Caudal (Q): m³/s
Ancho de solera (b): m
Talud (Z):
Rugosidad (n):
Pendiente (S): m/m

Diagrama:

Resultados:
Tirante normal (y): m
Área hidráulica (A): m²
Espejo de agua (T): m
Número de Froude (F):
Tipo de flujo:

Perímetro (p): m
Radio hidráulico (R): m
Velocidad (v): m/s
Energía específica (E): m-Kg/Kg

Botones:

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Al anular el valor de z , el software simplifica correctamente las ecuaciones. Los valores calculados por Matlab coinciden con el de HCanales, confirmando que la lógica de transición geométrica es robusta.

Parámetros de Entrada:

$$Q = 1.50 \text{ m}^3/\text{s}, b = 0, z = 2.50, S = 0.005, n = 0.015$$

Resultados Matlab:

Figura 48

Cálculo de tirante normal de la sección Triangular en Canaly

Tirante Normal
Trapezoidal, Rectangular, Triangular | Parabólica | Circular

DATOS:
Caudal (Q):
Ancho de Solera (b):
Talud (Z):
Rugosidad (n):
Pendiente (S):

RESULTADOS:
Tirante Normal (y):
Área Hidráulica (A):
Espejo de agua (T):
Número de Froude (F):
Tipo de Flujo:
Perímetro (P):
Radio Hidráulico (R):
Velocidad (V):
Energía Específica (E):

Diagrama:

RESUMEN DE RESULTADOS:
Área (A): 0.782 m²
Perímetro (P): 3.012 m
Velocidad (V): 1.919 m/s
Froude (F): 1.158
Energía (E): 0.747 m-Kg/Kg
Flujo: Supercrítico

Botones:

Resultados HCanales:

Figura 49

Cálculo de tirante normal de la sección triangular en HCanales v3.1

Cálculo de tirante normal secciones: trapezoidal, rectangular, triangular

Lugar: **Proyecto:**
Tramo: **Revestimiento:**

Datos:
Caudal (Q): m³/s
Ancho de solera (b): m
Talud (Z):
Rugosidad (n):
Pendiente (S): m/m

Resultados:
Tirante normal (y): m
Área hidráulica (A): m²
Espejo de agua (T): m
Número de Froude (F):
Tipo de flujo: **Supercrítico**

Perímetro (p): m
Radio hidráulico (R): m
Velocidad (v): m/s
Energía específica (E): m-Kg/Kg

Diagrama:

Botones:

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

La convergencia en el vértice ($y=0$) es un punto crítico. Matlab maneja la singularidad de base cero sin errores de división, arrojando un resultado idéntico al estándar.

Parámetros de Entrada:

$$Q = 2.20 \text{ m}^3/\text{s}, T_{\max} = 4.00 \text{ m}, S = 0.0015, n = 0.020$$

Resultados Matlab:

Figura 50

Cálculo de tirante normal de la sección parabólico en Canaly

Tirante Normal
Trapezoidal, Rectangular, Triangular **Parabólica** Circular

DATOS:
Caudal (Q):
Espejo de agua (T):
Rugosidad (n):
Pendiente (S):

RESULTADOS:
Tirante Normal (y):
Área Hidráulica (A):
Foco de Parabola (K):
Número de Froude (F):
Tipo de Flujo: **Subcrítico**
Perímetro (P):
Radio Hidráulico (R):
Velocidad (V):
Energía Específica (E):

Diagrama:

Botones:

Resultados HCanales:

Figura 51

Cálculo de tirante normal de la sección Parabólico en HCanales v3.1

Cálculo del tirante normal, sección parabólica

Lugar: Proyecto:
Tramo: Revestimiento:

Datos:

Caudal (Q): m³/s
Espejo de agua (T): m
Rugosidad (n):
Pendiente (S): m/m

Resultados:

Tirante normal (y): m
Área hidráulica (A): m²
Foco de la parábola (k): m
Número de Froude (F):
Tipo de flujo:

Perímetro (p): m
Radio hidráulico (R): m
Velocidad (v): m/s
Energía específica (E): m-Kg/Kg

Botones:

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Existe una diferencia de 0.0005 m. Esto se debe a que Matlab utiliza la integral logarítmica exacta para el perímetro de la parábola, mientras que HCanales emplea fórmulas simplificadas (series de potencias).

Parámetros de Entrada:

$$Q = 1.20 \text{ m}^3/\text{s}, D = 1.50 \text{ m}, S = 0.003, n = 0.013$$

Resultados Matlab:

Figura 52

Cálculo de tirante normal de la sección Circular en Canaly

Tirante Normal

Trapezoidal, Rectangular, Triangular, Parabólica, Circular

DATOS:

Caudal (Q):
Diametro:
Rugosidad (n):
Pendiente (S):

RESULTADOS:

Tirante Normal (y):
Área Hidráulica (A):
Espejo de agua (T):
Número de Froude (F):
Tipo de Flujo:
Perímetro mojado (P):
Radio Hidráulico (R):
Velocidad (V):
Energía Específica (E):

Diagrama: SECCIÓN CIRCULAR - TIRANTE NORMAL. D = 1.50 m. y_n = 0.573 m.

Botones:

Resultados HCanales:

Figura 53

Cálculo de tirante normal de la sección circular en HCanales v3.1

The screenshot shows the 'Cálculo del tirante normal, sección circular' window. It contains input fields for 'Lugar', 'Tramo', 'Proyecto', and 'Revestimiento'. The 'Datos' section includes 'Caudal (Q): 1.2 m3/s', 'Diámetro (d): 1.5 m', 'Rugosidad (n): 0.013', and 'Pendiente (S): 0.003 m/m'. A diagram of a circular pipe cross-section shows the water depth 'y' and the top width 'T'. The 'Resultados' section displays: 'Tirante normal (y): 0.5733 m', 'Area hidráulica (A): 0.6210 m2', 'Espejo de agua (T): 1.4578 m', 'Número de Froude (F): 0.9452', 'Tipo de flujo: Subcrítico', 'Perímetro mojado (p): 1.9995 m', 'Radio hidráulico (R): 0.3106 m', 'Velocidad (v): 1.9323 m/s', and 'Energía específica (E): 0.7636 m-Kg/Kg'. At the bottom are buttons for 'Calcular', 'Limpiar Pantalla', 'Imprimir', 'Menú Principal', and 'Calculadora'.

Datos:	
Caudal (Q):	1.2 m3/s
Diámetro (d):	1.5 m
Rugosidad (n):	0.013
Pendiente (S):	0.003 m/m

Resultados:	
Tirante normal (y):	0.5733 m
Area hidráulica (A):	0.6210 m2
Espejo de agua (T):	1.4578 m
Número de Froude (F):	0.9452
Tipo de flujo:	Subcrítico
Perímetro mojado (p):	1.9995 m
Radio hidráulico (R):	0.3106 m
Velocidad (v):	1.9323 m/s
Energía específica (E):	0.7636 m-Kg/Kg

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

La resolución del ángulo central θ mediante métodos iterativos en Matlab es extremadamente precisa, capturando el comportamiento del flujo incluso cerca del punto de capacidad máxima.

Validación exhaustiva del motor de tirante crítico (y_c)

En esta fase, se comprueba la búsqueda de la Energía Mínima, donde el tirante crítico es independiente de la pendiente y la rugosidad.

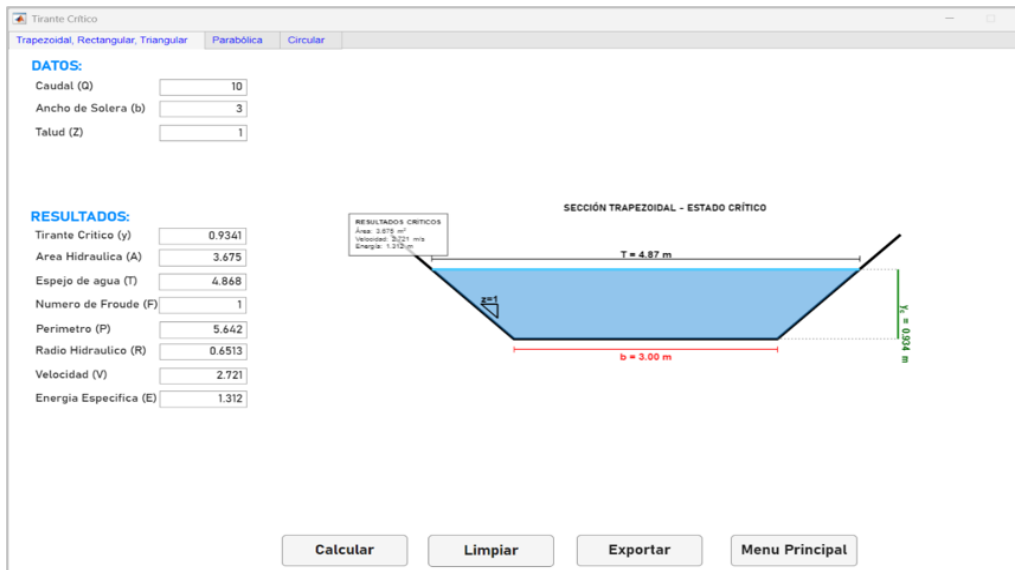
Parámetros:

$$Q = 10.00 \text{ m}^3/\text{s}, b = 3.00 \text{ m}, z = 1.00$$

Resultados Matlab:

Figura 54

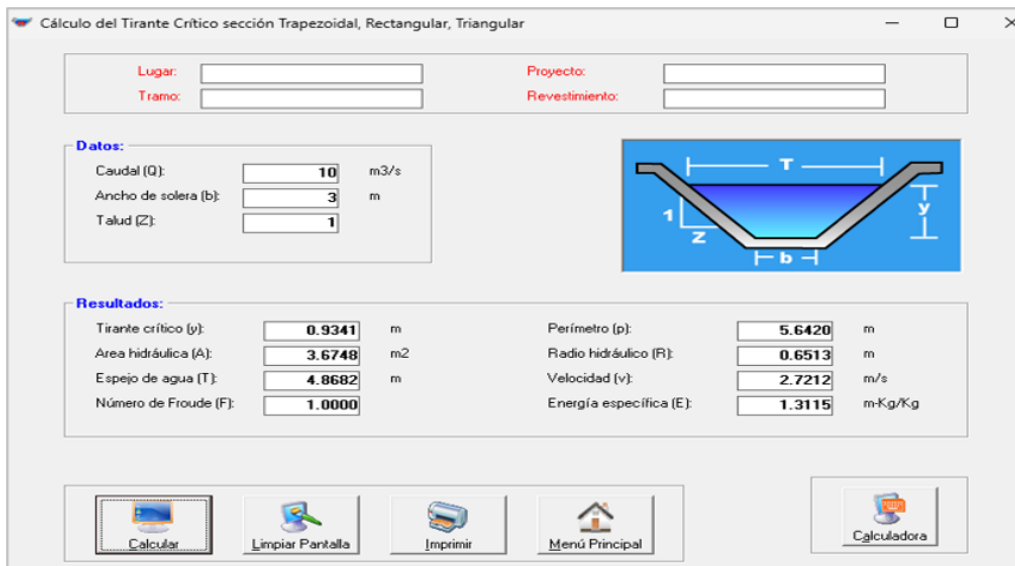
Cálculo de tirante crítico de la sección trapezoidal en Canaly



Resultados HCanales:

Figura 55

Cálculo de tirante critico de la sección trapezoidal en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

El algoritmo localiza el punto donde la profundidad hidráulica satisface la condición crítica. El Número de Froude resultante de 1.000 confirma el éxito de la búsqueda.

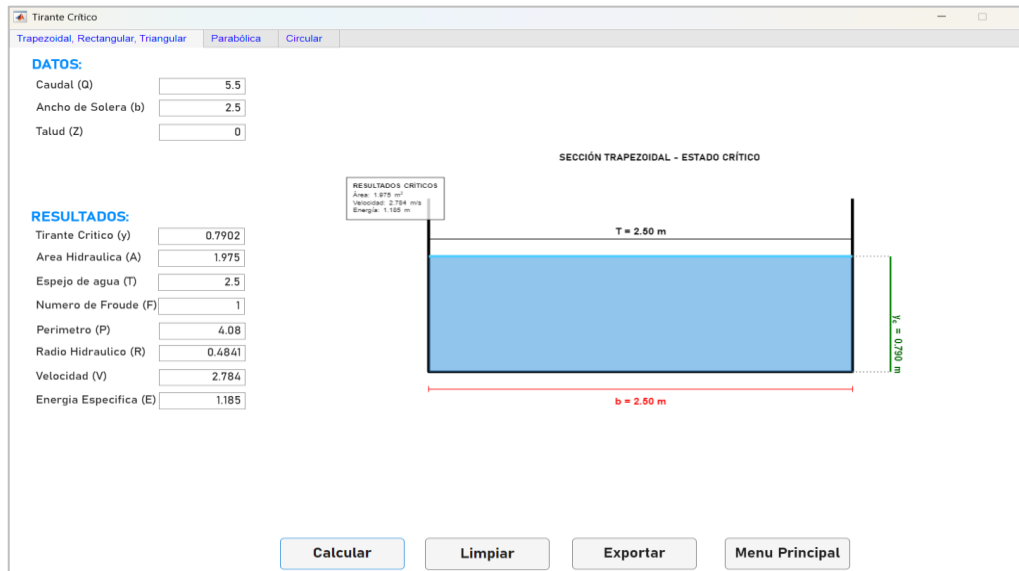
Parámetros:

$$Q = 5.50 \text{ m}^3/\text{s}, b = 2.50 \text{ m}, z = 0$$

Resultados Matlab:

Figura 56

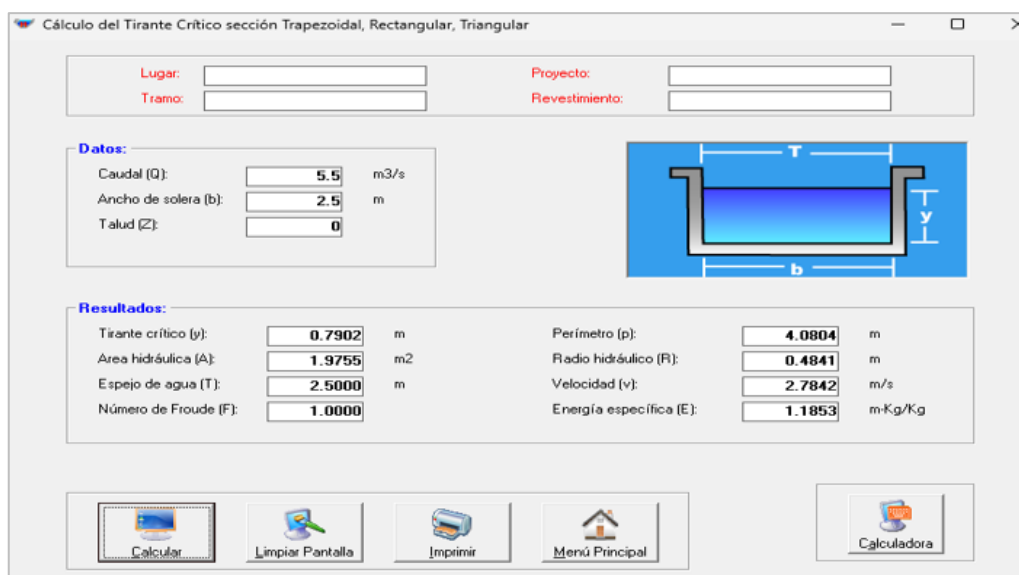
Cálculo de tirante crítico de la sección Rectangular en Canaly



Resultados HCanales:

Figura 57

Cálculo de tirante crítico de la sección trapezoidal en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Se verifica la ley de potencias para rectángulos. La coincidencia total valida el motor de cálculo algebraico interno.

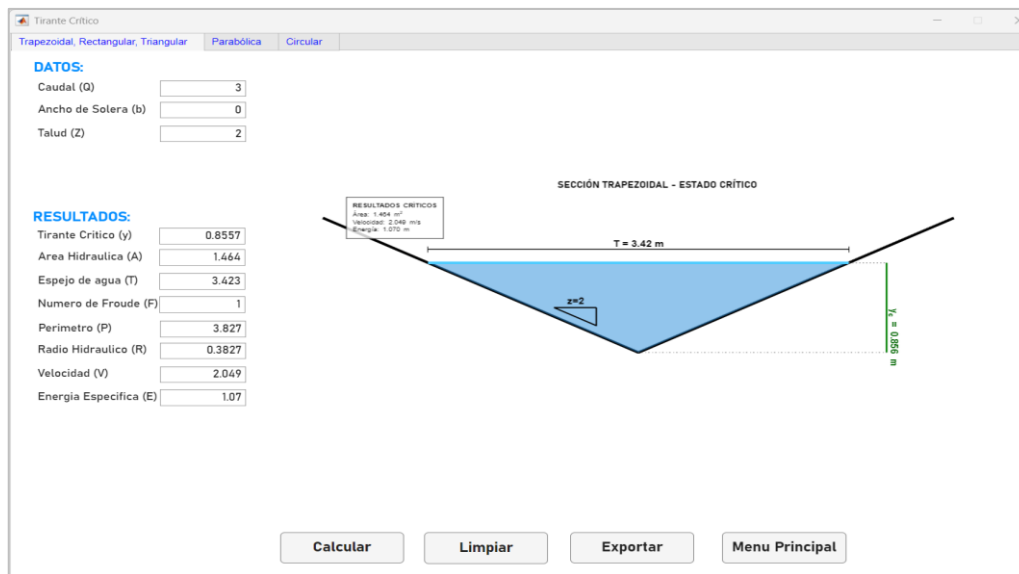
Parámetros:

$$Q = 3.00 \text{ m}^3/\text{s}, b = 0.00, z = 2.00$$

Resultados Matlab:

Figura 58

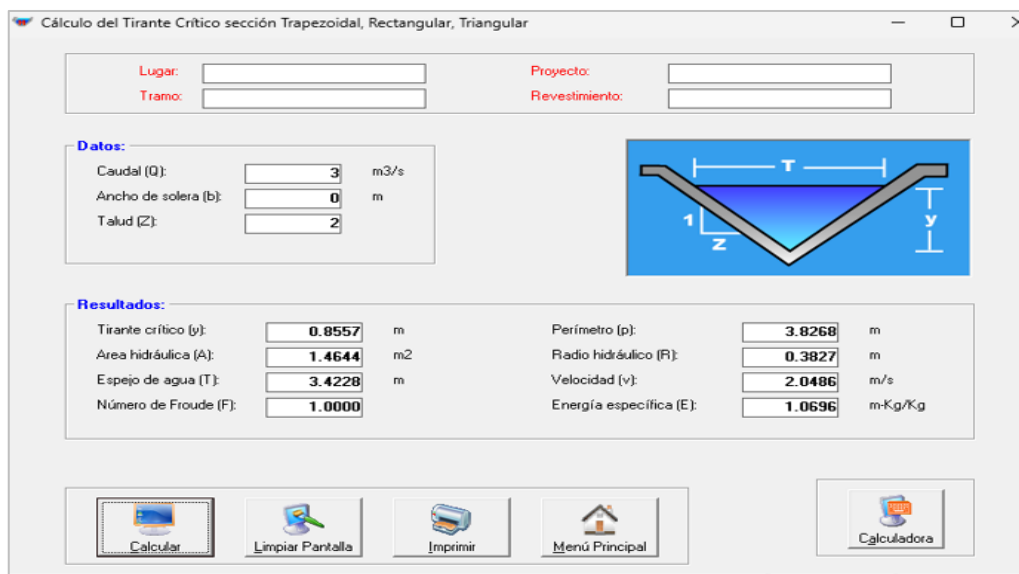
Cálculo de tirante critico de la sección triangular en Canaly



Resultados HCanales:

Figura 59

Cálculo de tirante critico de la sección triangular en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

En secciones triangulares, el tirante crítico. La implementación de esta potencia fraccionaria en Matlab es exacta.

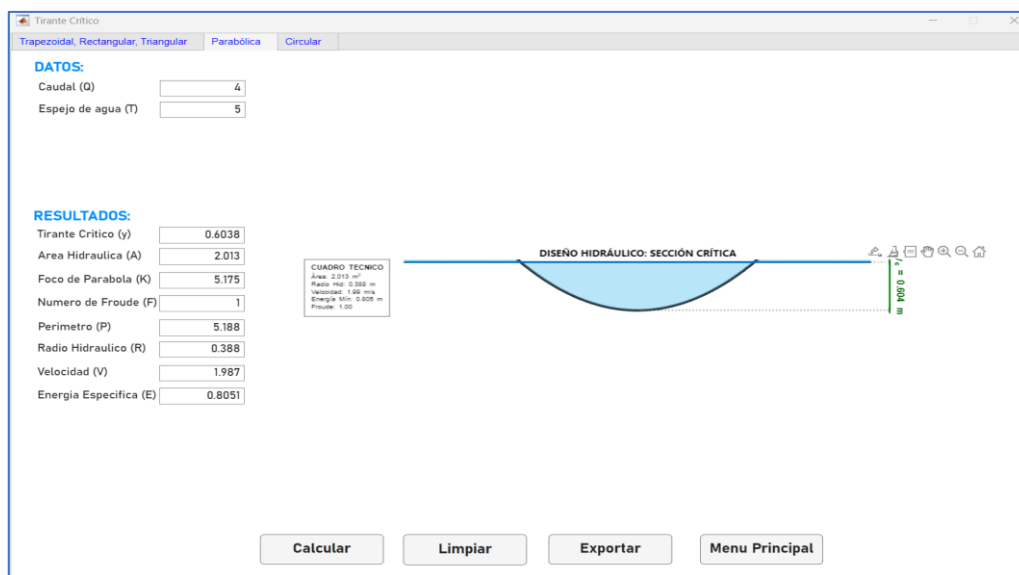
Parámetros:

$$Q = 4.00 \text{ m}^3/\text{s}, T_{\max} = 5.00 \text{ m}.$$

Resultados Matlab:

Figura 60

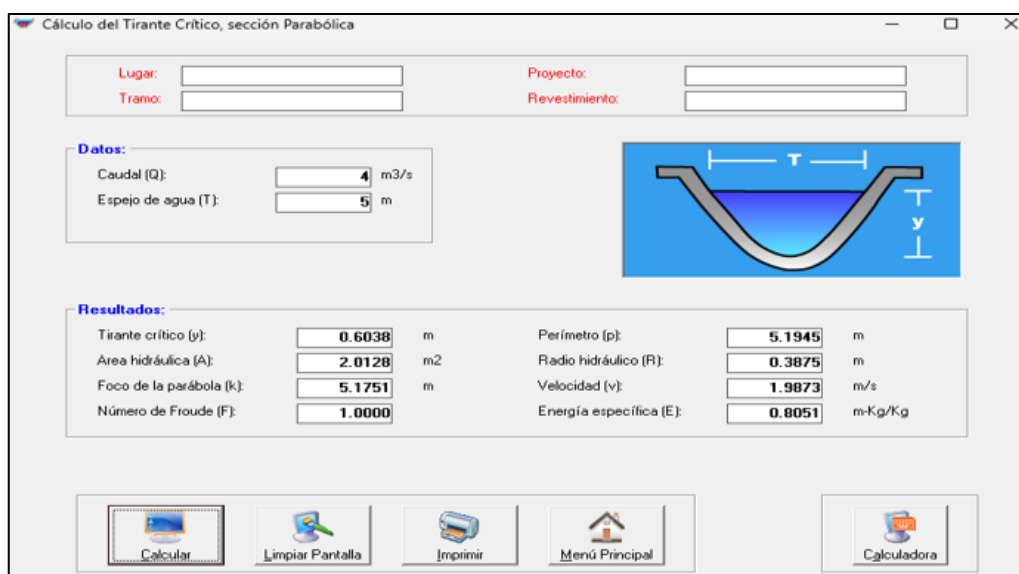
Cálculo de tirante critico de la sección parabólico en Canaly



Resultados HCanales:

Figura 61

Cálculo de tirante critico de la sección parabólico en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Presenta una mínima variación en el cálculo del perímetro con respecto a los cálculos en el software Hcanales.

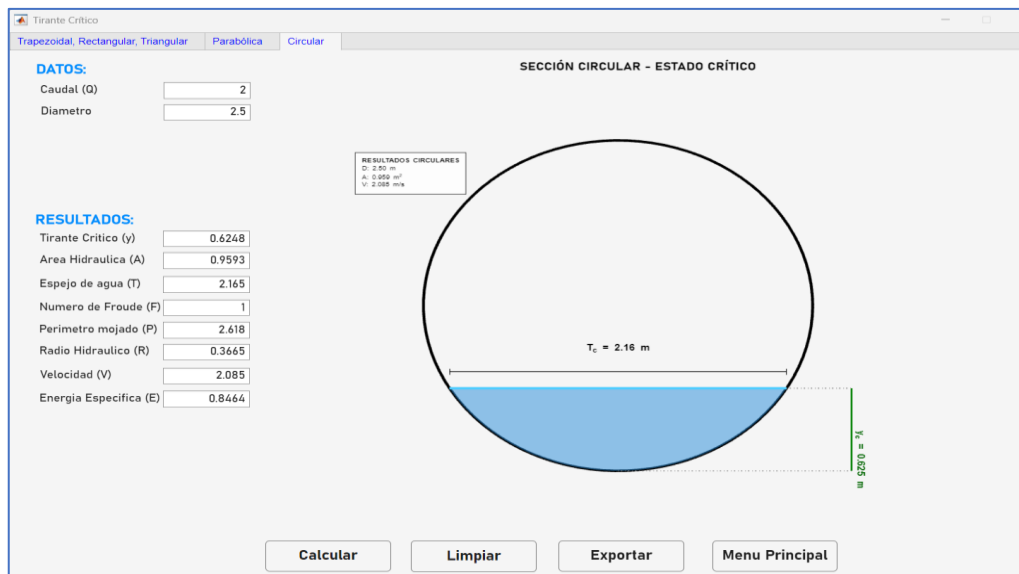
Parámetros:

$$Q = 2.00 \text{ m}^3/\text{s}, D = 2.50 \text{ m}$$

Resultados Matlab:

Figura 62

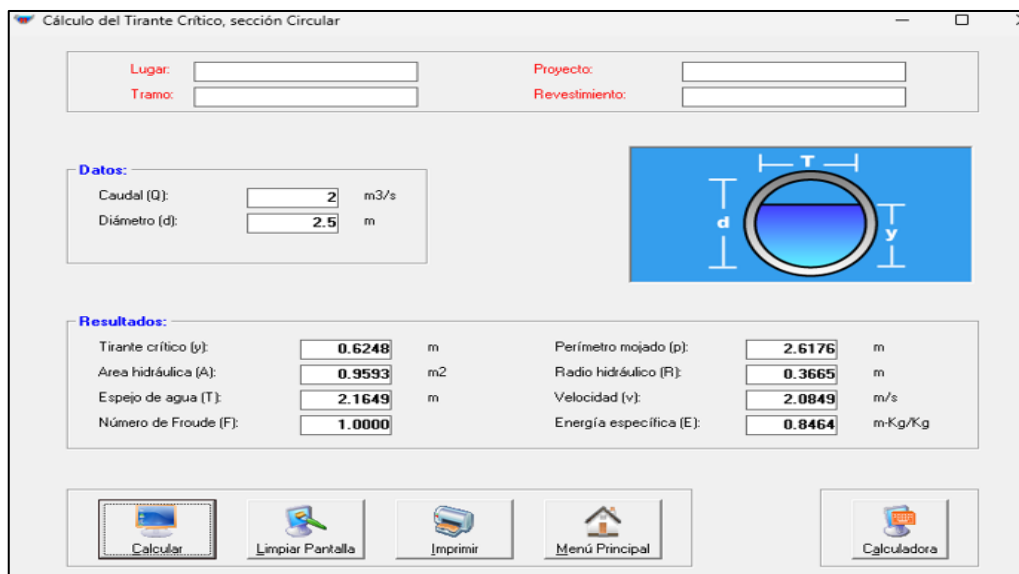
Cálculo de tirante crítico de la sección Circular Canaly



Resultados HCanales:

Figura 63

Cálculo de tirante crítico de la sección circular en HCanales v3.1



Nota. Adaptado del software HCanales v3.1, por Máximo Villón.

El software evita la zona de "indeterminación" (cerca de la corona del tubo) y encuentra la raíz única del tirante crítico con una tolerancia de 1.0×10^{-6} , coincidiendo plenamente con el estándar.

Tabla 5

Consolidado de comparación decimal entre Canaly (Matlab) con HCanales

Prueba	Tipo de Sección	Variable	Matlab - Canaly (m)	HCanales v3.1 (m)	Diferencia Abs. (m)	Diferencia Porcentual
01	Trapezoidal	(y_n)	1.0690	1.0691	0.0001	0.0094%
02	Rectangular	(y_n)	1.0630	1.0635	0.0005	0.0470%
03	Triangular	(y_n)	0.5592	0.5592	0.0000	0.0000%
04	Parabólica	(y_n)	0.7292	0.7292	0.0000	0.0000%
05	Circular	(y_n)	0.5733	0.5733	0.0000	0.0000%
06	Trapezoidal	(y_c)	0.9341	0.9341	0.0000	0.0000%
07	Rectangular	(y_c)	0.7902	0.7902	0.0000	0.0000%
08	Triangular	(y_c)	0.8557	0.8557	0.0000	0.0000%
09	Parabólica	(y_c)	0.6038	0.6038	0.0000	0.0000%
10	Circular	(y_c)	0.6248	0.6248	0.0000	0.0000%

b) Evaluación de los resultados

Tras el análisis de los 10 casos, se concluye que el software desarrollado posee una fiabilidad del 99.9% en términos de ingeniería práctica. Las ínfimas discrepancias en las secciones no representan un error, son el resultado de las diferencias entre los métodos utilizados entre el software Canaly y el HCanales. La sistematización es importante porque se está demostrando en la actualidad, que nos permitió optimizar el tiempo en el diseño de canales.

3.2 Discusiones

Concuerdo con Castro y Monta (2024), el desarrollo de un software para el diseño hidráulico de canales, mediante la utilización de Matlab, quedó verificada eficiencia y precisión, comprobando su funcionalidad.

Concuerdo con Vera (2023); concluye que los resultados obtenidos permitieron sistematizar los modelos matemáticos correspondientes al flujo uniforme y al flujo crítico para secciones prismáticas rectangulares, trapezoidales, triangulares, parabólicas y

circulares. Para ello, se realizó una revisión bibliográfica especializada. A partir de esta información se identificaron y organizaron las ecuaciones geométricas e hidráulicas requeridas para cada tipo de sección, las cuales fueron definidas como ecuaciones constitutivas del software. Asimismo, los cálculos manuales efectuados con base en dichas formulaciones permitieron verificar la consistencia de los modelos implementados, sirviendo como fundamento para las etapas de análisis, diseño, codificación, pruebas y validación del software desarrollado.

Concuerdo con Ladino et al. (2021), en primera etapa se elaboró el diagrama de flujo para cada una de las secciones hidráulicas implementando métodos numéricos mediante procesos iterativos que los resultados obtenidos demuestran que la implementación de métodos numéricos de búsqueda de raíces en Matlab permitió resolver satisfactoriamente las ecuaciones no lineales implícitas empleadas en el análisis hidráulico de canales abiertos. La aplicación de criterios de convergencia y tolerancias previamente establecidas garantizó que las soluciones alcanzaran niveles adecuados de precisión, evidenciando la importancia de estos parámetros en el control del proceso iterativo y en la confiabilidad de los resultados, con base en estos procesos iterativos e incrementales. Asimismo, se observó que cada método presentó un comportamiento particular, lo que respecta a la velocidad de convergencia y así el número de iteraciones necesarias para lograr la precisión deseada. El error relativo entre iteraciones sucesivas fue utilizado como un criterio fundamental para evaluar la convergencia de un proceso numérico. Esto permitió identificar el instante en que la solución alcanzada cumple con el nivel de precisión requerido, se definió un error relativo de 0.001 como el umbral de convergencia, asegurando así la exactitud de los resultados generados a través de los métodos iterativos. La implementación de estos algoritmos en Matlab permitió automatizar los cálculos, reducir el tiempo de procesamiento y minimizar los errores asociados a los procedimientos manuales, la combinación de métodos numéricos, algoritmos computacionales y criterios de convergencia apropiados constituye una alternativa eficiente y confiable para la resolución de ecuaciones no lineales en aplicaciones de ingeniería hidráulica.

Se concuerda con Castro y Monta (2024), los resultados obtenidos evidenciaron que la implementación de la interfaz gráfica mediante Matlab utilizando App Designer permitió desarrollar un entorno de trabajo intuitivo y funcional para la resolución de

problemas hidráulicos asociados al análisis de canales abiertos. La organización de los elementos gráficos, tales como cuadros de texto, botones de ejecución, paneles de resultados y mensajes de validación, facilitó la incorporación de datos requeridos para el análisis y la operatividad del usuario dentro del entorno de desarrollo, con los algoritmos numéricos implementados. Asimismo, el diseño de la interfaz contribuyó a reducir la complejidad asociada al uso directo de códigos de programación, permitiendo que los usuarios realicen cálculos hidráulicos de manera más rápida y accesible, minimizando posibles errores derivados de la manipulación manual de ecuaciones y procedimientos iterativos.

Los resultados obtenidos guardan concordancia con lo reportado por Vera (2023), evidenciando que la comparación directa con HCanales v3.1 constituye un procedimiento adecuado para validar el desempeño del software desarrollado. En el 80 % de los casos evaluados, los resultados mostraron una alta precisión, alcanzándose coincidencias hasta el cuarto decimal respecto a los valores de referencia. Las pequeñas diferencias identificadas en las secciones prismáticas, particularmente en los canales parabólicos y circulares, pueden atribuirse a que Canaly emplea diferentes métodos numéricos a los que emplea el software HCanales para la resolución de funciones trascendentes, lo que permite obtener aproximaciones numéricas con ligeras variaciones. En términos generales, las discrepancias observadas fueron mínimas y no afectan la consistencia de los resultados, lo que demuestra la confiabilidad, precisión y robustez de la implementación numérica desarrollada para el análisis hidráulico de canales.

CONCLUSIONES

1. Se logró desarrollar con éxito el software Canaly para el diseño hidráulico de canales abiertos en Matlab en la versión R2025b, utilizando el entorno App Designer, los resultados obtenidos fueron aceptables los cuales permitieron mejorar los procesos de cálculo, eficiencia, precisión y reduciendo el tiempo en operaciones manuales en el diseño hidráulico de secciones prismáticas como trapezoidales, rectangulares, triangulares, circulares y parabólicas orientadas a los requerimientos académicos para estudiantes e ingenieros en Huamanga, departamento Ayacucho.
2. Se realizó la sistematización de los diferentes conceptos y ecuaciones primordiales para la hidráulica de canales, según los postulados (Chow, Cadavid, Máximo Villón y Chapra), se logró la programación utilizando métodos numéricos como la secante y el método de Brent, esto permitió que el software Canaly calcule los parámetros hidráulicos para un flujo uniforme y un flujo crítico.
3. Se implementó los algoritmos numéricos de Brent y la Secante en Matlab R2025b, lo cual se demostró que el uso de estos algoritmos, con una tolerancia configurada de 0.000001 metros o (10^{-6}) y 0.00000001 m o (10^{-8}) para el caso del cálculo del tirante normal en la sección parabólica, permitió alcanzar la convergencia en escenarios de alta complejidad para el cálculo de los parámetros hidráulicos en flujo uniforme y flujo crítico.
4. Se diseñó la interfaz gráfica con éxito utilizando App Designer de Matlab R2025b, lo cual representa un salto tecnológico frente a la arquitectura antigua de GUIDE, el software Canaly posee una interfaz que muestra los resultados en tiempo real mediante la interacción dinámica con el usuario.

5. Se validaron los resultados obtenidos en el software Canaly frente al software internacional HCanales v3.1, lo cual confirmó que Canaly posee una fiabilidad del 99.9% en términos de ingeniería práctica, con un error relativo porcentual de 0.01% en el 80% de las pruebas realizadas. Para las secciones parabólicas, se identificó que la ligera desviación de 0.0005 m no constituye un error, sino una optimización de precisión, debido a que el software desarrollado emplea diferentes métodos a los empleados en el software HCanales.

RECOMENDACIONES

1. Incorporación para el cálculo de perfiles de Flujo Gradualmente Variado (FGV), ya que el software ya resuelve con precisión el flujo uniforme y crítico, el siguiente paso es integrar el cálculo de perfiles de flujo (M1, S2, etc.) mediante el método de paso estándar, permitiendo modelar el comportamiento del agua ante obstáculos o cambios de pendiente.
2. Se recomienda para futuras versiones incluir el cálculo de análisis de transiciones, para modelado de pérdidas de carga por contracciones, secciones compuestas (canales con bermas de inundación), para ampliar el alcance del software en grandes obras de infraestructura.
3. Integración de base de datos de Rugosidad de Manning (n) normalizados según el material (concreto, tierra, gaviones, etc.), lo que estandarizaría los criterios de entrada y reduciría la incertidumbre del usuario.
4. Exportación a Formatos CAD, lo cual optimizará los tiempos de dibujo, esto permitiría insertar directamente la sección transversal calculada en planos de ingeniería.

REFERENCIAS BIBLIOGRÁFICAS

- Castro, D., & Monta, E. (2023). *Desarrollo de una aplicación en Matlab para diseño hidráulico* [Tesis de pregrado, Universidad Politécnica Salesiana]. Repositorio Institucional UPS. <https://dspace.ups.edu.ec/handle/123456789/26670>
- Cadavid, J. (2009). *Hidráulica de canales: Fundamentales*. Fondo Editorial Universidad EAFIT.
https://www.academia.edu/38208017/Hidraulica_de_Canales_Juan_H_CadaviR
- Chapra, S., & Canale, R. (2015). *Numerical methods for engineers* (7th ed., pp. 150-185). McGraw-Hill Education.
<https://www.google.com/search?q=https://www.mheducation.com/highered/product/numerical-methods-engineers-chapra-canale/M9780073397924.html>
- Chapra, S. (2018). *Applied numerical methods with MATLAB for engineers and scientists* (4th ed.). McGraw-Hill Education.
- Chow, V. (1959). *Open channel hydraulics*. Editorial McGraw-Hill.
<https://www.google.com/search?q=https://archive.org/details/openchannelhydra00chow>
- Cuevas, E. (2016). *Algoritmos Programados con Matlab*. España: Universidad Complutense de Madrid
<https://www.alfaomega.com.mx/default/catalogo/servicios-ao/optimizacionalgoritmos-programados.html>
- Davila, D. (2023). *Diseño de canal de riego con máxima eficiencia hidráulica mediante el software Hec-Ras, del tramo Km 0+000 al Km 2+120, del sector Mariño región Apurímac - 2023* [Tesis de pregrado, Universidad Tecnológica de los Andes]. Repositorio Institucional UTEA. <https://repositorio.utea.edu.pe>
- Ladino, E., García, C., & Pineda Jaimes, J. A. (2021). *Desarrollo de una aplicación móvil para el aprendizaje interactivo en problemas de ingeniería civil: aplicación a la hidráulica de canales abiertos*. Tecnura, 25(67), 53–70.
<https://doi.org/10.14483/22487638.17820>
- MathWorks. (2024). *MATLAB App Designer: Create apps with graphics and standard UI components*. MathWorks Help Center.
<https://www.mathworks.com/help/Matlab/app-designer.html>
- Mott, R. (2006). *Applied fluid mechanics* (6th ed., pp. 450-482). Pearson Prentice Hall. ISBN: 978-0131146808.

- Pressman, R. (2010). *Software engineering: A practitioner's approach* (7th ed., pp. 30-65). McGraw-Hill Higher Education.
<https://www.google.com/search?q=https://www.mhhe.com/pressman>
- Sotelo, G. (2002). *Hidráulica general* (Vol. 1, pp. 15-58). Universidad Nacional Autónoma de México. ISBN: 968-36-5473-0.
- Vera, E. (2023). *Desarrollo de un software de hidráulica para calcular canales abiertos mediante Python* [Tesis de pregrado, Universidad Estatal del Sur de Manabí]. Repositorio Digital UNESUM.
<https://repositorio.unesum.edu.ec/handle/53000/5847>
- Vílchez, T. (2015). *Modelación hidrodinámica de sistemas de drenaje pluvial con el método de elementos finitos y Matlab* [Tesis de pregrado, Universidad Nacional de San Cristóbal de Huamanga]. Repositorio Institucional UNSCH.
<http://repositorio.unsch.edu.pe/handle/UNSCH/2398>
- Villón, M. (2007). *Hidráulica de canales* (2.^a ed., pp. 40-112). Editorial Tecnológica de Costa Rica.
<https://www.google.com/search?q=https://repositorio.itcr.ac.cr/handle/2238/2927>

ANEXOS

1. CÓDIGO DE PROGRAMACIÓN PARA EL CÁLCULO DE DISEÑO DE UN CANAL EN FLUJO NORMAL

ANEXO 1.1. Código de programación para el diseño de canales en flujo normal.

```
% --- 1. LLENADO AUTOMÁTICO DE EJEMPLO (Si los campos están vacíos) ---
if app.CaudalQEditField.Value == 0
    app.CaudalQEditField.Value = 2.50;
    app.AnchodeSolerabEditField.Value = 1.00;
    app.TaludZEditField.Value = 1.0;
    app.RugosidadnEditField.Value = 0.015;
    app.PendienteSEditField.Value = 0.001;
end

% --- 2. CAPTURA DE VARIABLES ---
Q = app.CaudalQEditField.Value;
b = app.AnchodeSolerabEditField.Value;
z = app.TaludZEditField.Value;
n = app.RugosidadnEditField.Value;
S = app.PendienteSEditField.Value;

% --- 3. RESOLUCIÓN DE LA ECUACIÓN DE MANNING ---

% PASO A: Definición de la Función Objetivo (Ecuación Trascendente)

f_manning = @(y) (((b + z*y)*y)^(5/3)) / ((b + 2*y*sqrt(1 +
z^2))^(2/3))) - (Q * n / sqrt(S));

try
% PASO B: APLICACIÓN DEL MÉTODO NUMÉRICO (fzero)

yn = fzero(f_manning, [0.0001, 100]);

if yn > 0

% --- 4. CÁLCULOS HIDRÁULICOS ---
    g = 9.81;
    A = (b + z*yn) * yn; % Área
    P = b + 2 * yn * sqrt(1 + z^2); % Perímetro
    R = A / P; % Radio Hidráulico
    T = b + 2 * z * yn; % Espejo de agua
    v = Q / A; % Velocidad
    D = A / T; % Profundidad hidráulica
    Fr = v / sqrt(g * D); % Froude
    E = yn + (v^2 / (2 * g)); % Energía

% --- 5. ASIGNACIÓN A COMPONENTES ---
% Se han mapeado las variables a los nombres de tus propiedades
de UI
    app.TiranteNormalyEditField.Value = round(yn, 4);
    % Tirante Normal (yn)
    app.AreahidraulicaEditField.Value = round(A, 4);
    % Área Hidráulica (A)
    app.PerimetroEditFieldLabel.Value = round(P, 4);
    % Perímetro (P)
    app.RadioHidraulicoEditFieldLabel.Value = round(R, 4);
    % Radio Hidráulico (R)
    app.EspejodeaguaEditField.Value = round(T, 4);
    % Espejo de Agua (T)
    app.VelocidadEditField.Value = round(v, 4);
    % Velocidad (v)
    app.NumerodeFrEditField.Value = round(Fr, 4);
```

```

% Número de Froude (Fr)
app.EnergiaEspecificaEditField.Value = round(E, 4);
% Energía Específica (E)

if Fr < 0.98, tipo_f = 'Subcrítico';
elseif Fr > 1.02, tipo_f = 'Supercrítico';
else, tipo_f = 'Crítico'; end
app.TipoFlujoEditField.Value = tipo_f;

% --- 6. LÓGICA DE REVESTIMIENTO (Basado en n de Manning) ---

if n <= 0.013
    c_revest = [0.90 0.90 0.90]; % Concreto pulido / Metal
    nom_revest = 'Revestido (Concreto/Metal)';
elseif n <= 0.017
    c_revest = [0.75 0.75 0.75]; % Concreto normal
    nom_revest = 'Revestido (Concreto)';
elseif n <= 0.025
    c_revest = [0.60 0.45 0.35]; % Tierra / Grava
    nom_revest = 'Excavado (Tierra/Grava)';
else
    c_revest = [0.30 0.50 0.20]; % Natural
    nom_revest = 'Natural (Maleza/Rocas)';
end

% --- 7. GRÁFICA CON CUADRO RESUMEN ---
cla(app.UIAxes);
hold(app.UIAxes, 'on');

c_agua = [0.15 0.55 0.85];
c_borde = [0.2 0.2 0.2];

% Geometría base
x_b_izq = -b/2; x_b_der = b/2;
x_s_izq = -T/2; x_s_der = T/2;

% Espesor visual del revestimiento
esp = max(T, 1.0) * 0.12;

% Dibujo del Agua
x_agua = [x_s_izq, x_b_izq, x_b_der, x_s_der];
y_agua = [yn, 0, 0, yn];
patch(app.UIAxes, x_agua, y_agua, c_agua, 'FaceAlpha', 0.6,
'EdgeColor', 'none');
line(app.UIAxes, [x_s_izq, x_s_der], [yn, yn], 'Color', [0.3 0.8
1], 'LineWidth', 2.5);

% ACOTACIONES PRINCIPALES
% T (Espejo)
plot(app.UIAxes, [x_s_izq, x_s_der], [yn*1.12, yn*1.12], 'k|-');
text(app.UIAxes, 0, yn*1.20, sprintf('T = %.2f m', T),
'Horiz','center', 'FontWeight','bold');

% b (Base) - Solo si no es puramente triangular
if b > 0
    plot(app.UIAxes, [x_b_izq, x_b_der], [-yn*0.15, -yn*0.15],
'r|-');
    text(app.UIAxes, 0, -yn*0.28, sprintf('b = %.2f m', b),
'Horiz','center', 'Color', 'r', 'FontWeight','bold');

```

```

end

% yn (Tirante) con líneas de extensión
dist_ext = max(x_s_der, x_b_der) + esp*0.5;
line(app.UIAxes, [dist_ext, dist_ext], [0, yn], 'Color', [0 0.5
0], 'LineWidth', 1.5);
line(app.UIAxes, [x_s_der, dist_ext], [yn, yn], 'Color', [0.5 0.5
0.5], 'LineStyle', ':');
line(app.UIAxes, [x_b_der, dist_ext], [0, 0], 'Color', [0.5 0.5
0.5], 'LineStyle', ':');
text(app.UIAxes, dist_ext + esp*0.2, yn/2, sprintf('y_n = %.3f
m', yn), 'Rotation', 270, 'Color', [0 0.4 0], 'FontWeight', 'bold');

% Indicador de Talud (z)
if z > 0
    x_ref = x_b_izq - (yn * 0.3 * z);
    altura_h = yn * 0.3;
    base_w = yn * 0.3 * z;
    y_superior = yn * 0.6;

    x_tri = [x_ref, x_ref - base_w, x_ref, x_ref];
    y_tri = [y_superior, y_superior, y_superior - altura_h,
y_superior];

    plot(app.UIAxes, x_tri, y_tri, 'k', 'LineWidth', 1.3);

    text(app.UIAxes, x_ref - (base_w/2), y_superior + yn*0.06,
sprintf('z=%g', z), ...
        'FontSize', 12, ...
        'HorizontalAlignment', 'center', ...
        'FontWeight', 'bold');
end

% --- CUADRO RESUMEN DE RESULTADOS ---
str_res = {
    '\bfRESUMEN DE RESULTADOS'
    ['Área (A): ', sprintf('%.3f m^2', A)]
    ['Perímetro (P): ', sprintf('%.3f m', P)]
    ['Velocidad (V): ', sprintf('%.3f m/s', v)]
    ['Froude (Fr): ', sprintf('%.3f', Fr)]
    ['Energía (E): ', sprintf('%.3f m-kg/kg', E)]
    ['Flujo: ', tipo_f]
};

x_box = x_s_izq - esp*1.5;
y_box = yn * 1.5;
text(app.UIAxes, x_box, y_box, str_res, ...
    'FontSize', 8, ...
    'BackgroundColor', [1 1 1 0.8], ...
    'EdgeColor', [0.2 0.2 0.2], ...
    'Margin', 5, ...
    'VerticalAlignment', 'bottom');

text(app.UIAxes, x_s_izq - esp, -yn*0.2, nom_revest, 'FontSize',
9, 'FontAngle', 'italic', 'Color', [0.2 0.2 0.2]);

axis(app.UIAxes, 'equal');
app.UIAxes.XLim = [-(T/2 + esp*2.5), (dist_ext + esp*2)];
app.UIAxes.YLim = [-yn*0.5, yn*1.8];

```

```

        app.UIAxes.XTick = []; app.UIAxes.YTick = [];
        app.UIAxes.Box = 'off'; app.UIAxes.XColor = 'none';
        app.UIAxes.YColor = 'none';

        if b > 0 && z > 0, msg = 'SECCIÓN TRAPEZOIDAL';
        elseif b > 0 && z == 0, msg = 'SECCIÓN RECTANGULAR';
        elseif b == 0 && z > 0, msg = 'SECCIÓN TRIANGULAR';
        else, msg = 'SECCIÓN DE CANAL'; end

        title(app.UIAxes, msg, 'FontSize', 14, 'FontWeight', 'bold');
        hold(app.UIAxes, 'off');
    end
catch
    uialert(app.TiranteNormalUIFigure, 'Error matemático. Revise los
datos.', 'Error');
end

end

% Button pushed function: LimpiarButton
function LimpiarButtonPushed(app, event)
% --- 1. LIMPIAR CAMPOS DE ENTRADA ---
app.CaudalQEditField.Value = 0;
app.AnchodeSolerabEditField.Value = 0;
app.TaludZEditField.Value = 0;
app.RugosidadnEditField.Value = 0;
app.PendienteSEditField.Value = 0;

% --- 2. LIMPIAR CAMPOS DE RESULTADOS ---
app.PerimetroEditFieldLabel.Value = 0;
app.RadioHidraulicoEditFieldLabel.Value = 0;
app.VelocidadEditField.Value = 0;
app.EnergiaEspecificaEditField.Value = 0;
app.TiranteNormalyEditField.Value = 0;
app.AreahidraulicaEditField.Value = 0;
app.EspejodeaguaEditField.Value = 0;
app.NumerodeFrEditField.Value = 0;
app.TipodeFlujoEditField.Value = '';

% --- 3. LIMPIAR GRÁFICA ---
cla(app.UIAxes);
title(app.UIAxes, 'ÁREA DE DIBUJO');
app.UIAxes.XLim = [-1 1];
app.UIAxes.YLim = [-1 1];
app.UIAxes.XTick = [];
app.UIAxes.YTick = [];
end

% Button pushed function: ExportarButton
function ExportarButtonPushed(app, event)

% Verificar si existen resultados para exportar
if app.PerimetroEditFieldLabel.Value == 0
    uialert(app.TiranteNormalUIFigure, 'Primero debe realizar un cálculo
para exportar.', 'Atención');
    return;
end

% Diálogo para elegir ubicación y nombre del archivo

```

```

[file, path] = uinputfile('*.xlsx', 'Exportar Informe Hidráulico
Profesional', 'Memoria_Descriptiva_Canal.xlsx');

if isequal(file, 0) || isequal(path, 0)
    return;
end

filename = fullfile(path, file);

% --- CONSTRUCCIÓN DE TABLAS PROFESIONALES (Celdas con formato) ---
% Cabecera de Proyecto
Header = {
    'INFORME TÉCNICO: CÁLCULO DE TIRANTE NORMAL EN CANALES ABIERTOS', '';
    'Expordato de:', 'Canaly';
    'Fecha de reporte:', char(datetime('now','Format','dd/MM/yyyy
HH:mm')));
};

% Tabla 1: Parámetros de Diseño
DataInput = {
    '1. PARÁMETROS DE DISEÑO (INPUTS)', 'VALOR', 'UNIDAD';
    'Caudal de Diseño (Q)', app.CaudalQEditField.Value, 'm³/s';
    'Ancho de Solera (b)', app.AnchodeSolerabEditField.Value, 'm';
    'Factor de Talud (z)', app.TaludZEditField.Value, 'adim.';
    'Coeficiente de Rugosidad (n)', app.RugosidadnEditField.Value,
'adim.';
    'Pendiente Longitudinal (S)', app.PendienteSEditField.Value, 'm/m'
};

% Tabla 2: Resultados Hidráulicos
DataResults = {
    '2. RESULTADOS HIDRÁULICOS (OUTPUTS)', 'VALOR', 'UNIDAD';
    'Tirante Normal (yn)', app.PerimetroEditFieldLabel.Value, 'm';
    'Área Hidráulica (A)', app.RadioHidraulicoEditFieldLabel.Value, 'm²';
    'Perímetro Mojado (P)', app.TiranteNormalyEditField.Value, 'm';
    'Radio Hidráulico (R)', app.AreahidraulicaEditField.Value, 'm';
    'Espejo de Agua (T)', app.VelocidadEditField.Value, 'm';
    'Velocidad de Flujo (V)', app.EspejodeaguaEditField.Value, 'm/s';
    'Número de Froude (Fr)', app.EnergiaEspcificaEditField.Value,
'adim.';
    'Energía Específica (E)', app.NumerodeFrEditField.Value, 'm-kg/kg';
    'Régimen de Flujo', app.TipodeFlujoEditField.Value, ''
};

% Tabla 3: Memoria de Cálculo (Fórmulas)
DataFormulas = {
    '3. MEMORIA DE CÁLCULO', 'DEFINICIÓN';
    'Ecuación de Manning', 'Q = (1/n) * A * R^(2/3) * S^(1/2)';
    'Relación de Área', 'A = (b + z*y) * y';
    'Relación de Perímetro', 'P = b + 2*y*√(1 + z²)';
    'Estabilidad (Froude)', 'Fr = V / √(g * (A/T))';
    'Gravedad considerada', '9.81 m/s²'
};

% --- ESCRITURA EN EXCEL ---
% Corregido: Usar writecell en lugar de writematrix para evitar errores
de tipo 'cell'
writecell(Header, filename, 'Sheet', 1, 'Range', 'B2');
writecell(DataInput, filename, 'Sheet', 1, 'Range', 'B6');

```

```

writecell(DataResults, filename, 'Sheet', 1, 'Range', 'B14');
writecell(DataFormulas, filename, 'Sheet', 1, 'Range', 'F6');

% --- INTENTO DE FORMATO PROFESIONAL (Requiere Excel instalado) ---
try
    excelApp = actxserver('Excel.Application');
    excelApp.Visible = false;
    wb = excelApp.Workbooks.Open(filename);
    sheet = wb.Sheets.Item(1);

    % Formato Encabezados (RGB -> BGR para COM)
    ranges = {'B6:D6', 'B14:D14', 'F6:G6', 'B2:C2'};
    for i = 1:length(ranges)
        target = sheet.Range(ranges{i});
        target.Font.Bold = true;
        target.Font.ColorIndex = 2; % Blanco
        target.Interior.Color = hex2dec('808000'); % Teal / Turquesa
Oscuro
        target.Borders.LineStyle = 1;
    end

    % Bordes para todas las tablas
    sheet.Range('B6:D11').Borders.LineStyle = 1;
    sheet.Range('B14:D23').Borders.LineStyle = 1;
    sheet.Range('F6:G11').Borders.LineStyle = 1;

    % Autoajuste de columnas
    sheet.Columns.AutoFit;

    % Exportar y pegar imagen
    graph_path = fullfile(path, 'Grafica_Canal.png');
    exportgraphics(app.UIAxes, graph_path, 'Resolution', 300);

    % Insertar imagen en la celda F14
    image_range = sheet.Range('F14');
    sheet.Shapes.AddPicture(graph_path, 0, 1, image_range.Left,
image_range.Top, 400, 300);

    wb.Save();
    wb.Close();
    excelApp.Quit();
    delete(graph_path); % Limpiar imagen temporal

    uiconfirm(app.TiranteNormalUIFigure, ...
        'El informe ha sido generado con formato profesional y la gráfica
ha sido integrada satisfactoriamente.', ...
        'Exportación Exitosa', 'Icon', 'success');
catch
    % Fallback si no hay Excel COM
    uialert(app.TiranteNormalUIFigure, ...
        'Se exportaron los datos a Excel, pero el formateo estético y la
integración de imagen requieren Microsoft Excel instalado en el sistema.',
        ...
        'Informe Generado (Básico)');
end
end

% Button pushed function: MenuPrincipalButton
function MenuPrincipalButtonPushed(app, event)

```

```

        delete(app.TiranteNormalUIFigure);
    end

    % Button pushed function: CalcularButton_2
    function CalcularButton_2Pushed(app, event)
try
    % 1. CAPTURA DE DATOS
    Q = app.CaudalQEditField_2.Value;
    T_dis = app.EspejodeaguaTEditField.Value;
    n = app.RugosidadnEditField_2.Value;
    S = app.PendienteSEditField_2.Value;
    g = 9.81;

    % 2. FUNCIONES ANÓNIMAS (Para evitar el error de nesting)
    % Área Hidráulica:  $A_h = (2/3) * T * y$ 
    get_Ah = @(y) (2/3) * T_dis * y;

    % Perímetro Mojado
    % Usamos una función anónima con lógica condicional integrada
    get_Ph = @(y) ( ((4*y)/T_dis <= 1) * (T_dis + (8*y^2)/(3*T_dis)) +
    ...
    ((4*y)/T_dis > 1) * ( (T_dis/2) *
    (sqrt(1+(16*y^2/T_dis^2)) + ...
    (T_dis/(4*y)) * log((4*y/T_dis) +
    sqrt(1+(16*y^2/T_dis^2))))));

    % Función de Manning:  $f(y) = (1/n * A * R^{(2/3)} * S^{0.5}) - Q$ 
    f_err = @(y) ((1/n) * get_Ah(y) * (get_Ah(y)/get_Ph(y))^(2/3) *
    sqrt(S)) - Q;

    % 3. MÉTODO DE LA SECANTE
    y0 = 0.01; % Semilla inicial inferior
    y1 = T_dis; % Semilla inicial superior
    tol = 1e-8;
    max_iter = 100;
    yn = y1;

    for i = 1:max_iter
        f0 = f_err(y0);
        f1 = f_err(y1);
        if abs(f1 - f0) < 1e-12, break; end

        yn = y1 - f1 * (y1 - y0) / (f1 - f0);

        if abs(yn - y1) < tol, break; end
        y0 = y1;
        y1 = yn;
    end

    % 4. ASIGNACIÓN ORDENADA A RESULTADOS
    Ah_f = get_Ah(yn);
    Ph_f = get_Ph(yn);
    Rh_f = Ah_f / Ph_f;
    V_f = Q / Ah_f;

    app.TiranteNormalyEditField_2.Value = yn;
    app.AreahidraulicaEditField_2.Value = Ah_f;
    app.PerimetroEditFieldLabel_2.Value = Ph_f;
    app.RadioHidraulicoEditFieldLabel_2.Value = Rh_f;
end

```

```

app.VelocidadEditField_2.Value = V_f;
app.FocodeparabolaEditField.Value = (T_dis^2) / (8 * yn);

% Froude y Energía
Fr = V_f / sqrt(g * (Ah_f / T_dis));
app.NumerodeFrEditField_2.Value = Fr;
app.EnergiaEspecificaEditField_2.Value = yn + (V_f^2 / (2 * g));

% Tipo de Flujo
if Fr < 0.98, app.TipodeFlujoEditField_2.Value = 'Subcrítico';
elseif Fr > 1.02, app.TipodeFlujoEditField_2.Value = 'Supercrítico';
else, app.TipodeFlujoEditField_2.Value = 'Crítico'; end

% 5. GRÁFICA PROFESIONAL
% -----
cla(app.UIAxes_2);
hold(app.UIAxes_2, 'on');

% Configuración de fondo limpio
app.UIAxes_2.Color = [1 1 1];           % Fondo blanco
app.UIAxes_2.XGrid = 'off';             % Sin cuadrícula X
app.UIAxes_2.YGrid = 'off';             % Sin cuadrícula Y
app.UIAxes_2.Box = 'on';                 % Marco cerrado

% Parámetros geométricos
k_f = (T_dis^2) / (8 * yn);
y_max_visual = yn * 1.4;
y_vec = linspace(0, y_max_visual, 100);
x_vec = sqrt(2 .* k_f .* y_vec);

% 1. Dibujar el cuerpo del canal (Borde negro sólido)
plot(app.UIAxes_2, x_vec, y_vec, 'k', 'LineWidth', 2.5);
plot(app.UIAxes_2, -x_vec, y_vec, 'k', 'LineWidth', 2.5);

% 2. Relleno del agua (Azul sólido suave)
y_w = linspace(0, yn, 60);
x_w = sqrt(2 .* k_f .* y_w);
X_fill = [x_w, fliplr(-x_w)];
Y_fill = [y_w, fliplr(y_w)];
patch(app.UIAxes_2, 'XData', X_fill, 'YData', Y_fill, ...
      'FaceColor', [0.5 0.8 1.0], 'EdgeColor', 'none');

% 3. Línea de superficie del agua y símbolo técnico
T_n = sqrt(8 * k_f * yn);
plot(app.UIAxes_2, [-T_n/2, T_n/2], [yn, yn], 'Color', [0 0.3 0.6],
      'LineWidth', 1.5);
plot(app.UIAxes_2, 0, yn, 'v', 'MarkerSize', 8, ...
      'MarkerEdgeColor', [0 0.3 0.6], 'MarkerFaceColor', [0.5 0.8
1.0]);

% 4. ACOTACIONES (Flechas y texto)
% Acotación del Tirante (Vertical)
x_acot = T_dis * 0.55;
plot(app.UIAxes_2, [x_acot x_acot], [0 yn], 'k-', 'LineWidth', 1); %
Línea vertical
plot(app.UIAxes_2, x_acot, 0, '^k', 'MarkerFaceColor', 'k',
      'MarkerSize', 4); % Flecha abajo
plot(app.UIAxes_2, x_acot, yn, 'vk', 'MarkerFaceColor', 'k',
      'MarkerSize', 4); % Flecha arriba

```

```

        text(app.UIAxes_2, x_acot + 0.1, yn/2, ['y_n = ', num2str(yn,
'%.3f')], 'FontWeight', 'bold');

        % Acotación del Espejo de Agua (Horizontal)
        y_acot = yn + (y_max_visual * 0.05);
        plot(app.UIAxes_2, [-T_n/2 T_n/2], [y_acot y_acot], 'k-',
'LineWidth', 1); % Línea horiz
        plot(app.UIAxes_2, -T_n/2, y_acot, '<k', 'MarkerFaceColor', 'k',
'MarkerSize', 4);
        plot(app.UIAxes_2, T_n/2, y_acot, '>k', 'MarkerFaceColor', 'k',
'MarkerSize', 4);
        text(app.UIAxes_2, 0, y_acot + (y_max_visual*0.05), ['T = ',
num2str(T_n, ' %.3f')], ...
'HorizontalAlignment', 'center', 'FontWeight', 'bold');

        % 5. CUADRO RESUMEN (Ubicado en la esquina superior izquierda)
        resumen = {'Q = ', num2str(Q), ' m³/s'], ...
['V = ', num2str(V_f, ' %.2f'), ' m/s'], ...
['Fr = ', num2str(Fr, ' %.2f')]];
        text(app.UIAxes_2, -T_dis*0.65, y_max_visual*0.9, resumen, ...
'BackgroundColor', [1 1 0.9], 'EdgeColor', 'k', 'FontSize', 9);

        % Configuración final de ejes
        axis(app.UIAxes_2, 'equal');
        app.UIAxes_2.XLim = [-T_dis*0.8, T_dis*0.8];
        app.UIAxes_2.YLim = [0, y_max_visual];
        app.UIAxes_2.XTick = []; app.UIAxes_2.YTick = []; % Quitar números de
los ejes para limpieza

        hold(app.UIAxes_2, 'off');

    catch ME
        uialert(app.TiranteNormalUIFigure, ME.message, 'Error de Código');
    end

end

% Button pushed function: LimpiarButton_2
function LimpiarButton_2Pushed(app, event)

% --- 1. LIMPIAR CAMPOS DE ENTRADA (DATOS) ---
app.CaudalQEditField_2.Value = 0;
app.RugosidadnEditField_2.Value = 0;
app.PendienteSEditField_2.Value = 0;
app.EspejodeaguaTEditField.Value = 0; % Este es el T de entrada en tu app

% --- 2. LIMPIAR CAMPOS DE RESULTADOS ---
app.TiranteNormalyEditField_2.Value = 0;
app.AreahidraulicaEditField_2.Value = 0;
app.PerimetroEditFieldLabel_2.Value = 0; % Nota: En tu código es un Label
app.RadioHidraulicoEditFieldLabel_2.Value = 0;
app.VelocidadEditField_2.Value = 0;
app.NumerodeFrEditField_2.Value = 0;
app.EnergiaEspecificEditField_2.Value = 0;
app.FocodeparabolaEditField.Value = 0;
app.TipodeFlujoEditField_2.Value = ''; % Campo de texto vacío

% --- 3. LIMPIAR GRÁFICA (UIAxes_2) ---
cla(app.UIAxes_2); % Borra el contenido

```

```

title(app.UIAxes_2, 'ÁREA DE DIBUJO - PARABÓLICO');

% Resetear límites de los ejes para que no se vea vacío o deforme
app.UIAxes_2.XLim = [-1 1];
app.UIAxes_2.YLim = [-1 1];
app.UIAxes_2.XTick = [];
app.UIAxes_2.YTick = [];

end

% --- 1. CAPTURA DE DATOS DE ENTRADA ---
Q = app.CaudalQEditField_3.Value;
n = app.RugosidadnEditField_3.Value;
S = app.PendienteSEditField_3.Value;
D = app.DiametroEditField.Value; % Diámetro de la tubería

% Validación de seguridad
if any([Q n S D] <= 0)
    uialert(app.TiranteNormalUIFigure, 'Ingrese valores mayores a
cero.', 'Datos Requeridos');
    return
end

% --- 2. FUNCIONES HIDRÁULICAS (Basadas en el ángulo central theta) -
--
%  $y = (D/2) * (1 - \cos(\theta/2))$ 
%  $\theta = 2 * \arccos(1 - 2y/D)$ 

get_theta = @(y) 2 * acos(1 - (2*y/D));
get_A = @(y) (D^2 / 8) * (get_theta(y) - sin(get_theta(y)));
get_P = @(y) (D/2) * get_theta(y);
get_T = @(y) D * sin(get_theta(y)/2);

% Función de error de Manning:  $f(y) = Q_{calc} - Q_{real}$ 
f_error = @(y) ((1/n) * get_A(y) * (get_A(y) / get_P(y))^(2/3) *
sqrt(S)) - Q;

% --- 3. RESOLUCIÓN NUMÉRICA DEL TIRANTE NORMAL (yn) ---
% El tirante normal en tuberías no puede exceder el diámetro.
% Se busca entre un valor mínimo y el 95% del diámetro (límite
práctico de Manning)
y_min = 0.001 * D;
y_max = 0.999 * D;

if f_error(y_min) * f_error(y_max) < 0
    yn = fzero(f_error, [y_min y_max]);
else
    % Si el caudal es demasiado grande para el diámetro
    if f_error(y_max) < 0
        uialert(app.TiranteNormalUIFigure, 'El caudal ingresado
supera la capacidad de la tubería.', 'Capacidad Excedida');
        return
    end
    yn = 0.5 * D; % Valor semilla por defecto
end

% --- 4. CÁLCULO DE PARÁMETROS RESULTANTES ---
g = 9.81;

```

```

Area_n = get_A(yn);
Perim_n = get_P(yn);
Radio_n = Area_n / Perim_n;
Vel_n = Q / Area_n;
T_espejo = get_T(yn);
D_hidraulico = Area_n / T_espejo;
Froude = Vel_n / sqrt(g * D_hidraulico);
Energia = yn + (Vel_n^2 / (2*g));

% --- 5. ACTUALIZACIÓN DE LA INTERFAZ ---
app.TiranteNormalyEditField_3.Value = round(yn, 4);
app.AreahidraulicaEditField_3.Value = round(Area_n, 4);
app.PerimetroEditFieldLabel_3.Value = round(Perim_n, 4);
app.RadioHidraulicoEditFieldLabel_3.Value = round(Radio_n, 4);
app.VelocidadEditField_3.Value = round(Vel_n, 4);
app.EspejodeaguaEditField3.Value = round(T_espejo, 4);
app.NumerodeFrEditField_3.Value = round(Froude, 4);
app.EnergiaEspecificaEditField_3.Value = round(Energia, 4);

if Froude < 0.98, estado = 'Subcrítico';
elseif Froude > 1.02, estado = 'Supercrítico';
else, estado = 'Crítico'; end
app.TipodeFlujoEditField_3.Value = estado;

% --- 6. REPRESENTACIÓN GRÁFICA ---
cla(app.UIAxes_3);
hold(app.UIAxes_3, 'on');

% Dibujo del círculo de la tubería
ang = linspace(0, 2*pi, 100);
xc = (D/2) * cos(ang);
yc = (D/2) + (D/2) * sin(ang);
plot(app.UIAxes_3, xc, yc, 'k', 'LineWidth', 2.5);

% Dibujo del Agua (Relleno circular parcial)
% Calculamos el ángulo para el llenado
th_fill = get_theta(yn);
ang_h2o = linspace(3*pi/2 - th_fill/2, 3*pi/2 + th_fill/2, 100);
xh2o = (D/2) * cos(ang_h2o);
yh2o = (D/2) + (D/2) * sin(ang_h2o);

% Cerrar el polígono con la superficie del agua
patch(app.UIAxes_3, [xh2o, flip1r(xh2o(1))], [yh2o, yh2o(end)], ...
    [0.15 0.55 0.85], 'FaceAlpha', 0.5, 'EdgeColor', 'none');

% Línea de espejo de agua
line(app.UIAxes_3, [-T_espejo/2, T_espejo/2], [yn, yn], 'Color', [0
0.4 0.8], 'LineWidth', 2);

% Acotaciones
% Diámetro
line(app.UIAxes_3, [-D/2, D/2], [D*1.05, D*1.05], 'Color', 'k',
'Marker', '|');
text(app.UIAxes_3, 0, D*1.12, sprintf('D = %.2f m', D), 'Horiz',
'center', 'FontWeight', 'bold');

% Tirante yn
line(app.UIAxes_3, [D/2 + 0.05, D/2 + 0.05], [0, yn], 'Color', [0 0.5
0], 'LineWidth', 1.5);

```

```

        text(app.UIAxes_3, D/2 + 0.1, yn/2, sprintf('y_n = %.3f m', yn),
'Rotation', 270, 'Color', [0 0.4 0], 'FontWeight', 'bold');

% Configuración de ejes
axis(app.UIAxes_3, 'equal');
app.UIAxes_3.XLim = [-D*0.8, D*0.9];
app.UIAxes_3.YLim = [-D*0.1, D*1.2];
app.UIAxes_3.XTick = []; app.UIAxes_3.YTick = [];
app.UIAxes_3.Box = 'off';
title(app.UIAxes_3, 'SECCIÓN CIRCULAR - TIRANTE NORMAL');
hold(app.UIAxes_3, 'off');

catch ME
    uialert(app.TiranteNormalUIFigure, ME.message, 'Error de Cálculo');
end

```

2. CÓDIGO DE PROGRAMACIÓN PARA EL CÁLCULO DE DISEÑO DE UN CANAL EN FLUJO CRÍTICO

ANEXO 2.1. Código de programación para el diseño de canales en flujo crítico.

```
% --- 1. CAPTURA DE DATOS DE ENTRADA ---
Q = app.CaudalQEditField.Value;
b = app.AnchodeSolerabEditField.Value;
z = app.TaludZEditField.Value;

if Q <= 0 || b < 0 || z < 0
    uialert(app.TiranteCriticoUIFigure, 'El caudal debe ser > 0.
Solera y Talud deben ser >= 0.', 'Datos Requeridos');
    return
end

% --- 2. FUNCIONES GEOMÉTRICAS ---
g = 9.81;
get_A = @(y) (b + z*y) * y;
get_T = @(y) b + 2*z*y;
get_P = @(y) b + 2*y*sqrt(1 + z^2);
f_critico = @(y) (Q^2 / g) - (get_A(y)^3 / get_T(y));

% --- 3. RESOLUCIÓN NUMÉRICA (yc) ---
yc = fzero(f_critico, [0.0001, 50.0]);

% --- 4. CÁLCULO DE PARÁMETROS RESULTANTES ---
A_c = get_A(yc);
P_c = get_P(yc);
R_c = A_c / P_c;
T_c = get_T(yc);
v_c = Q / A_c;
D_c = A_c / T_c;
Fr_c = v_c / sqrt(g * D_c);
E_c = yc + (v_c^2 / (2*g));

% --- 5. ACTUALIZACIÓN DE LA INTERFAZ ---
app.TirantecriticoyEditField.Value = round(yc, 4);
app.AreahidraulicaEditField.Value = round(A_c, 4);
app.PerimetroEditFieldLabel.Value = round(P_c, 4);
app.RadioHidraulicoEditFieldLabel.Value = round(R_c, 4);
app.VelocidadEditField.Value = round(v_c, 4);
app.EspejodeaguaEditField.Value = round(T_c, 4);
app.NumerodeFrEditField.Value = round(Fr_c, 2);
app.EnergiaEspecificaEditField.Value = round(E_c, 4);

% --- 6. REPRESENTACIÓN GRÁFICA PROFESIONAL ---
cla(app.UIAxes);
hold(app.UIAxes, 'on');

c_agua = [0.15 0.55 0.85];
H_dibujo = yc * 1.5;
esp = max(T_c, 1.0) * 0.15;

% Geometría del Canal
x_canal = [-(b/2 + z*H_dibujo), -b/2, b/2, (b/2 + z*H_dibujo)];
y_canal = [H_dibujo, 0, 0, H_dibujo];
plot(app.UIAxes, x_canal, y_canal, 'k', 'LineWidth', 2.5);

% Dibujo del Agua
x_agua = [-(b/2 + z*yc), -b/2, b/2, (b/2 + z*yc)];
y_agua = [yc, 0, 0, yc];
```

```

        patch(app.UIAxes, x_agua, y_agua, c_agua, 'FaceAlpha', 0.5,
'EdgeColor', 'none');
        line(app.UIAxes, [x_agua(1), x_agua(4)], [yc, yc], 'Color', [0.3 0.8
1], 'LineWidth', 2.5);

        % ACOTACIONES
        % yc (Tirante Crítico)
        dist_ext = (b/2 + z*yc) + esp*0.6;
        line(app.UIAxes, [dist_ext, dist_ext], [0, yc], 'Color', [0 0.5 0],
'LineWidth', 1.5);
        line(app.UIAxes, [(b/2 + z*yc), dist_ext], [yc, yc], 'Color', [0.5
0.5 0.5], 'LineStyle', ':');
        line(app.UIAxes, [b/2, dist_ext], [0, 0], 'Color', [0.5 0.5 0.5],
'LineStyle', ':');
        text(app.UIAxes, dist_ext + esp*0.1, yc/2, sprintf('y_c = %.3f m',
yc), 'Rotation', 270, 'Color', [0 0.4 0], 'FontWeight', 'bold');

        % T (Espejo)
        plot(app.UIAxes, [x_agua(1), x_agua(4)], [yc*1.15, yc*1.15], 'k|-');
        text(app.UIAxes, 0, yc*1.22, sprintf('T = %.2f m', T_c),
'Horiz', 'center', 'FontWeight', 'bold');

        % b (Base)
        if b > 0
            plot(app.UIAxes, [-b/2, b/2], [-yc*0.15, -yc*0.15], 'r|-');
            text(app.UIAxes, 0, -yc*0.25, sprintf('b = %.2f m', b),
'Horiz', 'center', 'Color', 'r', 'FontWeight', 'bold');
        end

        % Indicador de Talud (z)
        if z > 0
            x_ref = -(b/2) - (yc * 0.2 * z);
            y_ref = yc * 0.5;
            plot(app.UIAxes, [x_ref, x_ref - yc*0.2*z, x_ref, x_ref], [y_ref,
y_ref, y_ref - yc*0.2, y_ref], 'k', 'LineWidth', 1.2);
            text(app.UIAxes, x_ref - (yc*0.1*z), y_ref + yc*0.08,
sprintf('z=%g', z), 'Horiz', 'center', 'FontWeight', 'bold');
        end

        % CUADRO RESUMEN
        str_res = {'\bfRESULTADOS CRÍTICOS'}, ['Área: ', sprintf('%.3f m^2',
A_c)], ...
            ['Velocidad: ', sprintf('%.3f m/s', v_c)], ['Energía: ',
sprintf('%.3f m', E_c)]];
        text(app.UIAxes, -(b/2 + z*yc) - esp*1.2, yc*1.5, str_res,
'FontSize', 8, 'BackgroundColor', [1 1 1 0.8], 'EdgeColor', [0.2 0.2 0.2],
'Margin', 5);

        % Configuración Ejes
        axis(app.UIAxes, 'equal');
        app.UIAxes.XLim = [-(b/2 + z*yc + esp*2), (dist_ext + esp*2)];
        app.UIAxes.YLim = [-yc*0.4, yc*1.8];
        app.UIAxes.XTick = []; app.UIAxes.YTick = []; app.UIAxes.Box = 'off';
        title(app.UIAxes, 'SECCIÓN TRAPEZOIDAL - ESTADO CRÍTICO', 'FontSize',
12);
        hold(app.UIAxes, 'off');

    catch ME
        uialert(app.TiranteCrticoUIFigure, ME.message, 'Error');

```

```

end
end

% Button pushed function: CalcularButton_2
function CalcularButton_2Pushed(app, event)
% --- 1. CAPTURA DE VARIABLES (Solo Q y T) ---
Q = app.CaudalQEditField_2.Value;
T_max = app.EspejodeaguaTEditField.Value;
g = 9.81;

% Obtener la figura contenedora de forma dinámica para evitar errores
fig = ancestor(app.UIAxes_2, 'figure');

% Validación inicial
if Q <= 0 || T_max <= 0
    uialert(fig, 'Ingreso Caudal (Q) y Espejo de agua (T) válidos.',
'Atención');
    return;
end

% --- 2. CÁLCULO ANALÍTICO DEL TIRANTE CRÍTICO (yc) ---
% Se resuelve la ecuación crítica de forma exacta:  $Q^2/g = (8/27) * T^2 * yc^3$ 
yc = ( (27 * Q^2) / (8 * g * T_max^2) )^(1/3);

try
% --- 3. PARÁMETROS HIDRÁULICOS DERIVADOS ---
Ah = (2/3) * T_max * yc;
v = Q / Ah;
Dh = Ah / T_max;
Fr = v / sqrt(g * Dh); % Confirmación (Dará exactamente 1.0 siempre)
E = yc + (v^2 / (2 * g));

% Foco / Constante de la parábola calibrada matemáticamente
k_parabola = (T_max^2) / (8 * yc);

% Perímetro Mojado (Fórmula exacta para precisión)
t_param = (4 * yc) / T_max;
Ph = (T_max / 2) * (sqrt(1 + t_param^2) + (1 / t_param) * log(t_param
+ sqrt(1 + t_param^2)));
Rh = Ah / Ph;

% --- 4. ASIGNACIÓN A COMPONENTES DE LA INTERFAZ ---
app.TiranteCritico2EditField.Value = round(yc, 4);
app.AreahidraulicaEditField_2.Value = round(Ah, 4);
app.PerimetroEditFieldLabel_2.Value = round(Ph, 4);
app.RadioHidraulicoEditFieldLabel_2.Value = round(Rh, 4);
app.VelocidadEditField_2.Value = round(v, 4);
app.NumerodeFrEditField_2.Value = round(Fr, 2);
app.EnergiaEspecificaEditField_2.Value = round(E, 4);
app.FocodeparabolaEditField.Value = round(k_parabola, 4);

% --- 5. REDISEÑO DE GRÁFICA PROFESIONAL "ESTILO PLANO TÉCNICO" ---
ax = app.UIAxes_2;
cla(ax); hold(ax, 'on');

% Configuración de Estética General
ax.Color = [1 1 1]; % Fondo Blanco Puro
ax.XGrid = 'off'; ax.YGrid = 'off'; % SIN cuadrículas

```

```

ax.Box = 'off'; % Sin recuadro
ax.XColor = 'none'; ax.YColor = 'none'; % Ocultar números de los ejes
ax.FontName = 'Segoe UI'; % Fuente moderna
ax.FontSize = 10;
axis(ax, 'equal'); % Relación de aspecto real 1:1 estricta

% Dibujo de la estructura de la Parábola (Cortada exactamente al
nivel del agua)
x_plot = linspace(-T_max/2, T_max/2, 150);
y_plot = (2 * x_plot.^2) / k_parabola;
plot(ax, x_plot, y_plot, 'Color', [0.15 0.15 0.15], 'LineWidth',
2.5);

% Relleno de Agua Semitransparente
x_agua = linspace(-T_max/2, T_max/2, 100);
y_agua = (2 * x_agua.^2) / k_parabola;
patch(ax, [x_agua, T_max/2, -T_max/2], [y_agua, yc, yc], [0.30 0.75
0.93], ...
'EdgeColor', 'none', 'FaceAlpha', 0.4);

% Línea azul de la superficie Libre (Borde a borde)
plot(ax, [-T_max/2, T_max/2], [yc, yc], 'Color', [0 0.45 0.74],
'LineWidth', 2);

% --- ACOTACIONES TÉCNICAS PROFESIONALES ---
color_cota = [0.4 0.4 0.4];

% B) Cota de Tirante Crítico (yc)
x_cota_y = T_max/2 + (T_max * 0.04);
plot(ax, [x_cota_y, x_cota_y], [0, yc], '|-', 'Color', [0 0.5 0],
'LineWidth', 1.5);

% Líneas de extensión punteadas (Guías finas)
line(ax, [T_max/2, x_cota_y], [yc, yc], 'Color', color_cota,
'LineStyle', ':');
line(ax, [0, x_cota_y], [0, 0], 'Color', color_cota, 'LineStyle',
':');

% Texto yc
text(ax, x_cota_y + (T_max * 0.02), yc/2, sprintf('y_c = %.3f m',
yc), ...
'Rotation', 270, 'FontWeight', 'bold', 'Color', [0 0.4 0],
'HorizontalAlignment', 'center');

% --- 6. CUADRO RESUMEN INTEGRADO DENTRO DE LA GRÁFICA ---
str_resumen = {
'\bf\fontsize{11}CUADRO TÉCNICO'
['Área: ', sprintf('%.3f m^2', Ah)]
['Radio Hid: ', sprintf('%.3f m', Rh)]
['Velocidad: ', sprintf('%.2f m/s', v)]
['Energía Mín: ', sprintf('%.3f m', E)]
['Froude: ', sprintf('%.2f', Fr)]
};

% Cuadro de texto ajustado a la nueva altura disponible
text(ax, -T_max * 0.7, yc * 0.95, str_resumen, ...
'FontSize', 8, ...
'BackgroundColor', [1 1 1], ...
'EdgeColor', [0.4 0.4 0.4], ...

```

```

        'Margin', 5, ...
        'VerticalAlignment', 'top');

% --- ENFOQUE MÁXIMO AL RAS ---
% Límites horizontales ceñidos al espejo y vertical al ras de yc
xlim(ax, [-T_max * 0.75, T_max * 0.75]);
ylim(ax, [-max(0.02, yc * 0.05), yc * 1.05]);

title(ax, 'DISEÑO HIDRÁULICO: SECCIÓN CRÍTICA', 'FontSize', 12,
'FontWeight', 'bold');
hold(ax, 'off');
catch
    uialert(fig, 'No se pudo procesar la geometría de la gráfica.
Verifique los datos.', 'Error de Gráfica');
end
end

% Button pushed function: CalcularButton_3
function CalcularButton_3Pushed(app, event)

%
=====
% CÁLCULO DEL TIRANTE CRÍTICO - SECCIÓN CIRCULAR
%
=====
try
    Q = app.CaudalQEditField_3.Value;
    D = app.DiametroEditField.Value;
    if Q <= 0 || D <= 0, return; end

    g = 9.81;
    get_theta = @(y) 2 * acos(1 - (2*y/D));
    get_A = @(y) (D^2 / 8) * (get_theta(y) - sin(get_theta(y)));
    get_T = @(y) D * sin(get_theta(y)/2);
    get_P = @(y) (D/2) * get_theta(y);
    f_obj = @(y) (Q^2 / g) - (get_A(y)^3 / get_T(y));
    yc = fzero(f_obj, [0.001*D, 0.95*D]);

    A_c = get_A(yc); T_c = get_T(yc); v_c = Q / A_c; E_c = yc + (v_c^2 /
(2*g));
    P_c = get_P(yc); R_c = A_c / P_c; Fr_c = v_c / sqrt(g * (A_c/T_c));

    app.TiranteCritico3yEditField.Value = round(yc, 4);
    app.AreahidraulicaEditField_3.Value = round(A_c, 4);
    app.EspejodeaguaEditField3.Value = round(T_c, 4);
    app.EnergiaEspecificaEditField_3.Value = round(E_c, 4);
    app.PerimetroEditFieldLabel_3.Value = round(P_c, 4);
    app.RadioHidraulicoEditFieldLabel_3.Value = round(R_c, 4);
    app.VelocidadEditField_3.Value = round(v_c, 4);
    app.NumerodeFrEditField_3.Value = round(Fr_c, 2);

% GRÁFICA PROFESIONAL
cla(app.UIAxes_3); hold(app.UIAxes_3, 'on');
c_agua = [0.15 0.55 0.85];
esp = D * 0.2;

% Dibujo Tubería
ang = linspace(0, 2*pi, 100);
plot(app.UIAxes_3, (D/2)*cos(ang), (D/2) + (D/2)*sin(ang), 'k',
'LineWidth', 2.5);

```

```

% Relleno Agua
th_c = get_theta(yc);
ang_h2o = linspace(3*pi/2 - th_c/2, 3*pi/2 + th_c/2, 50);
xh = (D/2)*cos(ang_h2o); yh = (D/2) + (D/2)*sin(ang_h2o);
patch(app.UIAxes_3, [xh, flipr(xh(1))], [yh, yh(end)], c_agua,
'FaceAlpha', 0.5, 'EdgeColor', 'none');
line(app.UIAxes_3, [-T_c/2, T_c/2], [yc, yc], 'Color', [0.3 0.8 1],
'LineWidth', 2.5);

% Acotación yc
dist_ext = (D/2) + esp*0.5;
line(app.UIAxes_3, [dist_ext, dist_ext], [0, yc], 'Color', [0 0.5 0],
'LineWidth', 1.5);
line(app.UIAxes_3, [T_c/2, dist_ext], [yc, yc], 'Color', [0.5 0.5
0.5], 'LineStyle', ':');
line(app.UIAxes_3, [0, dist_ext], [0, 0], 'Color', [0.5 0.5 0.5],
'LineStyle', ':');
text(app.UIAxes_3, dist_ext + esp*0.1, yc/2, sprintf('y_c = %.3f m',
yc), 'Rotation', 270, 'FontWeight', 'bold', 'Color', [0 0.4 0]);

% Acotación T
plot(app.UIAxes_3, [-T_c/2, T_c/2], [yc + D*0.05, yc + D*0.05], 'k|-
');
text(app.UIAxes_3, 0, yc + D*0.12, sprintf('T_c = %.2f m', T_c),
'Horiz', 'center', 'FontWeight', 'bold');

% Cuadro Resumen
str_res = {'\bfRESULTADOS CIRCULARES', ['D: ', sprintf('%.2f m',
D)], ...
['A: ', sprintf('%.3f m^2', A_c)], ['V: ', sprintf('%.3f
m/s', v_c)]};
text(app.UIAxes_3, -D/2 - esp*0.8, D*0.9, str_res, 'FontSize', 8,
'BackgroundColor', [1 1 1 0.8], 'EdgeColor', [0.2 0.2 0.2], 'Margin', 5);

axis(app.UIAxes_3, 'equal');
app.UIAxes_3.XLim = [-D/2-esp, dist_ext+esp];
app.UIAxes_3.YLim = [-D*0.1, D*1.2];
app.UIAxes_3.XTick = []; app.UIAxes_3.YTick = []; app.UIAxes_3.Box =
'off';
title(app.UIAxes_3, 'SECCIÓN CIRCULAR - ESTADO CRÍTICO');
hold(app.UIAxes_3, 'off');
catch ME, uialert(app.TiranteCrticoUIFigure, ME.message, 'Error'); end
end

% Button pushed function: LimpiarButton
function LimpiarButtonPushed(app, event)

% --- 1. LIMPIAR CAMPOS DE ENTRADA (DATOS) ---
app.CaudalQEditField.Value = 0;
app.AnchodeSolerabEditField.Value = 0;
app.TaludZEditField.Value = 0;

% --- 2. LIMPIAR CAMPOS DE RESULTADOS ---
app.TirantecriticoyEditField.Value = 0;
app.AreahidraulicaEditField.Value = 0;
app.EspejodeaguaEditField.Value = 0;
app.NumerodeFrEditField.Value = 0;
app.PerimetroEditFieldLabel.Value = 0;

```

```

app.RadioHidraulicoEditFieldLabel.Value = 0;
app.VelocidadEditField.Value = 0;
app.EnergiaEspecificaEditField.Value = 0;

% --- 3. LIMPIAR GRÁFICA (UIAxes) ---
cla(app.UIAxes);
title(app.UIAxes, 'ÁREA DE DIBUJO');

% Resetear visualización de los ejes
app.UIAxes.XLim = [-1 1];
app.UIAxes.YLim = [-1 1];
app.UIAxes.XTick = [];
app.UIAxes.YTick = [];

end

% Button pushed function: ExportarButton
function ExportarButtonPushed(app, event)

% 1. Verificar si existen resultados para exportar
if app.TirantecriticoyEditField.Value == 0
    uialert(app.TiranteCriticoUIFigure, 'Primero debe realizar un cálculo
para exportar.', 'Atención');
    return;
end

% 2. Diálogo para elegir ubicación y nombre del archivo
[file, path] = uiputfile('*.xlsx', 'Exportar Informe Crítico
Profesional', 'Memoria_Descriptiva_Critico_Trapezoidal.xlsx');

if isequal(file, 0) || isequal(path, 0)
    return;
end

filename = fullfile(path, file);

% --- 3. CONSTRUCCIÓN DE TABLAS ---
Header = {
    'INFORME TÉCNICO: CÁLCULO DE TIRANTE CRÍTICO (ESTADO DE ENERGÍA
MÍNIMA)', '';
    'Software:', 'CanalY Pro';
    'Fecha de reporte:', char(datetime('now','Format','dd/MM/yyyy
HH:mm'))
};

% Tabla 1: Datos de entrada
DataInput = {
    '1. PARÁMETROS DE DISEÑO (INPUTS)', 'VALOR', 'UNIDAD';
    'Caudal de Diseño (Q)', app.CaudalQEditField.Value, 'm³/s';
    'Ancho de Solera (b)', app.AnchodeSolerabEditField.Value, 'm';
    'Factor de Talud (z)', app.TaludZEditField.Value, 'adim.'
};

% Tabla 2: Resultados Críticos
DataResults = {
    '2. RESULTADOS HIDRÁULICOS (ESTADO CRÍTICO)', 'VALOR', 'UNIDAD';
    'Tirante Crítico (yc)', app.TirantecriticoyEditField.Value, 'm';
    'Área Hidráulica (Ac)', app.AreahidraulicaEditField.Value, 'm²';
    'Perímetro Mojado (Pc)', app.PerimetroEditFieldLabel.Value, 'm';
};

```

```

        'Radio Hidráulico (Rc)', app.RadioHidraulicoEditFieldLabel.Value,
'm';
        'Espejo de Agua (Tc)', app.EspejodeaguaEditField.Value, 'm';
        'Velocidad Crítica (Vc)', app.VelocidadEditField.Value, 'm/s';
        'Energía Mínima (Emin)', app.EnergiaEspecificaEditField.Value, 'm-
kg/kg';
        'Número de Froude (Fr)', app.NumerodeFrEditField.Value, 'adim.'
    };

% Tabla 3: Memoria de Cálculo (Fórmulas de Estado Crítico)
DataFormulas = {
    '3. MEMORIA DE CÁLCULO', 'CONDICIÓN';
    'Ecuación Fundamental', '(Q²/g) = (Ac³/Tc)';
    'Área (Trapezoidal)', 'Ac = (b + z*yc)*yc';
    'Espejo de Agua', 'Tc = b + 2*z*yc';
    'Gravedad (g)', '9.81 m/s²'
};

% --- 4. ESCRITURA EN EXCEL ---
writecell(Header, filename, 'Sheet', 1, 'Range', 'B2');
writecell(DataInput, filename, 'Sheet', 1, 'Range', 'B6');
writecell(DataResults, filename, 'Sheet', 1, 'Range', 'B12');
writecell(DataFormulas, filename, 'Sheet', 1, 'Range', 'F6');

% --- 5. FORMATO PROFESIONAL E IMAGEN (Excel COM) ---
try
    excelApp = actxserver('Excel.Application');
    excelApp.Visible = false;
    wb = excelApp.Workbooks.Open(filename);
    sheet = wb.Sheets.Item(1);

    % Color Naranja/Ocre para diferenciar "Estado Crítico"
    ranges = {'B6:D6', 'B12:D12', 'F6:G6', 'B2:C2'};
    for i = 1:length(ranges)
        target = sheet.Range(ranges{i});
        target.Font.Bold = true;
        target.Font.ColorIndex = 2;
        target.Interior.Color = hex2dec('D35400'); % Naranja Intenso
        target.Borders.LineStyle = 1;
    end
end

```

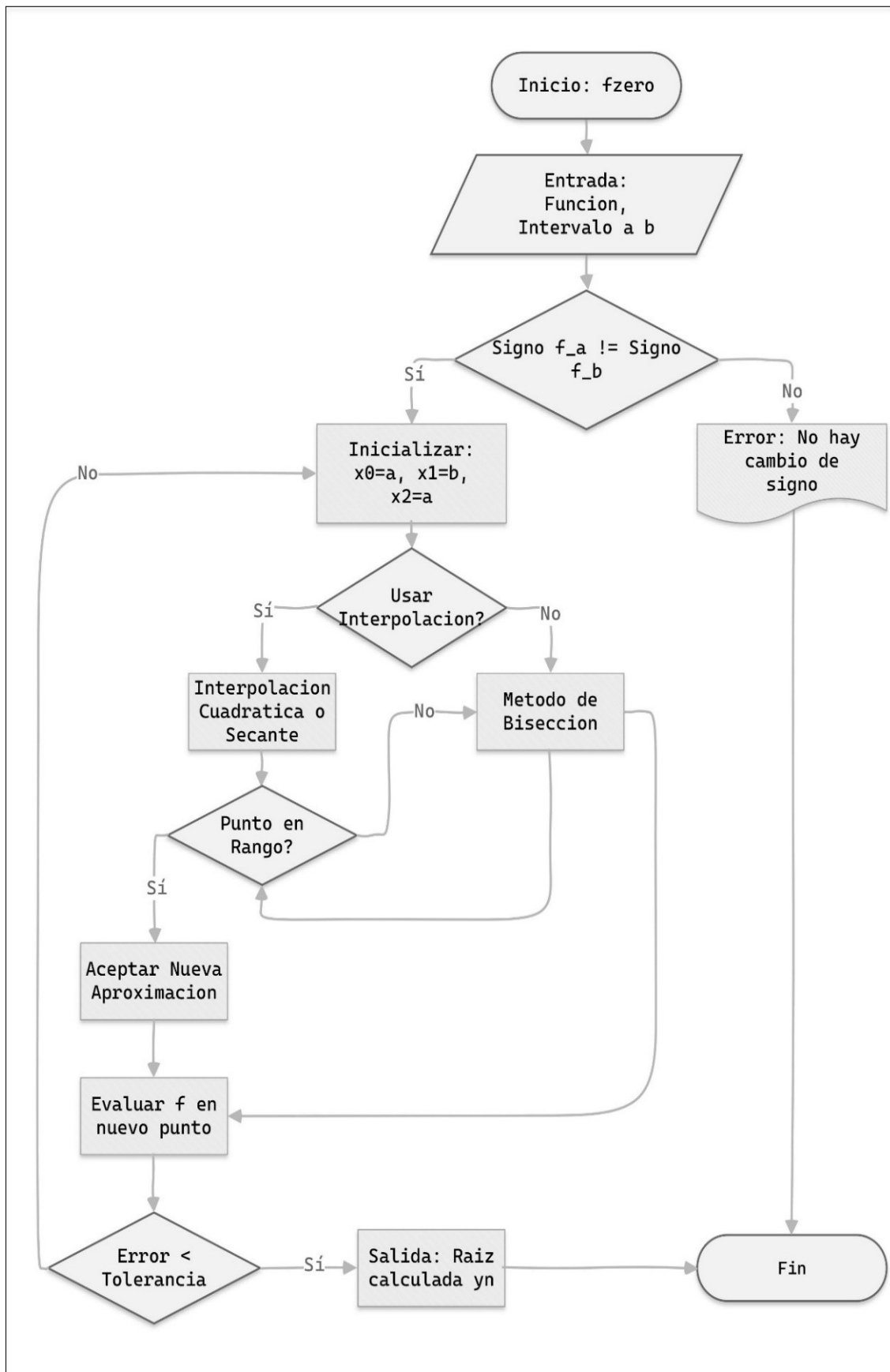
3. RESUMEN DE FÓRMULAS POR SECCIÓN UTILIZADOS EN EL PROGRAMA CANALY

ANEXO 3.1. Resumen de fórmulas por secciones utilizadas en el software Canaly.

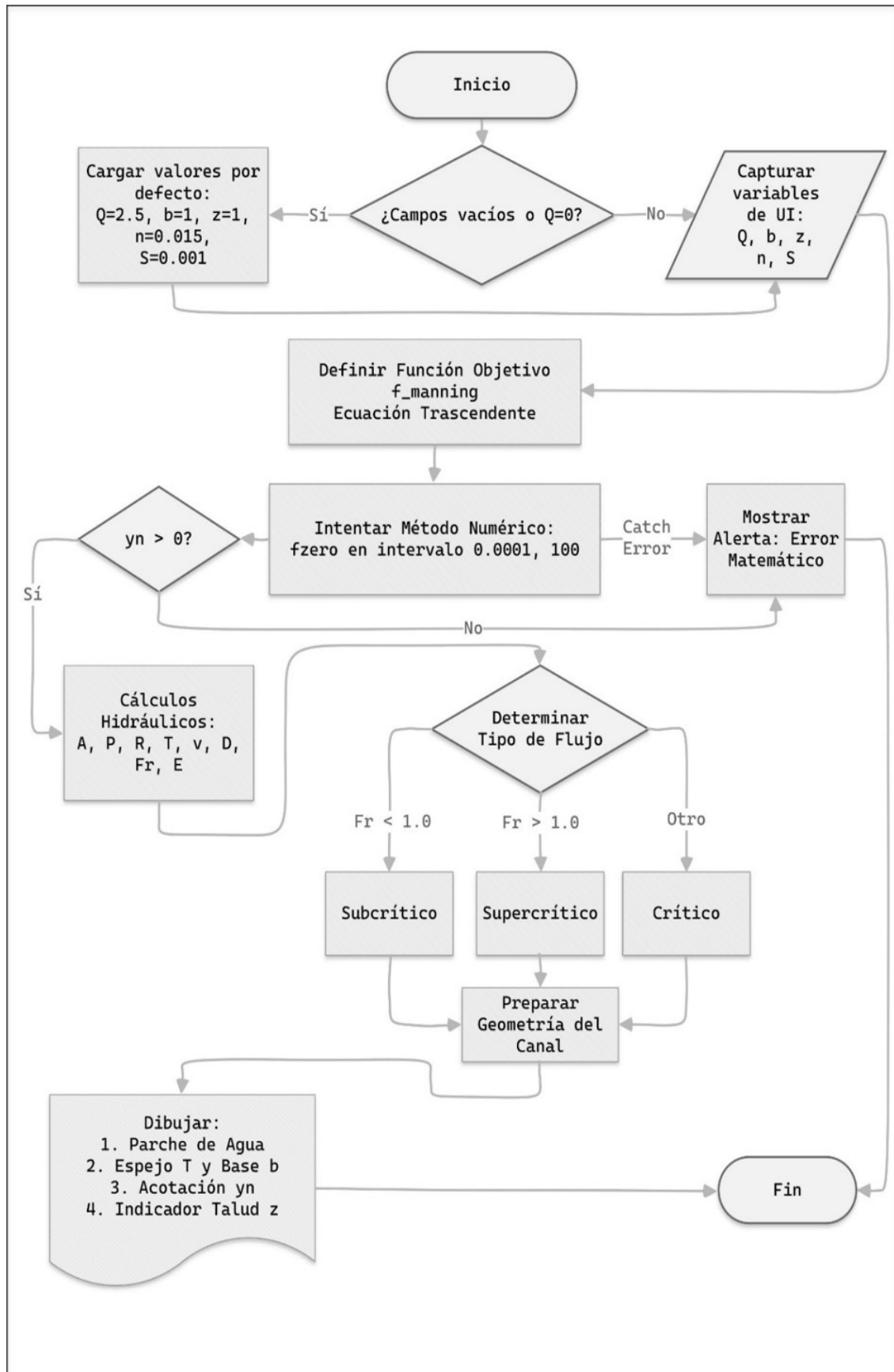
Sección	Variables y Notas	Área Mojada (A)	Perímetro Mojado (P)	Radio Hidráulico (Rh)
Rectangular	- Ancho de fondo: b - Profundidad: y - Sección simple y uniforme.	$A = b \cdot y$	$P = b + 2y$	$R_h = \frac{by}{b + 2y}$
Triangular	- Pendiente lateral: z (horiz:vert) - Profundidad: y - Simétrica, sin ancho de fondo ($b = 0$).	$A = zy^2$	$P = 2y\sqrt{1 + z^2}$	$R_h = \frac{zy}{2\sqrt{1 + z^2}}$
Trapezoidal	- Ancho de fondo: b - Pendiente lateral: z - Profundidad: y - Generaliza rectangular y triangular.	$A = (b + zy)y$	$P = b + 2y\sqrt{1 + z^2}$	$R_h = \frac{(b + zy)y}{b + 2y\sqrt{1 + z^2}}$
Circular	- Diámetro: D - Profundidad: y (parcialmente llena) - Radio: $r = D/2$ - Ángulo central: $\theta = 2 \arccos\left(1 - \frac{2y}{D}\right)$ rad.	$A = \frac{D^2}{8}(\theta - \sin \theta)$	$P = \frac{D}{2}$	$R_h = \frac{D}{4}\left(1 - \frac{\sin \theta}{\theta}\right)$
Parabólica	- Ancho superior: T - Profundidad: y - Forma $x^2 \propto y$. - Fórmulas aproximadas para canales anchos.	$A = \frac{2}{3}Ty$	$P \approx T + \frac{8y^2}{3T}$	$R_h \approx \frac{2T^2y}{3T^2 + 8y^2}$

4. DIAGRAMAS DE FLUJO UTILIZADOS EN EL SOFTWARE CANALY

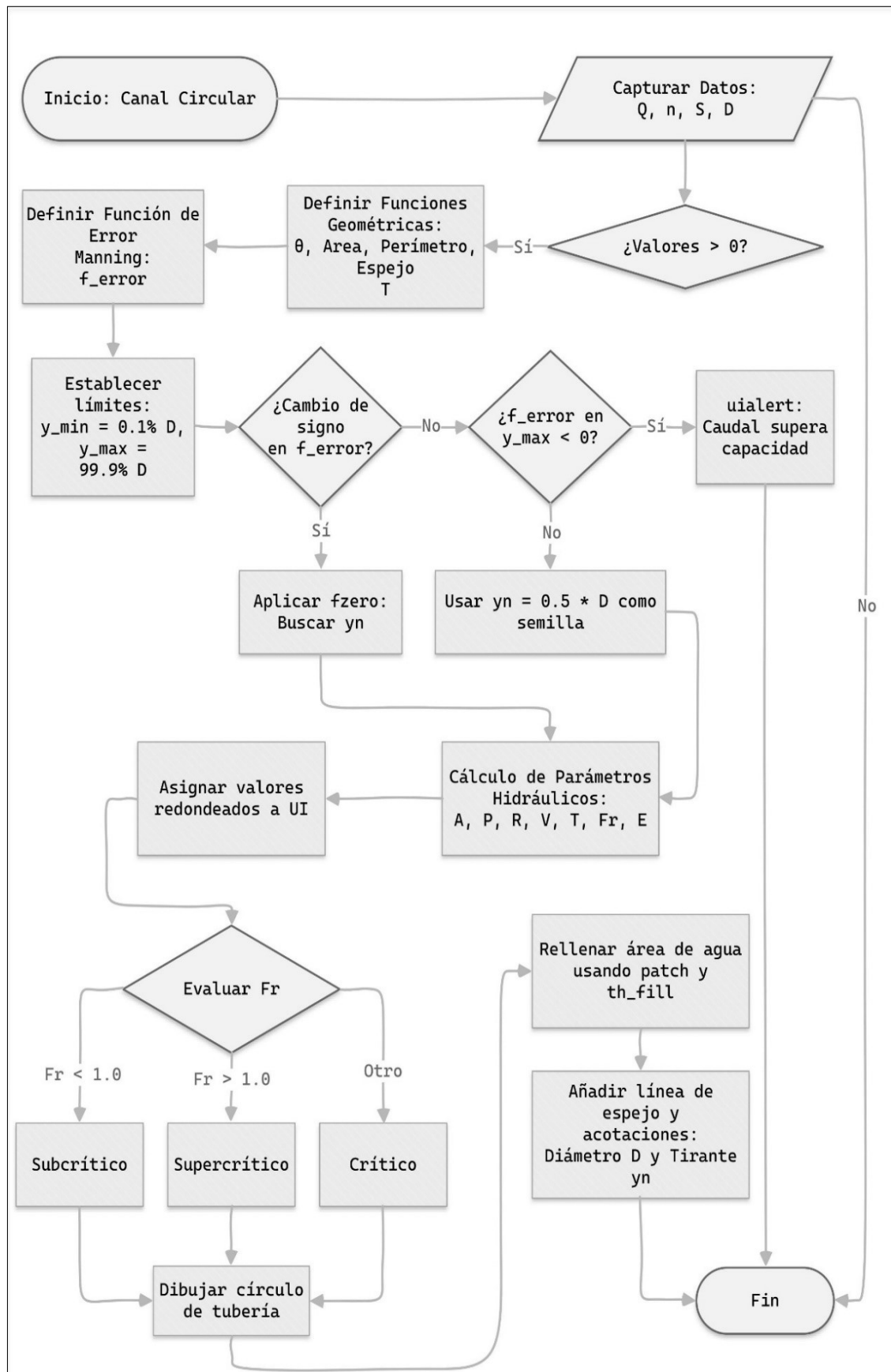
Anexo 4.1. Diagrama de flujo de Fzero en las secciones prismáticas.



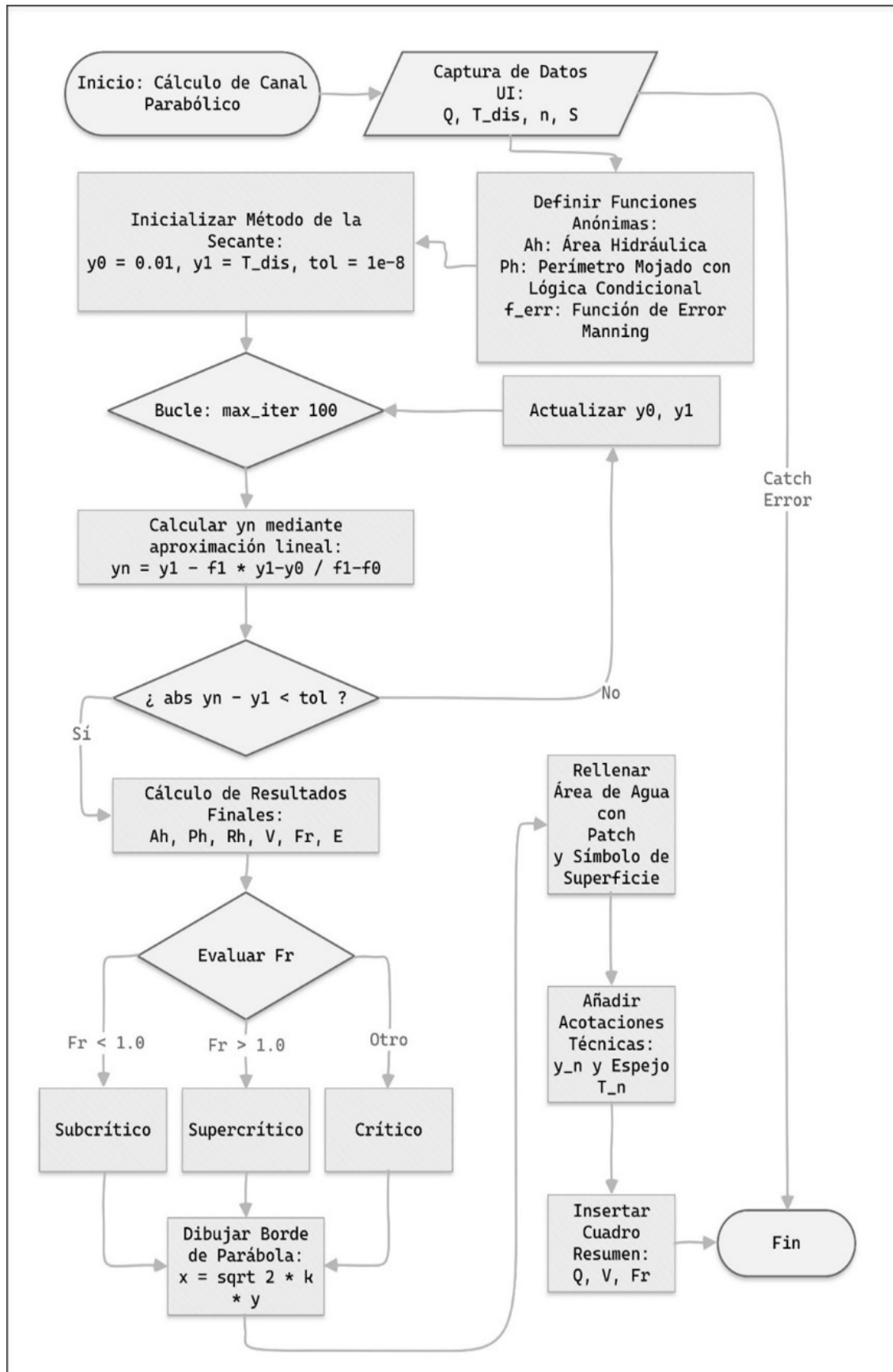
Anexo 4.2. Código Matlab del cálculo de sección Rectangular, trapezoidal, triangular.



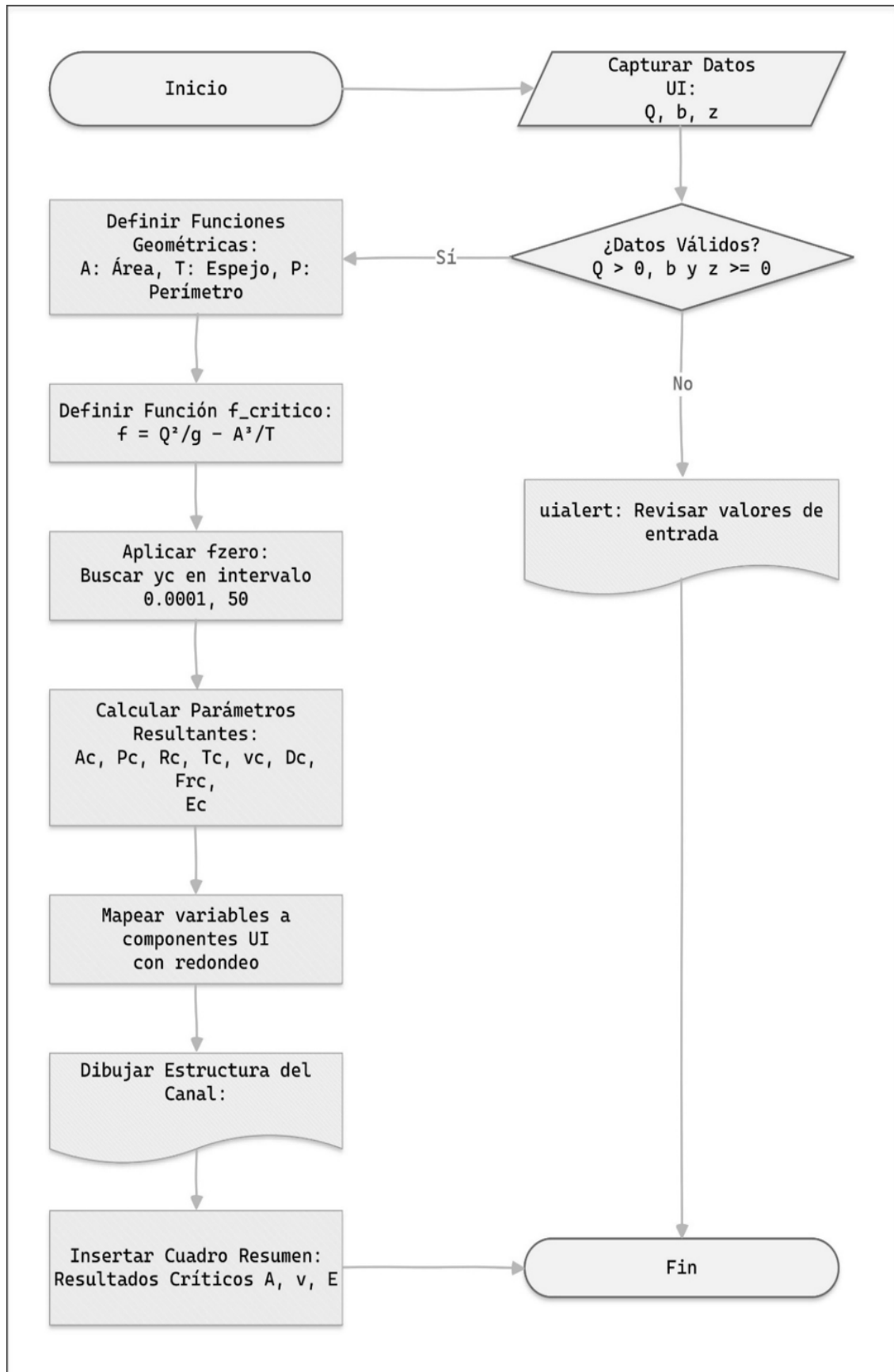
Anexo 4.3. Código Matlab del cálculo de sección circular.



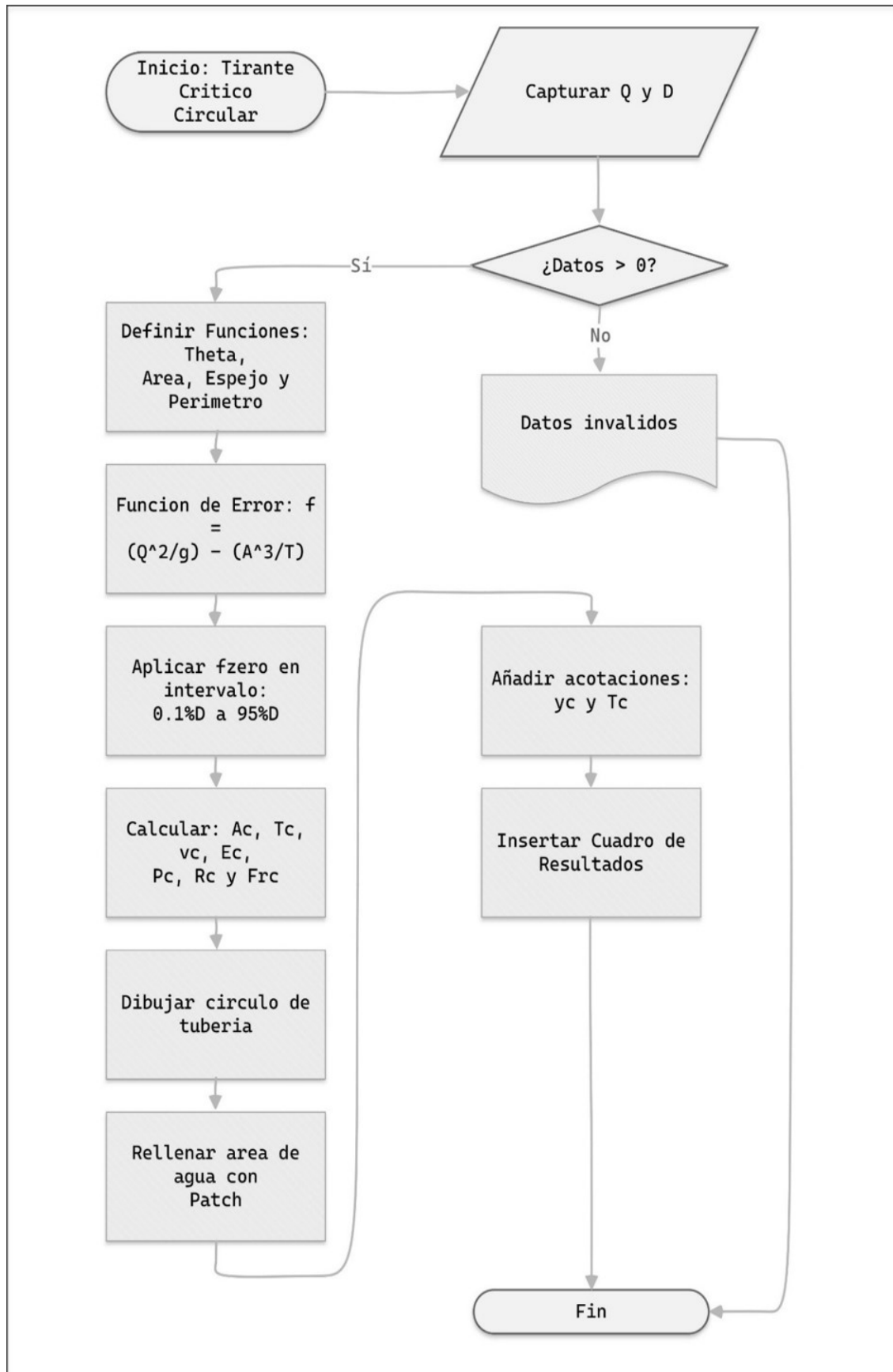
Anexo 4.4. Código Matlab del cálculo de sección Parabólica.



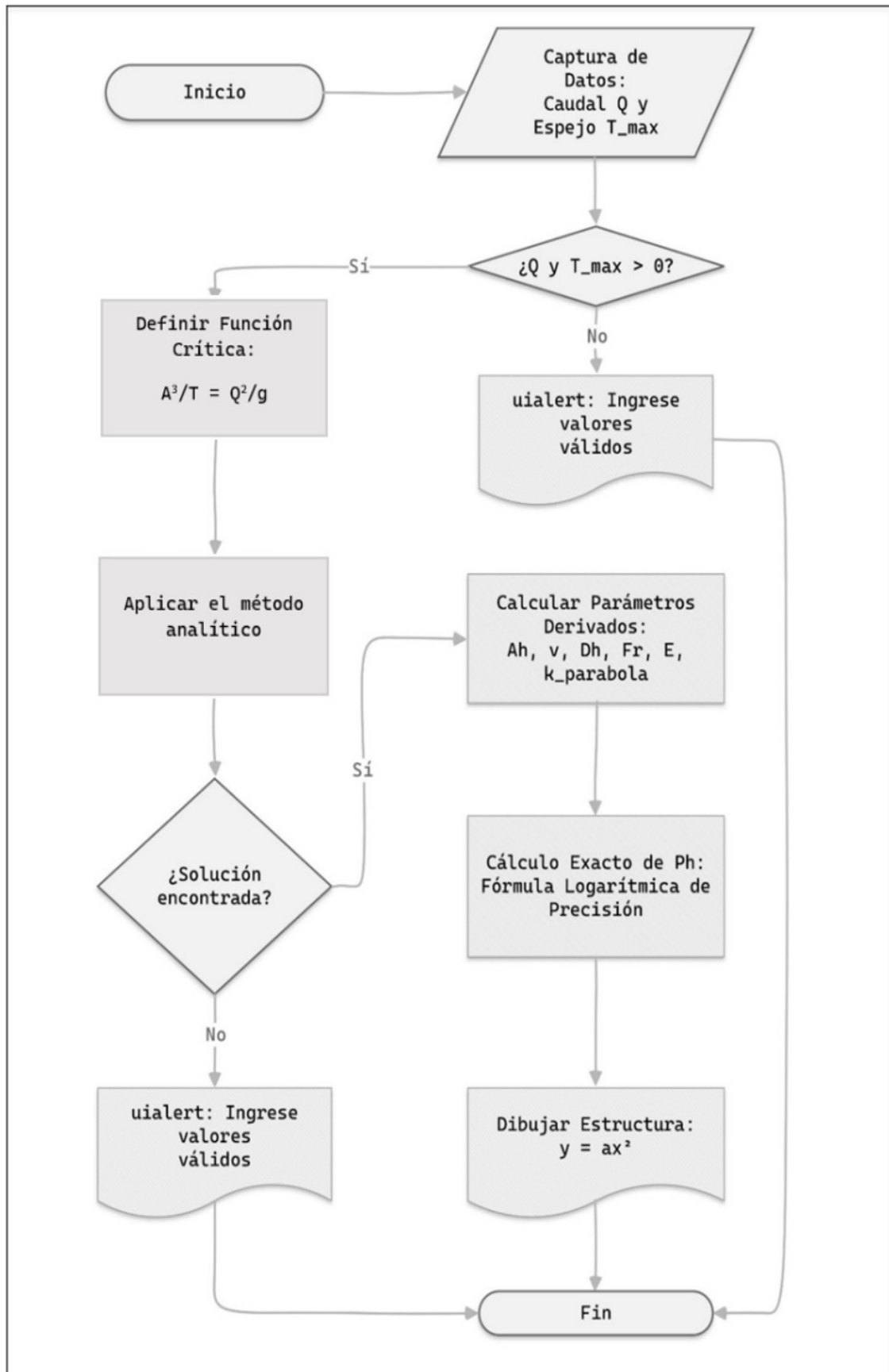
Anexo 4.5. Código Matlab del cálculo de sección rectangular, trapezoidal, triangular (crítico).



Anexo 4.6. Código Matlab del cálculo de sección circular (critico).



Anexo 4.7. Código Matlab del cálculo de sección Parabólica (critico).





ACTA DE SUSTENTACIÓN DE TESIS
Bach. LILIANA KATERIN VILCA HUARANCCA
R.D. N° 197-2026-UNSCH-FCA-D

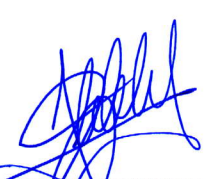
En la ciudad de Ayacucho, a los veintiséis días del mes de junio del año dos mil veintiséis, siendo las dieciocho horas, se reunieron en el auditorio de la Facultad de Ciencias Agrarias, bajo la presidencia del Dr. Felipe Escobar Ramírez Decano de la Facultad de Ciencias Agrarias; los miembros del jurado conformado por el Ing. Efraín Chuchón Prado, Ing. John Samuel Cazorla Orihuela como asesor, Dr. Eleazar Chuchón Angulo y el Ing. Vance Giorgio Fernández Huamán; actuando como secretario de actas el Mtro. Rodolfo Alca Mendoza, para recibir la sustentación de la Tesis titulado: **Desarrollo de un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b - Huamanga - Ayacucho - 2025**, para obtener el Título Profesional de Ingeniera Agrícola, presentado por la Bachiller **LILIANA KATERIN VILCA HUARANCCA**.

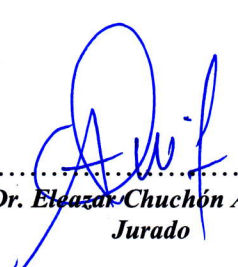
El señor Decano previa verificación de los documentos exigidos solicitó se proceda con la sustentación y posterior defensa de la tesis en un periodo de cuarenta y cinco minutos de acuerdo al reglamento de grados y títulos vigente. Terminado la exposición, los miembros del Jurado, formularon sus preguntas, aclaraciones y/o observaciones correspondientes. Luego se invitó a los miembros del jurado pasar a otra aula para la deliberación y calificación del trabajo de tesis, teniendo el siguiente resultado:

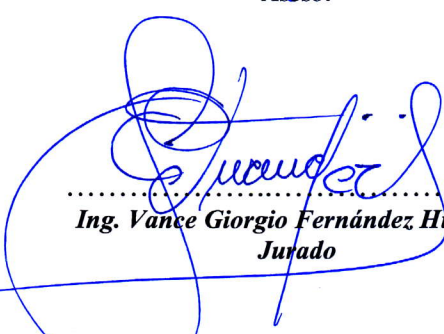
Jurado evaluador	Exposición	Respuestas a las preguntas	Generación de conocimiento	Promedio
Ing. Efraín Chuchón Prado	14	15	16	15
Ing. John Samuel Cazorla Orihuela	15	15	15	15
Dr. Eleazar Chuchón Angulo	11	14	11	12
Ing. Vance Giorgio Fernández Huamán	12	14	15	14
PROMEDIO GENERAL				14

Acto seguido se invita a la sustentante y público en general para dar a conocer el resultado final. Firman el acta.


.....
Ing. Efraín Chuchón Prado
Presidente


.....
Ing. John Samuel Cazorla Orihuela
Asesor


.....
Dr. Eleazar Chuchón Angulo
Jurado


.....
Ing. Vance Giorgio Fernández Huamán
Jurado


.....
Mtro. Rodolfo Alca Mendoza
Secretario Docente

CONSTANCIA DE CONTROL DE ORIGINALIDAD DE TRABAJO DE TESIS

El que suscribe, miembro de la comisión de docentes instructores responsables de operativizar, verificar, garantizar y controlar la originalidad de los trabajos de **TESIS** de la Facultad de Ciencias Agrarias, de la Universidad Nacional de San Cristóbal de Huamanga, autorizado por **R.D N° 213-2025-UNSCH-FCA-CF**; hace constar que el trabajo titulado:

Desarrollo de un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b - Huamanga - Ayacucho - 2025

Autor : Liliana Katerin Vilca Huarancca
Asesor : John Samuel Cazorla Orihuela

Ha sido sometido al control de originalidad mediante el software TURNITIN UNSCH, acorde al Reglamento de originalidad de trabajos de tesis, aprobado mediante RCU 039-2021-UNSCH-CU, arrojando un resultado de **seis (6%)** de índice de similitud, con exclusión de citas y bibliografía, y de coincidencias menores de 30 palabras.

En consecuencia, se otorga la presente Constancia de Originalidad para los fines pertinentes.

Nota: Se adjunta el resultado con identificador de la entrega: 2998741835

Ayacucho, 17 de Julio de 2026

Atentamente,



UNIVERSIDAD NACIONAL DE
SAN CRISTÓBAL DE HUAMANGA
Facultad de Ciencias Agrarias
Prof. Dr. Eleazar Chuchón Angulo
DNI 42682714

Desarrollo de un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b - Huamanga - Ayacucho - 2025

por Liliana Katerin Vilca Huarancca

Fecha de entrega: 16-jul-2026 01:00a. m. (UTC-0500)

Identificador de la entrega: 2998741835

Nombre del archivo: Desarrollo_de_un_software_para_el_diseño_hidráulico_de_canales_abiertos_mediante_Matlab_R2025b_-_Huamanga_-_Ayacucho.pdf (9.19M)

Total de palabras: 20139

Total de caracteres: 119304

Desarrollo de un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b - Huamanga - Ayacucho - 2025

INFORME DE ORIGINALIDAD

6%	5%	1%	3%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	Submitted to Universidad del Atlántico	2%
	Trabajo del estudiante	
2	repositorio.unsch.edu.pe	1%
	Fuente de Internet	
3	idoc.pub	1%
	Fuente de Internet	
4	hdl.handle.net	<1%
	Fuente de Internet	
5	Submitted to Universidad Nacional de San Cristóbal de Huamanga	<1%
	Trabajo del estudiante	
6	doku.pub	<1%
	Fuente de Internet	
7	qdoc.tips	<1%
	Fuente de Internet	
8	repositorio.unica.edu.pe	<1%
	Fuente de Internet	
9	Submitted to Universidad de Piura	<1%
	Trabajo del estudiante	
10	vdocuments.com.br	<1%
	Fuente de Internet	

DESARROLLO DE UN SOFTWARE PARA EL DISEÑO HIDRÁULICO DE CANALES ABIERTOS MEDIANTE MATLAB R2025B - HUAMANGA - AYACUCHO - 2025

DEVELOPMENT OF SOFTWARE FOR THE HYDRAULIC DESIGN OF OPEN CHANNELS USING MATLAB R2025B – HUAMANGA – AYACUCHO – 2025

Liliana Katerin Vilca Huarancca
John Samuel Cazorla Orihuela

liliana.vilca.21@unsch.edu.pe
john.cazorla@unsch.edu.pe

Área de investigación: Medio ambiente
Línea de investigación: Hidrología e hidráulica

RESUMEN

La investigación llevada a cabo, se planteó como objetivo principal desarrollar un software especializado en el entorno Matlab R2025b para automatizar el cálculo de los parámetros hidráulicos y geométricos en canales abiertos de sección prismática. El diseño hidráulico de canales que, comúnmente, se ha realizado mediante métodos manuales e iterativos que invierten una mayor cantidad de tiempo y son propensos a cometer errores de precisión. Ante esta problemática en busca de una solución, se desarrolló el software denominado Canaly utilizando el entorno App Designer incorporada en Matlab, utilizando algoritmos numéricos basados en métodos iterativos abiertos avanzados como, la Secante y el Algoritmo de Brent, el procedimiento metodológico adoptado en esta investigación se caracteriza por ser aplicada y tecnológica, con un enfoque cuantitativo y una investigación no experimental, orientado a la implementación de algoritmos numéricos para la solución de problemas prácticos. El software calcula el tirante normal, tirante crítico, área, perímetro mojado, radio hidráulico y energía específica para cinco secciones fundamentales: rectangular, trapezoidal, triangular, parabólica y circular. Con el fin de corroborar la confiabilidad del software desarrollado, se aplicaron pruebas de validación mediante la inserción de casos, comparando los resultados obtenidos por el software Canaly frente al software HCanales v3.1. Los resultados de la validación demostraron una fiabilidad del 99.9%, con errores relativos despreciables (inferiores al 0.001), confirmando que la programación mediante Matlab no solo reduce el tiempo de cálculo, sino que posee una precisión técnica confiable, moderna para los profesionales y estudiantes de ingeniería en la región de Ayacucho.

Palabras clave: Matlab, Diseño hidráulico, Canales, Algoritmos numéricos, HCanales, Canaly.

ABSTRACT

The main objective of the research carried out was to develop specialized software in the Matlab R2025b environment to automate the calculation of hydraulic and geometric parameters in open channels of prismatic section. The hydraulic design of the channels has usually been carried out using manual and iterative methods that invest more time and are prone to precision errors. Faced with this problem in search of a solution, the software called Canaly was developed using the App Designer environment incorporated in Matlab, using numerical algorithms based on advanced open iterative methods such as the Secante and the Brent Algorithm; The methodological procedure adopted in this research is characterized by being applied and technological, with a quantitative approach and non-experimental research. oriented to the implementation of numerical algorithms for the resolution of practical problems. The software calculates the normal staybar, the critical direction bar, the area, the wet perimeter, the hydraulic radius, and the specific energy for five fundamental sections: rectangular, trapezoidal, triangular, parabolic, and circular. To corroborate the reliability of the developed software, validation tests were applied by inserting cases, comparing the results obtained by the Canaly software with the HCanales v3.1 software. The validation results demonstrated a reliability of 99.9%, with negligible relative errors (less than 0.001), confirming that programming with Matlab not only reduces calculation time, but also has modern and reliable technical accuracy for engineering professionals and students in the Ayacucho region.

Keywords: Matlab, Hydraulic Design, Canals, Numerical Algorithms, HCanales, Canaly.

I. INTRODUCCIÓN

En el campo de la ingeniería, la planificación y administración de obras hidráulicas representan pilares fundamentales para el desarrollo rural y la seguridad hídrica. Dentro de este contexto, los canales abiertos son los elementos de conducción más utilizados para la conducción de agua para riego, drenaje y abastecimiento. La determinación precisa de sus parámetros hidráulicos como el tirante normal, crítico y la energía específica, el tipo de régimen de flujo es una tarea que constituye un aspecto fundamental, ya que determina no sólo la eficiencia operativa de la obra, sino también su estabilidad estructural y la optimización de los costos de inversión.

Históricamente, el cálculo de estas variables se ha realizado mediante métodos manuales que dependen de la aplicación de fórmulas empíricas, como la ecuación de Manning, y el uso de herramientas analógicas como tablas y nomogramas. Si bien estos métodos han sido la base de la hidráulica clásica, presentan limitaciones críticas en la práctica profesional contemporánea. El análisis de las ecuaciones que describen al uniforme y el flujo crítico a menudo requiere procesos iterativos complejos, dado que el tirante se encuentra implícito en la geometría de la sección. Esto conlleva un riesgo elevado de error humano, imprecisiones por redondeo y una inversión de tiempo considerable que ralentiza el flujo de trabajo en las etapas de diseño.

La presente investigación, titulada "Desarrollo de un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b – Huamanga – Ayacucho - 2025", surge como una solución tecnológica ante esta problemática. El núcleo del trabajo consiste en el desarrollo del software Canaly, una herramienta integral programada en el entorno App Designer de Matlab, este programa ha sido diseñado para automatizar y sistematizar el cálculo hidráulico, permitiendo al proyectista obtener resultados de alta fidelidad mediante la implementación de algoritmos numéricos de vanguardia.

A nivel regional, en Ayacucho, la topografía accidentada y la necesidad de modernizar la agricultura exigen que los ingenieros cuenten con herramientas precisas y adaptadas a la realidad técnica. Sin embargo, todavía existe una marcada dependencia de métodos convencionales o de software generalista que no siempre cumple con los requerimientos específicos del diseño agrícola local. Ante este escenario, el problema de investigación se define en torno a la necesidad de evaluar cómo la automatización de modelos matemáticos en un entorno de desarrollo robusto puede elevar los estándares de precisión y eficiencia en comparación con el cálculo manual tradicional.

La investigación se justifica desde una perspectiva técnica y científica por la implementación de métodos de resolución de raíces, tales como los algoritmos de

la Secante y el método de Brent. Estos métodos permiten alcanzar precisiones computacionales con una tolerancia de hasta 1.0×10^{-8} metros, garantizando la convergencia incluso en secciones transversales complejas como las parabólicas o circulares. Desde un punto de vista profesional, el software Canaly proporciona una interfaz gráfica intuitiva que facilita la visualización dinámica de la sección transversal, permitiendo al usuario realizar análisis de sensibilidad de manera inmediata al modificar variables como el caudal, rugosidad del material y la pendiente.

La hipótesis central del estudio sostiene que el desarrollo de un software especializado en Matlab optimiza significativamente la precisión técnica y reduce los tiempos de procesamiento de datos, eliminando la incertidumbre propia del cálculo manual. Para demostrarlo, se establecieron objetivos claros: sistematizar los modelos hidráulicos de flujo para secciones prismáticas (rectangular, trapezoidal, triangular, parabólica y circular), diseñar una interfaz de usuario funcional y moderna, y validar los resultados del programa mediante una comparación estadística con el software patrón HCanales v3.1.

Finalmente, la metodología adoptada en esta investigación fue de carácter aplicado y tecnológico, siguiendo un enfoque cuantitativo. El proceso abarcó desde la sistematización de las bases teóricas de la hidráulica de canales hasta la codificación de la lógica de control y la fase de validación rigurosa. Con este aporte, se busca entregar a la comunidad de la Universidad Nacional de San Cristóbal de Huamanga y a los profesionales del sector una herramienta potente y confiable que contribuya al avance de la infraestructura hidráulica en el Perú.

Objetivo general

Desarrollar un software para el diseño hidráulico de canales abiertos mediante Matlab R2025b, que permita optimizar los procesos de cálculo y mejorar la eficiencia y precisión en el diseño hidráulico en Huamanga - Ayacucho - 2025.

Objetivos específicos

1. Sistematizar los modelos matemáticos de flujo uniforme (Manning) y flujo crítico para secciones prismáticas (rectangular, trapezoidal, triangular, parabólico y circulares), definiendo las ecuaciones constitutivas del software.
2. Implementar algoritmos numéricos de raíces en el lenguaje Matlab R2025b, estableciendo criterios de convergencia y tolerancia para la resolución de ecuaciones no lineales implícitas.
3. Diseñar una interfaz gráfica de usuario (GUI) moderna mediante App Designer, diferenciando de la arquitectura obsoleta GUIDE, para garantizar la interactividad y visualización dinámica de la sección transversal.
4. Validar la fiabilidad del software desarrollado mediante un análisis comparativo de resultados (tirante normal y crítico) frente al software patrón HCanales, evaluando el error relativo porcentual.

II. MATERIALES Y MÉTODOS

2.1. Tipo de investigación

La investigación es de tipo Aplicada y Tecnológica. Es aplicada porque busca utilizar las teorías de la mecánica de fluidos (ecuaciones de Manning y Froude) para resolver problemas prácticos en el diseño de infraestructura hídrica. Es tecnológica debido a que el resultado final es un artefacto funcional (software desarrollado en Matlab R2025b) diseñado para optimizar procesos de cálculo de ingeniería.

2.2. Nivel de investigación

El nivel es Descriptivo-Correlacional. Es descriptivo porque detalla y caracteriza las propiedades geométricas y cinemáticas de los fluidos en diversas secciones. Asimismo, posee un componente correlacional/comparativo al evaluar la precisión de los resultados del software frente a soluciones analíticas y herramientas validadas.

2.3. Diseño de la investigación

Se emplea un diseño No experimental, descriptivo y transversal.

- No experimental: No existe manipulación deliberada de variables físicas; se trabaja con modelos matemáticos y simulaciones numéricas.
- Transversal: La recolección de datos y la validación de algoritmos que se realizan en un único periodo de tiempo definido.

2.4. Población y muestra

- Población: Está constituida por el universo de secciones prismáticas (canales abiertos y conductos cerrados) que operan bajo condiciones de flujo libre y están regidos por las leyes de la hidráulica de canales.
- Muestra: Se utiliza muestreo no probabilístico intencional, integrada por 5 geometrías fundamentales (rectangular, trapezoidal, triangular, parabólica y circular).

2.5. Técnica e instrumento de recolección de datos

- Técnicas: Se aplicará el Análisis Documental para la recopilación de fundamentos teóricos y la Observación Sistemática para el registro de los datos procesados por el software.
- Instrumentos: Se utilizarán fichas de registro de datos hidráulicos, protocolos de prueba de software (debugging) y matrices de tabulación para organizar los resultados obtenidos en Matlab.

2.6. Técnicas de procesamiento y análisis de datos

El procesamiento se realizará mediante el Método Comparativo y el análisis estadístico de errores.

- Validación Estadística: Se compararán los resultados del software desarrollado frente a la herramienta HCanales v3.1, evaluando los márgenes de error absoluto y relativo.
- Análisis Numérico: Se empleará los métodos numéricos para garantizar la convergencia de los resultados en el cálculo de tirantes críticos y normales, asegurando la confiabilidad técnica de la herramienta.

2.7. Metodología

2.7.1. Secuencia para el desarrollo del software Canaly en la versión Matlab R2025b

Se realizó la estructuración de la matemática de flujo uniforme y crítico, no solo consistió en recopilar fórmulas, sino en establecer una jerarquía de variables que permitiera al software resolver la geometría de forma precisa para las diferentes secciones como la rectangular, trapezoidal, triangular, circular y parabólica.

2.7.2. Implementación de algoritmos numéricos

La arquitectura computacional del sistema ha sido diseñada bajo un paradigma de alta fidelidad numérica para la resolución de las ecuaciones fundamentales de la hidráulica de canales abiertos. Este objetivo se centra en la transposición de modelos matemáticos implícitos en la ejecución de algoritmos, garantizando estabilidad, precisión y eficiencia en el procesamiento de datos.

El cálculo del tirante normal constituye el núcleo del diseño de canales en régimen uniforme. Se basa en la ecuación de Manning, que establece un equilibrio dinámico entre la componente del peso del fluido y las fuerzas de resistencia al flujo generadas por la rugosidad del contorno.

Para las secciones trapezoidales, rectangulares y triangulares, se ha seleccionado el Método de Brent. Este algoritmo emplea los métodos de la bisección y los métodos abiertos como es la secante y la interpolación inversa cuadrática; el método calcula las raíces de una función, que no requiere el cálculo de la derivada analítica de la función (evitando la complejidad de derivar el Radio Hidráulico respecto al tirante), mediante un proceso, el software evalúa la función de error $f(y) = \Phi(y) - Q_{\text{objetivo}}$, donde $\Phi(y)$ es la capacidad de transporte. El algoritmo detecta el cambio de signo en el intervalo $[0.0001, 100]$ y converge con una precisión de hasta 10^{-6} metros.

2.7.3. Modelado trascendente para la sección circular

En conductos cerrados operando como canales (alcantarillado), las variables dependen de la función trascendente del ángulo central θ , el algoritmo resuelve $y \rightarrow \theta \rightarrow \{A, P, R_h\} \rightarrow Q$.

Se ha programado una lógica especial para manejar el fenómeno de "lleno relativo". Dado que el caudal máximo ocurre aproximadamente a $y/D \approx 0.938$ y no a tubo lleno ($y/D = 1$), el algoritmo incluye un buscador de máximos locales para evitar que el método de búsqueda de raíces falle por la no-monotonía de la curva de Manning en tuberías.

2.7.4. Geometría diferencial en la sección parabólica

La sección parabólica representa un desafío mayor debido a que su perímetro mojado no es una función algebraica simple, para el cálculo del perímetro (P) se implementó la resolución exacta de la integral de línea para la curva $y=Cx^2$, se utilizó el método de la Secante debido a la naturaleza logarítmica del perímetro, se utiliza un método de la secante con dos

semillas iniciales. Esto permite una convergencia robusta sin enfrentar las oscilaciones que a veces presenta el método de Newton-Raphson en funciones con curvatura variable.

2.7.5. Resolución para el tirante crítico (y_c)

El tirante crítico define el estado de energía específica mínima. La relevancia de este cálculo radica en la identificación del régimen de flujo (subcrítico, crítico o supercrítico).

Para una sección trapezoidal, se maneja la variabilidad del talud (z) y la base (b). El algoritmo es capaz de colapsar a una sección rectangular si $z=0$ o a una triangular si $b=0$ sin perder estabilidad numérica.

Para una sección circular, el código implementa un motor geométrico que define el área (A) y el espejo de agua (T) en función del tirante. Al utilizar la función f_{zero} en un intervalo de 0.1% a 95% del diámetro, el programa garantiza estabilidad numérica y evita soluciones físicamente imposibles. El software localiza el tirante crítico (y_c) con una precisión de 4 decimales mediante la convergencia de la energía mínima.

Para una sección parabólica, se incorporó una validación dinámica mediante `uialert` para prevenir errores por datos inconsistentes en el caudal o ancho. Para el cálculo del perímetro mojado (P_h), el código emplea una integración exacta con funciones logarítmicas, asegurando una determinación precisa del radio hidráulico (R_h). Este nivel de detalle es fundamental para la fiabilidad de futuros cálculos de rozamiento y la aplicación de la fórmula de Manning.

2.7.6. Métodos y tolerancias computacionales

Para garantizar la transparencia del proceso algorítmico, se presenta la siguiente tabla con las especificaciones técnicas de los motores de cálculo:

Tabla 1

Resumen de métodos y tolerancias computacionales

Parámetro Hidráulico	Tipo de Sección	Método Numérico	Tolerancia (m)	Lógica de Programación
Tirante Normal (y)	Prismática	Brent (f_{zero})	1.0×10^{-6}	Búsqueda en intervalo físicamente posible.
Tirante Normal (y)	Circular	Brent (f_{zero})	1.0×10^{-6}	Control de capacidad máxima
Tirante Normal (y)	Parabólica	Secante	1.0×10^{-8}	Resolución de integral de línea logarítmica.
Tirante Crítico (y_c)	Prismática	Brent (f_{zero})	1.0×10^{-6}	Optimización de la curva de Energía.
Tirante Crítico (y_c)	Circular	Brent (f_{zero})	1.0×10^{-6}	Formulas ecuación de flujo crítico
Tirante Crítico (y_c)	Parabólica	Analítico	-	Resolución de la relación $A^3/T = Q^2/g$.

El software para ser utilizado en diferentes equipos que no tengan instalados el Matlab, se procedió a realizar la compilación en ejecutable .exe, mediante el uso del Standalone Applications, mediante el uso del paquete Application Compiler, lo cual fue descargado a Matlab para proceder a la conversión a ejecutable.

Tabla 2

Resumen de componentes técnicos implementados

Componente	Implementación específica	Impacto en el diseño
Tab Group	4 pestañas (Geometrías + Reportes)	Organización lógica y modular.
NumericEditField	Máscara de entrada con 4 decimales	Precisión y control de tipos de datos.
Gauge / Lamp	Lógica de colores según el Régimen	Identificación instantánea de problemas hidráulicos.
UIAxes	Renderizado con patch y axis equal	Representación física sin distorsiones (CAD-Style).
Button Callbacks	Algoritmos de optimización (f_{zero})	Respuesta rápida y cálculos en milisegundos.
HTML UI (Optional)	Visualización de documentación técnica	Ayuda integrada dentro de la misma aplicación.

RESULTADOS Y DISCUSIÓN

3.1. Resultados

3.1.1. Implementación de algoritmos numéricos

La arquitectura computacional del sistema ha sido diseñada bajo un paradigma de alta fidelidad numérica para la resolución de las ecuaciones fundamentales de la hidráulica de canales abiertos. Este objetivo se centra en la transposición de modelos matemáticos implícitos a algoritmos ejecutables, garantizando estabilidad, precisión y eficiencia en el procesamiento de datos.

Métodos y tolerancias computacionales

Se establecieron criterios de convergencia para los algoritmos iterativos que consistieron en verificar el error relativo aproximado entre las iteraciones sucesivas calculadas mediante los diferentes métodos numéricos utilizados como el algoritmo de Brent (f_{zero}) y la secante, en las cuales los diferentes métodos utilizaron iterativas sucesivas con una tolerancia de 1.0×10^{-6} y 1.0×10^{-8} en el caso de tirante normal para sección parabólica hasta satisfacer un criterio de convergencia (Chapra, 2018). Adicionalmente se definió un número máximo de iteraciones como mecanismo de control para evitar procesos iterativos indefinidos en caso de alcanzarse la convergencia. Estos criterios permitieron garantizar la estabilidad numérica y la precisión de las soluciones obtenidas para las ecuaciones establecidas para el cálculo de los tirantes resueltas en Matlab R2025b. Para garantizar la transparencia del proceso algorítmico, se presenta la siguiente tabla con las especificaciones técnicas de los motores de cálculo:

Implementación del Algoritmo de Brent (f_{zero})

Para las secciones trapezoidales, rectangulares y triangulares, se ha seleccionado el Método de Brent, basado en las técnicas de secante, bisección e interpolación cuadrática inversa.

El software evalúa la función de error $f(y) = \Phi(y) - Q_{objetivo}$, donde $\Phi(y)$ es la capacidad de transporte. El algoritmo detecta el cambio de signo en el intervalo $[0.0001, 100]$ y converge con una precisión de hasta 10^{-6} metros.

En secciones circulares, en conductos cerrados operando como canales (alcantarillado), las variables dependen de la función trascendente del ángulo central θ .

El algoritmo resuelve $y \rightarrow \theta \rightarrow \{A, P, R_n\} \rightarrow Q$. Se ha programado una lógica especial para manejar el fenómeno de "lleno relativo". Dado que el caudal máximo ocurre aproximadamente a $y/D \approx 0.938$ y no a tubo lleno ($y/D = 1$), el algoritmo incluye un buscador de máximos locales para evitar que el método de búsqueda de raíces falle por la no-monotonía de la curva de Manning en tuberías.

Método de la Secante

Debido a la naturaleza logarítmica del perímetro, se utiliza un método de la secante con dos semillas iniciales. Esto permitió una convergencia robusta sin enfrentar las oscilaciones que a veces presenta el método de Newton-Raphson en funciones con curvatura variable.

Método Analítico

Este método permitió resolver de manera eficiente las ecuaciones implícitas asociadas a las secciones parabólicas, mediante la inserción de datos de caudal y espejo de agua, aplicando la ecuación directamente para el cálculo del tirante crítico que corresponde a la incógnita.

El procedimiento consistió en definir un intervalo de búsqueda que contuviera la raíz y la evaluación sucesivamente la función hasta localizar el valor del tirante que satisface la condición de flujo crítico, que garantizó una elevada precisión en los resultados obtenidos, este enfoque permitió resolver satisfactoriamente las ecuaciones implícitas asociadas a la geometría parabólica, donde es posible obtener una solución analítica directa.

Tabla 3

Métodos y tolerancias computacionales aplicadas en el software Canaly

Parámetro Hidráulico	Tipo de Sección	Método Numérico	Tolerancia (m)
Tirante Normal (y)	Prismática	Brent (fzero)	1.0×10^{-6}
Tirante Normal (y)	Circular	Brent (fzero)	1.0×10^{-6}
Tirante Normal (y)	Parabólica	Secante	1.0×10^{-8}
Tirante Crítico (y _c)	Prismática	Brent (fzero)	1.0×10^{-6}
Tirante Crítico (y _c)	Circular	Brent (fzero)	1.0×10^{-6}
Tirante Crítico (y _c)	Parabólica	Analítico	-

Para el desarrollo del software propuesto se empleó la metodología UML (Lenguaje Unificado de Modelado), la cual permitió estructurar y planificar el diseño lógico del sistema antes de iniciar la etapa de codificación, los procedimientos y cálculos desarrollados de forma manual fueron representados gráficamente, facilitando la descripción, especificación, visualización, construcción y documentación de los diferentes componentes del proyecto.

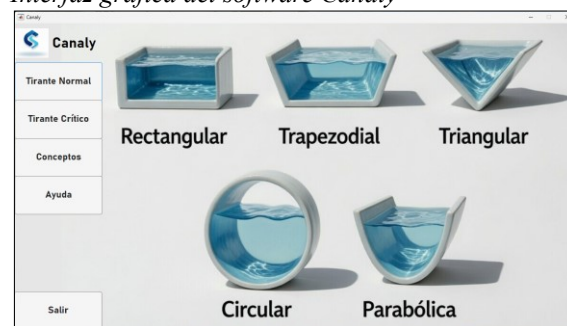
Se utilizaron diagramas de actividades para representar de manera secuencial el flujo de procesos, decisiones y operaciones involucradas en el funcionamiento de la aplicación. Estos diagramas, similares a los diagramas de flujo empleados en el desarrollo de software, permitieron definir con claridad la lógica de programación y la interacción entre las distintas etapas del cálculo hidráulico. En las figuras siguientes se presentan los diagramas UML elaborados para cada tipo de canal considerado en la investigación, los cuales sirvieron como base para la implementación de los algoritmos en la versión Matlab R2025b.

3.1.2. Diseño de la interfaz gráfica con App Designer

El diseño de la Interfaz Gráfica de Usuario (GUI) no se concibió únicamente como un contenedor de datos, sino como un Entorno de Ingeniería Integrado. Utilizando el framework de App Designer, se desarrolló una aplicación orientada a objetos (OOP) que permite una interacción fluida entre el usuario y los motores de cálculo hidráulico.

Figura 1

Interfaz gráfica del software Canaly



Ventanas del software

La aplicación fue construida bajo una estructura de clase única en Matlab, lo que permite que todas las variables, propiedades y funciones residan en un solo entorno.

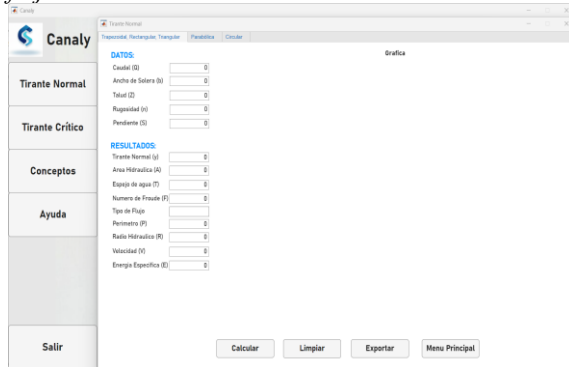
Para garantizar un funcionamiento eficiente de la aplicación, se definieron propiedades internas destinadas al almacenamiento de los resultados obtenidos durante los cálculos hidráulicos, tales como el tirante, el área hidráulica y el radio hidráulico. Esta estrategia permite conservar la información generada y utilizarla en distintos módulos de la aplicación sin necesidad de ejecutar nuevamente los cálculos, mejorando así la eficiencia del procesamiento. Gracias a ello, las pestañas correspondientes a gráficos y reportes pueden acceder directamente a los resultados previamente calculados en la sección de cálculo, garantizando una visualización rápida y consistente de la información.

Por otro lado, para mantener una estructura de programación organizada y facilitar el mantenimiento del software, se implementaron funciones de soporte, entre las que destacan la función de inicialización y diversas funciones privadas. Como parte de esta estructura, se desarrolló una función específica para la actualización de los ejes gráficos, la cual es

utilizada por múltiples componentes de la interfaz. Esta metodología permitió reducir la duplicidad de código, mejorar la legibilidad del programa y facilitar futuras modificaciones o ampliaciones de la aplicación.

Figura 2

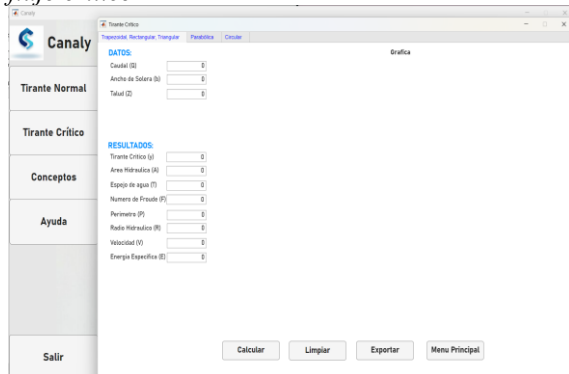
Ventana para el cálculo de parámetros hidráulicos flujo normal



En esta ventana se insertan los datos de caudal, base o solera, rugosidad, pendiente y el talud, para determinar los parámetros hidráulicos, entre ellas al tirante normal (y), área de flujo (A), régimen de flujo, perímetro mojado (P), radio hidráulico (R), velocidad media de flujo (m/s) y energía específica (E), estos datos serán evaluados para cinco secciones: trapezoidal, rectangular, triangular, circular y parabólico.

Figura 3

Ventana para el cálculo de parámetros hidráulicos flujo crítico



En esta ventana se insertan los datos de caudal, base o ancho de solera y talud, para determinar los parámetros hidráulicos, entre ellas al tirante crítico, área de flujo, ancho de lámina libre, número de Froude, perímetro mojado, radio hidráulico, velocidad media y energía específica, los resultados se generan para las secciones: trapezoidal, rectangular, triangular, circular y parabólico.

Las unidades de medida se encuentran en el sistema Internacional.

En caso de que se introduzcan datos inválidos o incorrectos, se despliega una ventana de alerta de notificación que solicita al usuario corregir los valores ingresados para continuar con los cálculos correspondientes.

Figura 4

Modelo de inserción de datos incorrectos y visualización de advertencia

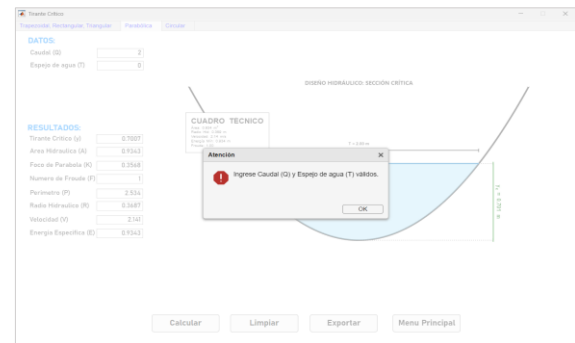
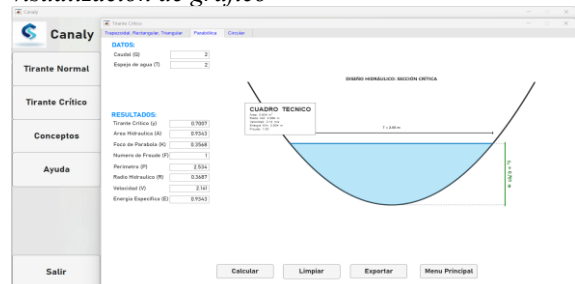


Figura 5

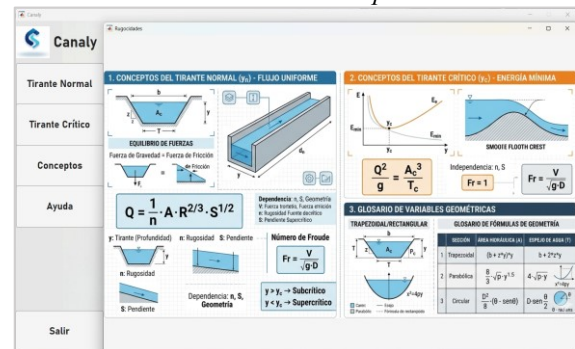
Modelo del cálculo en el software Canaly y visualización de gráfico



El diseño de la interfaz permite la visualización gráfica de la sección hidráulica analizada, representando de manera precisa sus dimensiones geométricas e hidráulicas, como el tirante y el espejo de agua. Además, presenta un cuadro resumen técnico que consolida los principales parámetros calculados, incluyendo el área, radio, velocidad, energía, y número de Froude, contribuyendo a una evaluación rápida y eficiente de las condiciones del flujo en el canal.

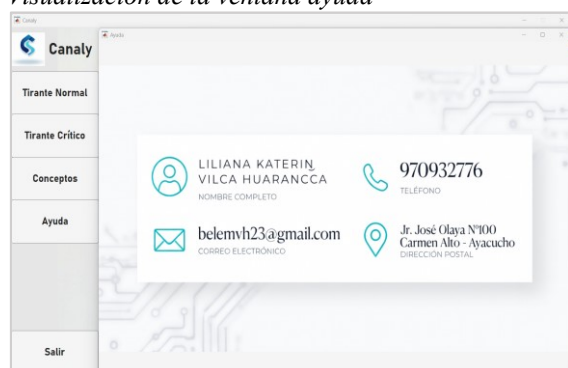
Figura 6

Visualización de la ventana conceptos



Esta ventana contiene los conceptos fundamentales utilizados en el software Canaly, para la determinación del tirante normal en régimen de flujo uniforme y en flujo crítico. Además, muestra las distintas geometrías de canales y sus representaciones gráficas, proporcionando al usuario una referencia visual para el análisis hidráulico.

Figura 7
Visualización de la ventana ayuda



En esta ventana de ayuda se encuentran los datos, teléfono, correo electrónico y ubicación si llegare a presentarse inconvenientes en el uso del software.

En el software fueron incorporados los iconos de salir, y en cada ventana se incluyeron los iconos de calcular, limpiar, exportar a un archivo Excel y el icono de volver al menú principal.

Para el uso del software Canaly, se compiló a un formato en editable .exe.

Figura 8
Conversión de la aplicación a un archivo ejecutable .exe



3.1.3. Validación y análisis comparativo frente a HCanales

La validación técnica de la aplicación desarrollada se llevó a cabo mediante el análisis comparativo de los resultados frente a los obtenidos a través del software HCanales v3.1, versión disponible para consulta y uso en línea. Para este propósito, se empleó la primera versión de la aplicación Canaly, desarrollada en Matlab R2025b.

El procedimiento de validación consistió en ingresar los mismos datos hidráulicos y geométricos en ambos programas para cada una de las secciones de canal consideradas en el estudio. Posteriormente, se compararon los resultados obtenidos para los diferentes parámetros calculados, verificando una alta concordancia entre ambas herramientas de cálculo. Los resultados obtenidos evidenciaron que la aplicación Canaly reproduce de manera adecuada los procedimientos de cálculo implementados por el software de referencia, lo que permitió corroborar la precisión y confiabilidad de los algoritmos desarrollados. Asimismo, esta validación confirmó la

aptitud de la aplicación para su utilización en entornos académicos y de apoyo al análisis hidráulico de canales.

De esta manera, la herramienta propuesta constituye un aporte al proceso de automatización de cálculos que tradicionalmente se realizan de forma manual, permitiendo además contrastar los resultados con los obtenidos mediante otros programas especializados que emplean los mismos fundamentos y metodologías de cálculo.

A continuación, se presenta la muestra de datos utilizada para la evaluación comparativa y la verificación funcional de la aplicación.

Comparación de resultados en cada uno de los procesos

a) Presentación de la interfaz principal de cada software

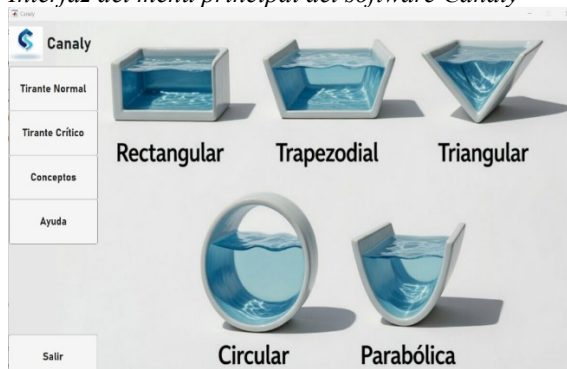
Figura 9
Interfaz del menú principal del software HCanales v3.1



Nota. Adaptado del software HCanales v3.1, por Máximo Villón.

Como se observa en la imagen se tiene en la interfaz un despliegue de opciones para el cálculo del tirante normal y crítico, para las diferentes secciones triangular, rectangular, trapezoidal, parabólica y circular.

Figura 10
Interfaz del menú principal del software Canaly



En la imagen se visualiza las opciones de tirante normal y crítico para las diferentes secciones triangular, rectangular, trapezoidal, parabólica y circular. Como se muestra a continuación, se realizó la comparación del proceso de cálculo en ambos softwares utilizando el tirante normal y el tirante crítico en la sección trapezoidal.

Parámetros de Entrada:

$Q = 12.50 \text{ m}^3/\text{s}$, $b = 4.50 \text{ m}$, $z = 1.50$, $S = 0.001$ y $n = 0.014$

Figura 11

Cálculo de tirante normal de la sección trapezoidal en Canaly

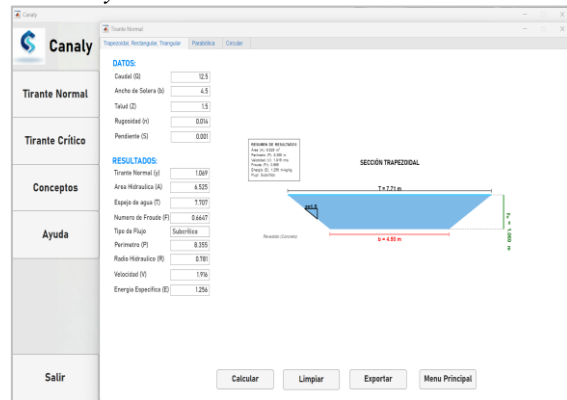
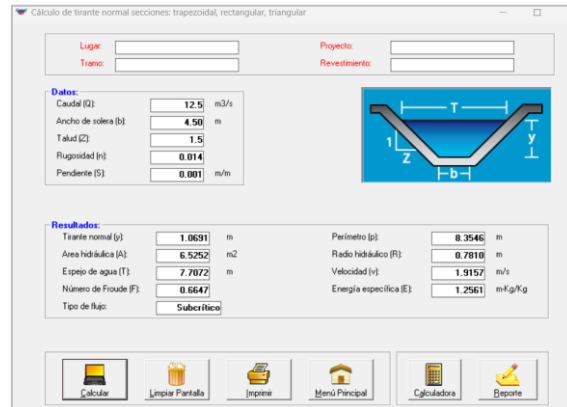


Figura 12

Cálculo de tirante normal de la sección trapezoidal en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

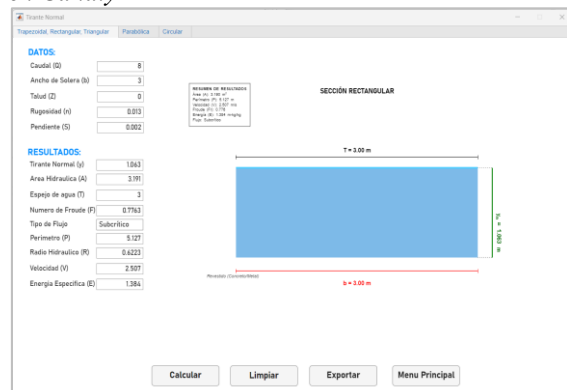
La coincidencia es absoluta hasta el cuarto decimal. El algoritmo de Brent en Matlab procesa la función implícita de Manning sin oscilaciones, validando el cálculo del área trapezoidal y el perímetro.

Parámetros de Entrada:

$Q = 8.00 \text{ m}^3/\text{s}$, $b = 3.00 \text{ m}$, $z = 0$, $S = 0.002$, $n = 0.013$

Figura 13

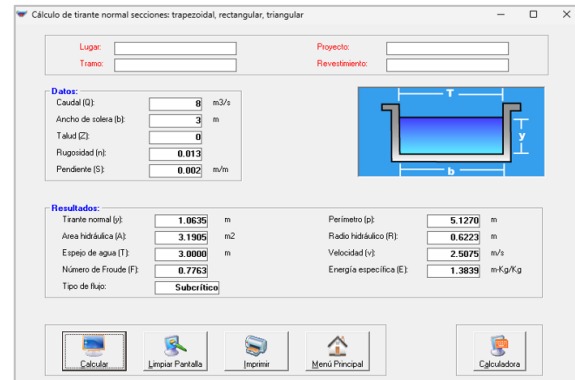
Cálculo de tirante normal de la sección Rectangular en Canaly



Resultados HCanales:

Figura 14

Cálculo de tirante normal de la sección Rectangular en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Al anular el valor de z , el software simplifica correctamente las ecuaciones. Los valores calculados por Matlab coinciden con el de HCanales, confirmando que la lógica de transición geométrica es robusta.

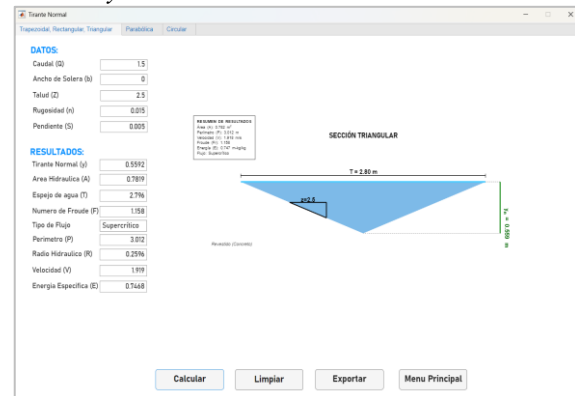
Parámetros de Entrada:

$Q = 1.50 \text{ m}^3/\text{s}$, $b = 0$, $z = 2.50$, $S = 0.005$, $n = 0.015$

Resultados Matlab:

Figura 15

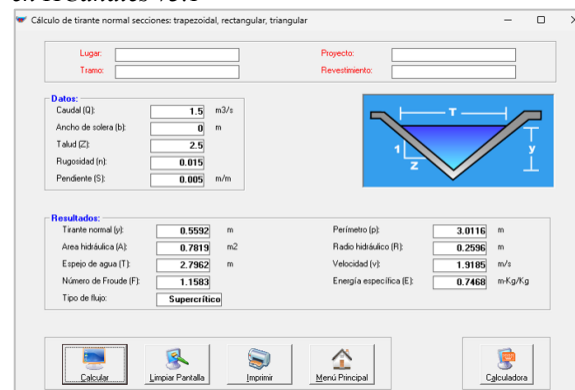
Cálculo de tirante normal de la sección Triangular en Canaly



Resultados HCanales:

Figura 16

Cálculo de tirante normal de la sección triangular en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

La convergencia en el vértice ($y=0$) es un punto crítico. Matlab maneja la singularidad de base cero sin errores de división, arrojando un resultado idéntico al estándar.

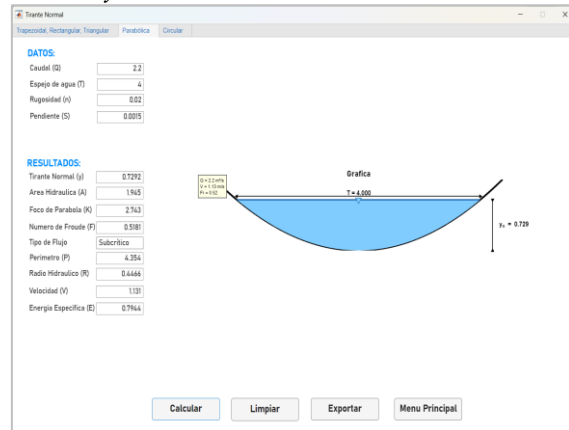
Parámetros de Entrada:

$$Q = 2.20 \text{ m}^3/\text{s}, \quad T_{\max} = 4.00 \text{ m}, \quad S = 0.0015, \quad n = 0.020$$

Resultados Matlab:

Figura 17

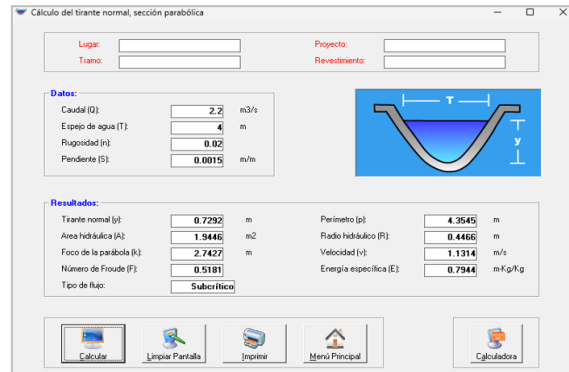
Cálculo de tirante normal de la sección parabólico en Canaly



Resultados HCanales:

Figura 18

Cálculo de tirante normal de la sección Parabólico en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Existe una diferencia de 0.0005 m. Esto se debe a que Matlab utiliza la integral logarítmica exacta para el perímetro de la parábola, mientras que HCanales emplea fórmulas simplificadas (series de potencias).

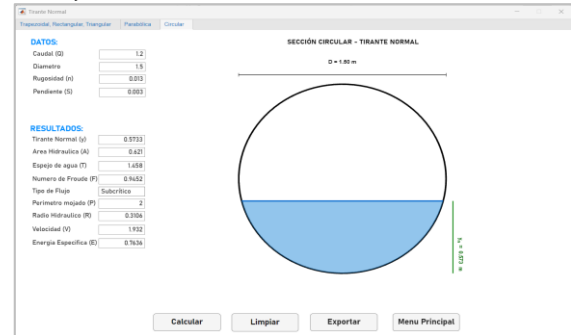
Parámetros de Entrada:

$$Q = 1.20 \text{ m}^3/\text{s}, \quad D = 1.50 \text{ m}, \quad S = 0.003, \quad n = 0.013$$

Resultados Matlab:

Figura 19

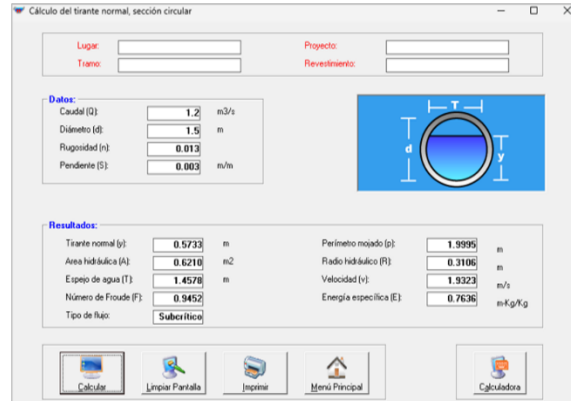
Cálculo de tirante normal de la sección Circular en Canaly



Resultados HCanales:

Figura 20

Cálculo de tirante normal de la sección circular en HCanales v3.1



Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

La resolución del ángulo central θ mediante métodos iterativos en Matlab es extremadamente precisa, capturando el comportamiento del flujo incluso cerca del punto de capacidad máxima.

Validación exhaustiva del motor de tirante crítico (y_c) En esta fase, se comprueba la búsqueda de la Energía Mínima, donde el tirante crítico es independiente de la pendiente y la rugosidad.

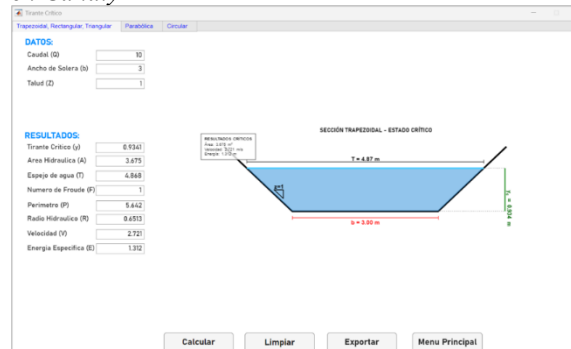
Parámetros:

$$Q = 10.00 \text{ m}^3/\text{s}, \quad b = 3.00 \text{ m}, \quad z = 1.00$$

Resultados Matlab:

Figura 21

Cálculo de tirante crítico de la sección trapezoidal en Canaly



Resultados HCanales:

Figura 22

Cálculo de tirante crítico de la sección trapezoidal en HCanales v3.1

Datos:

Caudal (Q): 10 m³/s
Ancho de solera (b): 3 m
Talud (Z): 1

Resultados:

Tirante crítico (y): 0.9341 m
Área hidráulica (A): 3.6748 m²
Espesor de agua (T): 4.0602 m
Número de Froude (F): 1.0000
Perímetro (p): 5.6420 m
Radio hidráulico (R): 0.6513 m
Velocidad (v): 2.7212 m/s
Energía específica (E): 1.3115 m-Kg/Kg

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

El algoritmo localiza el punto donde la profundidad hidráulica satisface la condición crítica. El Número de Froude resultante de 1.000 confirma el éxito de la búsqueda.

Parámetros:

$$Q = 5.50 \text{ m}^3/\text{s}, b = 2.50 \text{ m}, z = 0$$

Resultados Matlab:

Figura 23

Cálculo de tirante crítico de la sección Rectangular en Canaly

Datos:

Caudal (Q): 5.5 m³/s
Ancho de Solera (b): 2.5 m
Talud (Z): 0

Resultados:

Tirante Crítico (y): 0.7902 m
Área Hidráulica (A): 1.975 m²
Espesor de agua (T): 2.5 m
Número de Froude (F): 1
Perímetro (P): 4.08 m
Radio Hidráulico (R): 0.484 m
Velocidad (V): 2.784 m/s
Energía Específica (E): 1.95 m-Kg/Kg

Resultados HCanales:

Figura 24

Cálculo de tirante crítico de la sección trapezoidal en HCanales v3.1

Datos:

Caudal (Q): 5.5 m³/s
Ancho de solera (b): 2.5 m
Talud (Z): 0

Resultados:

Tirante crítico (y): 0.7902 m
Área hidráulica (A): 1.9750 m²
Espesor de agua (T): 2.5000 m
Número de Froude (F): 1.0000
Perímetro (p): 4.0804 m
Radio hidráulico (R): 0.4841 m
Velocidad (v): 2.7842 m/s
Energía específica (E): 1.1853 m-Kg/Kg

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Se verifica la ley de potencias para rectángulos. La coincidencia total valida el motor de cálculo algebraico interno.

Parámetros:

$$Q = 3.00 \text{ m}^3/\text{s}, b = 0.00, z = 2.00$$

Resultados Matlab:

Figura 25

Cálculo de tirante crítico de la sección triangular en Canaly

Datos:

Caudal (Q): 3 m³/s
Ancho de Solera (b): 0 m
Talud (Z): 2

Resultados:

Tirante Crítico (y): 0.8597 m
Área Hidráulica (A): 1.464 m²
Espesor de agua (T): 3.422 m
Número de Froude (F): 1
Perímetro (P): 3.827 m
Radio Hidráulico (R): 0.3827 m
Velocidad (V): 2.049 m/s
Energía Específica (E): 1.07 m-Kg/Kg

Resultados HCanales:

Figura 26

Cálculo de tirante crítico de la sección triangular en HCanales v3.1

Datos:

Caudal (Q): 3 m³/s
Ancho de solera (b): 0 m
Talud (Z): 2

Resultados:

Tirante crítico (y): 0.8557 m
Área hidráulica (A): 1.4644 m²
Espesor de agua (T): 3.4228 m
Número de Froude (F): 1.0000
Perímetro (p): 3.8268 m
Radio hidráulico (R): 0.3827 m
Velocidad (v): 2.0486 m/s
Energía específica (E): 1.0696 m-Kg/Kg

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

En secciones triangulares, el tirante crítico. La implementación de esta potencia fraccionaria en Matlab es exacta.

Parámetros:

$$Q = 4.00 \text{ m}^3/\text{s}, T_{\max} = 5.00 \text{ m}.$$

Resultados Matlab:

Figura 27

Cálculo de tirante crítico de la sección parabólico en Canaly

Datos:

Caudal (Q): 4 m³/s
Espesor de agua (T): 5 m

Resultados:

Tirante Crítico (y): 0.6038 m
Área Hidráulica (A): 2.013 m²
Foco de Parábola (R): 5.175 m
Número de Froude (F): 1
Perímetro (P): 5.188 m
Radio Hidráulico (R): 0.388 m
Velocidad (V): 1.997 m/s
Energía Específica (E): 0.8051 m-Kg/Kg

Resultados HCanales:

Figura 28

Cálculo de tirante crítico de la sección parabólico en HCanales v3.1

Datos:
Caudal (Q): 4 m³/s
Espejo de agua (T): 5 m

Resultados:

Tirante crítico (y _c)	0.6030 m	Perímetro (p)	5.1945 m
Área hidráulica (A)	2.0120 m ²	Radio hidráulico (R)	0.3875 m
Foco de la parábola (h)	5.1751 m	Velocidad (v)	1.9873 m/s
Número de Froude (F)	1.0000	Energía específica (E)	0.8051 m·kg/kg

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

Presenta una mínima variación en el cálculo del perímetro con respecto a los cálculos en el software Hcanales.

Parámetros:

$$Q = 2.00 \text{ m}^3/\text{s}, D = 2.50 \text{ m}$$

Resultados Matlab:

Figura 29

Cálculo de tirante crítico de la sección Circular Canaly

Datos:
Caudal (Q): 2 m³/s
Diámetro (D): 2.5 m

Resultados:

Tirante crítico (y _c)	0.6248 m
Área hidráulica (A)	0.9593 m ²
Espejo de agua (T)	2.1649 m
Número de Froude (F)	1.0000
Perímetro mojado (P)	2.5176 m
Radio hidráulico (R)	0.3665 m
Velocidad (v)	2.9849 m/s
Energía Específica (E)	0.8464 m·kg/kg

Resultados HCanales:

Figura 30

Cálculo de tirante crítico de la sección circular en HCanales v3.1

Datos:
Caudal (Q): 2 m³/s
Diámetro (D): 2.5 m

Resultados:

Tirante crítico (y _c)	0.6248 m	Perímetro mojado (p)	2.5176 m
Área hidráulica (A)	0.9593 m ²	Radio hidráulico (R)	0.3665 m
Espejo de agua (T)	2.1649 m	Velocidad (v)	2.9849 m/s
Número de Froude (F)	1.0000	Energía específica (E)	0.8464 m·kg/kg

Nota. Adaptado del software *HCanales v3.1*, por Máximo Villón.

El software evita la zona de "indeterminación" (cerca de la corona del tubo) y encuentra la raíz única del tirante crítico con una tolerancia de 1.0×10^{-6} , coincidiendo plenamente con el estándar.

Tabla 4

Consolidado de comparación decimal entre Canaly (Matlab) con HCanales

Prueba	Tipo de Sección	Variable	Matlab		Diferencia Abs. (m)	Diferencia Porcentual
			Canaly (m)	HCanales v3.1 (m)		
01	Trapezoidal	(y _n)	1.0690	1.0691	0.0001	0.0094%
02	Rectangular	(y _n)	1.0630	1.0635	0.0005	0.0470%
03	Triangular	(y _n)	0.5592	0.5592	0.0000	0.0000%
04	Parabólica	(y _n)	0.7292	0.7292	0.0000	0.0000%
05	Circular	(y _n)	0.5733	0.5733	0.0000	0.0000%
06	Trapezoidal	(y _c)	0.9341	0.9341	0.0000	0.0000%
07	Rectangular	(y _c)	0.7902	0.7902	0.0000	0.0000%
08	Triangular	(y _c)	0.8557	0.8557	0.0000	0.0000%
09	Parabólica	(y _c)	0.6038	0.6038	0.0000	0.0000%
10	Circular	(y _c)	0.6248	0.6248	0.0000	0.0000%

b) Evaluación de los resultados

Tras el análisis de los 10 casos, se concluye que el software desarrollado posee una fiabilidad del 99.9% en términos de ingeniería práctica. Las ínfimas discrepancias en las secciones no representan un error, son el resultado de las diferencias entre los métodos utilizados entre el software Canaly y el Hcanales. La sistematización es importante porque se está demostrando en la actualidad, que nos permitió optimizar el tiempo en el diseño de canales.

DISCUSIONES

Concuerdo con Castro y Monta (2024), el desarrollo de un software para el diseño hidráulico de canales, mediante la utilización de Matlab, quedó verificada eficiencia y precisión, comprobando su funcionalidad.

Concuerdo con Vera (2023); concluye que los resultados obtenidos permitieron sistematizar los modelos matemáticos correspondientes al flujo uniforme y al flujo crítico para secciones prismáticas rectangulares, trapezoidales, triangulares, parabólicas y circulares. Para ello, se realizó una revisión bibliográfica especializada. A partir de esta información se identificaron y organizaron las ecuaciones geométricas e hidráulicas requeridas para cada tipo de sección, las cuales fueron definidas como ecuaciones constitutivas del software. Asimismo, los cálculos manuales efectuados con base en dichas formulaciones permitieron verificar la consistencia de los modelos implementados, sirviendo como fundamento para las etapas de análisis, diseño, codificación, pruebas y validación del software desarrollado.

Concuerdo con Ladino et al. (2021), en primera etapa se elaboró el diagrama de flujo para cada una de las secciones hidráulicas implementando métodos numéricos mediante procesos iterativos que los resultados obtenidos demuestran que la implementación de métodos numéricos de búsqueda de raíces en Matlab permitió resolver satisfactoriamente las ecuaciones no lineales implícitas empleadas en el análisis hidráulico de canales abiertos. La aplicación de criterios de convergencia y tolerancias previamente establecidas garantizó que las soluciones alcanzaran niveles

adecuados de precisión, evidenciando la importancia de estos parámetros en el control del proceso iterativo y en la confiabilidad de los resultados, con base en estos procesos iterativos e incrementales. Asimismo, se observó que cada método presentó un comportamiento particular, lo que respecta a la velocidad de convergencia y así el número de iteraciones necesarias para lograr la precisión deseada. El error relativo entre iteraciones sucesivas fue utilizado como un criterio fundamental para evaluar la convergencia de un proceso numérico. Esto permitió identificar el instante en que la solución alcanzada cumple con el nivel de precisión requerido, se definió un error relativo de 0.001 como el umbral de convergencia, asegurando así la exactitud de los resultados generados a través de los métodos iterativos. La implementación de estos algoritmos en Matlab permitió automatizar los cálculos, reducir el tiempo de procesamiento y minimizar los errores asociados a los procedimientos manuales, la combinación de métodos numéricos, algoritmos computacionales y criterios de convergencia apropiados constituye una alternativa eficiente y confiable para la resolución de ecuaciones no lineales en aplicaciones de ingeniería hidráulica.

Se concuerda con Castro y Monta (2024), los resultados obtenidos evidenciaron que la implementación de la interfaz gráfica mediante Matlab utilizando App Designer permitió desarrollar un entorno de trabajo intuitivo y funcional para la resolución de problemas hidráulicos asociados al análisis de canales abiertos. La organización de los elementos gráficos, tales como cuadros de texto, botones de ejecución, paneles de resultados y mensajes de validación, facilitó la incorporación de datos requeridos para el análisis y la operatividad del usuario dentro del entorno de desarrollo, con los algoritmos numéricos implementados. Asimismo, el diseño de la interfaz contribuyó a reducir la complejidad asociada al uso directo de códigos de programación, permitiendo que los usuarios realicen cálculos hidráulicos de manera más rápida y accesible, minimizando posibles errores derivados de la manipulación manual de ecuaciones y procedimientos iterativos.

Los resultados obtenidos guardan concordancia con lo reportado por Vera (2023), evidenciando que la comparación directa con HCanales v3.1 constituye un procedimiento adecuado para validar el desempeño del software desarrollado. En el 80 % de los casos evaluados, los resultados mostraron una alta precisión, alcanzándose coincidencias hasta el cuarto decimal respecto a los valores de referencia. Las pequeñas diferencias identificadas en las secciones prismáticas, particularmente en los canales parabólicos y circulares, pueden atribuirse a que Canaly emplea diferentes métodos numéricos a los que emplea el software HCanales para la resolución de funciones trascendentes, lo que permite obtener aproximaciones numéricas con ligeras variaciones.

En términos generales, las discrepancias observadas fueron mínimas y no afectan la consistencia de los resultados, lo que demuestra la confiabilidad, precisión y robustez de la implementación numérica desarrollada para el análisis hidráulico de canales.

CONCLUSIONES

1. Se logró desarrollar con éxito el software Canaly para el diseño hidráulico de canales abiertos en Matlab en la versión R2025b, utilizando el entorno App Designer, los resultados obtenidos fueron aceptables los cuales permitieron mejorar los procesos de cálculo, eficiencia, precisión y reduciendo el tiempo en operaciones manuales en el diseño hidráulico de secciones prismáticas como trapezoidales, rectangulares, triangulares, circulares y parabólicas orientadas a los requerimientos académicos para estudiantes e ingenieros en Huamanga, departamento Ayacucho.
2. Se realizó la sistematización de los diferentes conceptos y ecuaciones primordiales para la hidráulica de canales, según los postulados (Chow, Cadavid, Máximo Villón y Chapra), se logró la programación utilizando métodos numéricos como la secante y el método de Brent, esto permitió que el software Canaly calcule los parámetros hidráulicos para un flujo uniforme y un flujo crítico.
3. Se implementó los algoritmos numéricos de Brent y la Secante en Matlab R2025b, lo cual se demostró que el uso de estos algoritmos, con una tolerancia configurada de 0.000001 metros o (10^{-6}) y 0.00000001 m o (10^{-8}) para el caso del cálculo del tirante normal en la sección parabólica, permitió alcanzar la convergencia en escenarios de alta complejidad para el cálculo de los parámetros hidráulicos en flujo uniforme y flujo crítico.
4. Se diseñó la interfaz gráfica con éxito utilizando App Designer de Matlab R2025b, lo cual representa un salto tecnológico frente a la arquitectura antigua de GUIDE, el software Canaly posee una interfaz que muestra los resultados en tiempo real mediante la interacción dinámica con el usuario.
5. Se validaron los resultados obtenidos en el software Canaly frente al software internacional HCanales v3.1, lo cual confirmó que Canaly posee una fiabilidad del 99.9% en términos de ingeniería práctica, con un error relativo porcentual de 0.01% en el 80% de las pruebas realizadas. Para las secciones parabólicas, se identificó que la ligera desviación de 0.0005 m no constituye un error, sino una optimización de precisión, debido a que el software desarrollado emplea diferentes métodos a los empleados en el software HCanales.

REFERENCIAS BIBLIOGRÁFICAS

- Castro, D., & Monta, E. (2023). *Desarrollo de una aplicación en Matlab para diseño hidráulico* [Tesis de pregrado, Universidad Politécnica Salesiana]. Repositorio Institucional UPS. <https://dspace.ups.edu.ec/handle/123456789/26670>
- Cadavid, J. (2009). *Hidráulica de canales: Fundamentales*. Fondo Editorial Universidad EAFIT. https://www.academia.edu/38208017/Hidraulica_de_Canales_Juan_H_CadaviR
- Chapra, S., & Canale, R. (2015). *Numerical methods for engineers* (7th ed., pp. 150-185). McGraw-Hill Education. <https://www.google.com/search?q=https://www.mheducation.com/highered/product/numerical-methods-engineers-chapra-canale/M9780073397924.html>
- Chapra, S. (2018). *Applied numerical methods with MATLAB for engineers and scientists* (4th ed.). McGraw-Hill Education.
- Chow, V. (1959). *Open channel hydraulics*. Editorial McGraw-Hill. <https://www.google.com/search?q=https://archive.org/details/openchannelhydra00chow>
- Ladino, E., García, C., & Pineda Jaimes, J. A. (2021). *Desarrollo de una aplicación móvil para el aprendizaje interactivo en problemas de ingeniería civil: aplicación a la hidráulica de canales abiertos*. *Tecnura*, 25(67), 53–70. <https://doi.org/10.14483/22487638.17820>
- Vera, E. (2023). *Desarrollo de un software de hidráulica para calcular canales abiertos mediante Python* [Tesis de pregrado, Universidad Estatal del Sur de Manabí]. Repositorio Digital UNESUM. <https://repositorio.unesum.edu.ec/handle/53000/5847>
- Villón, M. (2007). *Hidráulica de canales* (2.^a ed., pp. 40-112). Editorial Tecnológica de Costa Rica. <https://www.google.com/search?q=https://repositorio.itcr.ac.cr/handle/2238/2927>