



Escola Tècnica Superior d'Enginyeries
Industrial i Aeronàutica de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Curso: Grado en ingeniería en tecnologías aeroespaciales

Estudiante: Andrés Adam Alberdi

Título del proyecto:

Estudio del desarrollo de una nueva aplicación Android para flujo compresible

“App4Comp”

Director: Pedro Javier Gamez Montero

Tutor: Daniel Garcia-Almiñana

Fecha de entrega: 12 de junio de 2015

Contenido: Documento 1 - Memoria

Este proyecto fue realizado por:

Andrés Adam Alberdi

Director:

Pedro Javier Gamez Montero

Tutor:

Daniel Garcia-Almiñana

Trabajo Final de Grado

**ETSEIAT-UPC
12 de junio de 2015**

Agradecimientos

En primer lugar debo agradecer a mi director de proyecto, Xavier Gamez, por su excelente trabajo de guía, consejo y motivación para realizar este TFG. Él se lleva el mérito de proponer la idea de realizar una aplicación de flujo compresible, idea que me contagió y que me hizo dedicarme a ello como si hubiera sido mía.

Por supuesto, gracias a Mercedes Herreras, que ha sido un contacto vital y una fuente de ayuda infalible durante la realización del proyecto, y con la motivación por el trabajo que desprende, un ejemplo a seguir.

Gracias a mis compañeros Oriol Crespo y Joan Pallejà por dejarse convencer para trabajar en Matlab con algo relacionado con mi TFG, para después dejarme aprovechar el código surgido de sus notables mentes para ello.

Debo agradecer también a Laura Molero, por su trabajo probando la App y su inagotable energía que le hace estar siempre presente en mi vida, solo que cambiando de plano.

Muchas gracias a los demás compañeros, amigos y familia que han hecho su aportación en forma de apoyo en este trabajo, y por supuesto gracias al lector, pues todo esto lo escribí para ser leído.

Contenido

Contenido	i
Lista de Figuras	iii
Lista de tablas	v
1. Introducción. Planteamiento del problema y justificación.	1
1.1. Objetivo del proyecto	2
1.2. Alcance del proyecto.....	2
1.3. Requisitos.....	3
2. Estado del arte	4
2.1. El flujo compresible en la ingeniería	4
2.2. Lenguajes de programación para Android disponibles.....	5
2.3. Aplicaciones de flujo compresible en el mercado	7
3. Planteamiento de la solución. Funciones de la App, teoría y lenguaje de programación.	9
3.1. Decisión de funciones	9
3.2. Introducción teórica al flujo compresible	10
3.2.1. ¿Qué es?.....	10
3.2.2. <i>Introducción analítica y flujo isentrópico</i>	11
3.2.3. <i>Ondas de choque normales</i>	15
3.2.4. <i>Flujo isentrópico en conductos de área variable. Toberas.</i>	16
3.2.5. <i>Flujo de Fanno</i>	21
3.3. App Inventor	24
3.3.1. <i>Funcionamiento</i>	24
3.3.2. <i>Compilación y prueba del programa</i>	29
3.3.3. <i>Ventajas, inconvenientes y limitaciones</i>	29
4. Cumplimiento del alcance. Programación de la App	32
4.1. Primera pantalla	32
4.2. Atmósfera estándar internacional ISA	33
4.3. Flujo isentrópico	37

4.4.	Ondas de choque normales	41
4.5.	Tobera convergente-divergente	44
4.5.1.	<i>Modo tobera adaptada</i>	45
4.5.2.	<i>Modo genérico</i>	51
4.6.	Flujo de Fanno	58
4.7.	Dificultades encontradas.....	63
4.8.	Análisis de resultados	64
4.8.1.	<i>Atmósfera estándar internacional</i>	64
4.8.2.	<i>Flujo isentrópico</i>	66
4.8.3.	<i>Onda de choque normal</i>	69
4.8.4.	<i>Tobera convergente-divergente</i>	71
<i>Modo tobera adaptada</i>	71	
<i>Modo genérico</i>	73	
4.8.5.	<i>Flujo de Fanno</i>	75
5.	Conclusiones	78
5.1.	Ampliaciones y trabajo futuro.....	79
6.	Condiciones de software	81
6.1.	Licencia App Inventor 2.....	81
6.1.1.	<i>Requisitos para ordenador</i>	81
6.1.2.	<i>Requisitos para dispositivo Android</i>	81
6.2.	Licencia Matlab	82
6.2.1.	<i>Requisitos</i>	82
6.3.	Licencia Microsoft Word.....	82
6.3.1.	<i>Requisitos</i>	82
7.	Planificación	83
8.	Análisis medioambiental	85
8.1.	Referente al desarrollo del proyecto	85
8.2.	Referente a la aplicación del proyecto.....	85
	Bibliografía	86

Lista de Figuras

Ilustración 2.1. Diagramas comparadores de aplicaciones	8
Ilustración 3.1. Coeficiente de resistencia en función del Mach	12
Ilustración 3.2. Ejemplo de onda de choque normal (Fuente: http://www.psr.d.hawaii.edu [3])	15
Ilustración 3.3. Volumen de control para flujo cuasi-unidimensional	17
Ilustración 3.4. Distribución de presión en una tobera. (Fuente: http://www.engapplets.vt.edu [4])	19
Ilustración 3.5. Parte "Designer" de App Inventor.....	25
Ilustración 3.6. Ejemplo de visualización de componentes	27
Ilustración 3.7. Visualización de la parte "Blocks" de App Inventor	29
Ilustración 4.1. Primera pantalla de la aplicación	33
Ilustración 4.2. Presentación de la función ISA	35
Ilustración 4.3. Guía en ISA	36
Ilustración 4.4. Presentación de la función flujo isentrópico	38
Ilustración 4.5. Elección de propiedad dimensional y presentación de la función flujo isentrópico	39
Ilustración 4.6. Problema inverso en la función flujo isentrópico	39
Ilustración 4.7. Ilustración de onda de choque normal	41
Ilustración 4.8. Función onda de choque normal, sin y con inputs dimensionales	42
Ilustración 4.9. Guía en onda de choque normal.....	42
Ilustración 4.10. Ejemplo de tobera convergente-divergente	44
Ilustración 4.11. Presentación del modo tobera adaptada	45
Ilustración 4.12. Guía práctica en tobera convergente-divergente	46
Ilustración 4.13. Relación de áreas en función del número de Mach	47
Ilustración 4.14. Presentación del modo genérico de la función tobera convergente-divergente.....	52
Ilustración 4.15. Problema ejemplo del modo genérico	53
Ilustración 4.16. Esquema del flujo de Fanno.....	58
Ilustración 4.17. Presentación del flujo de Fanno	60

Ilustración 4.18. Guía en el flujo de Fanno	61
Ilustración 4.19. Gráfico de la temperatura en función de la altitud	65
Ilustración 4.20. Gráficos de la presión y densidad en función de la altitud	65
Ilustración 4.21. Gráfica de temperatura en función del número de Mach	67
Ilustración 4.22. Gráficas de presión y densidad en función del número de Mach	67
Ilustración 4.23. Gráfica de la temperatura de estancamiento en función del número de Mach	68
Ilustración 4.24. Gráficas de la presión y densidad de estancamiento en función del número de Mach	69
Ilustración 4.25. Gráfica del Mach después de la onda en función del anterior a la onda	70
Ilustración 4.26. Gráfica de la temperatura en función del Mach 1	70
Ilustración 4.27. Gráfica de la presión estática P2 en función del Mach 1	70
Ilustración 4.28. Gráfica de la presión de estancamiento P0 2 en función del Mach 1	70
Ilustración 4.29. Gráfica de la densidad estática p2 en función del Mach 1	70
Ilustración 4.30. Gráfica de la densidad de estancamiento p0 2 en función del Mach 1 .	70
Ilustración 4.31. Gráfica de la velocidad v2 en función del Mach 1	71
Ilustración 4.32. Gráfica del área de garganta A*2 en función del Mach 1	71
Ilustración 4.33. Números de Mach en función de la relación de áreas	72
Ilustración 4.34. Gráficas comparativas de relaciones de propiedades	72
Ilustración 4.35. Gráficas comparativas del modo adaptada con inputs de relaciones de presión y temperatura	73
Ilustración 4.36. Gráfica comparativa del flujo másico en modo genérico	74
Ilustración 4.37. Gráficas comparativas de las presiones límite del modo genérico	74
Ilustración 4.38. Gráfica comparativa del área de onda de choque en función de la presión P2	75
Ilustración 4.39. Gráfica comparativa de la longitud en función del Mach de entrada	76
Ilustración 4.40. Gráficas comparativas de relaciones de presión y temperatura en función del Mach de entrada	76
Ilustración 4.41. Gráficas comparativas de relaciones de densidad y presión de remanso en función del Mach de entrada	76
Ilustración 8.1. Diagrama de Gantt del proyecto.	84

Lista de tablas

Tabla 2.1. Comparativa de lenguajes de programación	6
Tabla 2.2. Comparación de aplicaciones.....	8
Tabla 3.1. Comportamiento del flujo de Fanno.....	22
Tabla 4.1. Valores de referencia de la atmósfera ISA (1976).....	34
Tabla 4.2. Comparación de los resultados de temperatura en función de la altitud.....	65
Tabla 4.3. Comparación de los resultados de presión en función de la altitud	66
Tabla 4.4. Comparación de los resultados de densidad en función de la altitud	66
Tabla 4.5. Comparación de relaciones de temperatura en función del número de Mach 67	
Tabla 4.6. Comparación de relaciones de presión en función del número de Mach.....	68
Tabla 4.7. Comparación de relaciones de densidad en función del número de Mach.....	68
Tabla 4.8. Resultados de las relaciones de propiedades en ondas de choque normales71
Tabla 4.9. Tabla comparativa del modo adaptada con la relación de áreas como input .	73

1. Introducción. Planteamiento del problema y justificación.

El uso de la mecánica de fluidos en aplicaciones de ingeniería es, en muchos ámbitos, indiscutible. A pesar de ser un campo muy amplio, cuando se analizan los problemas a resolver, se puede observar que se suele partir de unos **problemas básicos o “tipo”**, y de allí se extienden otros problemas más complejos. Es por eso que estos problemas sencillos acaban siendo la espina dorsal de los conocimientos de la mecánica de fluidos en flujo compresible, y su conocimiento y familiaridad con ellos resulta básica para cualquier persona implicada en dicho ámbito. Por otra parte, como sucede en muchos aspectos de la ingeniería, un problema complejo suele reducirse a un problema más sencillo para tener una primera aproximación e idea inicial, antes de tratarlo en mayor profundidad.

En el trabajo con el flujo compresible puede surgir la necesidad de resolver uno de estos problemas sencillos de forma inmediata. Aunque el problema a resolver sea rápido, sin excesiva complejidad y muy probablemente otro problema similar ya ha sido resuelto por el ingeniero, su solución suele involucrar fórmulas y procedimientos, algunos de ellos iterativos, que hacen imposible su solución mental, especialmente en la rama de flujo compresible. Por ello, **el ingeniero tiene que desplazarse** del “sitio de trabajo” al “sitio de cálculo”, para resolverlo con papel y calculadora o con un programa de ordenador. Con la tecnología y potencia de cálculo en los terminales móviles actuales, **este problema resulta ya anticuado**.

El mismo inconveniente también se relaciona con los estudiantes. Deben aprender a realizar este tipo de problemas, pero una vez asimilado el concepto se pueden convertir en pasos intermedios para resolver problemas más complicados. Al tener ya los conceptos teóricos asimilados, éstos se vuelven repetitivos y pesados, y se beneficiarían de tener un **método para resolverlos rápidamente**, o para comprobar fácilmente sus resultados.

De estos problemas sale la idea de programar una aplicación para móviles que resuelva los problemas clásicos de flujo compresible, orientándola tanto a estudiantes de ingeniería como a ingenieros ya graduados, para agilizar el trabajo, disponer de una comprobación rápida de resultados y hacer posible la resolución de problemas fuera del lugar de cálculo, teniendo una herramienta en el bolsillo capaz de realizarlos.

De esta forma, las dos ideas principales de este proyecto son hacer el trabajo más rápido y hacerlo más sencillo, enfocándose a estudiantes e ingenieros.

1.1. Objetivo del proyecto

El principal objetivo de este trabajo es el **desarrollo de una aplicación para dispositivos Android para flujo compresible**, como por ejemplo flujos adiabáticos e isentrópicos estacionarios, ondas de choque normales, operaciones de toberas convergentes y divergentes y flujo compresible por un conducto con fricción. Dicha aplicación será programada en código libre.

Además, el proyecto tiene como objetivos adicionales:

- Estudiar todas las aplicaciones Android gratuitas relacionadas con el flujo compresible.
- Introducir una nueva aplicación Android gratuita para aplicaciones de flujo compresible en el mercado.
- Preparar una guía de usuario introductoria a la aplicación con problemas clásicos de mecánica de fluidos solucionados para principiantes, principalmente estudiantes de grado.
- Practicar el desarrollo de proyectos de ingeniería combinando trabajo técnico con aspectos económicos y medioambientales.

1.2. Alcance del proyecto

Una vez definidos los objetivos, se pueden enumerar los aspectos hasta donde el proyecto se plantea llegar:

1. Estudiar las aplicaciones Android relacionadas y elaborar una descripción, pros y contras.

2. Estudiar el software gratuito disponible para programar la aplicación Android.
3. Trabajar en el desarrollo de la aplicación para aplicaciones en, como mínimo, flujo estacionario, adiabático e isentrópico, onda de choque normal, operación de toberas convergentes y divergentes y flujo compresible en un conducto con fricción.
4. Analizar simulaciones de fluidos relacionadas con problemas reales en el campo de la ingeniería.
5. Preparar una guía básica, con explicaciones y consideraciones, para preparar a los potenciales usuarios, especialmente estudiantes de grado.
6. Determinar la viabilidad del proyecto a realizar.

1.3. Requisitos

- Software de programación en lenguaje para dispositivos Android.
- Dispositivo Android con acceso al mercado de aplicaciones.
- Ordenador con conexión a Internet.
- Conocimientos teóricos de flujo compresible: resolución de problemas de flujo isentrópico y toberas, ondas de choque y principios de flujo no isentrópico.

2. Estado del arte

2.1. El flujo compresible en la ingeniería

Un régimen compresible en un flujo se alcanza de forma relativamente fácil en el caso de los gases, como se verá más adelante. Por esta razón, el estudio del movimiento de fluidos teniendo en cuenta su compresibilidad es muy utilizado en muchos ámbitos de la ingeniería, en prácticamente cualquier aplicación que trate con gases, e incluso en algunos líquidos.

Por otra parte, el estudio del flujo compresible puede llegar a ser muy complejo, tanto que sobrepasaría la dificultad teórica de un proyecto de final de grado y exigiría una potencia de cálculo superior a la de un teléfono móvil. Es por eso que se tiene que establecer un nivel al cual se adaptará la aplicación. Dado que es una aplicación pensada en su mayor parte **para estudiantes** y ámbitos relacionados con la educación, resulta evidente que el nivel de la aplicación debería adaptarse al nivel teórico que poseen los estudiantes.

Como referencia se toma el grado de ingeniería en tecnologías aeroespaciales. Al acabar, los alumnos tienen conocimientos, como mínimo, de:

- **Flujo isentrópico y ecuaciones adiabáticas.**
- Flujo isentrópico en **conductos de sección variable**, aproximación unidimensional.
- **Ondas de choque normales y oblicuas**, en casos unidimensionales y bidimensionales respectivamente.
- Principios básicos de **toberas en cohetes** y reacciones químicas producidas.
- Método de volúmenes finitos para **flujos no isentrópicos** en conductos de área tanto constante como variable.

Después de la asignatura de mecánica de fluidos en dicha carrera, los conocimientos sobre flujo compresible que adquieren resultan indispensables para las asignaturas de aerodinámica, sistemas propulsivos, propulsión y dinámica de gases y transferencia de calor y masa.

En el mundo aeronáutico, por lo tanto, el análisis del flujo compresible es básico para los sectores de aerodinámica y diseño de vehículos, propulsión, y cualquier análisis con gases a velocidades moderadas o líquidos a velocidades elevadas.

2.2. Lenguajes de programación para Android disponibles

A la hora de empezar una aplicación sin tener conocimiento previo sobre programación en dispositivos móviles, siempre surge la misma duda: ¿Qué lenguaje de programación utilizar? Para bien o para mal, hay una notable variedad de lenguajes disponibles, cada uno con sus ventajas e inconvenientes. La confusión del potencial desarrollador viene porque muchas veces esas diferencias son en complicados procedimientos y rutinas, y las páginas de discusión de Internet se llenan de palabras y siglas desconocidas y desconcertantes.

Es por eso que, primero que todo, se establece un sesgo en los lenguajes Android considerados “disponibles”: deben ser gratuitos, y aunque puedan usarse en diferentes plataformas (iOS, Windows, etc.) deben de funcionar tan bien en Android como cualquier otra aplicación “única” para Android.

A continuación se enumerarán las principales opciones de lenguajes de programación acordes con ese criterio. No se entrará en detalles muy específicos sobre cada lenguaje, pues una investigación profunda sobre ellos está más allá de la profundidad buscada en este proyecto. Una descripción más desarrollada y genérica se puede encontrar en la web “Androideity” [1]

- **Java:** es el lenguaje más usado para Android, y potenciado enormemente por la empresa Google. Se le llama el lenguaje nativo de Android, ya que tiene una plataforma totalmente optimizada para este lenguaje. Con un lenguaje nativo se consigue un acceso total al dispositivo, pero no puede llevarse a otros sistemas operativos. En lo que respecta al lenguaje, Java es un lenguaje de complejidad moderada pero ofrece una potencia enorme, tanto que se ha vuelto de obligado aprendizaje para cualquier desarrollador de aplicaciones de semiprofesional en adelante.
- **C/C++:** es una buena opción para aquellos que quieren programar en un lenguaje más habitual de ordenadores y hacer aplicaciones

multiplataforma. Está la posibilidad de programar aplicaciones nativas, y su rendimiento en dispositivos móviles está en aumento. Sin embargo, no resulta tan potente como el Java.

- **In Design CS6:** es un lenguaje orientado principalmente a redactores de revistas y publicaciones. Tiene una buena y rápida funcionalidad gráfica, de forma que permite crear aplicaciones centradas en la presentación de contenidos más que en potencia de cálculo. Al estar dirigida a no programadores, resulta muy fácil de aprender.
- **Google App Inventor:** es una alternativa a los lenguajes de programación convencionales, creado en colaboración por el MIT y la empresa Google. Está pensada para acercar la creación de aplicaciones móviles a gente sin conocimientos de programación. Con ella se programa por bloques, sin escribir una sola línea de código, cosa que al “recién llegado” le parecerá muy intuitivo, pero a un programador habitual lo puede dejar descolocado. Para ser un lenguaje por bloques y dirigido a no programadores tiene un gran potencial, pero no se llegará tan lejos como el Java o C/C++.

En la siguiente tabla se resumen las características de los lenguajes de programación estudiados:

Lenguaje	Complejidad relativa	Potencia de cálculo	Potencial (Funciones posibles)
Java	Alta	Alta	Alto
C/C++	Media	Alta	Medio-Alto
In Design CS6	Baja	Baja	Bajo
App Inventor	Baja	Media	Medio

Tabla 2.1. Comparativa de lenguajes de programación

Dado que este proyecto no requiere una gran variedad de funciones posibles, sino principalmente cálculo interno del dispositivo para resolver ecuaciones de flujo compresible, y una complejidad baja del lenguaje es un factor muy positivo, la opción de **utilizar App Inventor** resulta adecuada. En cuanto a su potencia de cálculo, puede no ser su mejor cualidad, pero se espera que cumpla bien con lo que se le pide.

De esta forma, este proyecto puede ser, en cierto modo y además de un proceso para crear una aplicación Android útil para el trabajo con flujo compresible, una

investigación sobre el potencial y el rendimiento de App Inventor, a fin de servir de **guía** para alguien que quiera crear una aplicación posteriormente.

2.3. Aplicaciones de flujo compresible en el mercado

Aunque su uso puede que no esté generalizado, actualmente se puede encontrar cierta variedad de aplicaciones para móvil, también llamadas “Apps”, relacionadas con el flujo compresible. La mayoría de ellas calculan los parámetros del flujo isentrópico estacionario y toberas convergentes y divergentes. Algunas van más allá y cubren una gran variedad de aplicaciones, mientras que otras se centran en unos pocos.

Una lista de las aplicaciones relacionadas con el flujo compresible gratuitas que se encuentran en la *Play Store*:

- Compresible Flow Tool (DroidFlo)
- Compressible Flow / Compressible Flow Pro (aero-calculator)
- Compressible Flow Tool (Zdeiter)
- Compressible Nozzle Free (Mobile Education by Nilesh)
- Aerodynamics Tool Free (Mobile Education by Nilesh)
- GDGalc (Jawad Kaddouri)
- Normal Shock Tables (aero-calculator)
- Isentropic Flow Tool (aero-calculator)

Se ha omitido las aplicaciones que pueden tener relación con el flujo compresible, pero que su punto de vista es diferente al de este proyecto: aplicaciones de petróleo y máquinas fluídicas como bombas, compresores, etc.

En la página siguiente se muestra una tabla con los usos de cada aplicación, seleccionando los que están al alcance del fondo teórico de este proyecto, y un diagrama más enfocado a su uso y la experiencia de usuario.

Tabla 2.2. Comparación de aplicaciones

Apps	Flujo isentrópico estacionario	Tobera convergente divergente	Onda de choque normal	Onda de choque oblicua	Otros flujos no isentrópicos (rozamiento)	Astμόfera ISA	Propiedades de gases guardadas	Expansión de Prandtl
Compressible Flow Tool (DroidFlo)	✓	✓	✓	✓				✓
Compressible Flow Pro	✓		✓	✓			✓	
Compressible Flow Tool (Zdeiter)	✓	✓	✓	✓		✓	✓	
Compressible Nozzle Free		✓						
Aerodynamics Tool Free	✓		✓	✓				✓
GDCalc	✓	✓	✓	✓				✓
Normal Shock Tables			✓					
Isentropic Flow Tool	✓							

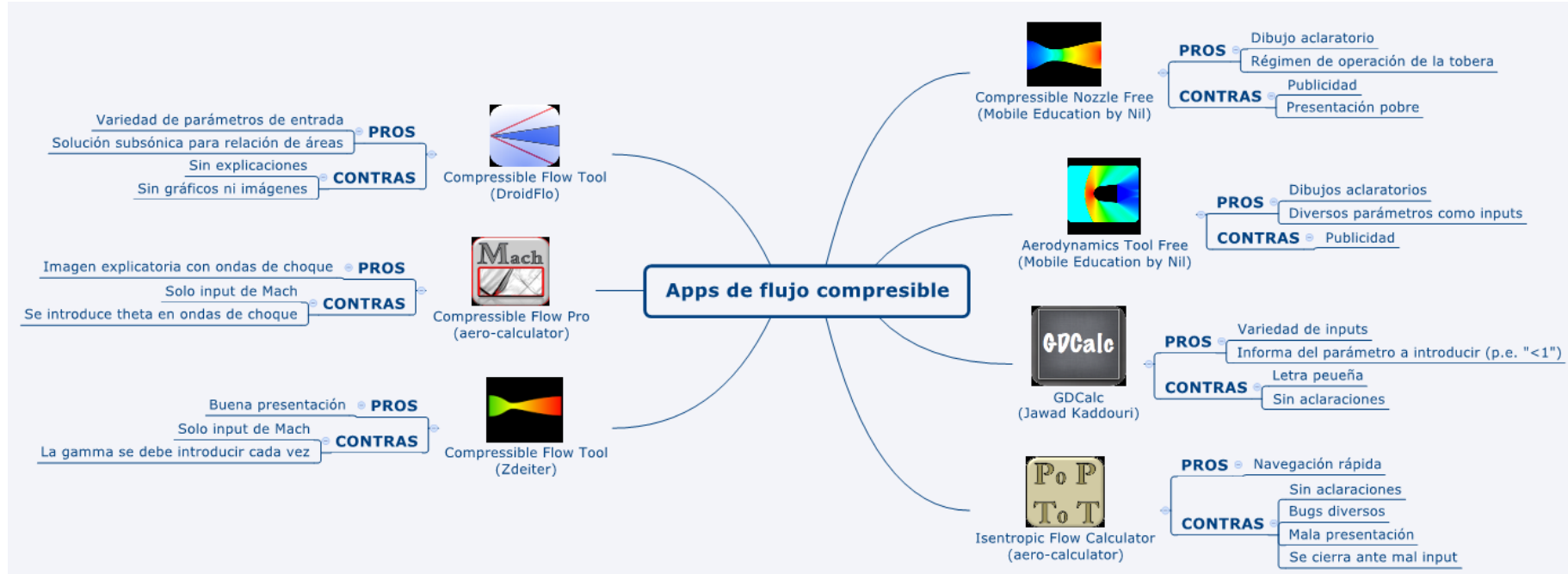


Ilustración 2.1. Diagramas comparadores de aplicaciones.

3. Planteamiento de la solución. Funciones de la App, teoría y lenguaje de programación.

3.1. Decisión de funciones

Una vez visto el problema y evaluado las diversas aplicaciones de flujo compresible, el objetivo del proyecto es claro: crear una aplicación móvil gratuita que supere los puntos débiles de las demás aplicaciones del mercado.

A la hora de decidir **qué hará la App**, y para cumplir su propósito inicial, se valoran principalmente dos aspectos:

- Debe cubrir los **máximos posibles**, si no todos, de los **aspectos** típicos de la mecánica de fluidos de los **estudiantes de ingeniería**: flujo isentrópico, ondas de choque, conductos de área variable, etc.
- Debe resultar más útil y práctica que las aplicaciones del mercado. Esto puede ser mediante:
 - Agrupar más aspectos en una sola App.
 - Realizar más funciones en un solo aspecto, como poder realizar cálculos dimensionales o casos específicos.
 - Al usar cálculos dimensionales, poder elegir la unidad a utilizar.
 - Resultar más intuitiva y fácil de utilizar. Mostrar con mayor claridad la notación utilizada y el fenómeno físico tratado, con dibujos, por ejemplo.

Así pues, y valorando también la dificultad teórica y tiempo disponible, se decide que la App incluirá los siguientes casos:

- **Atmósfera estándar internacional ISA**: pese a no ser exactamente flujo compresible (pues ni siquiera se contempla movimiento en ella), es muy común para un estudiante encontrarse problemas en los que debe calcular las condiciones ISA a cierta altura. Y aún más común no recordar las fórmulas de memoria. Como resultará útil para la App y es de cálculo sencillo, es un caso perfecto para el aprendizaje del programador.

- **Flujo isentrópico:** ejemplo claro y básico del flujo compresible, en el que es muy habitual olvidar la fórmula exacta. Relaciona propiedades de estancamiento con estáticas en función de su número de Mach.
- **Onda de choque normal:** tratada como una discontinuidad del fluido, relaciona las propiedades antes y después. Como en los casos anteriores, resulta un clásico de la mecánica de fluidos para estudiantes.
- **Tobera convergente-divergente:** es un caso común en las App de flujo compresible, pero no en mucho detalle. En este proyecto se propone, además de tratar la tobera de forma adimensional y adaptada, crear una función dimensional que calcule su régimen de operación y sus parámetros de salida.
- **Flujo de Fanno:** no es un caso que se enseñe en la carrera de ingeniería en tecnologías aeroespaciales, pero es un buen paso inicial hacia el trato con flujo compresible no isentrópico, enfocándose un poco más hacia el mundo laboral, o estudiantes de otras carreras.

A fin de dirigir la aplicación a un mayor número de usuarios, se decide hacer la aplicación en inglés. Esto no debería representar un problema dado el público objetivo de la App, que son estudiantes y profesionales con formación científica, en los que el idioma inglés suele ser un requisito básico.

3.2. Introducción teórica al flujo compresible

3.2.1. ¿Qué es?

Comenzando por las definiciones más básicas, se le llama flujo a todo aquel fluido que se haya en movimiento. A su vez, un fluido se trata de algo que fluye: una sustancia gaseosa o líquida.

El flujo compresible se define como *flujo de densidad variable*. Surge como contraste al flujo incompresible, donde la densidad se asume constante. Obviamente, en la práctica tal suposición es falsa, puesto que todo fluido es compresible en mayor o menor medida, pero aplicar la condición de incompresibilidad a líquidos y a gases en determinadas circunstancias da buenos

resultados, ya que la variación de densidad en ellos es suficientemente pequeña comparada con su valor de densidad inicial o promedio.

El trato al fluido en régimen compresible resulta más sencillo e intuitivo. Sin embargo, si su velocidad aumenta por encima de cierto valor, esta suposición ya no es válida. Para el aire, por ejemplo, la velocidad a la que debe ser considerado compresible está alrededor de los 40 m/s. A la vista de las aplicaciones actuales en los ámbitos de la mecánica de fluidos, los usos del flujo compresible son evidentes: la velocidad “frontera” se supera con muchísima frecuencia: flujo alrededor de los aviones, a través de motores a reacción o de hélice, en compresores y turbinas, en gasoductos, etc.

3.2.2. Introducción analítica y flujo isentrópico

En aquellos casos en los que la variación de densidad de un flujo es tan pequeña comparada con su densidad inicial o promedio que se puede suponer despreciable se puede aplicar la ecuación de Bernoulli:

$$p + \frac{1}{2}\rho V^2 = \text{constante} \quad (3.1)$$

Donde p es la presión del líquido, ρ es la densidad del fluido y V su velocidad.

La compresibilidad de una sustancia, τ , se define de la siguiente forma:

$$\tau = -\frac{1}{v} \frac{dv}{dp} \quad (3.2)$$

Donde v es su volumen específico. De aquí que grandes valores de compresibilidad impliquen altas variaciones de volumen ante una variación de presión. Si se transforma dicha definición con las hipótesis de compresión isoterma e isentrópica, y se tiene en cuenta que $\rho = 1/v$, se llega a la ecuación

$$d\rho = \rho\tau dp \quad (3.3)$$

Aunque la ecuación de Bernoulli no siempre sea aplicable, sí marca una tendencia: altas diferencias de presión corresponden a altas diferencias de

velocidades. Con esto y la ecuación anterior se puede intuir que cuanto más rápido sea un flujo, más importante es la compresibilidad. Más específicamente, se estudia la compresibilidad de un flujo en función de su velocidad adimensional: el número de Mach, M o Ma , definido como

$$M = \frac{V}{a} \tag{3.4}$$

Donde V es la velocidad del flujo y a es la velocidad del sonido en el medio, que depende de la temperatura del fluido. De hecho, se puede demostrar que con las ecuaciones (3.5) y (3.6) se llega a (3.7).

$$\rho V dV = -dp \tag{3.5}$$

(Conservación de momento)

$$\frac{dp}{p} = \frac{\gamma d\rho}{\rho} \tag{3.6}$$

(Flujo isentrópico)

$$-M^2 \frac{dV}{V} = \frac{\gamma d\rho}{\rho} \tag{3.7}$$

Donde γ es el coeficiente de dilatación adiabática del gas, $\gamma=c_p/c_v$

Para ilustrar el efecto del Mach sobre el flujo, se puede observar el siguiente gráfico, que muestra el coeficiente de resistencia en función de éste. A medida que los efectos de compresibilidad son más notables el coeficiente de resistencia aumenta de forma notable.

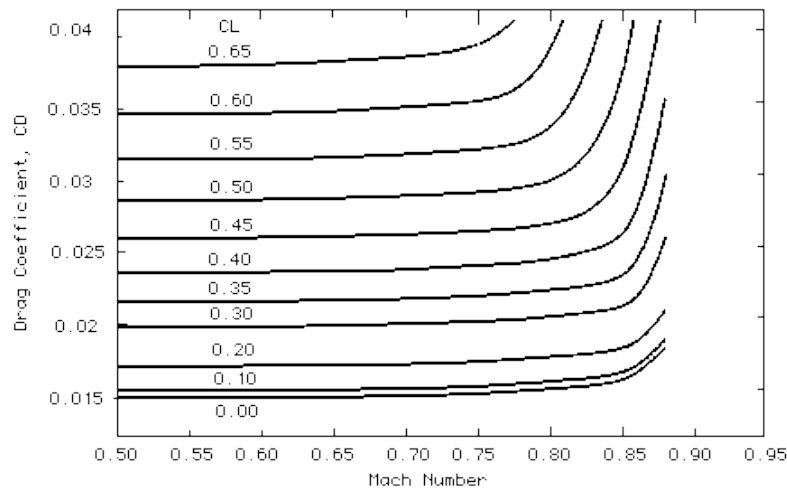


Ilustración 3.1. Coeficiente de resistencia en función del Mach

Así pues, se pueden definir regiones por sus efectos de compresibilidad en función del Mach:

- **Flujo incompresible** ($M < 0.3$): Los efectos de compresibilidad son despreciables.
- **Flujo subsónico** ($0.3 < M < 0.8$): Los efectos de variación de densidad son importantes, pero no aparecen ondas de choque.
- **Flujo sónico** ($M = 1.0$): Flujo sónico, aparece la condición de bloqueo en un conducto.
- **Flujo transónico** ($0.8 < M < 1.2$): Aparecen ondas de choque que separan regiones subsónicas y supersónicas en un flujo.
- **Flujo supersónico** ($1.2 < M < 3.0$): Hay ondas de choque pero ya no existen regiones subsónicas (ondas de choque oblicuas).
- **Flujo hipersónico** ($M > 3.0$): Las ondas de choque y otros cambios que experimenta el flujo son especialmente fuertes.

Para tratar el flujo compresible de forma analítica se parte de las dos leyes de la termodinámica, la definición de entalpía y las propiedades de gases ideales $du = c_v dT$ y $dh = c_p dT$, siendo u la energía interna y h la entalpía. La demostración se puede encontrar en el libro "Modern Compressible Flow" de Anderson [2], capítulo 3, *One-Dimensional Flow*. Para la utilidad de este trabajo es conveniente recordar la siguiente ecuación (3.8), que relaciona la variación de entropía de un estado 1 a 2 con su temperatura y presiones.

$$\int_1^2 ds = \int_1^2 \frac{c_p dT}{T} - r \ln \left(\frac{p_2}{p_1} \right) \quad (3.8)$$

Siendo r la constante universal de los gases, dividida entre la masa molar del gas en cuestión. Si c_p no es constante, la integral se realiza de forma numérica. Si se considera constante porque no hay unas variaciones de temperatura importantes (gas perfecto), la integral resulta fácil de realizar de forma analítica:

$$s_2 - s_1 = c_p \ln \left(\frac{T_2}{T_1} \right) - r \ln \left(\frac{p_2}{p_1} \right) \quad (3.9)$$

$$s_2 - s_1 = c_v \ln \left(\frac{T_2}{T_1} \right) - r \ln \left(\frac{p_2}{p_1} \right) \quad (3.10)$$

Si se considera flujo isentrópico, se obtienen conocidas las relaciones de un gas perfecto para flujo isentrópico:

$$\frac{p_2}{p_1} = \left(\frac{\rho_2}{\rho_1}\right)^\gamma = \left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \quad (3.11)$$

El flujo isentrópico implica que no hay transferencia de calor entre el sistema y su entorno (adiabático) y su procedimiento se lleva a cabo de forma reversible. Esto es, por supuesto, una idealización, pero resulta útil para hacer aproximaciones a problemas reales de forma rápida.

La velocidad del sonido en un gas se define como la velocidad a la que una perturbación de presión se propaga por él. Suponiendo que se hace de forma isentrópica, se puede llegar a la ecuación

$$a = \sqrt{\gamma r T} \quad (3.12)$$

Con todo flujo compresible se pueden definir unas *condiciones de estancamiento o reposo*, que serían las que tendría el flujo si se llevase de forma isentrópica al reposo, o a condiciones de $V=0$ y $M=0$. Se denominan por el subíndice cero: T_0 , P_0 , a_0 ... De esta forma, partiendo de la ecuación de la energía sin adición de calor

$$c_p T_1 + \frac{u_1^2}{2} = c_p T_2 + \frac{u_2^2}{2} \quad (3.13)$$

Donde u es la velocidad del flujo, se puede llegar fácilmente a

$$c_p T + \frac{u^2}{2} = c_p T_0 \quad (3.14)$$

Y de aquí, a las relaciones de temperatura, presión y densidad en función del Mach

$$\frac{T_0}{T} = 1 + \frac{\gamma - 1}{2} M^2 \quad (3.15)$$

$$\frac{P_0}{P} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}} \quad (3.16)$$

$$\frac{\rho_0}{\rho} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{1}{\gamma-1}} \quad (3.17)$$

Y por último,

$$\frac{a^2}{\gamma - 1} + \frac{u^2}{2} = \frac{a_0^2}{\gamma - 1} \quad (3.18)$$

3.2.3. Ondas de choque normales

Una onda de choque es una variación irreversible de las propiedades de un fluido en un espacio muy pequeño, de forma que son consideradas normalmente como discontinuidades. En una onda de choque, el flujo perpendicular a ésta pasa de ser supersónico a subsónico. Puede venir ocasionada, por ejemplo, por un obstáculo en el flujo o por la adaptación de la presión de descarga de una tobera a la presión ambiente.

Las ondas de choque normales se producen, como su nombre sugiere, de forma aproximadamente perpendicular al flujo incidente. Como se tratan como discontinuidades, las propiedades a tener en cuenta son antes de la onda de choque (1) y después (2). Se trata de un proceso en el que no se añade ni se retira calor, por lo tanto es adiabático.

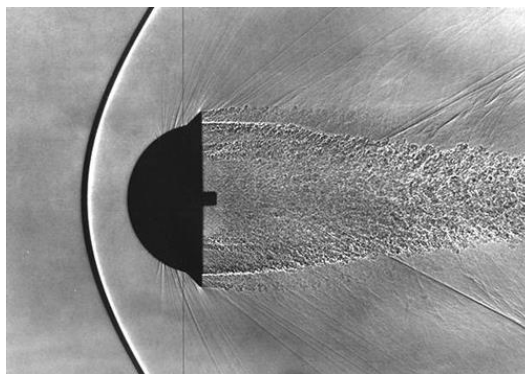


Ilustración 3.2. Ejemplo de onda de choque normal
(Fuente: <http://www.psr.d.hawaii.edu> [3])

Mediante las ecuaciones de continuidad, momento y energía, y con las suposiciones de gas ideal y caloríficamente perfecto, se puede demostrar que

$$M_2^2 = \frac{1 + \frac{\gamma - 1}{2} M_1^2}{\gamma M_1^2 - \frac{\gamma - 1}{2}} \quad (3.19)$$

Y también por lo que respecta a las relaciones de presión, temperatura, densidad y velocidad:

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma + 1} (M_1^2 - 1) \quad (3.20)$$

$$\frac{T_2}{T_1} = \frac{h_2}{h_1} = \left[1 + \frac{2\gamma}{\gamma + 1} (M_1^2 - 1) \right] \left[\frac{2 + (\gamma - 1)M_1^2}{(\gamma + 1)M_1^2} \right] \quad (3.21)$$

$$\frac{\rho_2}{\rho_1} = \frac{u_1}{u_2} = \frac{(\gamma + 1)M_1^2}{2 + (\gamma - 1)M_1^2} \quad (3.22)$$

Con éstas condiciones de después de la onda de choque y el Mach, se pueden calcular las condiciones de remanso de después de la onda de choque. En particular, la temperatura de remanso se mantiene constante.

Como ecuación explícita, se tiene:

$$\frac{p_{02}}{p_{01}} = \frac{\rho_{02}}{\rho_{01}} = \left(\frac{\rho_2}{\rho_1} \right)^{\frac{\gamma}{\gamma - 1}} \left(\frac{p_1}{p_2} \right)^{\frac{1}{\gamma - 1}} \quad (3.23)$$

3.2.4. Flujo isentrópico en conductos de área variable. Toberas.

En esta sección se analiza el problema de un flujo compresible a lo largo de un conducto de sección variable de forma cuasi-unidimensional, es decir, que las variables del fluido solo varían en función de una sola coordenada, en este caso la x . Esto viene de la suposición que el fluido tiene idénticas propiedades a lo largo de una misma sección (valor de x). Por supuesto, esto es una aproximación a la física real del problema. Aquí se explicarán las ecuaciones que se usarán en el programa y su origen. Una demostración más detallada se puede encontrar, de nuevo, en “Modern Compressible Flow” de Anderson [2].

Un ejemplo de volumen de control a analizar puede ser el siguiente:

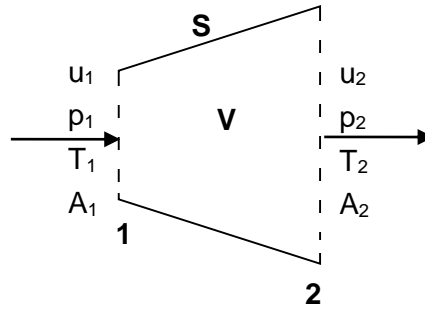


Ilustración 3.3. Volumen de control para flujo cuasi-unidimensional

Una ecuación fundamental en este tipo de problema es la ecuación de continuidad,

$$-\oint_S \rho V \cdot dS = \frac{\partial}{\partial t} \iiint_V \rho dV \quad (3.24)$$

Asumiendo flujo estacionario y propiedades constantes en cada sección, se llega a

$$\rho_1 u_1 A_1 = \rho_2 u_2 A_2 \quad (3.25)$$

Por otra parte la forma integral de la ecuación de conservación de la cantidad de movimiento es

$$\iint_S (\rho V \cdot dS) V + \iiint_V \frac{\delta(\rho V)}{\delta t} dV = \iiint_V \rho f dV - p dS \quad (3.26)$$

De nuevo asumiendo flujo estacionario y sin fuerzas másicas ni de fricción, se llega a

$$p_1 A_1 + \rho_1 u_1^2 A_1 + \int_{A_1}^{A_2} p dA = p_2 A_2 + \rho_2 u_2^2 A_2 \quad (3.27)$$

De esta forma se está asumiendo un flujo sin fricción en su conducto, condición indispensable para considerarlo isentrópico.

De la ecuación de conservación de la masa (3.28) se puede obtener

$$\frac{d\rho}{\rho} + \frac{du}{u} + \frac{dA}{A} = 0 \quad (3.28)$$

Combinándola con la ecuación (3.27), suponiendo flujo isentrópico y recordando que $h=c_p T$, se puede llegar a la ecuación que relaciona el área con la velocidad,

$$\frac{dA}{A} = (M^2 - 1) \frac{du}{u} \quad (3.29)$$

Esta ecuación es importante porque resulta fácil ver que si el Mach es próximo a 0, $A^*u=\text{constante}$, ecuación típica de flujo incompresible. Si el Mach está entre 0 y 1, una ampliación del área resulta en una disminución de velocidad. En cambio, si es mayor que 1, un aumento del área provoca un aumento de la velocidad. Cuando el Mach es 1, se trata de un caso límite en la que dicha ecuación señala un máximo o mínimo de la distribución de áreas, en la que solo resulta físicamente posible un mínimo.

De estos resultados se desprende que para acelerar un fluido desde el reposo o desde velocidades bajas hasta otras supersónicas, se le debe hacer pasar por un conducto convergente al principio y después divergente, en el que se alcancen condiciones sónicas ($M=1$) en su sección mínima. Las condiciones en esta sección, llamada *garganta*, se denotan por un asterisco (por ejemplo, A^* , u^* , etc.).

Combinando las ecuaciones vistas hasta ahora y las hipótesis de flujo estacionario e isentrópico se puede llegar a la ecuación que relaciona las áreas en función del Mach,

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{\gamma + 1}{\gamma - 1}} \quad (3.30)$$

Que no tiene solución analítica si lo que se tiene es la geometría de la tobera. Para este caso hay dos soluciones positivas, una subsónica y una supersónica.

La siguiente imagen permite ilustrar los modos de trabajo en los que se encuentra una tobera:

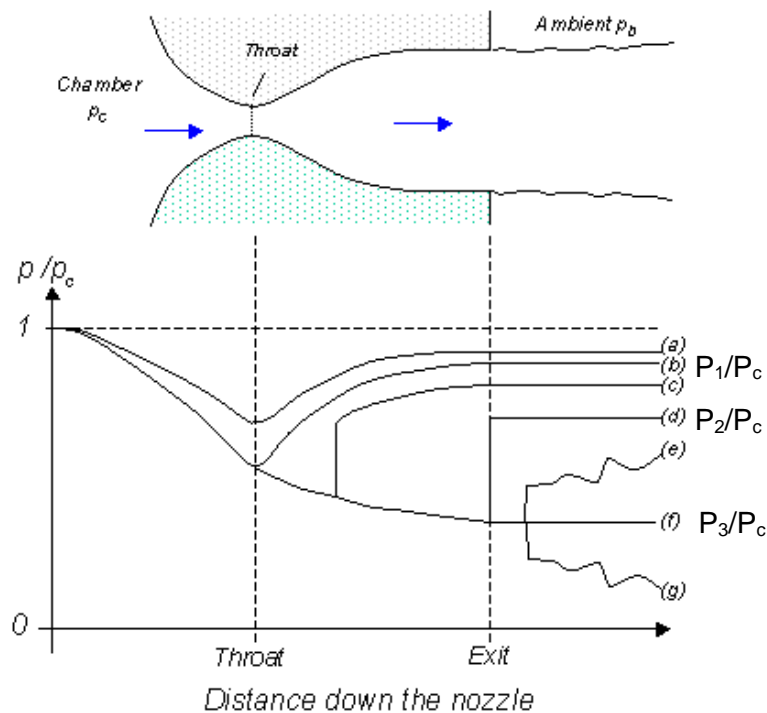


Ilustración 3.4. Distribución de presión en una tobera.
 (Fuente: <http://www.engapplets.vt.edu> [4])

En función de la relación de presiones P_{salida}/P_c , donde P_{salida} es la presión ambiente del lado derecho de la tobera, y P_c la presión de remanso de la izquierda de la tobera.

P_{salida}/P_c :

- $> P_1/P_c$: caso a. La tobera no alcanza condiciones sónicas, y el flujo se frena al llegar a la parte divergente.
- $= P_1/P_c$: caso b. La tobera alcanza condiciones sónicas en la garganta, pero no se llega a acelerar en la parte divergente. Solución subsónica de (3.30).
- $P_2/P_c < P/P_c < P_1/P_c$: caso c. La tobera está en condiciones sónicas pero su aceleración isentrópica sería excesiva para la presión ambiental de descarga, por lo que experimenta una onda de choque en algún punto de su tramo divergente, y descarga a presión ambiente.
- $= P_2/P_c$: caso d. La tobera experimenta una onda de choque justo en su salida.

- $P_3/P_c < P/P_c < P_2/P_c$: caso e. Ninguna onda de choque es capaz de producir la expansión necesaria, por lo que el flujo se comprime en el exterior mediante complejas ondas de choque oblicuas, que no se estudiarán en este proyecto.
- $= P_3/P_c$: caso f. La expansión isentrópica y supersónica de la tobera es la adecuada para la presión de descarga. Solución supersónica de (3.30).
- $< P_3/P_c$: caso g. La presión de salida es menor que la de diseño, pero la tobera sigue descargando a la presión P_3 . Fuera de ella, el flujo se expande mediante complejas ondas supersónicas hasta alcanzar la presión ambiente.

Para calcular la expansión después de la onda de choque en los casos d y e, la expresión (3.30) ya no es válida con esa área de garganta, pues los valores de remanso han cambiado. Se puede calcular una nueva área de garganta, que correspondería a las condiciones sónicas de las nuevas condiciones de remanso,

$$\frac{A_2^*}{A_1^*} = \frac{M_2}{M_1} \left[\frac{(2 + (\gamma - 1)M_1^2)}{(2 + (\gamma - 1)M_2^2)} \right]^{\frac{1}{2}(\gamma+1)} \quad (3.31)$$

Para el caso a, en el que el flujo no llega a condiciones sónicas en la garganta, la expresión (3.30) tampoco es válida. Realizando el mismo procedimiento para llegar a ella pero entre dos secciones cualesquiera, se llega a

$$\frac{A_1}{A_2} = \frac{M_2}{M_1} \left[\frac{\left(1 + \frac{(\gamma - 1)}{2} M_1^2\right)}{\left(1 + \frac{(\gamma - 1)}{2} M_2^2\right)} \right]^{\frac{\gamma+1}{2(\gamma-1)}} \quad (3.32)$$

La cual es una expresión que puede ser útil para calcular relaciones de Mach entre dos secciones cualesquiera, sin necesitar que una de las dos esté en condiciones sónicas.

Más adelante se utilizará esta teoría y estas ecuaciones en el caso de la tobera convergente divergente. En concreto, es importante recordar la ecuación (3.30), que relaciona una relación de áreas en la que el área del denominador tiene condiciones sónicas, con el número de Mach en la del numerador, y que tiene dos soluciones: subsónica y supersónica. Además, también serán importante los regímenes de operación de la tobera explicados con la Ilustración 3.4.

3.2.5. Flujo de Fanno

Una vez visto el efecto que tienen las variaciones de área sobre un flujo compresible despreciando la fricción del fluido, se puede analizar qué influencia tiene este último efecto en el flujo. Sin embargo, este tema se puede volver muy complejo si se mezclan conceptos, por lo que se limita a tratar solamente un conducto recto por el que pasa un gas perfecto, despreciando las variaciones de área y la transferencia de calor. A este caso se le llama *flujo de Fanno*. Yendo más allá se sobrepasaría la complejidad adecuada para tratarse de un aspecto más de entre los que hay en este proyecto.

El análisis completo de este caso se puede consultar en el libro “Mecánica de Fluidos”, de Frank M. White [5]. Aquí se hace, como en los apartados anteriores, un resumen de los aspectos clave de la teoría.

Las hipótesis básicas tomadas son, para el estudio analítico:

- Flujo adiabático estacionario unidimensional
- Gas perfecto (calores específicos constantes)
- Conducto recto de área constante
- El trabajo motor de posibles partes móviles es despreciable
- Las variaciones de energía potencial son despreciables
- El esfuerzo en la pared responde a correlaciones de coeficientes de fricción de Darcy

Con las hipótesis tomadas, se deben plantear ecuaciones suficientes para tratar el problema y determinar las propiedades del flujo: densidad, presión, temperatura y velocidad. Para ello se plantean las ecuaciones diferenciales de continuidad, cantidad de movimiento y energía. En la cantidad de movimiento se debe tener en cuenta también la fuerza que hace la tubería sobre el fluido, por lo que aparece un término de fricción, siendo la quinta incógnita. Para determinar el problema se añade la ley de los gases perfectos y la suposición de que el esfuerzo de fricción local se puede obtener a partir del coeficiente de fricción de Darcy. Al ser un nuevo concepto en este proyecto, se plantea la ecuación de la forma:

$$\tau_w = \frac{1}{8}f\rho V^2 = \frac{1}{8}f\gamma p M^2 \quad (3.33)$$

Donde la f es el coeficiente de fricción de Darcy. Dicho coeficiente relaciona el esfuerzo cortante de fricción con las propiedades de rugosidad del material y

viscosidad del fluido, este último a través del número de Reynolds. La ecuación para obtener f es la siguiente:

$$\frac{1}{f^{\frac{1}{2}}} = -2 \log \left(\frac{\frac{\varepsilon}{d}}{3.7} + \frac{2.51}{Re_d f^{\frac{1}{2}}} \right) \quad (3.34)$$

Sin embargo, se puede observar que la variable f no es aislable, por lo que se puede optar por una expresión alternativa:

$$\frac{1}{f^{\frac{1}{2}}} \approx -1.8 \log \left(\frac{6.9}{Re_d} + \left(\frac{\frac{\varepsilon}{d}}{3.7} \right)^{1.11} \right) \quad (3.35)$$

En ambas expresiones, ε es la rugosidad del material, d es el diámetro hidráulico de la tubería y Re el número de Reynolds, calculado en este caso como

$$Re = \frac{\rho V d}{\mu} \quad (3.36)$$

Donde V es la velocidad del fluido y μ la viscosidad absoluta del fluido.

Continuando con el análisis del flujo de Fanno, teniendo ya todas las ecuaciones iniciales, se pueden utilizar para llegar a relaciones diferenciales más prácticas, y se llega a ver el distinto comportamiento del fluido a lo largo de la tubería dependiendo de si es subsónico o supersónico:

Propiedad	Subsónico	Supersónico
ρ	Disminuye	Aumenta
ρ	Disminuye	Aumenta
V	Aumenta	Disminuye
ρ_0, p_0	Disminuye	Disminuye
T	Disminuye	Aumenta
Ma	Aumenta	Disminuye
Entropía	Aumenta	Aumenta

Tabla 3.1. Comportamiento del flujo de Fanno

Debido a este comportamiento, el flujo siempre tiende a $Ma=1$, independientemente de si es subsónico o supersónico. Por ello hay una longitud L^* para la cual, con un Ma de entrada determinado, el flujo llega a condiciones sónicas en la salida, y se cumple lo siguiente:

$$\frac{\bar{f}L^*}{D} = \frac{1 - M^2}{\gamma M^2} + \frac{\gamma + 1}{2\gamma} \ln\left(\frac{(\gamma + 1)M^2}{2 + (\gamma - 1)M^2}\right) \quad (3.37)$$

Donde \bar{f} es el coeficiente de fricción medio entre 0 y L^* . Cuando un conducto no es suficientemente largo como para que se llegue a condiciones sónicas, su longitud L puede ser calculada por

$$\frac{\bar{f}L}{D} = \left(\frac{\bar{f}L^*}{D}\right)_1 - \left(\frac{\bar{f}L^*}{D}\right)_2 \quad (3.38)$$

Donde las condiciones 1 y 2 son las de entrada y salida del conducto, respectivamente.

Utilizando las ecuaciones obtenidas en el proceso anterior, se pueden obtener otras más útiles para este caso, que relacionan sus las condiciones de un Mach determinado M con las condiciones sónicas:

$$\frac{p}{p^*} = \frac{1}{M} \left[\frac{\gamma + 1}{2 + (\gamma - 1)M^2} \right]^{\frac{1}{2}} \quad (3.39)$$

$$\frac{\rho}{\rho^*} = \frac{V^*}{V} = \frac{1}{M} \left[\frac{\gamma + 1}{2 + (\gamma - 1)M^2} \right]^{\frac{-1}{2}} \quad (3.40)$$

$$\frac{T}{T^*} = \frac{a^2}{a^{*2}} = \frac{\gamma + 1}{2 + (\gamma - 1)M^2} \quad (3.41)$$

$$\frac{p_0}{p_0^*} = \frac{\rho_0}{\rho_0^*} = \frac{1}{M} \left[\frac{\gamma + 1}{2 + (\gamma - 1)M^2} \right]^{\frac{-1}{2} \frac{\gamma + 1}{\gamma - 1}} \quad (3.42)$$

Para obtener una relación entre puntos de diferentes Mach 1 y 2 que no son sónicos, basta con hacer

$$\frac{p_2}{p_1} = \frac{p_2}{p^*} \frac{p^*}{p_1} \quad (3.43)$$

Podría darse el caso en el que el conducto es más largo que su longitud L^* para convertir el flujo en sónico. En este caso se produce un *bloqueo por fricción*. En el caso de entrada supersónica se podría producir una onda de choque, y en el caso subsónico el flujo se ralentiza. Sin embargo, este caso no será tratado en este proyecto.

3.3. App Inventor

Como se ha comentado con anterioridad, se elige como lenguaje de programación a utilizar la plataforma App Inventor, ya que se dirige a un público sin conocimientos previos de programación para dispositivos móviles y se enfoca más hacia la función a realizar, no tanto hacia la programación en sí.

App Inventor es una plataforma de Google Labs y respaldada por el Instituto Tecnológico de Massachusetts que permite crear aplicaciones de software para dispositivos de sistema operativo Android. Está orientada a un **público sin conocimientos previos de programación**, de forma que se pueda utilizar y programar de la forma más rápida posible, incluso sin necesidad de descargar un software para el ordenador, y resultado tan **intuitiva** que no necesite un aprendizaje previo extenso para programar cosas sencillas. De esta forma se reduce el tiempo de trabajo de forma significativa, siempre orientándose a aplicaciones sin mucha complejidad. Aunque en este proyecto se hable siempre de esta plataforma como “App Inventor”, en realidad su nombre correcto es “App Inventor 2”, una segunda versión del ya obsoleto App Inventor Classic, cuyo uso ya está prácticamente desaparecido.

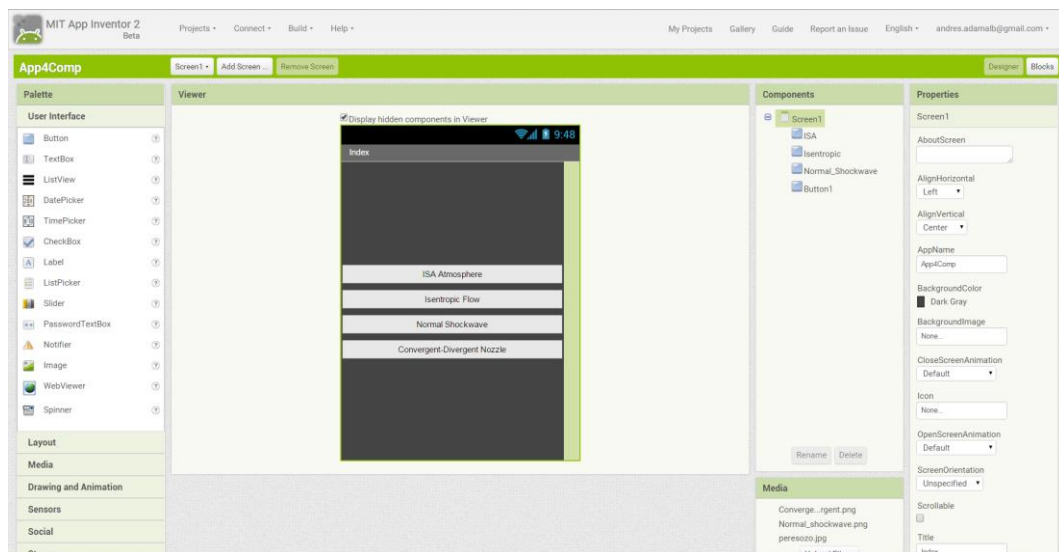
La principal página web sobre App Inventor es <http://appinventor.mit.edu/> [6], en donde se encuentran explicaciones sobre su funcionamiento y tutoriales para aprender a utilizarlo.

Lo primero que llama la atención de App Inventor es su forma de trabajar con él. Las aplicaciones se diseñan desde el navegador web, chrome o firefox, en la web <http://ai2.appinventor.mit.edu/> [7]. Al ser una aplicación Google, hace falta una cuenta Gmail para entrar. El trabajo se guarda automáticamente a medida que se trabaja, y permite añadir “checkpoints” para volver a ellos después, que es básicamente guardar el proyecto con otro nombre y seguir trabajando con el original.

3.3.1. Funcionamiento

Designer

La plataforma tiene dos partes: “Designer” y “Blocks”. La parte de “Designer” es para la **presentación** de la app móvil: botones, texto a mostrar e introducir, sonidos, etc. Se ve así:



Il·lustració 3.5. Parte "Designer" de App Inventor

Éste es el punto fuerte de App Inventor. Se arrastran los componentes a elegir desde el menú izquierdo ("Palette") a la visualización de la App, y solo se tienen que organizar. Cada componente (botón, lista, cuadro de texto, etiqueta...) tiene ya su comportamiento implícito, y no se puede modificar. La organización en la pantalla resulta muy intuitiva y bastante inequívoca respecto a su versión real en un terminal móvil.

El menú de la derecha, llamado "Components", permite seleccionar, cambiar el nombre o eliminar los componentes ya incluidos en la aplicación.

El menú de más a la derecha, "Properties", sirve para editar las propiedades de la pantalla en general (color de fondo, alineación de los componentes, icono de la aplicación, orientación de la pantalla, etc.) o de un componente en particular (color, tamaño y estilo de la letra y demás funciones específicas de cada componente).

Los componentes más importantes para este proyecto, que serán utilizados más adelante, son:

- **Button:** un botón, cuenta como seleccionado cuando el usuario lo aprieta y lo suelta.
- **Textbox:** cuadro de texto, permite introducir texto como tal, o solo números. También se puede mostrar un texto por él, pero por defecto es editable.

- **Label:** etiqueta, simplemente para mostrar un texto escrito no editable.
- **Listpicker:** es un componente de apariencia idéntica a la de un botón, pero cuando se pulsa aparece un menú para seleccionar entre diferentes opciones previamente elegidas. Por defecto, una vez seleccionada una opción el menú se cierra y el usuario no observa ningún cambio en el *listpicker*, por lo que sería muy útil señalar la opción escogida, por ejemplo, modificando el texto de éste.
- **Notifier:** es un elemento no visible para el usuario, pero que tiene que estar ahí para realizar cualquier advertencia, o abrir un cuadro de diálogo. El componente solo se coloca para *poder* hacerlo y elegir el color predeterminado con el que aparecerá, pero el texto se elige en la parte de bloques.
- **Arrangements:** son la espina dorsal de la organización de los componentes. Por defecto, la pantalla de la aplicación organiza los componentes de forma vertical y alineada a la izquierda. Aunque se puede modificar para alinearlos al centro o a la derecha, sigue habiendo solo un componente encima de otro, nunca al lado. Esto resulta muy poco práctico, y para eso están los *arrangements*, ya sean verticales, horizontales o tablas. Por ejemplo, si se coloca un *arrangement* horizontal, permite colocar elementos uno al lado de otro. Se puede introducir un *arrangement* dentro de otro, de forma que con un poco de tiempo y práctica se puede conseguir una organización correcta de los componentes.

Un ejemplo de algunos de estos componentes es el siguiente:

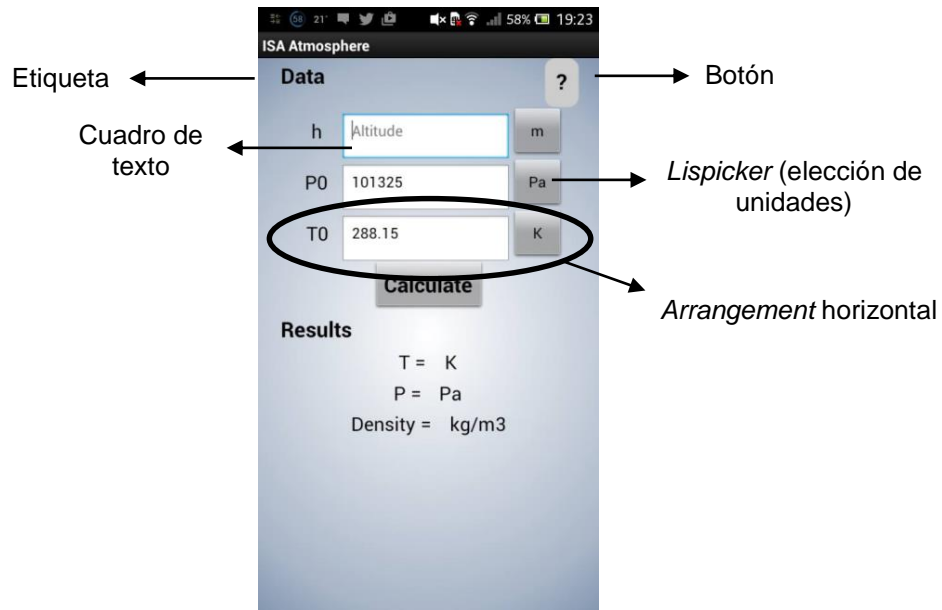


Ilustración 3.6. Ejemplo de visualización de componentes

La notación por defecto de los decimales de la aplicación es de la forma inglesa, que separa los enteros de los decimales por un punto. Por simplicidad, se adopta esa notación también en la App creada y, como extensión, en la memoria del proyecto.

Un último aspecto a comentar es que la aplicación también permite crear diferentes pantallas. Esto, a efectos prácticos, es una aplicación totalmente nueva y diferente de la otra, que ni siquiera comparte la parte de *blocks*.

Blocks

La parte de “Blocks” es para el funcionamiento de la app en sí. A diferencia de otros lenguajes de programación escritos en código, App Inventor utiliza **bloques**. Una secuencia de acciones será una secuencia de bloques acoplados entre ellos, provocados por una acción determinada. Este tipo de accionamiento tiene sentido si se analiza el entorno en el que se utiliza la App creada: el usuario no quiere tener un programa de cálculo sino una aplicación móvil, y ni siquiera tiene acceso al código de ésta. Así, en vez de hacer un código lineal, App Inventor crea series de bloques orientados a su interacción con el usuario.

Cada bloque representa una función a realizar y se clasifican en diferentes tipos: control, matemáticas, texto, etc. En general, los bloques van desde condiciones *if* y lógica *true* o *false* hasta análisis de un texto, como puede ser evaluar la longitud de un texto o analizar si un cierto texto contiene algún otro texto en él. Destaca la ausencia de tablas de más de una dimensión. En su lugar hay listas, que pueden ser vistas como vectores en otros lenguajes de programación, de 1xn componentes.

Tienen especial repetición en este proyecto varios tipos de bloques:

- **Inicialización de variables:** como en otros lenguajes de programación, se deben declarar las variables que se van a utilizar y darles un valor inicial. Se inician cuando se abre la pantalla. Normalmente se les llama *global*, que significa que son comunes en todo el programa (sin extenderse a las otras pantallas). Las variables locales también existen, aunque son de un uso menos extendido, y se usan principalmente en las funciones (procesos).
- **Inicios de secuencia:** son de la forma “When...”, son los que detectan un acontecimiento y desencadenan la acción asociada.
- **Bloques matemáticos:** marcados en azul, se puede incluir uno dentro de otro para crear ecuaciones más complejas.
- **Bloques condicionales *if*, *else if*, *else*:** evalúa una condición. A diferencia de otros lenguajes, que tienen el *switch*, éste es el único condicional existente.
- **Procedimientos:** también llamados “funciones” por analogía con otros programas como Matlab, ejecutan un procedimiento o devuelven un valor cuando son llamados y, en algunos casos, se les envían valores necesarios. Resultan muy útiles para ejecutar algo desde diferentes bloques de secuencia o repetidas veces.

La parte “Blocks” de App Inventor se ve así:

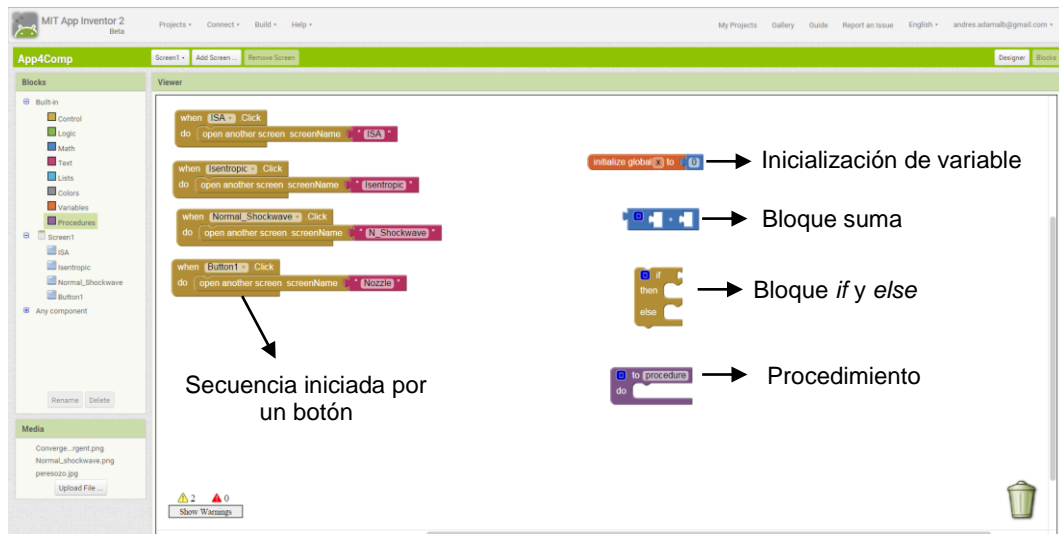


Ilustración 3.7. Visualización de la parte "Blocks" de App Inventor

3.3.2. Compilación y prueba del programa

A diferencia de lenguajes como C, no hay un compilador como tal y la posibilidad de ejecutar el programa para probarlo, sino que se realiza con un dispositivo Android o un emulador en el ordenador. La prueba en el dispositivo Android se hace a través de la App "AI2 Companion", con la cual se escanea un código QR y se necesita conexión Wi-Fi. De esta forma el dispositivo se sincroniza con el trabajo realizado en el ordenador en tiempo real, y se puede probar en cualquier momento. El emulador funciona de la misma forma, pero su instalación puede dar problemas, y su visualización no siempre es cómo se vería en un dispositivo, así que resulta mucho más eficiente trabajar con un dispositivo Android.

3.3.3. Ventajas, inconvenientes y limitaciones

Viendo el funcionamiento de la aplicación, las ventajas son evidentes: el ámbito de presentación e interacción con el usuario es muy sencillo, intuitivo y pocas cosas pueden fallar. En este aspecto, puede ser más rápido que programar en código con cualquier otro lenguaje.

En general, la plataforma entera tiene un **funcionamiento intuitivo y de aprendizaje rápido**. Por supuesto, no se trata de verlo y dominarlo, pero resulta

muy útil cuando el usuario no está familiarizado con la programación móvil y quiere crear una aplicación sencilla en poco tiempo.

Por la parte de “Blocks”, no se ven muchas limitaciones a un nivel de programación ocasional, sin entrar en trabajo de punteros. La parte matemática está completa, con algunas funciones útiles como devolver el residuo de una división, y la parte de control tiene suficientes funciones como para hacer posible muchas aplicaciones, incluyendo la de este proyecto.

Los inconvenientes pueden comenzar por su parte gráfica. Al tener unos componentes determinados, no se pueden crear nuevos ni modificar su comportamiento interno, lo cual hace que la plataforma esté **limitada en su vertiente gráfica**. Sin embargo, esto no es muy influyente en este proyecto, donde la funcionalidad es más importante que su presentación.

Respecto a su vertiente funcional (bloques), si bien no está limitado en capacidad, sí lo está en rapidez de programación y optimización. Acoplar diferentes bloques para **crear ecuaciones complejas resulta difícil, confuso y lento**, y una vez creada es prácticamente ilegible para una comprobación del código. Y cuando se produce algún error es difícil encontrarlo, ya que el programa solo señala las incongruencias de bloques asociados, no cuando alguna variable da algún valor imposible (infinito, por ejemplo) y, como se prueba en un dispositivo móvil desde el punto de vista del usuario de la App, se tiene un acceso limitado a los resultados internos desde el punto de vista del “tester”.

Respecto a la optimización, se puede destacar la falta de tablas de más de una dimensión y la ausencia absoluta de estructuras, que son ambas sustituidas por listas de 1xn componentes que no tienen que ser necesariamente del mismo tipo, pero su uso resulta muchas veces incómodo, ya que necesita más bloques y hace el código más grande y difícil de leer. Por otra parte, el usuario de App Inventor no tiene idea alguna de qué función requiere más tiempo de cálculo por parte del dispositivo. Por ejemplo, por regla general y con la posible excepción de Matlab, se conoce que la función “switch” es más rápida que la función “if”. En App Inventor no hay “switch”, pero salen dudas como “¿Qué es más rápido, plantar la fórmula entera en el bloque principal o utilizar una función a llamar?”, “¿Utilizar variables globales o locales?”, sobre las que no hay información. Al final el programa puede cargar mucho al dispositivo, pero **resulta muy difícil su optimización**.

Otra cosa a comentar como limitación es su precisión. Una comprobación de ella se puede encontrar en <http://www.exploringbinary.com/numbers-in-app-inventor->

[are-stored-as-floating-point/](#) , publicada por Rick Reagan [8]. En ella se demuestra que la precisión del programa se vuelve nula a partir de 10^{-17} , e introducir números decimales de más de 16 dígitos da problemas. Son números pequeños, pero pueden afectar en cálculos como una derivada numérica. Por el otro extremo, el programa también da problemas con números enteros de más de 17 o 18 cifras.

Por último, aunque su uso en un navegador resulte muy sencillo y la disposición de los bloques muy visual, eso se complica cuando el programa tiene muchos cálculos detrás. La disposición de los bloques por la pantalla puede ser errática, y es importante que el programador los mantenga ordenados. Si otra persona tiene que retomar el trabajo dejado por el primero, deberá buscar cada bloque moviéndose por la pantalla, desorientándose fácilmente y pudiendo haber bloques encima de otros, ocultándolos. También, cuando el programa tiene una gran cantidad de bloques, **su funcionamiento se vuelve lento**. Moverse por la parte de bloques resulta muy pesado entonces, ya que el movimiento pasa de ser fluido a dar saltos.

4. Cumplimiento del alcance. Programación de la App

La programación de la aplicación móvil “App” de este proyecto se realiza con la plataforma de programación App Inventor. Como ya se ha explicado anteriormente, dicho programa está dedicado a programadores novatos, sin grandes conocimientos de programación, y no está enfocado a realizar procedimientos matemáticos complejos, por lo que un programa relativamente sencillo en lenguajes como C++ o Matlab aumenta su dificultad en esta plataforma.

Pese a ser intuitivo, el autor de este proyecto desconocía este lenguaje anteriormente y su proceso de aprendizaje se llevó a cabo en gran parte al mismo tiempo que se programaba esta aplicación, comenzando por las partes más sencillas para ir pasando a los procedimientos más complejos. Esta es la razón por la que el código algunas veces podrá parecer torpe a aquel lector experimentado en App Inventor. Aunque se ruega perdón por herir cualquier sensibilidad profesional, se valora que el código *funcione* al fin y al cabo, y que este proyecto pueda servir de ayuda y mejora a futuros programadores en App Inventor.

A continuación se explicará cada función de la aplicación móvil. Siguiendo el estilo de App Inventor, se separan las explicaciones en *Presentación*, que es la parte gráfica de la App y su interacción con el usuario, y *Operaciones*, que es su funcionamiento interno y cálculos que realiza. Hay procedimientos y explicaciones que se podrían repetir en cada función. Para evitar esto, se explican en el apartado 4.2 con la función ISA, y de allí se adquiere el comportamiento general de la App.

4.1. Primera pantalla

En el momento de hacer la App, se desconoce cuál será la función más utilizada por los usuarios, y cada uno puede tener un “favorito” diferente. Así pues, resulta difícil y poco práctico decidir una función “inicial” que se abra al iniciar la aplicación móvil. Así pues, se destina la primera pantalla a un menú de opciones para ir a la

función deseada. Cuando se selecciona una, se abre en una nueva pantalla. Para retroceder al menú, basta con apretar el botón “Atrás” que tienen todos los terminales móviles Android.

La presentación de la primera pantalla es la siguiente:

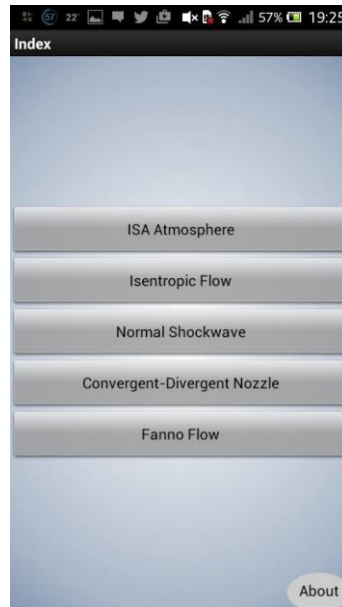


Ilustración 4.1. Primera pantalla de la aplicación

4.2. Atmósfera estándar internacional ISA

La atmósfera internacional ISA es un modelo de la atmósfera terrestre que permite obtener valores de presión, temperatura, densidad y viscosidad en función de la altitud. Es un marco de referencia para la navegación aérea y cálculos aerodinámicos.

En esta aplicación se calculan los valores de presión, temperatura y densidad en función de la altitud. Aunque la atmósfera ISA llega hasta la mesopausa, en la App solo se calculan valores desde los -1000 hasta los 20000 metros sobre el nivel del mar. Dado que el rango de vuelo de los aviones comerciales está alrededor de los 10500 metros y el público objetivo de la App móvil, este rango debería ser suficiente. Los valores de referencias de la ISA se obtienen de la norma ISO relacionada [9] y son, hasta la estratosfera:

Capa	Altura geopotencial	Gradiente térmico a (K/km)	Temperatura base T_0	Presión base P_0
Troposfera	0.0	-6.5	288.15	101325
Tropopausa	11000	0.0	216.65	22632
Estratosfera	20000	+1.0	216.65	5474.9

Tabla 4.1. Valores de referencia de la atmósfera ISA (1976)

Donde la altura geopotencial significa una altura suponiendo que el factor de la

Los valores de referencia se relacionan con las siguientes ecuaciones:

Para $a=0$

$$T(h) = T_0 \quad (4.1)$$

$$P(h) = P_0 e^{-\frac{g}{rT}(h-h_0)} \quad (4.2)$$

$$\rho(h) = \rho_0 e^{-\frac{g}{rT}(h-h_0)} \quad (4.3)$$

Para $a \neq 0$

$$T(h) = T_0 + a(h - h_0) \quad (4.4)$$

$$P(h) = P_0 \left(\frac{T(h)}{T_0} \right)^{-\frac{g}{ar}} \quad (4.5)$$

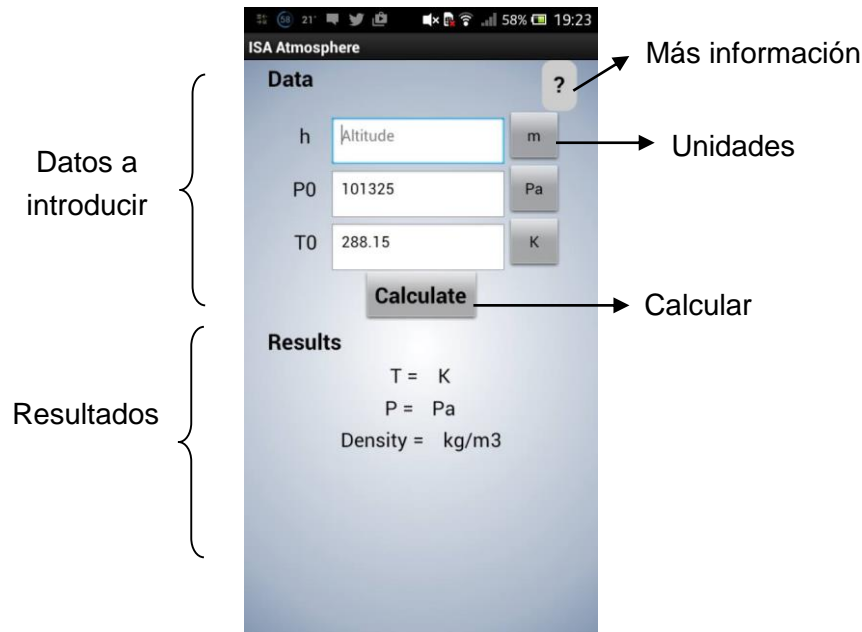
$$\rho(h) = \rho_0 \left(\frac{T(h)}{T_0} \right)^{-\frac{g}{ar}-1} \quad (4.6)$$

Donde g es la constante de gravedad, 9.81 N/kg y r la constante individual del aire, $287 \frac{\text{m}^2}{\text{s}^2\text{K}}$.

En ciertas condiciones, la atmósfera ISA puede variar de valores base en altitud 0 metros, por lo que resultará útil que el usuario pueda modificarlos.

Presentación

La función de atmósfera ISA se ve de la siguiente forma:



Il·lustració 4.2. Presentació de la funció ISA

En su interacción con el usuario, la función ISA funciona de la siguiente manera:

- Cuando el usuario abre la función, los valores predeterminados de P_0 y T_0 ya están introducidos. Puede modificarlos haciendo click en ellos.
- Cuando haya introducido todos los datos, aprieta el botón “Calculate” para obtener los resultados.
- Si alguna entrada de datos (“input”) es incorrecta (vacía, altitud fuera de rango o presión negativa) no se realizan los cálculos y el usuario recibe un mensaje informándole de ello.
- Para más información, el usuario puede hacer click en el botón “Más información”.
- Las unidades de los inputs pueden ser cambiadas individualmente haciendo click en ellas, y el usuario puede elegir entre varias de ellas.
- Las unidades de los resultados coinciden con las de los inputs.
- Los resultados solo varían si el botón de calcular ha sido apretado y los cálculos han sido realizados.

Estos criterios de funcionamiento, excepto tener los valores de presión y temperatura predeterminados se repiten en las demás funciones de la App, y no serán explicadas de nuevo en cada una.

La información mostrada con el botón de más información se muestra de la siguiente forma:

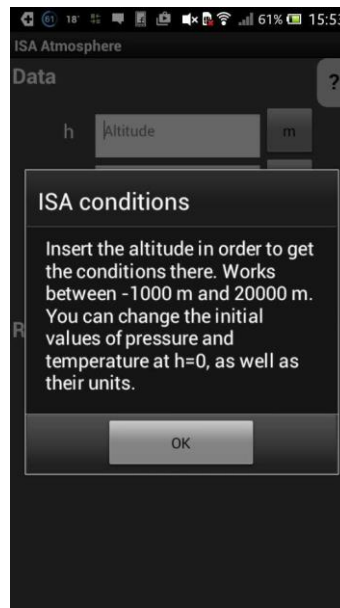
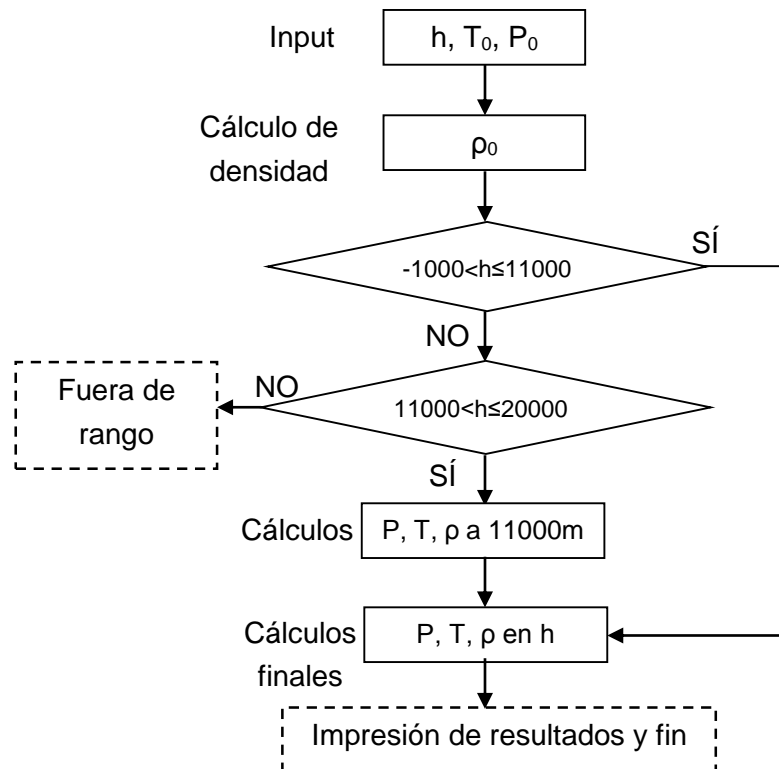


Ilustración 4.3. Guía en ISA

Más aspectos de la presentación de la función ISA se pueden encontrar en los anexos, en el apartado A.1.1.

Operaciones

Dado que los valores de presión y temperatura a altitud 0 de la aplicación pueden variar, el proceso de cálculo en la región de 11000 m hasta 20000 m no es inmediato, sino que se deben calcular las propiedades a los 11000 metros con el primer tramo y de allí utilizar los valores iniciales para este segundo tramo. El procedimiento a seguir es, por tanto:



El programa, además, debe comprobar que los datos introducidos sean correctos y que realmente *haya* datos introducidos. Además, debe convertir los datos introducidos a unidades del sistema internacional (SI) y, al acabar, devolverlos a las unidades introducidas por el usuario.

Los bloques de la aplicación se pueden encontrar en los anexos, en el apartado A.1.2.

4.3. Flujo isentrópico

Esta función relaciona propiedades estáticas y estancamiento a través del número de Mach. Es un programa sencillo pero útil, que usa la teoría del apartado 3.2.2.

Dado que en la mayoría de problemas de estudiantes no se opera con relaciones sino con valores dimensionales, resulta de especial utilidad incluir la posibilidad de utilizarlos en el cálculo de la aplicación y no tener que realizar cálculos posteriores.

El usuario debe introducir el valor del número de Mach y coeficiente de dilatación adiabática del gas, γ . Además, puede introducir valores de presión, temperatura y densidad de estancamiento, eligiendo las unidades, y se le darán las estáticas, y viceversa.

Presentación

La función de flujo isentrópico se ve de la siguiente manera:

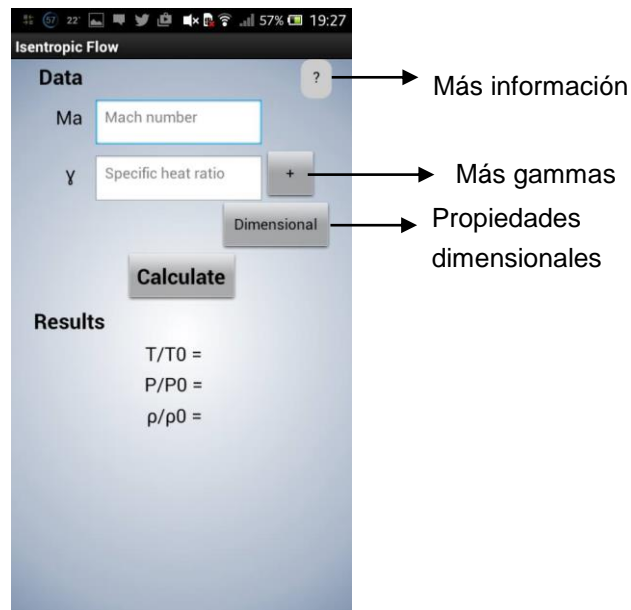
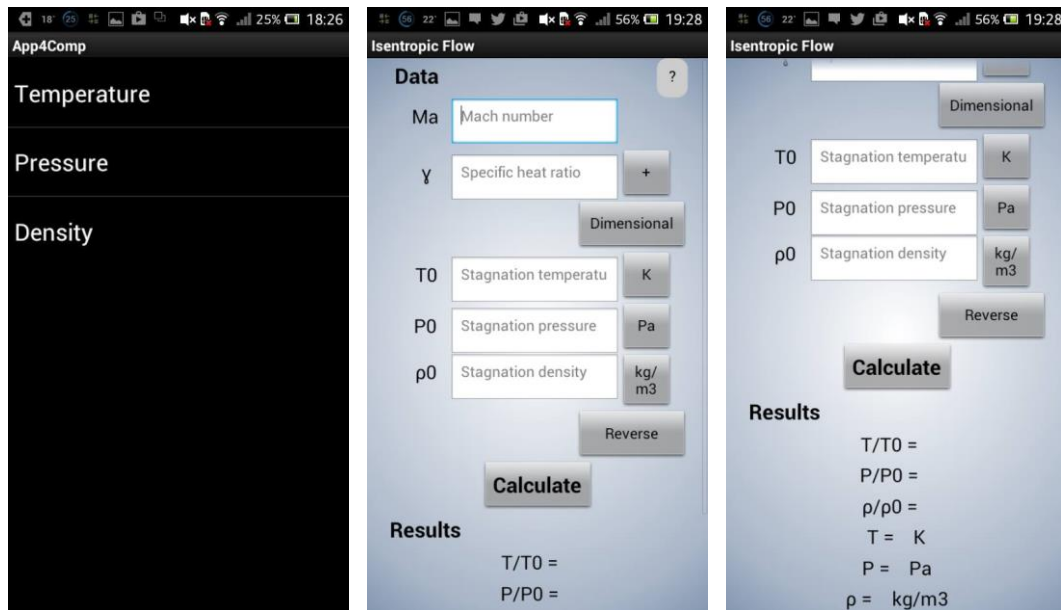


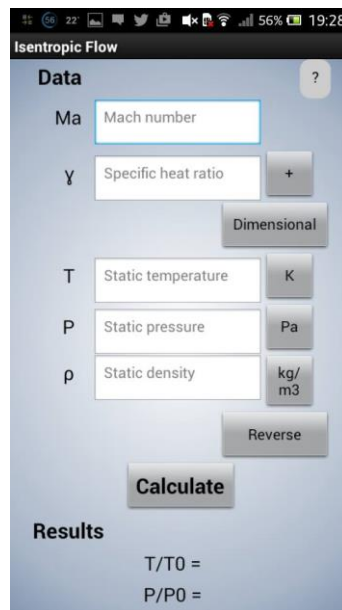
Ilustración 4.4. Presentación de la función flujo isentrópico

Con el botón “Dimensional”, se puede elegir una propiedad para operarla de forma dimensional:



Il·lustració 4.5. Elección de propiedad dimensional y presentación de la función flujo isentrópico.

Como se ha mencionado, la función también calcula propiedades de estancamiento a partir de las estáticas. Para ello se aprieta el botón “Reverse”, y las propiedades dimensionales a introducir pasan a ser las estáticas:



Il·lustració 4.6. Problema inverso en la función flujo isentrópico

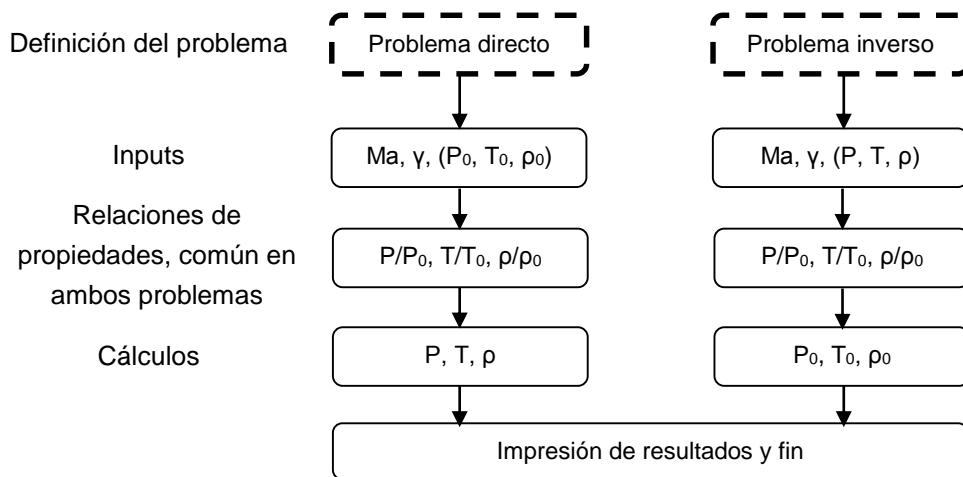
Algo que se introduce a partir de esta función en adelante y que puede resultar útil es el botón “Más gammas”, representado por un “+”. Al pulsarlo se despliega una

lista de gases con sus respectivos valores de coeficiente de dilatación adiabática. Al elegir uno, la casilla de γ se rellena con el valor escogido. La lista y su visualización en la App se encuentran en los anexos, en el apartado A.2.1.

Operaciones

El proceso de cálculo de esta función usa la teoría del apartado 3.2.2, en particular las ecuaciones (3.12), (3.15), (3.16), (3.17). Estas tres últimas ecuaciones aprovechan que las relaciones de propiedades estáticas y de estancamiento de la densidad y presión son la relación de temperaturas elevada a ciertos exponentes.

El proceso de cálculo es el siguiente:



Los bloques y sus explicaciones se pueden encontrar en los anexos, apartado A.2.2.

4.4. Ondas de choque normales

En el cálculo de las ondas de choque normales, se piden como inputs el coeficiente de dilatación adiabática del gas, γ , y un número de Mach. Las relaciones de propiedades del gas se pueden obtener a partir del Mach antes de la onda de choque (1) o después (2). Así pues, el usuario debe introducir alguno de los dos Mach y el factor γ para calcularlas.

Además, es útil que el programa calcule propiedades dimensionales y no solo relaciones de propiedades, como con el flujo isentrópico.

La notación y ejemplo utilizado es el siguiente:

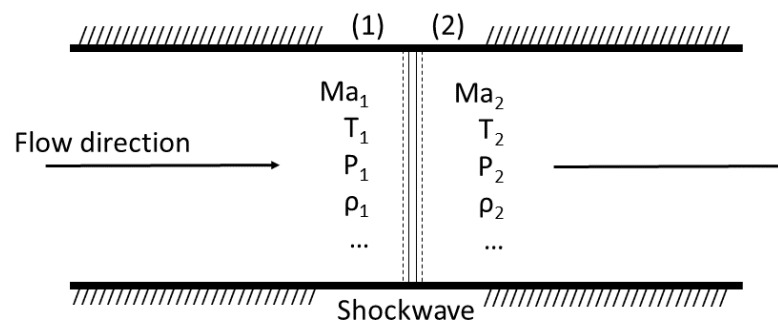


Ilustración 4.7. Ilustración de onda de choque normal

Presentación

La función de ondas de choque normales se ve de la siguiente manera:

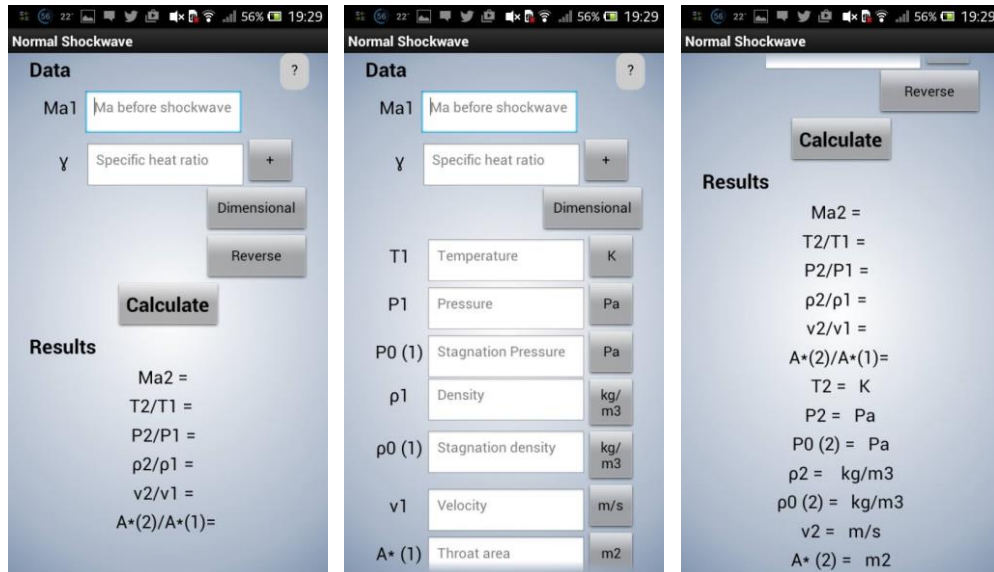


Ilustración 4.8. Función onda de choque normal, sin y con inputs dimensionales

El comportamiento de esta función es muy similar a la de flujo isentrópico. En este caso, la opción de invertir el problema no se limita solo al caso dimensional. Las unidades posibles son las mismas, y las del área de garganta pueden ser m^2 , cm^2 , mm^2 o in^2 .

En este caso se necesita una aclaración de la notación más clara, y para eso el botón de más información (marcado por “?”) abre una nueva pantalla con el dibujo anterior y una explicación, que se ve de la forma siguiente:

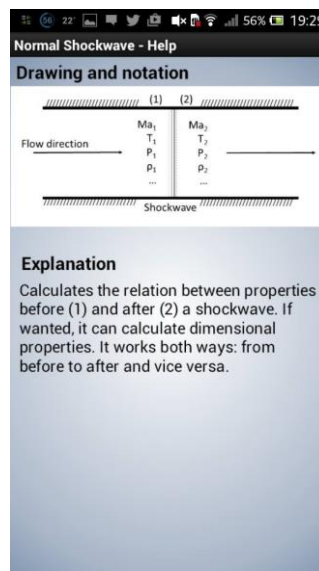
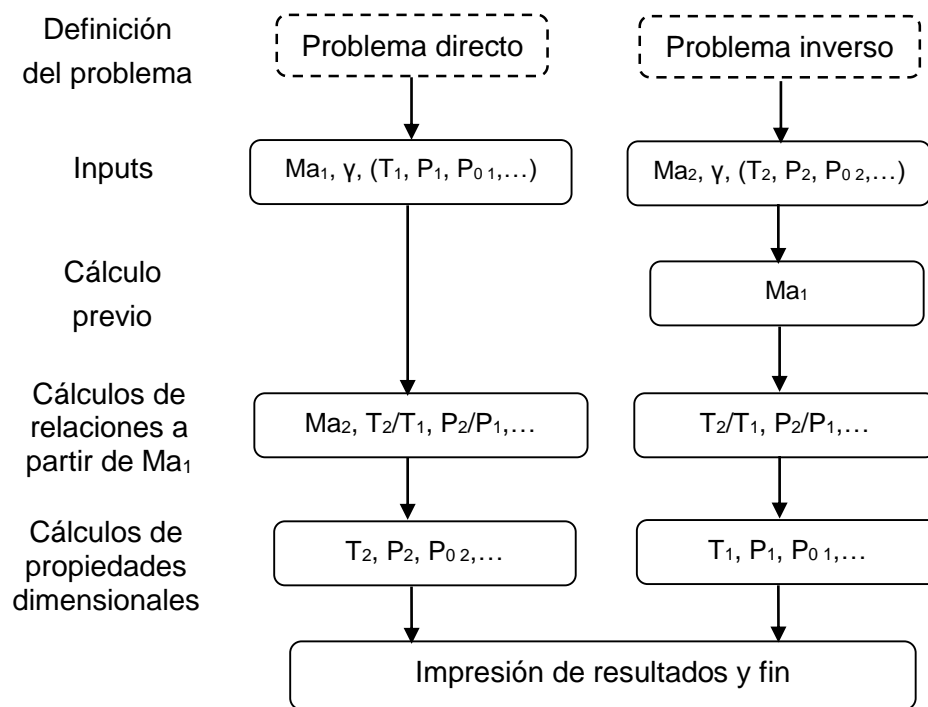


Ilustración 4.9. Guía en onda de choque normal

Operaciones

Las ecuaciones que gobiernan las ondas de choque normales, y por lo tanto las utilizadas en esta parte, son las obtenidas en el apartado 3.2.3.

El proceso de cálculo es el siguiente:



Además, como en los demás casos, se comprueban los inputs y se adaptan las unidades utilizadas.

Los bloques principales y funciones utilizadas se encuentran en los anexos, en el apartado A.3.1.

4.5. Tobera convergente-divergente

Esta es la función más compleja de la aplicación, basada en la teoría explicada en el apartado 3.2.4, y combina las ecuaciones de flujo isentrópico y onda de choque utilizadas hasta ahora.

La representación gráfica de la función podría tener la siguiente forma, aclarando la notación utilizada:

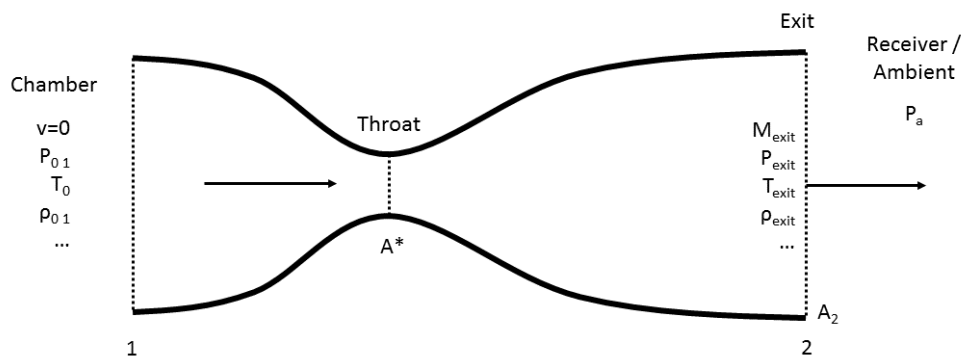


Ilustración 4.10. Ejemplo de tobera convergente-divergente

En la mayoría de Apps móviles sobre flujo compresible (Compressible Flow Tool [10], GDCalc [11]...) se trata el caso de toberas convergentes-divergentes como una pequeña extensión del flujo isentrópico, añadiendo al cálculo la relación entre relación de áreas (área de sección entre área de garganta) y número de Mach, y tratando siempre el caso adimensional. Hace falta solo un input (relación de presiones, temperatura, áreas, etc.), y los demás se calculan a partir de éste. De esta forma, se trata a la tobera como si siempre estuviese exactamente adaptada, y las condiciones de salida coinciden exactamente con las del ambiente receptor. Esta forma es útil y sencilla, pero incompleta respecto a la teoría involucrada en una tobera convergente-divergente.

Sería realmente útil una función que calculase el régimen de operación de la tobera, el flujo másico y las condiciones de salida dados unos datos de entrada como las condiciones de estancamiento en la cámara y la relación de áreas. Para eso hace falta operar con cantidades dimensionales, e incorporar algo que no ofrece ninguna otra aplicación móvil: en caso de una onda de choque, calcular en qué área se produce. Para ello hace falta un proceso iterativo, y es probablemente el motivo por el que no se incluye esta característica en otras aplicaciones.

Sería muy conveniente, entonces, dividir esta función en dos: los cálculos suponiendo tobera totalmente adaptada, adimensionales, y el caso genérico, dimensional y completo. El usuario selecciona el modo deseado con una opción en la parte superior de la pantalla.

4.5.1. Modo tobera adaptada

Presentación

La presentación de esta función cambia ligeramente respecto de las anteriores, por una parte porque el autor de este proyecto ya tiene un mejor conocimiento de App Inventor y puede organizarla mejor, y por otra parte porque hay una gran cantidad de inputs y outputs, especialmente en el modo genérico, y resulta de mayor utilidad dedicar más esfuerzo a la organización.

Al seleccionar el modo tobera adaptada, la opción preseleccionada como input es la relación P_2/P_1 , y se muestra de la siguiente forma:

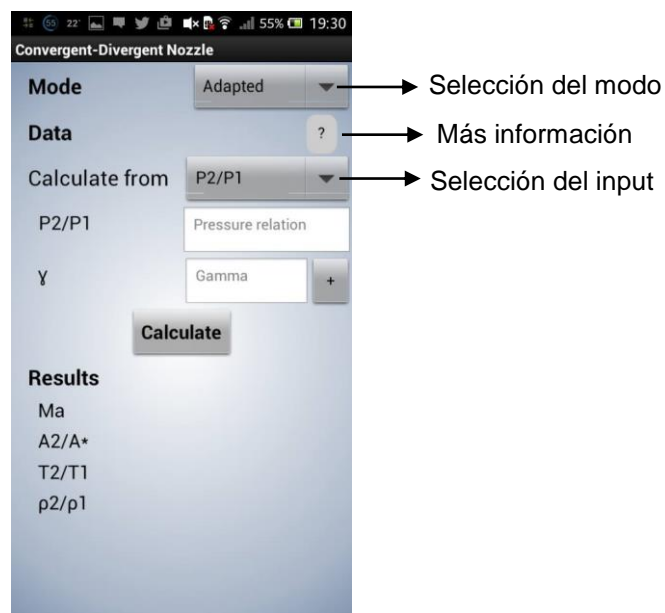


Ilustración 4.11. Presentación del modo tobera adaptada

El usuario puede seleccionar entre los inputs P_2/P_1 , A_2/A^* , T_2/T_1 y Mach de salida. De ellos, el caso más particular es el de A_2/A^* , pues tiene dos soluciones y se deben mostrar ambas. Las imágenes de esto se muestran en los anexos, apartado A.4.1.

El botón de más información muestra lo mismo para ambos modos de la función, que se abre en otra pantalla, y muestra lo siguiente:

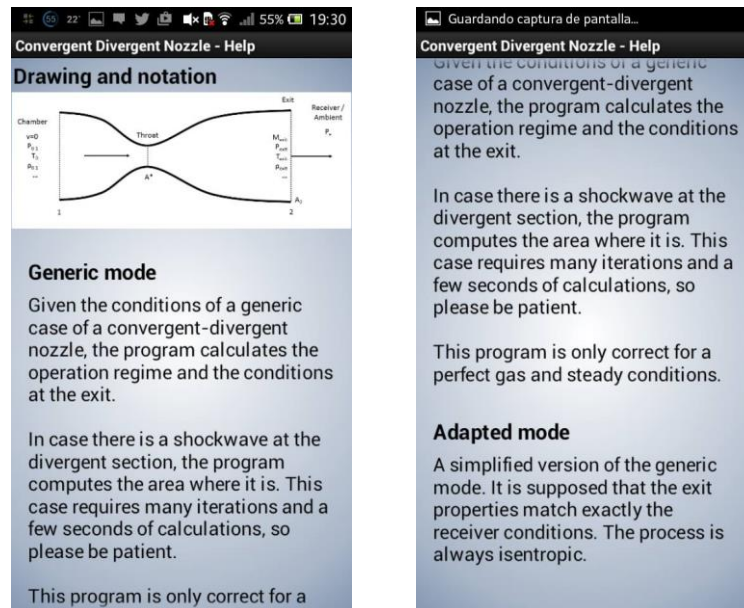
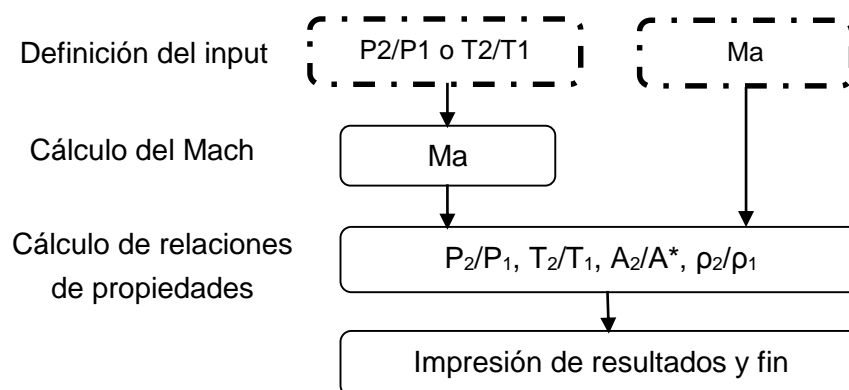


Ilustración 4.12. Guía práctica en tobera convergente-divergente

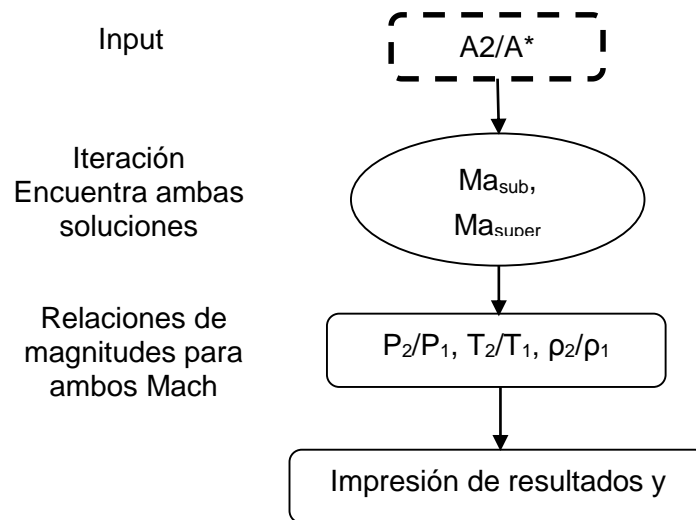
Igual que en las funciones anteriores, el usuario puede elegir un valor de gamma predeterminado con el botón marcado por el “+”.

Operaciones

El proceso de cálculo del modo tobera adaptada es el siguiente:



Para el caso en el que el input sea A_2/A^* , el proceso es un poco diferente:



Por claridad, se repite la ecuación (3.30), de forma aislada:

$$\frac{A}{A^*} = \sqrt{\frac{1}{M^2} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{\gamma + 1}{\gamma - 1}}} \quad (4.7)$$

Que tiene una representación gráfica como la siguiente imagen:

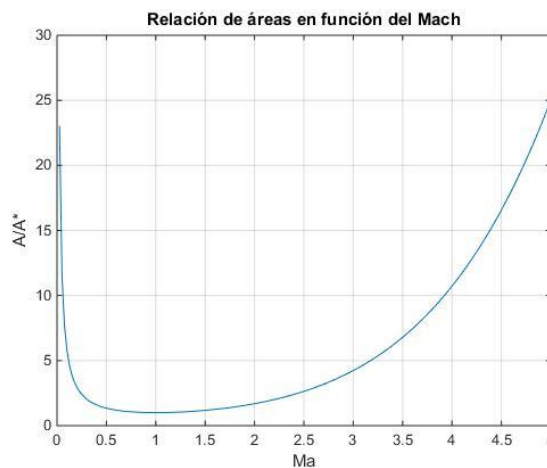


Ilustración 4.13. Relación de áreas en función del número de Mach

Viendo la forma de la gráfica, para encontrar el número de Mach dada la relación de áreas se puede utilizar un método Newton-Raphson, por el que se ensayan valores de la forma

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (4.8)$$

De esta forma se encuentra un valor x tal que $f(x)=0$. Así, la función a ensayar debe ser de la forma

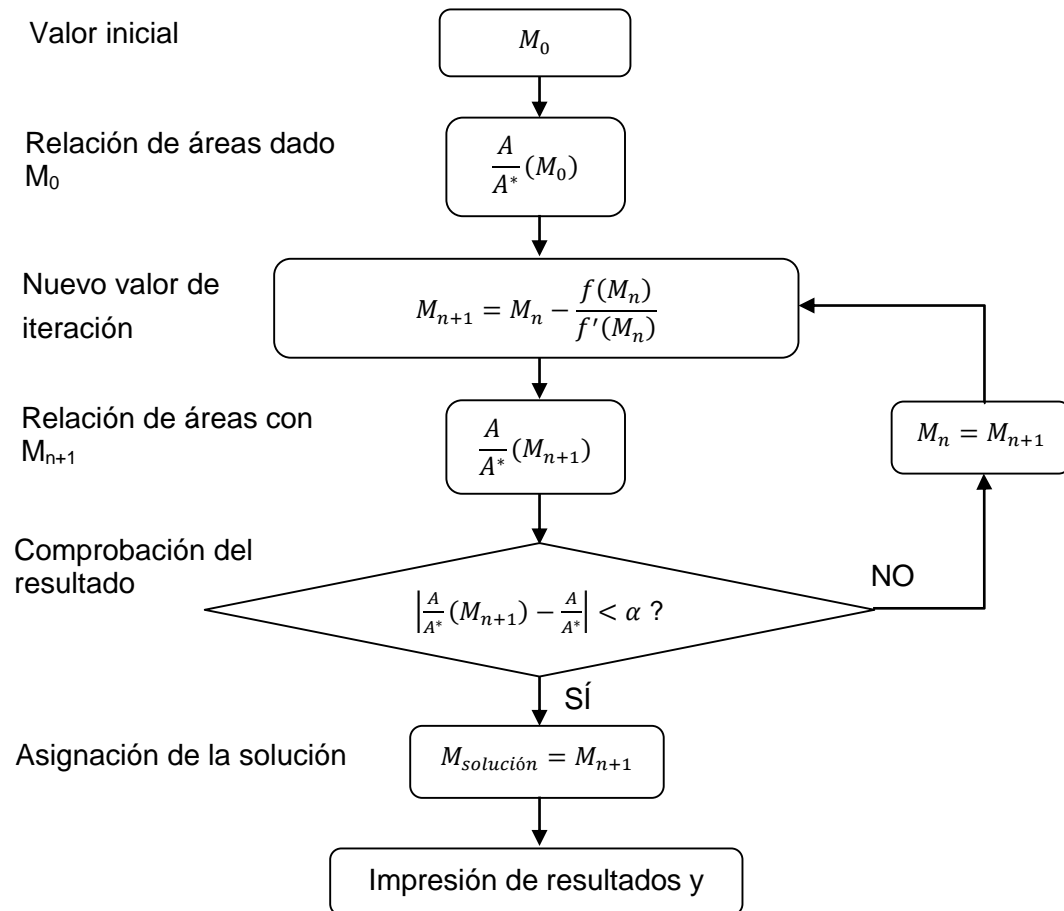
$$f(x) = \sqrt{\frac{1}{M^2} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{\gamma + 1}{\gamma - 1}} - \frac{A}{A^*}} \quad (4.9)$$

Y la derivada $f'(x)$, de forma numérica, es de la forma

$$f'(x) \approx \frac{f(x + \partial) - f(x)}{\partial} \quad (4.10)$$

Para utilizar el método Newton-Raphson para encontrar el valor deseado (subsónico o supersónico) se le asigna un valor inicial a la iteración más cercano al deseado. En concreto, para el subsónico se comienza a iterar desde un valor próximo a 0, y para el supersónico por un valor superior a 1 como, por ejemplo, 10. Viendo la Ilustración 4.13, el único valor inicial que daría problemas sería $Ma=1$, pues su derivada en este punto es 0.

El diagrama representativo de Newton-Raphson para este caso, con la función $f(x)$ mencionada antes:



Donde el valor α es una tolerancia del resultado elegida por el programador, teniendo en cuenta la velocidad de convergencia y la exactitud del resultado.

Los bloques principales del programa se pueden encontrar en los anexos, en el apartado A.4.1.

Por lo que respecta a los valores numéricos de la iteración, se establecen:

- Valor inicial de iteración para Mach subsónico: 10^{-12}
- Valor inicial de iteración para Mach supersónico: 10
- Tolerancia de la iteración Newton-Raphson: 10^{-12}
- Valor “ δ ” para la derivada numérica: 10^{-5}

Estos valores no son únicos, pero se ha comprobado que funcionan correctamente. Para aquellos que necesitan ser pequeños, 10^{-12} es un valor

suficientemente pequeño para dar resultados correctos. Sin embargo, la función “derivative_for_Arel” falla con un intervalo delta tan pequeño, por lo que se hace más grande y se consiguen resultados igualmente correctos.

Otro aspecto que se puede ver en el código es que solo usa el algoritmo para resolver problemas con relaciones de áreas menores a 5000. Esto es porque el programa, puede que por culpa de un valor inicial, o puede que por el propio funcionamiento de App Inventor, falla cuando intenta utilizar el algoritmo con valores a partir de, como mínimo, 5500, quedándose “congelado” e incapaz de seguir funcionando. Aunque estas relaciones son mucho mayores a las habituales en los problemas relacionados, se quiere que el programa funcione de forma tan correcta como sea posible, y en caso de no poder solucionarlo, evitar el cálculo y que el programa falle. Es por esta razón que para relaciones de áreas superiores a 5000 y valor gamma 1.4 se usan las soluciones aproximadas de la ecuación (4.7):

$$Ma_{subsónico} = \begin{cases} \frac{1 + 0.27 \left(\frac{A}{A^*}\right)^{-2}}{1.728 \left(\frac{A}{A^*}\right)} ; 1.34 < \frac{A}{A^*} < \infty \\ 1 - 0.88 \left(\ln \left(\frac{A}{A^*}\right)\right)^{0.45} ; 1.0 < \frac{A}{A^*} < 1.34 \end{cases} \quad (4.11)$$

$$Ma_{supersónico} = \begin{cases} \left[216 \frac{A}{A^*} - 254 \left(\frac{A}{A^*}\right)^{\frac{2}{3}} \right]^{\frac{1}{5}} ; 2.9 < \frac{A}{A^*} < \infty \\ 1 + 1.2 \left(\frac{A}{A^*} - 1\right)^{\frac{1}{2}} ; 1.0 < \frac{A}{A^*} < 2.9 \end{cases} \quad (4.12)$$

La concordancia de estas fórmulas aproximadas con la forma exacta no es muy buena a altos valores de relaciones de áreas. Es por ello que se informa al usuario con el mensaje “Changing to approximated formulas”.

Por otra parte, no se encuentran fórmulas aproximadas para valores de gamma diferentes de 1.4, y no tendría sentido hacer fórmulas para cada valor. Por eso la operación se cancela, informando al usuario de ello, si la relación de áreas es

superior a 5000 y la gamma diferente de 1.4. No es útil para resolver el problema, pero evita que el programa se quede congelado.

4.5.2. Modo genérico

En el modo genérico es el modo que aparece por defecto cuando se selecciona la función de tobera convergente-divergente. Se trata el problema de una tobera de geometría definida, separando una cámara de un gas conocido en condiciones de estancamiento, presión y temperatura conocidas, que descarga hacia un ambiente de presión conocida. Utiliza la teoría y suposiciones del apartado 3.2.4 (cuasi-unidimensional, estacionario, sin rozamiento con las paredes, etc.), por lo que el programa solo necesita los datos del área de salida A_2 y el área de garganta A^* para la definición de la tobera.

Así pues, los inputs que necesita el programa son:

- Presión de la cámara P_{01}
- Presión del ambiente receptor P_a
- Área de garganta A^*
- Área de salida A_2
- Temperatura de la cámara T_0
- Coeficiente de dilatación adiabática y del gas
- Constante específica del gas r .

Y los outputs, aunque podrían ser otros, se eligen como:

- Régimen de operación
- En caso de onda de choque, área en la que se produce
- Flujo másico
- Presión de salida
- Temperatura de salida
- Densidad de salida

Como se ha comentado antes, la capacidad de encontrar el área de la onda de choque es la característica más especial de esta función. Se planteó la posibilidad de incorporar una función para introducir la geometría de la tobera y que el programa encontrase la coordenada en la que se produce (por ejemplo, dar como output la distancia respecto a la garganta), pero complicaría notablemente el

programa. La forma más fácil de hacerlo sería que el usuario introdujese la geometría por puntos y que el programa interpolase entre ellos de forma lineal. Esto resulta muy tedioso para el usuario si tiene que introducir los puntos uno a uno, y aunque existe la posibilidad de importar ficheros de datos, no es fácil hacerlo con App Inventor. Finalmente, como el cálculo para encontrar la posición dado el área no es complejo, se deja que eso lo haga el usuario en un cálculo posterior.

Presentación

El aspecto en la App del modo genérico es el siguiente:

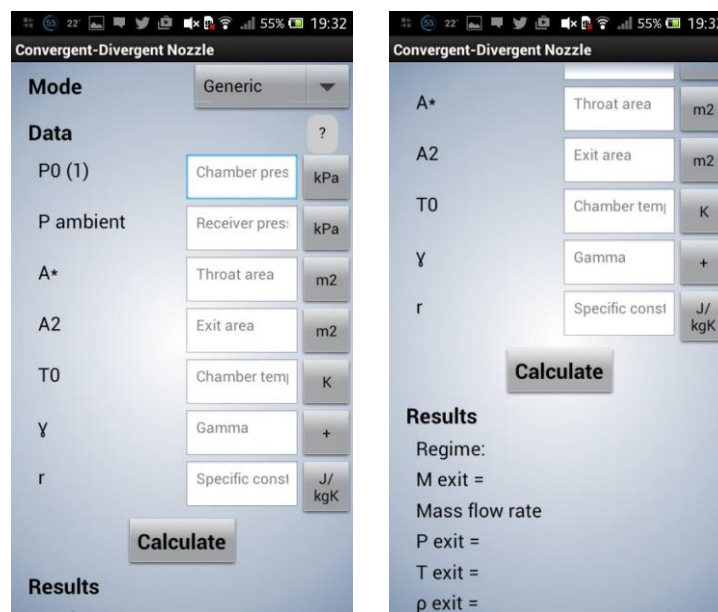


Ilustración 4.14. Presentación del modo genérico de la función tobera convergente-divergente

Como en las funciones anteriores, el usuario puede pedir más información (se muestra la pantalla de la Ilustración 4.12) y cambiar las unidades de las propiedades introducidas. La novedad en este modo es que cuando se selecciona la constante de dilatación adiabática del gas γ , automáticamente se rellena también la casilla de la constante específica del gas r .

En cuanto a las unidades de los resultados, son las mismas que los inputs. En caso de haber dos magnitudes del mismo tipo, se decide:

- El output de presión tiene las unidades de P_a
- El output de área de onda de choque tiene las unidades de A_2

Un problema ejemplo, solucionado por la App, se vería de la siguiente forma:

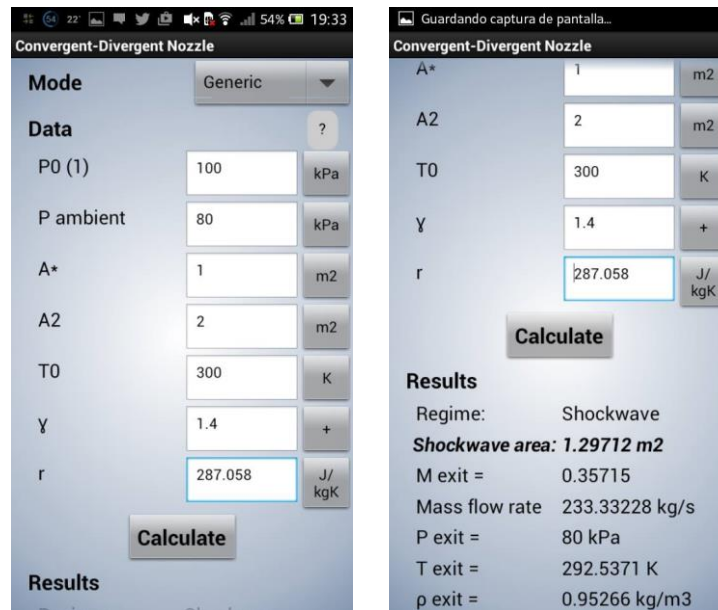


Ilustración 4.15. Problema ejemplo del modo genérico

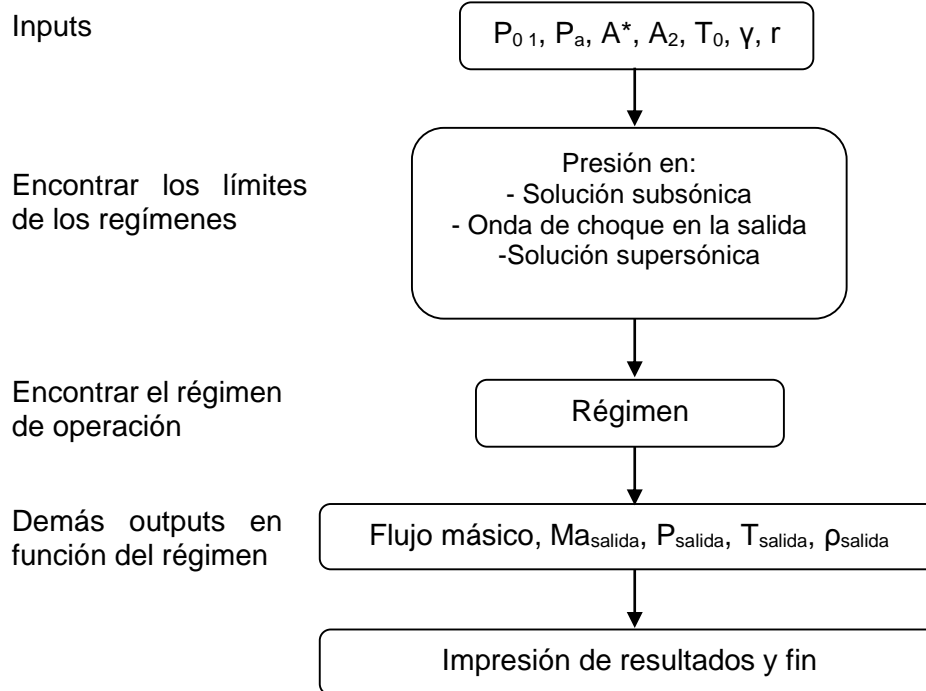
En caso de que la tobera no opere en régimen de onda de choque, el resultado de “Shockwave area” desaparecería.

Un aspecto que merece ser mencionado de este modo es su velocidad, más lenta en comparación con las otras funciones. En concreto, el problema ejemplo tardó unos 7 segundos con un modelo de móvil Sony Xperia. El motivo se explicará más adelante, en la parte de “operaciones”.

Operaciones

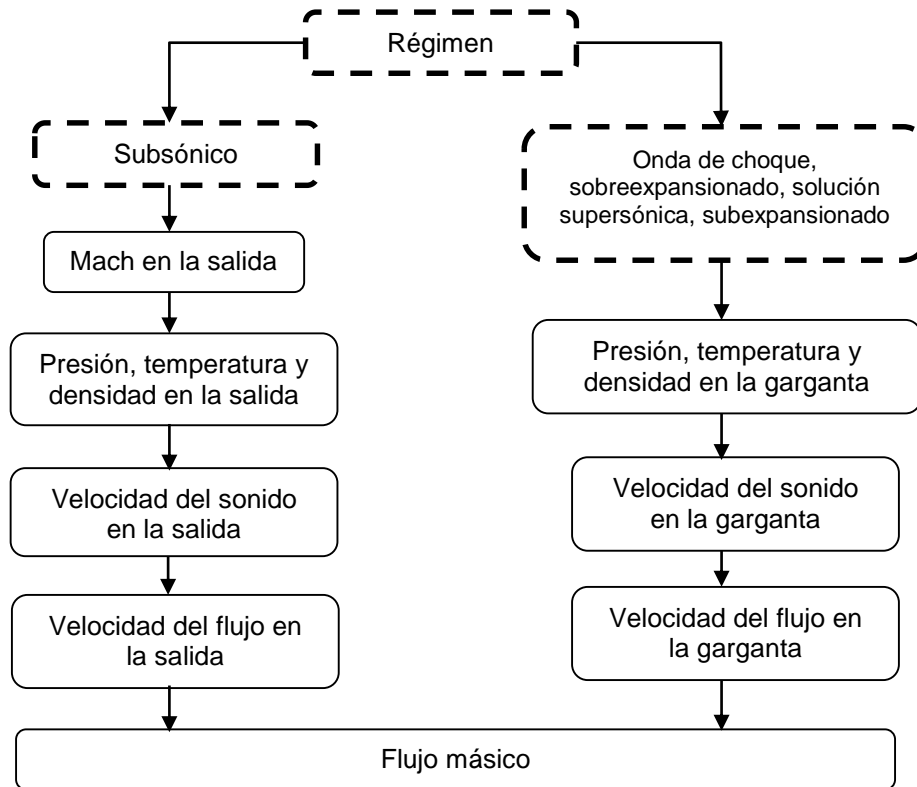
El funcionamiento de este modo es el más largo y complejo de la App. Es por eso que su diagrama de bloques se separa en varios más pequeños según su nivel de profundidad, o algunos para casos particulares.

El funcionamiento general es así:



Para encontrar los límites de los regímenes se necesita el Mach en la salida, y por tanto ambas soluciones de la ecuación (4.7). Esto es de nuevo la iteración por Newton-Raphson utilizada en el modo tobera adaptada.

Una vez obtenidos los límites y el régimen correspondiente siguiendo la teoría de la Ilustración 3.1, el flujo másico se encuentra de la forma:



Se puede apreciar que, en lo que respecta al flujo másico, el único caso en el que varía es en el subsónico. En los demás se mantiene constante, independientemente del régimen de operación. A esto se le llama que la tobera está bloqueada.

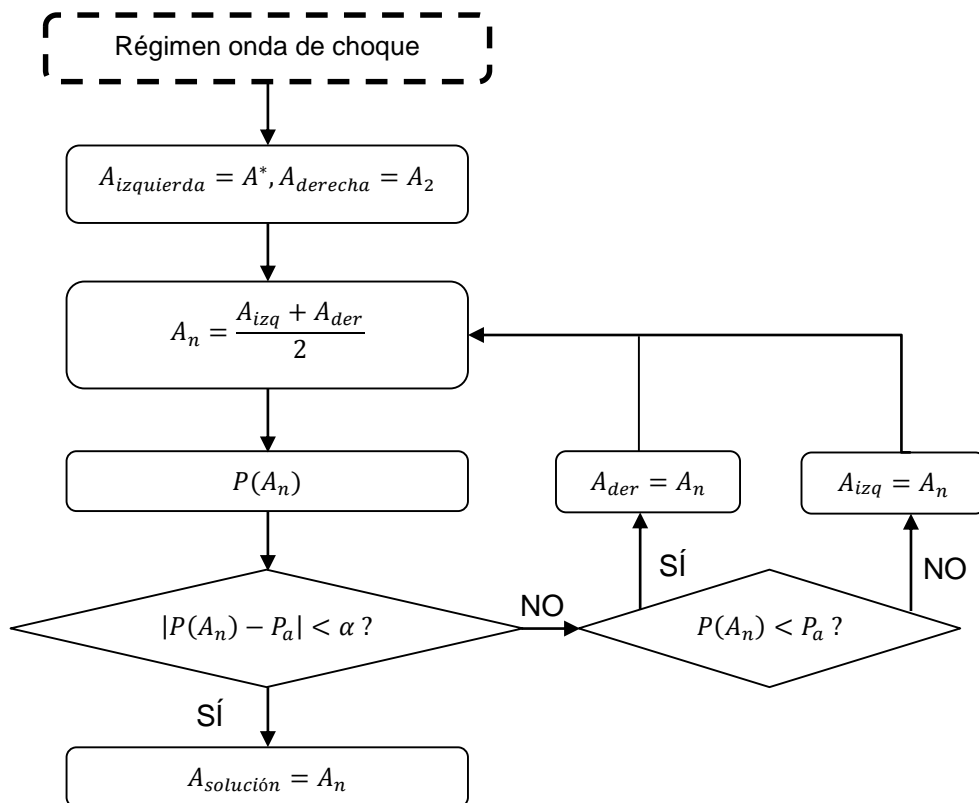
En los regímenes de subexpansionada, solución supersónica (adaptada) y sobreexpandida, la tobera tiene las mismas propiedades en la salida, por lo que éstas se realizan en un apartado común. El caso más particular es el de la onda de choque, que como es un proceso no isentrópico y su relación de números de Mach depende del Mach incidente, debe encontrarse el área donde se produce para calcular las condiciones de salida.

El proceso para encontrar el área de la onda de choque es iterativo. No obstante, y como se explicará más adelante en el apartado de dificultades encontradas, el proceso no puede ser como el Newton-Raphson utilizado para encontrar el Mach dada la relación de áreas, sino que se utiliza otro limitando los valores a ensayar en el dominio posible. Aprovecha también el hecho de que solo existe un área de onda de choque que sea solución del problema, y que la derivada de la función a

solucionar es siempre del mismo signo. Esto, explicado de forma más detallada, es:

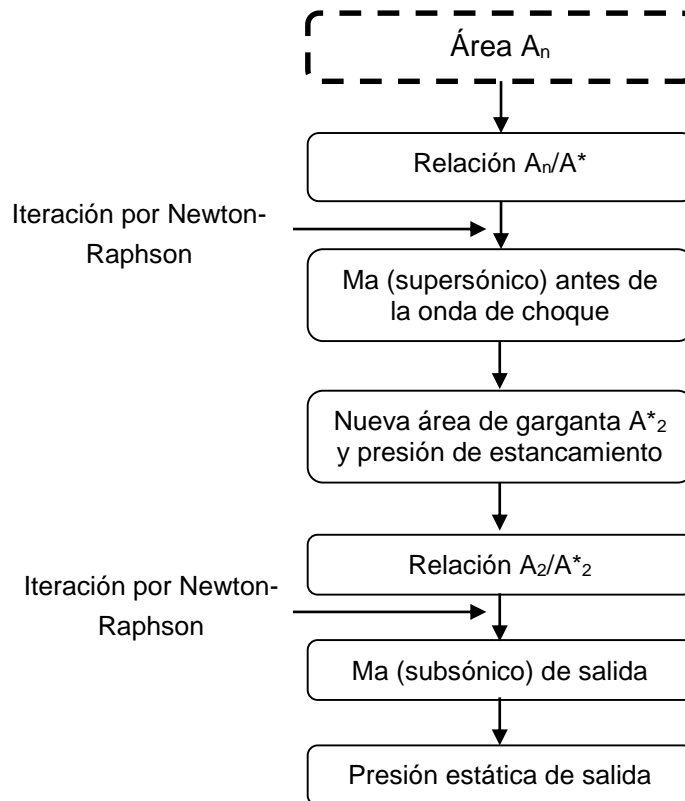
- La condición a evaluar es que la presión de salida dada un área de onda de choque debe ser igual a la presión del ambiente receptor P_2 . De forma que $P(A)=P_2$.
- Viendo el significado físico de la ecuación (y sino, una representación matemática), se observa que $P'(A)$ es siempre negativa, como mínimo para valores de área entre área de la garganta A^* y área de salida A_2 .
- Esto significa que, si se ensaya un valor de A y se obtiene $P(A)$, la relación $P(A)/P_2$ da información inequívoca sobre si el área solución es mayor o menor que la ensayada.
- En particular, para un valor A_0 y $P(A_0)$, si $P(A_0) < P_a$, $A_{solución} < A_0$, y viceversa.

Sabiendo esto se puede entender el algoritmo utilizado:



Es un proceso más lento que Newton-Raphson, pero asegura la convergencia para este tipo de problema.

Y para calcular la presión en la salida dado el área de la onda de choque, se hace:



Se puede observar entonces que, por cada iteración para encontrar la posición de la onda de choque, hay dos procesos iterativos en ella para encontrar un número de Mach dada la relación de áreas. Esta es la razón por la cual este modo es lento en comparación con las otras funciones.

Los bloques principales y explicaciones relacionadas de este modo se pueden encontrar, de nuevo, en los anexos, en el apartado A.4.2.

La tolerancia para la iteración encontrando el área de la onda de choque es del 0.01% de P_a . Esto asegura una buena exactitud de resultados cuando se opera con presiones bajas, mientras que mejora la velocidad de convergencia cuando se opera con presiones altas.

Igual que el modo tobera adaptada, la iteración para encontrar el Mach a partir de la relación de áreas no funciona para valores de relaciones mayores de 5000, por lo que se usan las fórmulas aproximadas para el caso del aire y se cancela para otros valores de gamma.

4.6. Flujo de Fanno

En esta función se trata el caso de un conducto recto y adiabático, con un flujo estacionario de un gas perfecto, sin despreciar la fricción. Solo se analiza el caso en el cual el conducto es igual de largo o menor que aquél que hace que las condiciones en la salida sean sónicas. De otra forma, el programa no dará resultados. Se basa en la teoría del apartado 3.2.5.

El dibujo y notación es el siguiente:

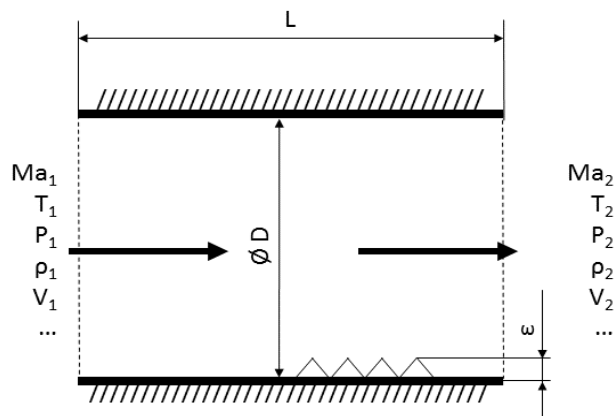


Ilustración 4.16. Esquema del flujo de Fanno

En esta última función del programa se optimiza la simplicidad de cálculo, pues las restricciones de tiempo del proyecto obligaron a programarlo en poco tiempo. Es por ello que se optó por el caso más sencillo de flujo de Fanno: El usuario introduce datos para calcular el Mach de entrada y el de salida, junto con los otros parámetros, y el programa calcula la longitud del conducto, la longitud que haría que el conducto se bloquease por fricción, y las relaciones de propiedades entre la entrada y la salida.

Como resumen, el programa necesita:

- Diámetro hidráulico del conducto D .
- Constante adiabática del gas γ .
- Números de Mach de entrada y salida o, si el usuario lo prefiere:
 - Constante específica del gas r .
 - Temperatura estática del gas T .
 - Velocidad del gas V .

- Coeficiente de fricción medio f_o , si el usuario lo prefiere:
 - Rugosidad del material ϵ .
 - Número de Reynolds en la entrada o , si el usuario lo prefiere:
 - Densidad del gas en la entrada ρ_1 .
 - Velocidad del gas en la entrada V_1 .
 - Viscosidad absoluta del gas μ .

Con esto, el programa proporciona:

- Longitud del conducto L .
- Longitud para el bloqueo por fricción L^* .
- Relaciones entre la entrada y la salida de las propiedades:
 - Presión estática.
 - Densidad estática.
 - Temperatura estática.
 - Velocidad.
 - Presión de remanso.
 - Densidad de remanso.

Se puede recordar que la temperatura de remanso, al ser un problema sin transferencia de calor, se mantiene invariable.

Presentación

El aspecto que tiene la aplicación es el siguiente:

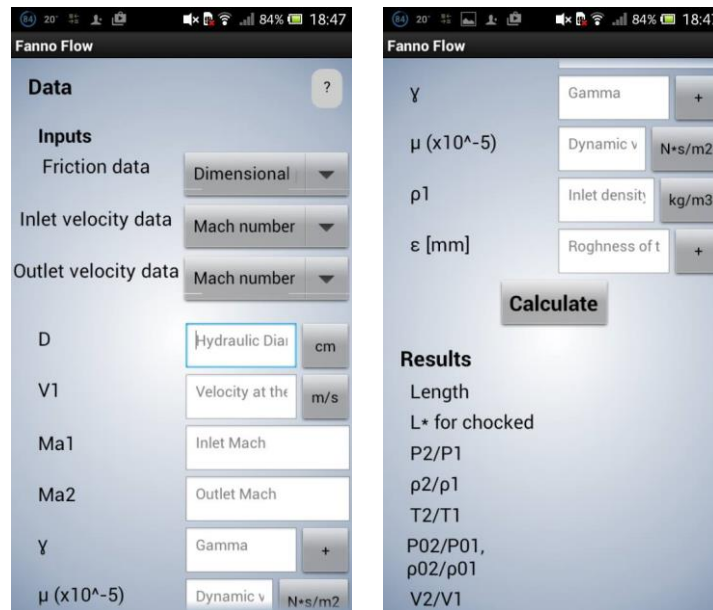
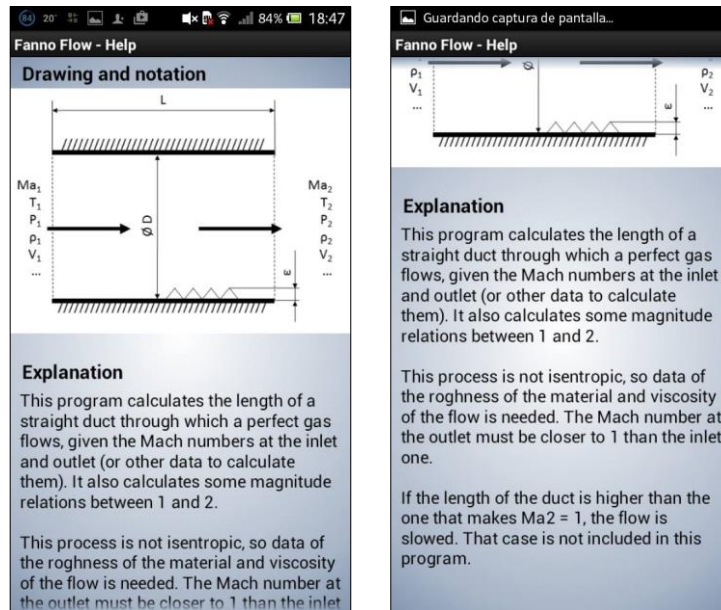


Ilustración 4.17. Presentación del flujo de Fanno

En “Friction data” el usuario puede elegir si introducir el factor f , número de Reynolds y rugosidad o magnitudes dimensionales en lugar del número de Reynolds. En las otras opciones a elegir, puede elegir entre introducir el número de Mach o la velocidad dimensional en su lugar, junto a la constante específica del gas y la temperatura para calcularlo.

Como en las funciones anteriores, el usuario puede escoger un gas desde el botón “+” de la casilla de γ , el cual rellena a su vez las casillas de r y μ . De una forma similar se trata el caso de la rugosidad ε , en donde puede elegir algunos materiales. La lista de ellos se encuentra en los anexos, en el apartado A.5.1.

Al apretar el botón “?”, se abre una nueva pantalla donde se muestra lo siguiente:



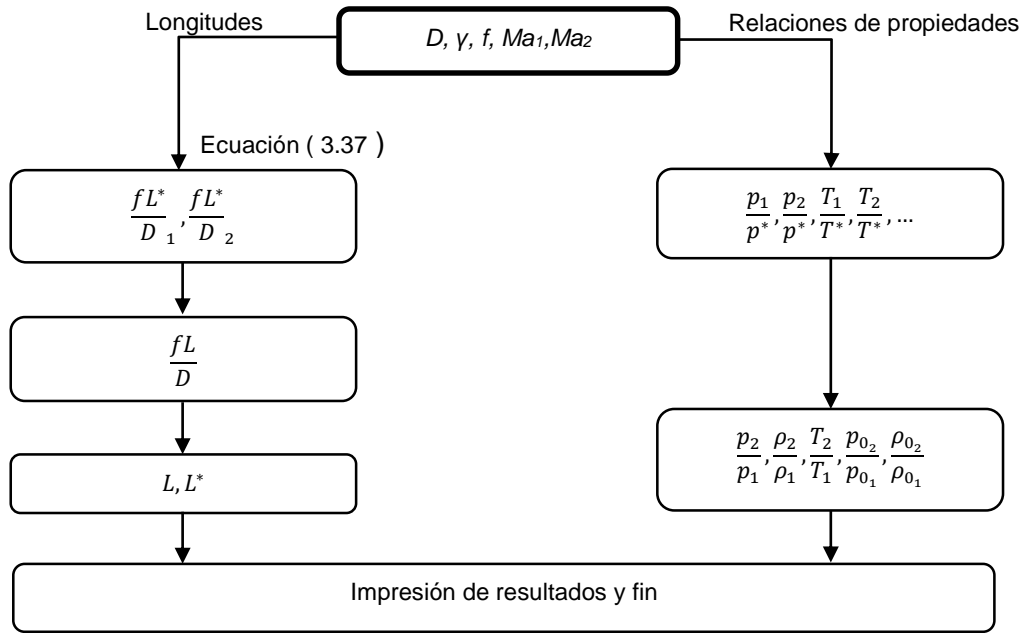
Il·lustració 4.18. Guia en el flujo de Fanno

Operaciones

A la hora de implementar la resolución del problema se han tomado consideraciones con el coeficiente de fricción de Darcy f que se deben mencionar:

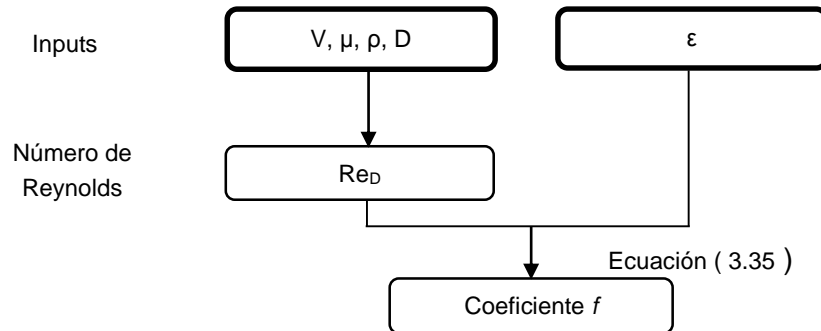
- Para obtenerlo, se utiliza la fórmula aproximada (3.35), para obtener f de forma explícita y no tener que iterar para encontrarla.
- Se usa la aproximación $f_1 = \bar{f}$. Esto significa que se aproxima que el coeficiente de fricción inicial para V_1 se mantiene constante.

El procedimiento a realizar depende ligeramente de los inputs del problema. El caso más sencillo es cuando el usuario introduce los números de Mach tanto a la entrada como a la salida, y el coeficiente de fricción medio. Partiendo de ello, se realiza el siguiente procedimiento:



En el caso de que el usuario dé la velocidad dimensional, constante r y temperatura, se añade un cálculo previo para encontrar el número de Mach.

En el caso de que el usuario dé otros inputs diferentes del factor f , el proceso para encontrarlo es el siguiente:



El código implementado para realizar esta función se encuentra en los anexos, en el apartado A.5.2.

4.7. Dificultades encontradas

Además de los problemas relacionados con las limitaciones y desventajas de App Inventor que ya fueron planteados en la explicación de la plataforma, se encontraron algunos problemas:

- Al asignar a una variable un valor de un cuadro de texto, y **éste está en blanco**, el programa da error y se cierra. Esto pasa incluso si dicha variable no se usa posteriormente. Esto resulta inconveniente al trabajar con propiedades dimensionales, cuyo uso es opcional. El problema se soluciona comprobando que el cuadro de texto no esté vacío antes de asignar un valor.
- La plataforma App Inventor **no tiene una precisión muy grande**, y da problemas en las comparaciones de valores. Por ejemplo una condición if que se ejecute cuando un valor sea menor o igual a 1, no se ejecuta cuando el valor comparado es 1. Esto pasa porque en realidad, el valor comparado es 1.00(...)1, debido a la precisión de App Inventor, y no se ejecuta la condición cuando debería hacerlo. Esto se soluciona poniendo, siguiendo el ejemplo, que la condición se ejecute cuando el valor comparado es menor o igual a 1.00001.
- Siguiendo con la precisión de la plataforma App Inventor, ésta falla cuando en una iteración **el valor de tolerancia es demasiado pequeño**. Por ejemplo, cuando se opera con presiones del orden de 10^5 y se establece una tolerancia de 10^{-12} . Evidentemente es una tolerancia ridículamente pequeña para este caso. Esto se soluciona estableciendo una tolerancia de iteración en función del orden de magnitud de los valores utilizados, como en el apartado “Operaciones” del modo genérico del apartado 4.8.4.
- El problema más complicado fue encontrado al intentar implementar una **iteración por Newton-Raphson** para encontrar el área de la onda de choque. El método divergía cuando se introducían relaciones de área superiores a 3, porque se ensayaban áreas negativas que hacían divergir el método. Al intentar limitar esto y establecer que cuando se fuese a ensayar un área negativa se utilizase el área de la garganta en su lugar, el siguiente valor de iteración era un área mucho mayor que el área de salida, que a su vez tenía un valor de iteración siguiente negativo, siendo en consecuencia el área de garganta. Así, el programa entraba en un bucle

infinito. Limitar superiormente el área a ensayar al área de salida no hizo nada por solucionar este bucle, por lo que se tuvo que cambiar el método de iteración al utilizado actualmente.

4.8. Análisis de resultados

El análisis de resultados consiste básicamente en la comprobación de los resultados obtenidos con el programa, y en caso de que no coincidan con la solución exacta, en el análisis de su precisión. Los resultados se comprueban con el programa Matlab, cuyo código se adjunta con el proyecto en los anexos, en el apartado B.

4.8.1. Atmósfera estándar internacional

En este apartado se ha implementado un código en Matlab en el que calculan los parámetros de presión, densidad y temperatura a diferentes alturas, igual que en la aplicación. La utilidad de Matlab es que puede calcular fácilmente los parámetros en una gran cantidad de alturas y hacer una gráfica de forma que quede prácticamente continua. Esto se compara con los resultados que da la aplicación, y se verifica que el código implementado coincida con la metodología teórica planeada. Después, se comparan algunos resultados con otra fuente para verificar que los resultados sean correctos.

Con el programa de Matlab los resultados coinciden hasta el último decimal que muestra la aplicación, 10^{-5} . A continuación se muestran gráficas comparativas de resultados, con los valores iniciales T_0 y P_0 por defecto:

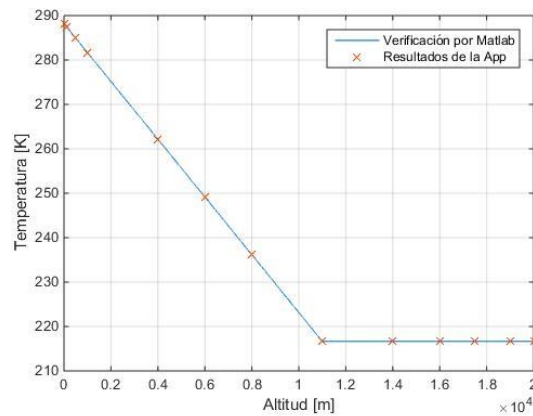


Ilustración 4.19. Gráfico de la temperatura en función de la altitud

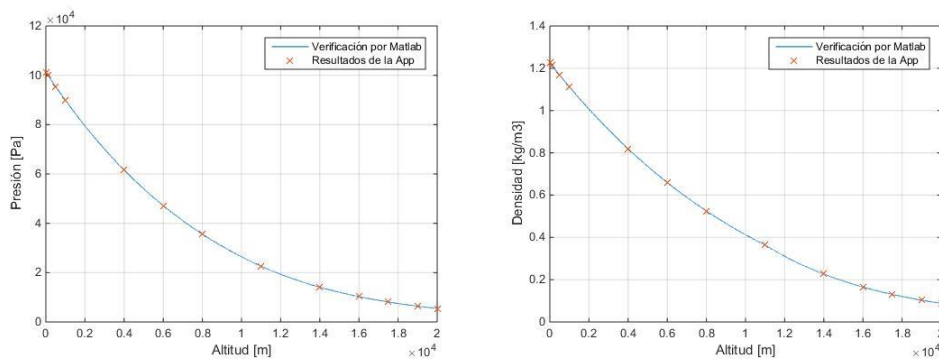


Ilustración 4.20. Gráficos de la presión y densidad en función de la altitud

A la vista de los resultados obtenidos, se puede afirmar que los resultados de ambos programas coinciden de forma muy satisfactoria.

Comparando los resultados del programa con la web <http://www.digitaldutch.com/atmoscalc/> [12] se obtiene, redondeando los números al mínimo número de decimales ofrecido por alguna de las dos fuentes y calculando su diferencia porcentual respecto a la de la App:

Altitud (m)	Temperatura de la App	Temperatura de la web	Diferencia (%)
0	288.15	288.15	0
5000	255.65	255.65	0
11000	216.65	216.65	0
15000	216.65	216.65	0
20000	216.65	216.65	0

Tabla 4.2. Comparación de los resultados de temperatura en función de la altitud.

Altitud (m)	Presión de la App	Presión de la web	Diferencia (%)
0	101325.00	101325.00	0
5000	54019.90	54019.90	0
11000	22632.06	22632.10	$1.77 \cdot 10^{-4}$
15000	12044.56	12044.60	$3.32 \cdot 10^{-4}$
20000	5474.88	5474.89	$1.83 \cdot 10^{-4}$

Tabla 4.3. Comparación de los resultados de presión en función de la altitud

Altitud (m)	Densidad de la App	Densidad de la web	Diferencia (%)
0	1.225	1.225	0
5000	0.73612	0.73612	0
11000	0.36392	0.36392	0
15000	0.19367	0.19367	0
20000	0.08803	0.08803	0

Tabla 4.4. Comparación de los resultados de densidad en función de la altitud

A la vista de los resultados obtenidos, se puede afirmar que tanto la metodología como los valores obtenidos son correctos.

4.8.2. Flujo isentrópico

De forma similar al caso anterior, se comprueba la tendencia con un código de Matlab, y luego se verifican los resultados con la web <http://www.dept.aoe.vt.edu/~devenpor/aoe3114/calc.html>, un código creado por William Devenport, de la universidad Virginia Tech [13].

La comparación con Matlab se realiza con las propiedades dimensionales, partiendo de $T_0=288.15$ K, $P_0=101325$ Pa y $\rho_0=1.225$ kg/m³ coincide de forma muy exacta con los resultados obtenidos por la App, y tiene la forma siguiente:

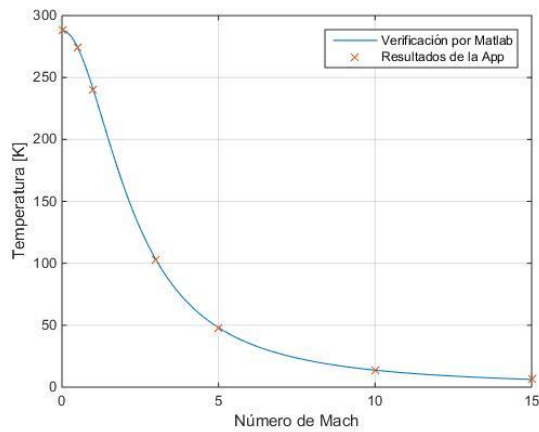


Ilustración 4.21. Gráfica de temperatura en función del número de Mach

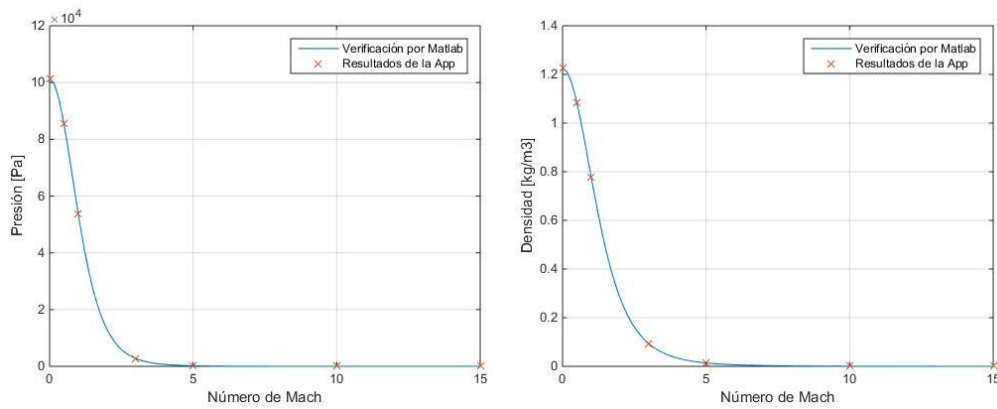


Ilustración 4.22. Gráficas de presión y densidad en función del número de Mach

La comprobación con la web mencionada se lleva a cabo con las relaciones de propiedades, ya que no ofrece propiedades dimensionales. Los resultados son los siguientes:

Mach	T/T ₀ de la App	T/T ₀ de la web	Diferencia (%)
0	1	1	0
0.5	0.95238	0.95238	0
1	0.83333	0.83333	0
5	0.16667	0.16667	0

Tabla 4.5. Comparación de relaciones de temperatura en función del número de Mach

Mach	P/P ₀ de la App	P/P ₀ de la web	Diferencia (%)
0	1	1	0
0.5	0.84302	0.84302	0
1	0.52828	0.52828	0
5	0.00189	0.00189	0

Tabla 4.6. Comparación de relaciones de presión en función del número de Mach

Mach	ρ / ρ_0 de la App	ρ / ρ_0 de la web	Diferencia (%)
0	1	1	0
0.5	0.88517	0.88517	0
1	0.63394	0.63394	0
5	0.01134	0.01134	0

Tabla 4.7. Comparación de relaciones de densidad en función del número de Mach

Respecto a la función inversa, las relaciones no cambian pero sí lo hacen las propiedades dimensionales. Se compara de nuevo con el código de Matlab:

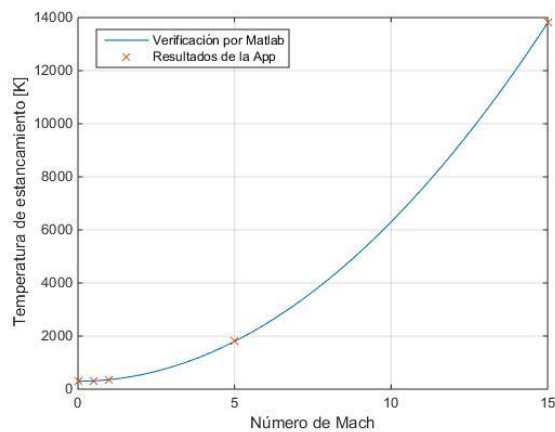


Ilustración 4.23. Gráfica de la temperatura de estancamiento en función del número de Mach

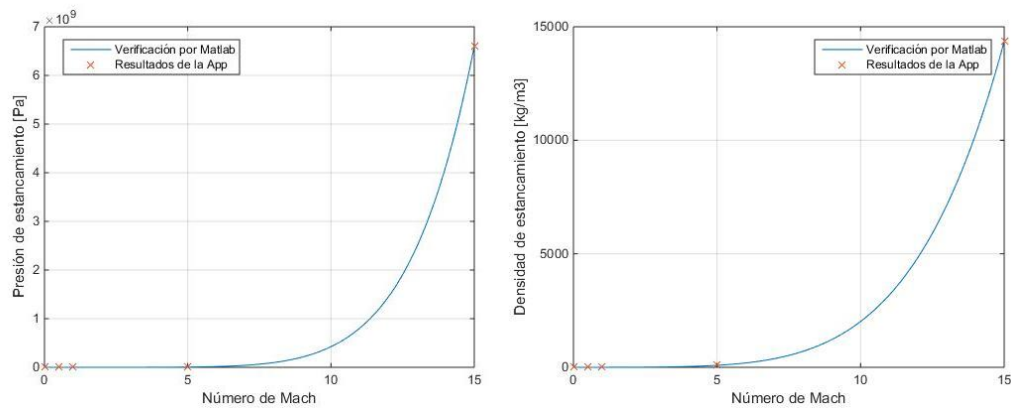


Ilustración 4.24. Gráficas de la presión y densidad de estancamiento en función del número de Mach

Los resultados de nuevo se ajustan bien al cálculo teórico, y se puede concluir que son correctos.

4.8.3. Onda de choque normal

Igual que en el apartado anterior, se verifica la tendencia correcta con un programa de Matlab, utilizando valores dimensionales. Los valores introducidos son:

- $T1 = 300\text{K}$
- $P1 = 101325\text{ Pa}$
- $P0(1) = 101325\text{ Pa}$
- $\rho1 = 1.225\text{ kg/m}^3$
- $\rho0(1) = 1.225\text{ kg/m}^3$
- $v1 = 100\text{ m/s}$
- $A^*(1) = 1\text{ m}^2$

Las gráficas comparativas son las siguientes:

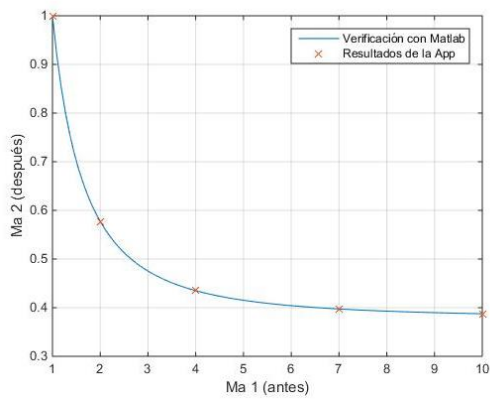


Ilustración 4.25. Gráfica del Mach después de la onda en función del anterior a la onda

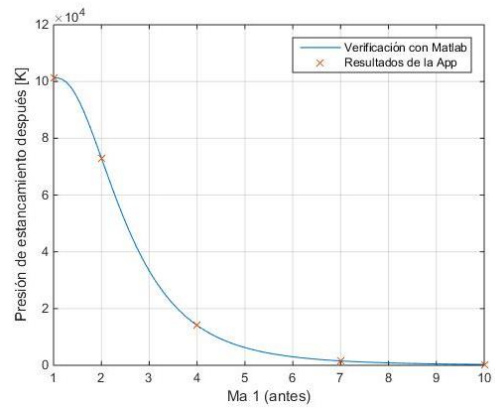


Ilustración 4.28. Gráfica de la presión de estancamiento P0 2 en función del Mach 1

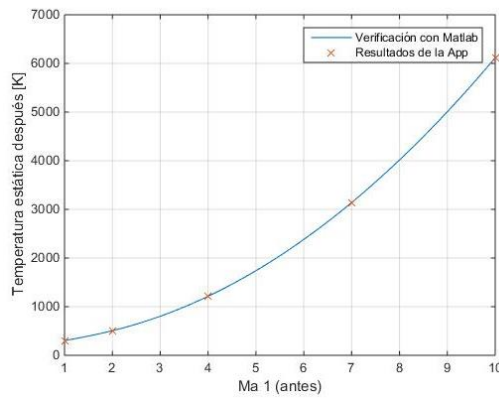


Ilustración 4.26. Gráfica de la temperatura en función del Mach 1

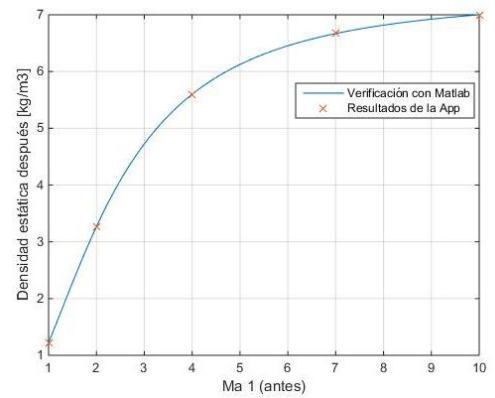


Ilustración 4.29. Gráfica de la densidad estática p2 en función del Mach 1

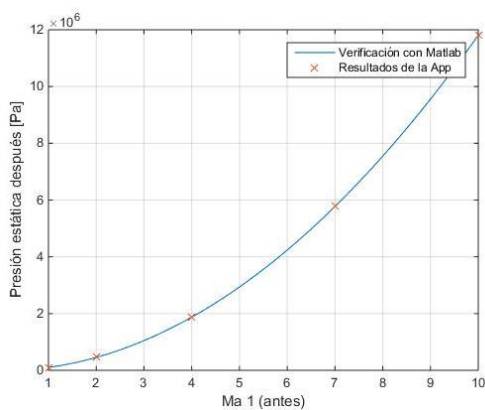


Ilustración 4.27. Gráfica de la presión estática P2 en función del Mach 1

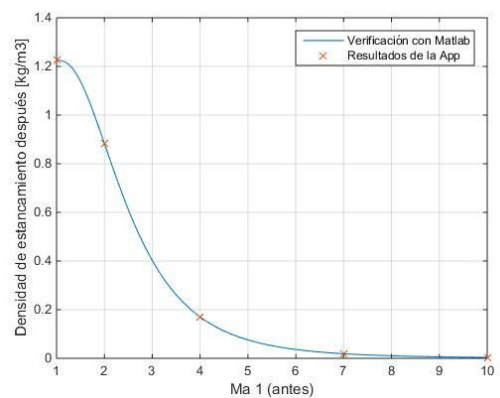


Ilustración 4.30. Gráfica de la densidad de estancamiento p0 2 en función del Mach 1

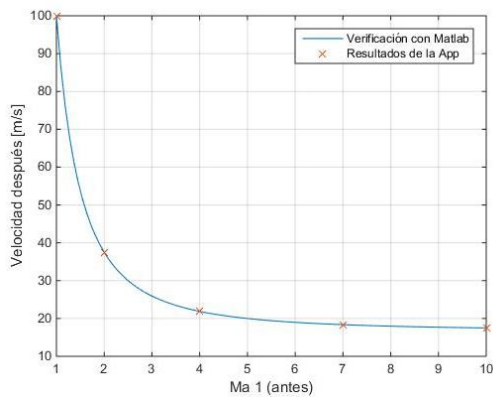


Ilustración 4.31. Gráfica de la velocidad v_2 en función del Mach 1

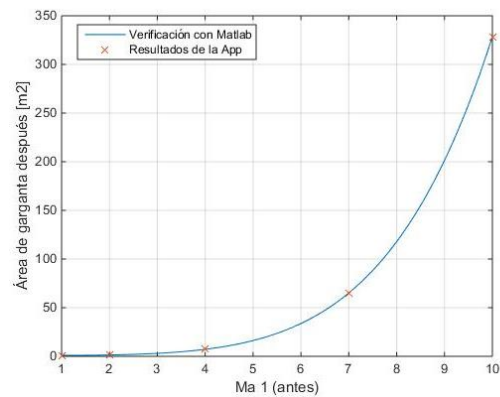


Ilustración 4.32. Gráfica del área de garganta A^*_2 en función del Mach 1

A continuación se comparan los números de forma más exacta con el libro “Mecánica de Fluidos” de Frank M. White [5], en concreto con los datos de la tabla de relaciones para un gas perfecto, onda de choque normal, apéndice B.

Los resultados coinciden hasta el cuarto decimal, que es la precisión de la tabla del libro consultado. Se puede comprobar que los resultados obtenidos son estos:

Ma 1	Ma 2	P2/P1	ρ_2/ρ_1	T2/T1	A*2/A*1
1.2	0.8422	1.5133	1.3416	1.1280	1.0073
1.8	0.6165	3.6133	2.3592	1.5316	1.2305
3	0.4752	10.3333	3.8571	2.6790	3.0456
4	0.4350	18.5	4.5714	4.0469	7.2069

Tabla 4.8. Resultados de las relaciones de propiedades en ondas de choque normales

A la vista de los resultados obtenidos, se puede concluir que la App proporciona resultados correctos.

4.8.4. Tobera convergente-divergente

Modo tobera adaptada

A este modo se le introducen relaciones de propiedades, que pueden ser presión, temperatura, áreas o número de Mach en la salida, y calcula las demás.

Siguiendo la tendencia de las verificaciones anteriores, primero se comparan los resultados con un programa de Matlab, cuyo código se encuentra en los anexos. Primero se analiza el caso de tener como input la relación de áreas, y se comparan los resultados de números de Mach, y relaciones de presión, temperatura y densidad para el caso supersónico. En los demás inputs, como el proceso para encontrar las relaciones de propiedades es idéntico y a partir del Mach, se verifica solo la obtención del número de Mach.

Relación de áreas

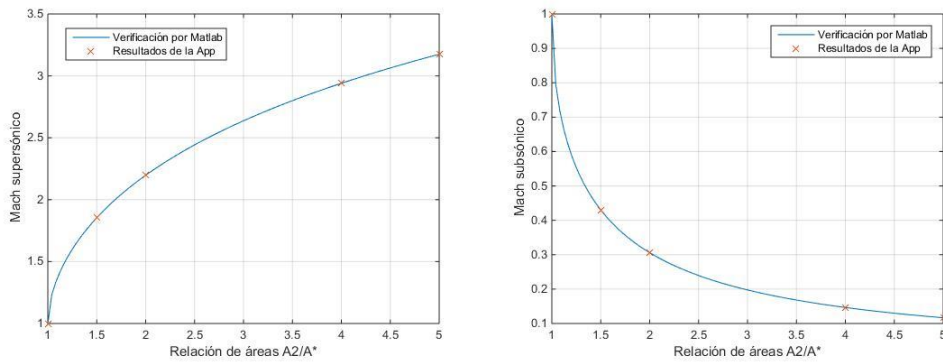


Ilustración 4.33. Números de Mach en función de la relación de áreas

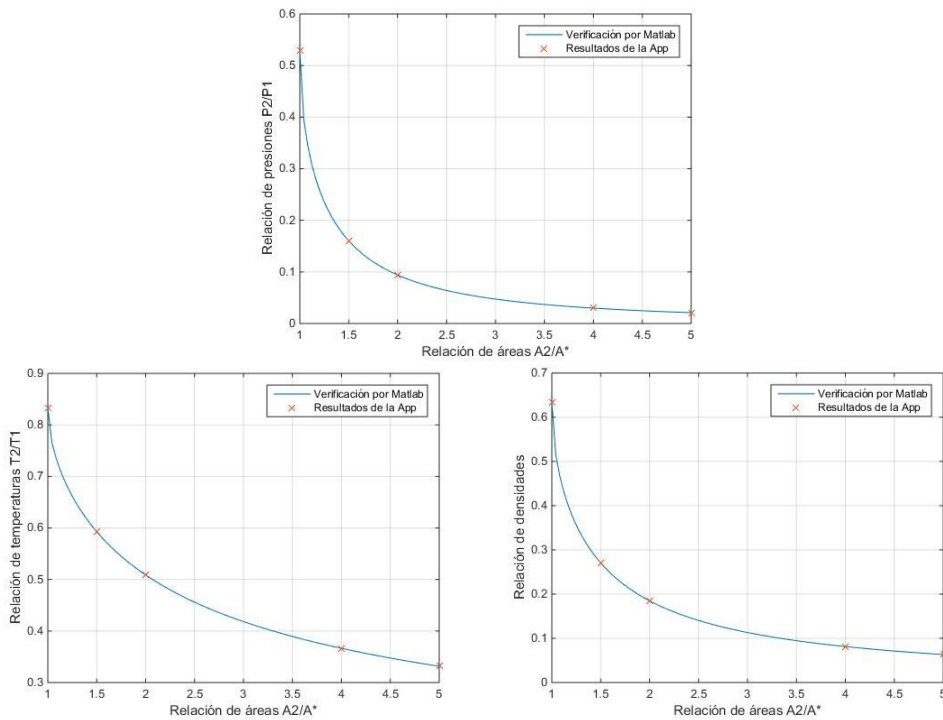


Ilustración 4.34. Gráficas comparativas de relaciones de propiedades

Comparando con una fuente externa, de nuevo con la web del apartado de flujo isentrópico [13], los resultados coinciden hasta el último decimal dado por la aplicación, siendo:

$A2/A^*$	Ma super	Ma sub	P2/P1	T2/T1	$\rho2/\rho$
1.5	1.85412	0.43026	0.16018	0.59257	0.27031
2	2.19720	0.30590	0.09393	0.50877	0.18463
5	3.17478	0.11669	0.02099	0.33158	0.06331

Tabla 4.9. Tabla comparativa del modo adaptada con la relación de áreas como input

Relaciones de presión y temperatura

Se comprueba la obtención correcta del Mach de salida a partir de las relaciones de presión y temperatura. El resto del proceso es una repetición de las funciones utilizadas con la relación de áreas como input, así que no merecen volverse a comprobar.

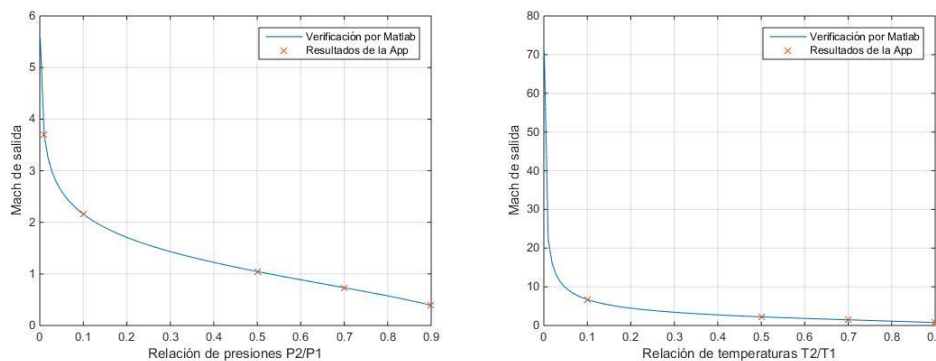


Ilustración 4.35. Gráficas comparativas del modo adaptada con inputs de relaciones de presión y temperatura

Comparando con la misma web, los resultados vuelven a coincidir hasta el último decimal dado por la App.

Modo genérico

En este apartado hay muchas cosas a evaluar, por lo que se comprueban aspectos clave. Los códigos de Matlab utilizados para estas verificaciones, fueron creados por Oriol Crespo y Joan Pallejà, estudiantes del grado de ingeniería en tecnologías aeroespaciales, además del autor del proyecto.

Primero, se evalúa el flujo másico. Evaluando el flujo másico se comprueba también la correcta determinación de los regímenes subsónico y supersónico, que también sirve para determinar la presión de salida, y se verifica que se cumpla la teoría respecto al bloqueo de la tobera. En este caso se ha comparado con un programa de Matlab por los valores de $P_{01}=100\text{kPa}$, $T_1=300\text{K}$, $A^*=1\text{ m}^2$, $A_2=2\text{ m}^2$, aire. La gráfica obtenida tiene la siguiente forma:

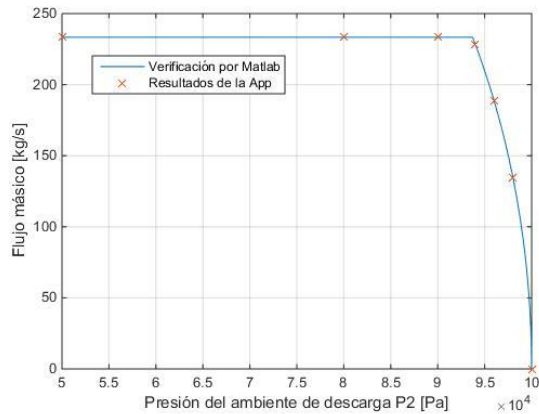


Ilustración 4.36. Gráfica comparativa del flujo másico en modo genérico

Una vez comprobado el flujo másico y por lo tanto la presión límite para que la tobera esté bloqueada, es útil comprobar las otras presiones límite, que con la presión con onda de choque en la salida y la presión de diseño, o de solución supersónica. No son resultados que dé la App, pero se puede modificar ligeramente para que dé los siguientes resultados, con una presión $P_{01}=100\text{ kPa}$ y aire:

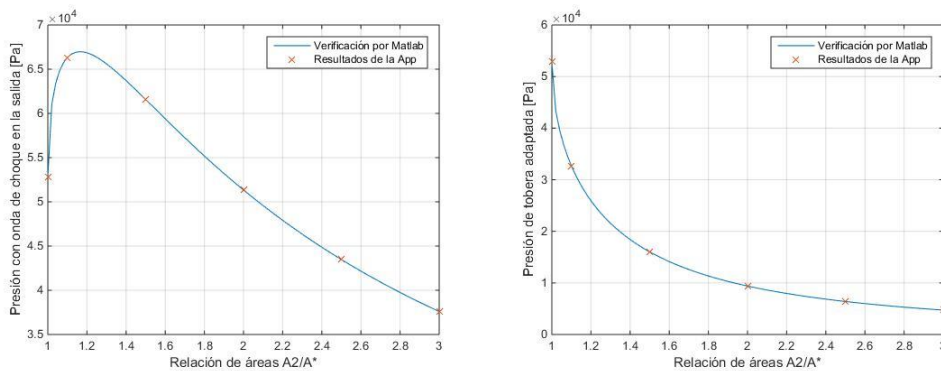


Ilustración 4.37. Gráficas comparativas de las presiones límite del modo genérico

Finalmente falta por comprobar el cálculo del área de la onda de choque. En las comprobaciones anteriores, aunque no se vean los números exactos, se obtuvo que para una relación de áreas de 2, habrá una onda de choque entre las presiones de salida $P_a=93716$ Pa y 51340 Pa. Calculando la posición de la onda de choque entre esas presiones se obtiene:

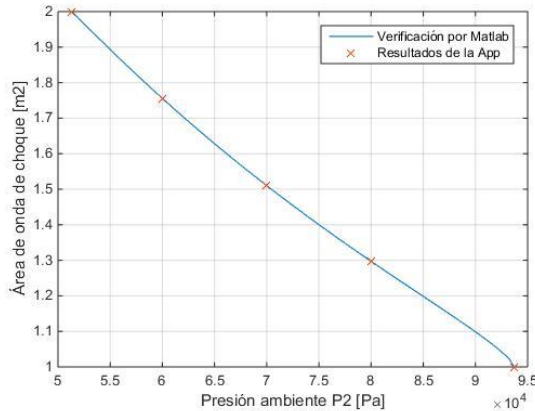


Ilustración 4.38. Gráfica comparativa del área de onda de choque en función de la presión P_2

Al ser propiedades dimensionales, es muy difícil comprobar que se cumplen los resultados con una fuente externa, ya que es raro encontrar un programa genérico como éste. Sin embargo, la comprobación de resultados se realiza con los ejemplos relacionados del libro *Mecánica de fluidos*, de Frank M. White [5], y dan correctamente.

4.8.5. Flujo de Fanno

Por último se comprueba la función de flujo de Fanno. En primer lugar se comparan los resultados de un código de Matlab y la App con el problema siguiente:

Se tiene un conducto de 2cm de diámetro y coeficiente de fricción medio 0.024, por el cual pasa aire seco. Se tiene que el Mach en la salida es 0.9, calcular la longitud y relaciones de propiedades en función del Mach de entrada.

Las gráficas comparativas son las siguientes:

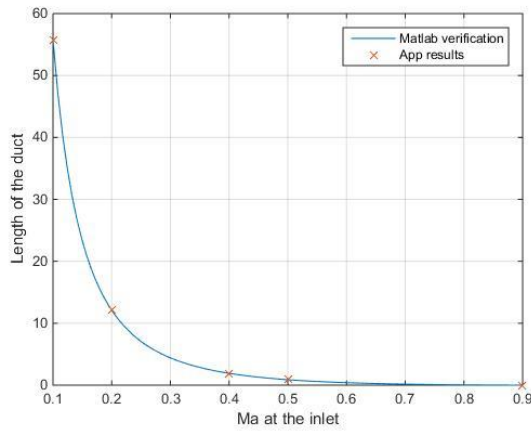


Ilustración 4.39. Gráfica comparativa de la longitud en función del Mach de entrada

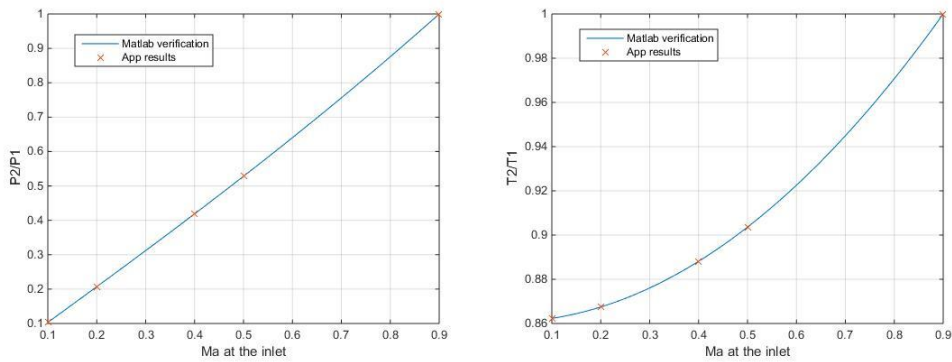


Ilustración 4.40. Gráficas comparativas de relaciones de presión y temperatura en función del Mach de entrada

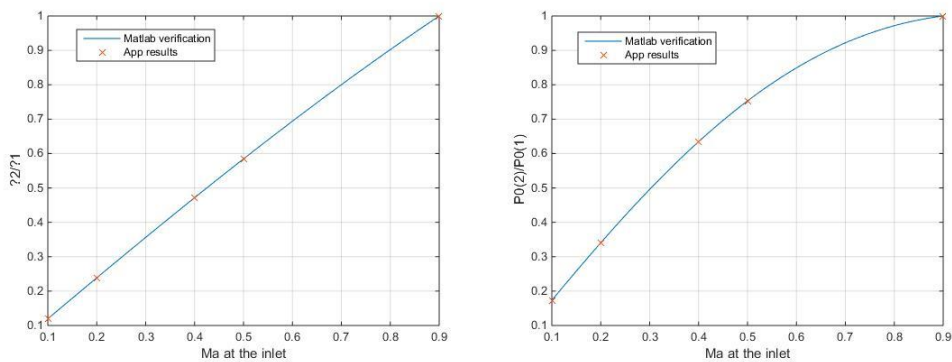


Ilustración 4.41. Gráficas comparativas de relaciones de densidad y presión de remanso en función del Mach de entrada

Los resultados obtenidos se comparan de nuevo con la web de Devenport [13]. Coinciden hasta el decimal 10^{-5} , por lo que se puede afirmar que esta función de la App da resultados correctos.

5. Conclusiones

En este proyecto se planteó un problema relacionado con el flujo compresible en el ámbito de la mecánica de fluidos, consistente en que los problemas planteados en él, si bien se reducen muchas veces a casos simples para obtener primeras aproximaciones, implican el uso de fórmulas complicadas, valores tabulados e incluso iteraciones, que hacen imposible su solución rápida. Hoy en día ese problema parece anticuado, pues los terminales móviles ofrecen una potencia de cálculo suficiente para resolver este tipo de problemas.

Es por ello que se propuso y se llevó a cabo el estudio de las soluciones para este problema, **analizando las aplicaciones** gratuitas existentes en la Play Store para sistemas operativos Android. La conclusión general que se pudo obtener de dicha investigación es que sí existen algunas aplicaciones que buscan cubrir esta necesidad, pero éstas presentan desventajas y debilidades que pueden ser superadas por una nueva aplicación a desarrollar.

A continuación, con el objetivo de programar una nueva aplicación de flujo compresible, se estudió el software gratuito disponible para crearla.

Una vez analizado el estado del arte y los posibles lenguajes de programación, se **llevó a cabo la programación de la aplicación** en la plataforma App Inventor, llamada "**App4Comp**". Ésta aplicación fue desarrollada desde cero, y tiene las siguientes funciones:

- Atmósfera estándar internacional
- Flujo isentrópico
- Onda de choque normal
- Tobera convergente-divergente
- Flujo de Fanno

Como aspecto más destacable de la App, se puede mencionar el hecho de que la App trabaja con aspectos dimensionales de las toberas convergentes-divergentes, llegando a calcular el área de la onda de choque cuando ésta se produce.

Una debilidad que se observó analizando las aplicaciones ya existentes en el mercado es que éstas resultaban poco claras respecto al problema tratado, resultando confusas para un usuario nuevo. Cubriendo este problema, se realiza una **guía básica de la App**, no como documento aparte porque no forma parte de la dinámica de cualquier aplicación móvil, sino como globos y pantallas explicativos incluidos en cada función.

App4Comp puede ser encontrada en la web <https://sites.google.com/site/app4comp> y descargada por cualquier dispositivo Android de forma gratuita.

Por último, se estudiaron los aspectos económicos, ambientales y temporales del proyecto, para determinar su viabilidad ante un posible inversor o proyectista.

Como conclusión final, se puede afirmar que este proyecto ha cumplido con sus objetivos y alcance planteado inicialmente, en el tiempo marcado para ello. Como producto más útil de éste trabajo queda la aplicación “App4Comp”, tanto su aspecto funcional como su código. Por una parte, para realizar su función inicial solucionando problemas clásicos de la mecánica de fluidos. Por la otra, servir como precedente y guía para nuevos programadores de App Inventor, que podrán consultar su código libremente.

5.1. Ampliaciones y trabajo futuro

Si bien la App creada cumple con los objetivos planteados inicialmente, no tiene por qué estar finalizada. Hay diversos aspectos que pueden ser mejorados o añadidos, tanto de programación como nuevas funciones de mecánica de fluidos. Éstos pueden ser, principalmente:

- **Mejorar la iteración** por Newton-Raphson que obtiene el número de Mach a partir de la relación de áreas, por la ecuación (4.7), planteado en el apartado 4.5.1, para que obtenga ambas soluciones con relaciones de área superiores a 500.
- Añadir una función de **ondas de choque oblicuas**.
- **Ampliar** la función de **flujo de Fanno** para casos de conducto bloqueado.

-
- Mejorar la **presentación y dinámica global** de la App, para hacerla más fácil de usar y mejor presentada.
 - Solucionar cualquier fallo, imprecisión o “bug” (fallo en la programación) que se pueda encontrar durante su uso.

En cuanto a la investigación del mercado de aplicaciones relacionadas, se debe mantener al día de las nuevas actualizaciones y Apps que puedan surgir, así como adaptar la aplicación a las nuevas versiones Android que surgirán en el futuro.

6. Condiciones de software

Para la realización de este proyecto se han utilizado los siguientes programas:

- **App Inventor 2:** programación de toda la App y comprobación del código en tiempo real.
- **Matlab R2014b:** desarrollo de los códigos de verificación de resultados y comparación de lenguajes de programación.
- **Microsoft Word 2013:** redacción de los documentos del proyecto.

6.1. Licencia App Inventor 2

Para llevar a cabo la programación entera de la aplicación móvil se ha utilizado la plataforma App Inventor 2 para ordenador. La prueba del código en tiempo real con esta aplicación se puede hacer con un dispositivo Android por red wifi, cable USB o por emulador en el ordenador. Cada opción tiene sus requisitos, y en este proyecto se ha utilizado la opción recomendada, un dispositivo Android por conexión wifi.

6.1.1. Requisitos para ordenador

Para utilizar la plataforma App Inventor 2 no hacen falta grandes requisitos, ya que se utiliza en un navegador web y es gratuita:

- Navegador Mozilla Firefox o Google Chrome, en versión compatible con la última de App Inventor.
- Cuenta Gmail de usuario, gratuita. Sin embargo, habilitarla para poder subir Apps al mercado cuesta 22.75 €.

6.1.2. Requisitos para dispositivo Android

A su vez, el dispositivo Android tampoco necesita grandes requisitos:

- Sistema operativo Android de versión 4.0 o posterior.
- Conexión a red wifi.
- 2.66 MB de espacio libre.

6.2. Licencia Matlab

Para la realización del proyecto se ha utilizado el programa Matlab, de MathWorks, en la versión R2014b. Dicho programa tiene un coste en su licencia que debe ser incluido en los aspectos económicos del proyecto, y unos requisitos previos para el ordenador en el que se usa.

6.2.1. Requisitos

La versión disponible en el mercado es la R2015b, que tiene los siguientes requisitos previos:

- Sistema operativo compatible. En Windows, tienen versiones compatibles desde la versión XP.
- Procesador Intel o AMD x86 que soporte el set de instrucciones SSE2.
- 1 GB de espacio libre para la instalación única de Matlab, 3-4 GB libres para una instalación típica.
- 2 GB de RAM.

El coste de la licencia, para el programa Matlab en sí en la versión “Home”, es de 105€. Se escoge la versión “Home” dado el carácter laboral del proyecto, más que educacional, y que su trabajo es para la comprobación de resultados y no una publicación oficial.

6.3. Licencia Microsoft Word

El programa Microsoft Word se adquiere dentro del paquete de Microsoft Office 2013, que tiene ciertos requisitos y un coste económico.

6.3.1. Requisitos

La licencia de Microsoft Office 365, válida por un año, tiene los siguientes requisitos:

- Procesador a 1 gigahercios (GHz) o más rápido, x86 o x64 bits con conjunto de instrucciones SSE2.
- Sistema operativo compatible.
- 3.0 GB de espacio disponibles.

El conjunto tiene un coste de 69€.

7. Planificación

Para llevar a cabo este proyecto se realizó una planificación temporal, entregada con la carta de proyecto ("Project charter"). Sin embargo, la realización final resultó ser diferente de la planificación inicial. En la siguiente página se muestra el diagrama de Gantt del trabajo realizado. Después del último día de defensa del trabajo final de grado, el 3 de julio de 2015, se planifica el trabajo propuesto en el apartado de conclusiones. Este trabajo se considera que necesita, aproximadamente:

- **Onda de choque oblicua:** 20 horas, 4 días.
- **Ampliación del flujo de Fanno:** 20 horas, 4 días.
- **Mejorar iteración Newton-Raphson:** 10 horas, 2 días.
- **Solución de bugs y aspectos formales:** 10 horas, 2 días.

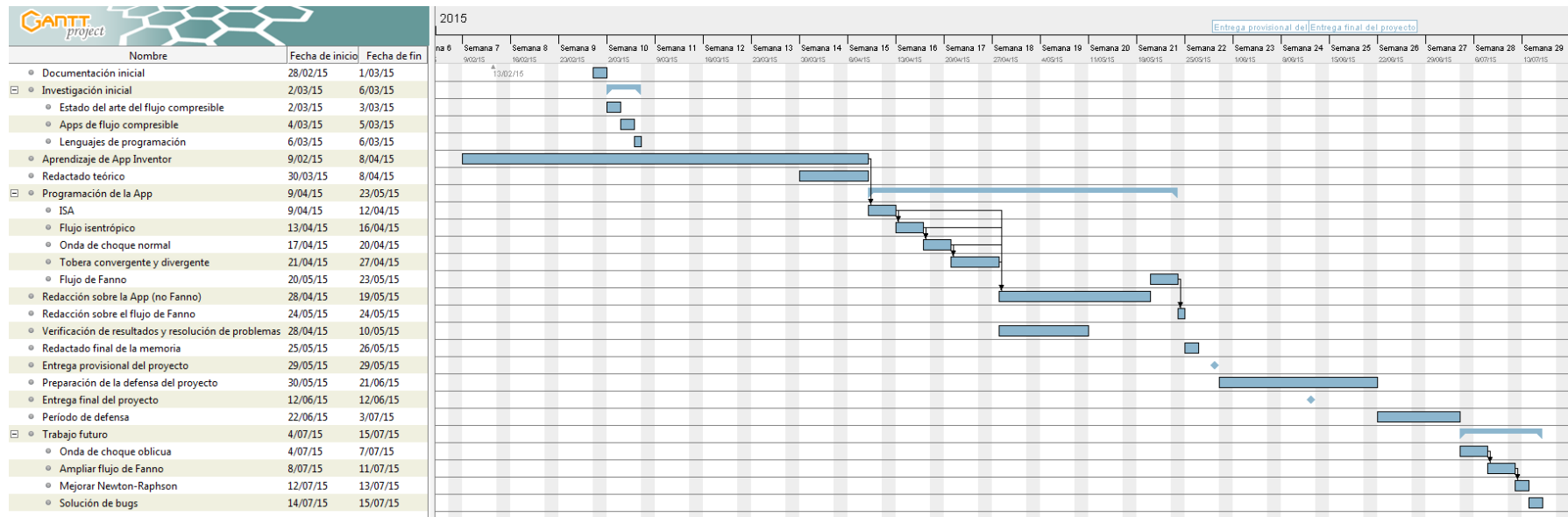


Ilustración 7.1. Diagrama de Gantt del proyecto.

8. Análisis medioambiental

8.1. Referente al desarrollo del proyecto

El desarrollo de este proyecto no ha supuesto un gran impacto ambiental, con ninguna actividad que repercuta directamente en el medio ambiente. Se pueden considerar como impacto indirecto el consumo de papel y el gasto energético, que comporta una emisión de CO₂ y contribuye al calentamiento global. Para minimizar este impacto ambiental, y de paso reducir el coste económico, se tomaron las siguientes consideraciones durante el proyecto:

- El papel utilizado para anotaciones y cálculos se aprovechaba de hojas reutilizadas.
- Se evita imprimir en papel hasta la última entrega, definitiva.
- Se trabaja en un ordenador portátil en modo economizador, reduciendo su consumo.

En total, el coste en papel de este proyecto es de no más de 200 hojas de papel, muchas menos si se realiza impresión en doble cara. A su vez, el gasto energético del ordenador portátil es de 31 kWh que, según Endesa, corresponden a **7.81 kg de CO₂** a 0.252 kg CO₂/kWh [14].

8.2. Referente a la aplicación del proyecto

La aplicación del proyecto no tiene ningún impacto directo sobre el medio ambiente. Se podría considerar un impacto indirecto el gasto energético en los teléfonos móviles. Sin embargo, el consumo de un teléfono móvil es muy bajo, el público objetivo es reducido y su uso puntual, por lo que las emisiones de CO₂ resultantes por la App "App4Comp" resultan despreciables.

Bibliografía

- [1] «Androideity.» [En línea]. Disponible en: <http://androideity.com/2012/07/16/5-lenguajes-para-programar-en-android/>. [Accedido: 16-mar-2015].
- [2] J. D. Anderson, *Modern Compressible Flow*, 3ª ed. McGraw-Hill, 2003.
- [3] «Planetary Science Research Discoveries.» [En línea]. Disponible en: http://www.psrdr.hawaii.edu/Oct05/chondrules_impacts.html. [Accedido: 16-mar-2015].
- [4] W. Devenport, R. Kapania, R. Kamal, y K. Singh, «Nozzle Applets.» [En línea]. Disponible en: <http://www.engapplets.vt.edu/fluids/CDnozzle/cdinfo.html>. [Accedido: 28-mar-2015].
- [5] F. M. White, *Mecánica de Fluidos*, 5ª ed. Madrid: McGraw-Hill, 2004.
- [6] MIT App Inventor, «MIT App Inventor.» [En línea]. Disponible en: <http://appinventor.mit.edu/explore/>. [Accedido: 26-may-2015].
- [7] MIT App Inventor, «MIT App Inventor 2 Beta.» [En línea]. Disponible en: <http://ai2.appinventor.mit.edu/>. [Accedido: 26-may-2015].
- [8] R. Reagan, «Exploring binary», 2014. [En línea]. Disponible en: <http://www.exploringbinary.com/numbers-in-app-inventor-are-stored-as-floating-point/>. [Accedido: 09-abr-2015].
- [9] ISO, «Standard Atmosphere», en *ISO 2533*, 1975.
- [10] Z. Etier, «Compressible Flow Tool.» 2012.
- [11] J. Kaddouri, «GDCalc.» 2013.
- [12] Digital Dutch, «1976 Standard Atmosphere Calculator.» [En línea]. Disponible en: <http://www.digitaldutch.com/atmoscalc/>. [Accedido: 11-may-2015].
- [13] W. Devenport, «Compressible Aerodynamics Calculator.» [En línea]. Disponible en: <http://www.dept.aoe.vt.edu/~devenpor/aoe3114/calc.html>. [Accedido: 11-may-2015].
- [14] Endesa Educa, «Presentación: La emisión de CO2 en la generación y el hogar», 2009. [En línea]. Disponible en:

<http://es.slideshare.net/EndesaEduca/endesa-educa-co2centralesyhogar>.
[Accedido: 25-may-2015].

- [15] J. A. Dean, *Lange's Handbook of Chemistry*, 15^a ed. McGraw-Hill, 1999.
(En anexos)