



UNIVERSIDAD POLITÉCNICA DE MADRID

**Escuela Técnica Superior
de Ingenieros Navales**

MADRID

TRABAJO DE FIN DE GRADO EN
ARQUITECTURA NAVAL

GAN - 348

Implementación de modelos semi-empíricos de predicción de resistencia al
avance dentro del software libre FreeCAD.



Autor:

Juan Manuel Muñoz Godin

Tutor:

Antonio Souto Iglesias

Co-tutor:

José Luis Cercós Pita

Junio 2023



*A mi familia y amigos,
por apoyarme siempre.*



Especificaciones

A continuación, se adjunta la propuesta de TFG aprobada por la COA en fecha 1 de marzo de 2023.



PROPUESTA DE TRABAJO FIN DE GRADO ARQUITECTURA NAVAL



D./D^a profesor/a adscrito/a al departamento de la E.T.S. de Ingenieros Navales de la U.P.M. propone el siguiente TRABAJO FIN DE GRADO para alumnos/as del título de Graduado/a en Arquitectura Naval, para su evaluación por la C.O.A. y posterior aprobación, si procede.

Cotutor/a D./D^a Empresa

Correo electrónico profesional Cotutor/a

Avalista D./D^a (en caso de ser necesario)

Título del trabajo propuesto:

Descripción del TRABAJO: (Indíquese seguidamente si esta información se encuentra en documento anexo claramente indentificado)

Objetivo:

El objetivo del presente Trabajo Fin de Grado es la implementación de modelos semi-empíricos de predicción de potencia en el software libre FreeCAD. Se implementarán el método de Holtrop y el método de Amadeo utilizando Python como lenguaje de programación en soporte Windows. El resultado más importante del trabajo será un módulo externo en el software libre FreeCAD capaz de calcular la predicción de la resistencia al avance para distintos tipos de barcos. Se realizará una validación de los cálculos mediante la comparación de los resultados con la implementación del método de Holtrop en el módulo Resistance de Maxsurf.

Método y fases del trabajo:

El trabajo se dividirá en las siguientes etapas:

- 1.- Estado del arte de los modelos semi-empíricos y software asociado. Selección de métodos a implementar, Holtrop y Amadeo. Identificación de inputs para los métodos a implementar.
- 2.- El software FreeCAD.
- 3.- Elementos necesarios para la implementación de código libre en FreeCAD y su instalación: ecosistema FreeCAD; python; repositorio; Qt.
- 4.- Creación de una nueva plantilla de herramienta en la GUI de FreeCAD.
- 5.- Implementación a partir de datos sin geometría.
- 6.- Comparación con Maxsurf Resistance.
- 7.- Obtención de dimensiones principales y ratios desde FreeCAD.
- 8.- Implementación a partir de datos desde geometría.
- 9.- Postprocesado en FreeCAD.
- 10.- Conclusiones y definición de futuras líneas de trabajo.



PROPUESTA DE TRABAJO FIN DE GRADO
ARQUITECTURA NAVAL



Medios específicos necesarios:

Overleaf, Github, docker, QT, Python, FreeCAD.

Estructura de capítulos, planos o documentos que constituyen el cuerpo del trabajo:

Los que se deducen de las fases de trabajo más los apéndices que sean necesarios.

Propuesta de Tribunal, si lo considera el Tutor/a:

Presidente:

Vocal:

Secretario:

Asignación propuesta (táchese lo que no proceda):

1. Para el/la alumno/a D/D^a Juan Manuel Muñoz Godin
2. Libre

Firmado: Madrid, a 21 de febrero de 2022

Tutor/a

Cotutor/a

Alumno/a

Avalista

Firmado por SOUTO IGLESIAS
ANTONIO - DNI 52493731A el día
21/02/2023 con un certificado
emitido por AC Sector Público

MUÑOZ
GODIN JUAN
MANUEL -
50362437G

Firmado
digitalmente por
MUÑOZ GODIN
JUAN MANUEL -
50362437G
Fecha: 2023.02.21
13:54:30 +01'00'



PROPUESTA DE TRABAJO FIN DE GRADO ARQUITECTURA NAVAL



El **Trabajo Fin de Grado** permite adquirir las competencias necesarias para la redacción y desarrollo de proyectos de **Arquitectura Naval** (Disposición general, formas de carena, cálculo de hidrostáticas, estabilidad, estructura del buque, habilitación de espacios, etc.) que forman parte de las actividades de construcción, montaje, transformación, explotación, mantenimiento, reparación, o desguace de buques y embarcaciones. Permite además formación en el manejo de especificaciones, reglamentos y normas de obligado cumplimiento de su área de actividad y capacidad de analizar y valorar el impacto social y ambiental de las soluciones técnicas.

Competencias sobre las que trata el TFG:

A) C Generales

CG 2.

Que los estudiantes lleguen a saber aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CG 3.

Que los estudiantes alcancen la capacidad de reunir e interpretar datos relevantes (preferentemente dentro del área de la Arquitectura Naval) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o técnica

CG 4.

Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CG 5.

Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

(De las siguientes competencias, resaltar las que considere adecuadas para trabajar en el TFG. Las marcadas en rojo y con la palabra NO, no son evaluables en este Grado IM)

B) C Específicas

Nº de la competencia específica	Competencia	En su caso, para las profesiones reguladas ¿está recogida entre las que se regulan para el acceso a la profesión? (SI / NO)	Nivel de competencia que se alcanzará	¿El Trabajo Fin de Grado permitirá desarrollar esta competencia? (SI / NO)
CE 1	Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal; geometría; geometría diferencial; cálculo diferencial e integral; ecuaciones diferenciales y en derivadas parciales; métodos numéricos; algorítmicos numéricos; estadísticos y optimización.	SI	3	NO
CE 2	Comprensión y dominio de los conceptos básicos sobre las leyes generales de la mecánica, termodinámica, campos y ondas y electromagnetismo y su aplicación para la resolución de problemas propios de la ingeniería	SI	3	NO
CE 3	Conocimientos básicos de la química general, química orgánica e inorgánica y sus aplicaciones en la ingeniería.	SI	3	NO



PROPUESTA DE TRABAJO FIN DE GRADO ARQUITECTURA NAVAL



Nº de la competencia específica	Competencia	En su caso, para las profesiones reguladas ¿está recogida entre las que se regulan para el acceso a la profesión? (SI / NO)	Nivel de competencia que se alcanzará	¿El Trabajo Fin de Grado permitirá desarrollar esta competencia? (SI / NO)
CE 4	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.	SI	3	SI
CE 5	Capacidad de visión espacial y conocimiento de las técnicas de representación gráfica, tanto por métodos tradicionales de geometría métrica y geometría descriptiva, como mediante las aplicaciones de diseño asistido por ordenador.	SI	3	SI
CE 6	Conocimiento adecuado del concepto de empresa, marco institucional y jurídico de la empresa. Organización y gestión de empresas.	SI	3	NO
CE 7	Conocimiento de los conceptos fundamentales de la mecánica de fluidos y de su aplicación a las carenas de buques y artefactos, y a las máquinas, equipos y sistemas navales	SI	3	NO
CE 8	Conocimiento de la ciencia y tecnología de materiales y capacidad para su selección y para la evaluación de su comportamiento.	SI	3	NO
CE 9	Conocimiento de la teoría de circuitos y de las características de las máquinas eléctricas y capacidad para realizar cálculos de sistemas en los que intervengan dichos elementos.	SI	3	NO
CE 10	Conocimiento de la teoría de automatismos y métodos de control y de su aplicación a bordo.	SI	3	NO
CE 11	Conocimiento de las características de los componentes y sistemas electrónicos y de su aplicación a bordo	SI	3	NO
CE 12	Conocimiento de la elasticidad y resistencia de materiales y capacidad para realizar cálculos de elementos sometidos a sollicitaciones diversas	SI	3	NO
CE 13	Conocimiento de la mecánica y de los componentes de máquinas	SI	3	NO
CE 14	Conocimiento de la termodinámica aplicada y de la transmisión del calor	SI	3	NO
CE 15	Conocimiento de las características de los sistemas de propulsión naval	SI	2	NO
CE 16	Capacidad para la realización del cálculo y control de vibraciones y ruidos a bordo de buques y artefactos	SI	3	NO
CE 17	Conocimiento de los sistemas para evaluación de la calidad, y de la normativa y medios relativos a la seguridad y protección ambiental	SI	3	NO



PROPUESTA DE TRABAJO FIN DE GRADO ARQUITECTURA NAVAL



Nº de la competencia específica	Competencia	En su caso, para las profesiones reguladas ¿está recogida entre las que se regulan para el acceso a la profesión? (SI / NO)	Nivel de competencia que se alcanzará	¿El Trabajo Fin de Grado permitirá desarrollar esta competencia? (SI / NO)
CE 18	Capacidad para la realización de cálculos de geometría de buques y artefactos, flotabilidad y estabilidad	SI	3	SI
CE 19	Conocimiento de la hidrodinámica naval aplicada	SI	3	SI
CE 20	Conocimiento de las características de los materiales estructurales navales y de los criterios para su selección	SI	2	NO
CE 21	Conocimiento de los procedimientos y sistemas que se emplean para el control de la corrosión marina	SI	3	NO
CE 22	Capacidad para el diseño y cálculo de estructuras navales	SI	3	NO
CE 23	Capacidad para el diseño y cálculo de los espacios habitables de los buques y artefactos marinos, y de los servicios que se disponen en dichos espacios	SI	3	NO
CE 24	Capacidad para la integración a bordo de los sistemas propulsores, teniendo en cuenta su empacho, peso, cargas dinámicas, impacto en la estanqueidad, el espacio necesario para su mantenimiento, etc.	SI	2	NO
CE 25	Capacidad para la integración a bordo de los sistemas auxiliares teniendo en cuenta su empacho, peso, cargas dinámicas, impacto en la estanqueidad, el espacio necesario para su mantenimiento, etc.	SI	3	NO
CE 26	Capacidad para la integración a bordo de los sistemas eléctricos teniendo en cuenta su empacho, peso, cargas dinámicas, impacto en la estanqueidad, el espacio necesario para su mantenimiento, etc.	SI	3	NO
CE 27	Capacidad para la integración a bordo de los sistemas electrónicos de control y de navegación, teniendo en cuenta su empacho, peso, impacto en la estanqueidad, el espacio necesario para su mantenimiento, etc.	SI	3	NO
CE 28	Conocimiento de los métodos de proyecto de Arquitectura Naval.	SI	3	NO
CE 29	Conocimiento de los procesos de construcción naval	SI	3	NO
CE 30	Conocimiento de los fundamentos del tráfico marítimo para su aplicación a la distribución de los espacios del buque.	SI	2	NO



PROPUESTA DE TRABAJO FIN DE GRADO
ARQUITECTURA NAVAL



CE 31	Capacidad para integrar las competencias anteriores en el proyecto, la construcción y la reparación de las Estructuras Marinas.	SI	3	NO
-------	---	----	---	----

C) C Transversales

Nº de la competencia transversal	Competencia	Nivel que se alcanzará	Competencia en el trabajo fin de grado
CT UPM 1	Uso de la lengua inglesa	2	NO
CT UPM 2	Trabajo en equipo	3	NO
CT UPM 3	Comunicación oral y escrita	3	SI
CT UPM 4	Uso de las TIC	3	SI
CT UPM 5	Creatividad	2	SI
CT UPM 6	Liderazgo de equipos	2	NO
CT UPM 7	Organización y planificación	2	SI
CT UPM 8	Respeto ambiental	3	NO



Resumen

En los últimos años, con el auge de la tecnología, el uso de software en el sector naval ha experimentado un crecimiento exponencial. En consecuencia, han aumentado las herramientas con capacidad de realizar predicciones de resistencia al avance en buques, facilitando y agilizando dicha labor. Sin embargo, el software actualmente disponible en el mercado es privado, implicando que el usuario desconozca el código de funcionamiento y deba asumir un coste económico elevado. Por ello, en el presente Trabajo Fin de Grado se pretende implementar en software libre varios modelos semi-empíricos de predicción de resistencia al avance.

En este caso, se escoge *FreeCAD* como software base para introducir las nuevas funcionalidades, debido a que ya existe un módulo externo que cuenta con la capacidad de realizar numerosos cálculos de arquitectura naval. En primer lugar, se detalla el ecosistema de software y configuración inicial necesaria.

A continuación, se explica la creación de una nueva plantilla de herramienta, lo que constituye la primera etapa. En ella se incluyen todos los trabajos necesarios, entre los que se pueden destacar la creación de la interfaz gráfica de usuario o la programación del método semi-empírico.

Posteriormente, se expone la creación e implementación de dos variantes para calcular la predicción de resistencia al avance. En la primera variante se crean y verifican funciones para obtener las dimensiones y características del buque a través de un modelo en tres dimensiones. En la segunda variante, por el contrario, el usuario introduce manualmente los valores pertinentes.

La siguiente etapa se refiere al postprocesado de los datos, en ella se incluyen los trabajos a realizar para la presentación de los resultados obtenidos. En este caso se basa en tres gráficas y una hoja de cálculo que recoge todos los resultados.

Por último, para verificar las funcionalidades introducidas, se comparan los valores obtenidos, tanto en *FreeCAD* como en *MAXSURF*, de predicciones de resistencia al avance en diversos buques.

Finalmente, tras realizar las acciones pertinentes, se integran las nuevas funcionalidades en el código de *FreeCAD Ship*, pudiendo acceder a ellas desde el módulo externo Ship, tras descargar *FreeCAD* en: "https://www.freecad.org/downloads.php?lang=es_ES". De esta manera, se introduce en el mercado una solución de software libre, disponible para todos los sistemas operativos y de calidad, al alcance de todos los usuarios.



Abstract

In recent years, with the rise of technology, the use of software in the naval sector has experienced exponential growth. As a result, the number of tools capable of predicting the forward resistance in ships has increased, facilitating and speeding up this task. However, the software currently available on the market is proprietary, meaning that the user does not know the operating code and must assume a high economic cost. Therefore, this project aims to implement in open source software several semi-empirical methods for the prediction of forward resistance.

In this case, FreeCAD is chosen as the base software to introduce the new functionalities, since there is already an external module that has the capacity to perform numerous naval architecture calculations. First, the software ecosystem and initial configuration required is detailed.

Next, the creation of a new tool template is explained, which is the first step. This includes all the necessary work, including the creation of the graphical user interface or the programming of the semi-empirical method, among other works.

Subsequently, the creation and implementation of two variants for calculating the forward resistance prediction is presented. In the first variant, functions are created and verified to obtain the dimensions and characteristics of the ship through a three-dimensional model. In the second variant, on the other hand, the user manually enters the relevant values.

The next stage concerns the post-processing of the data, which includes the work to be carried out for the presentation of the results obtained. In this case it is based on three graphs and a spreadsheet that collects all the results.

Finally, to verify the introduced functionalities, the values obtained, both in FreeCAD and in MAX-SURF, of forward resistance predictions in different vessels are compared.

Finally, after performing the pertinent actions, the new functionalities are integrated into the FreeCAD Ship code, and can be accessed from the external Ship module after downloading FreeCAD in: "https://www.freecad.org/downloads.php?lang=es_ES". In this way, a free software solution is introduced in the market, available for all operating systems and of high quality, within the reach of all users.



Agradecimientos

En este trabajo quiero expresar mi agradecimiento a la Escuela Técnica Superior de Ingenieros Navales (ETSIN) de la Universidad Politécnica de Madrid por la formación académica recibida a lo largo de estos años. Además, agradecer a todos los profesores que han formado parte de esta bonita etapa.

Agradecer a mi familia y amigos, por haberme apoyado durante estos últimos cuatro años, pero sobretodo, por haber estado en los momentos más difíciles, os quiero.

Por último, quiero dar las gracias, en especial, a mi tutor de TFG, Antonio Souto, por la cercanía y confianza mostrada en mí a lo largo de todo el proyecto, y a mi co-tutor, José Luis Cercós, por haberme guiado y enseñado los fundamentos necesarios para trabajar con *FreeCAD*. Muchas gracias por vuestro tiempo y dedicación.





Índice general

Resumen	II
Resumen	X
Abstract	XII
Agradecimientos	XIV
1. Introducción.	3
1.1. Motivación.	3
1.2. Estado del arte.	3
1.3. Selección de métodos a implementar.	5
1.4. Software asociado.	6
1.4.1. NavCad - HydroComp.	7
1.4.2. MAXSURF Resistance.	7
2. FreeCAD.	9
2.1. Introducción.	9
2.2. FreeCAD Ship.	9
3. Ecosistema de software necesario.	11
3.1. Repositorio.	11
3.2. Entorno <i>FreeCAD</i>	12
3.3. Python.	13
3.4. QT.	14
4. Creación de una nueva plantilla de herramienta.	15
4.1. Introducción.	15
4.2. Pasos previos.	16
4.3. Icono.	16



4.4. Interfaz gráfica de usuario.	17
4.5. Creación del menú y barra de herramientas.	21
4.6. Integración de la interfaz gráfica de usuario.	22
4.7. Programación del método semiempírico.	25
5. Predicción de resistencia al avance sin geometría.	31
5.1. Introducción.	31
5.2. Implicaciones de su implementación.	31
5.3. Flujo de trabajo para su implementación.	32
6. Predicción de resistencia al avance con geometría.	33
6.1. Introducción.	33
6.2. Tratamiento de datos.	33
6.3. Hidrostáticas.	34
6.4. Cálculos en <i>FreeCAD</i>	35
6.4.1. Amadeo.	35
6.4.2. Holtrop-Mennen.	40
6.5. Verificación de las funciones.	44
7. Postprocesado en <i>FreeCAD</i>.	47
7.1. Introducción.	47
7.2. Obtención de la resistencia al avance.	47
7.2.1. Extracción de los valores.	48
7.3. Presentación de los resultados.	48
7.3.1. Gráficas.	48
7.3.2. Hoja de cálculo.	51
8. Comparación con <i>MAXSURF</i> de las resistencias obtenidas.	53
8.1. Introducción.	53
8.2. Ejemplo numérico.	53
8.3. Análisis de los diversos buques.	54
8.3.1. Series 60 de la universidad de Iowa.	55
8.3.2. Petrolero de productos de 40.000 TPM.	57
8.3.3. Buque granelero de 35.000TPM.	60
8.3.4. Buque portacontenedores de 4600 TEUS.	64
8.3.5. Buque tipo ro-ro.	67
8.4. Discusión de los valores obtenidos.	70



9. Conclusiones y líneas de trabajo futuras.	73
9.1. Conclusiones.	73
9.2. Líneas futuras de trabajo.	74
Bibliografía	75
A. Instalación detallada del ecosistema necesario.	77
B. Repositorio FreeCAD Ship.	81
C. Creación de una nueva plantilla de herramienta.	87
D. Verificación del método de Amadeo.	107
E. Verificación de las características principales obtenidas.	117
F. Postprocesado.	121
G. Comparación de valores obtenidos en FreeCAD con Maxsurf	145
H. Manual de usuario.	165





Lista de figuras

2.1. Funcionalidades integradas en <i>FreeCAD Ship</i> previamente a la inclusión de las herramientas correspondientes a la predicción de la resistencia al avance.	10
3.1. Selección del ambiente de trabajo Ship desde la pestaña de escenarios	13
4.1. Icono método de Amadeo.	16
4.2. Icono método de Holtrop-Mennen.	16
4.3. Interfaz gráfica del método de Amadeo	18
4.4. Interfaz gráfica del método de Holtrop-Mennen	20
4.5. Imagen parcial de la ventana principal de <i>FreeCAD</i> con las nuevas funcionalidades incluidas.	22
4.6. Interfaz gráfica del método de Holtrop-Mennen integrada en <i>FreeCAD</i>	24
4.7. Interfaz gráfica del método de Amadeo integrada en <i>FreeCAD</i>	25
4.8. Gráfica resistencia vs velocidad obtenida mediante método de Amadeo en el buque 1_A.	28
4.9. Gráfica resistencia vs velocidad obtenida mediante método de Holtrop-Mennen en el buque 1_H.	30
6.1. Interfaz gráfica del método de Amadeo tras la obtención de datos del modelo 3D.	39
6.2. Semi-ángulo de entrada, obtenida de [1]	42
6.3. Interfaz gráfica del método de Holtrop-Mennen tras la obtención de datos del modelo 3D.	43
6.4. Interfaz gráfica del método de Amadeo con buque rápido.	45
6.5. Interfaz gráfica del método de Holtrop-Mennen con buque rápido.	45
7.1. Resistencia obtenida a través del método de Amadeo en el buque 1A.	49
7.2. Potencias obtenidas a través del método de Amadeo en el buque 1A.	49
7.3. Coeficientes resistivos obtenidos a través del método de Amadeo en el buque 1A.	50
7.4. Resistencia obtenida a través del método de Holtrop-Mennen en el buque 1H.	50
7.5. Potencias obtenidas a través del método de Holtrop-Mennen en el buque 1H.	50



7.6. Coeficientes resistivos obtenidos a través del método de Holtrop-Mennen en el buque 1H.	51
7.7. Ventana principal de <i>FreeCAD</i> con hoja de cálculo obtenidas a través del método de Amadeo en el buque 1A.	52
7.8. Ventana principal de <i>FreeCAD</i> con hoja de cálculo obtenidas a través del método de Holtrop-Mennen en el buque 1H.	52
8.1. Gráfica comparativa de resistencia vs velocidad obtenida para el buque S60 a través de <i>MAXSURF</i> y <i>FreeCAD</i>	56
8.2. Gráfica comparativa de potencia vs velocidad obtenida para el buque S60 a través de <i>MAXSURF</i> y <i>FreeCAD</i>	56
8.3. Gráfica de coeficientes resistivos vs velocidad obtenida para el buque S60 a través de <i>FreeCAD</i>	57
8.4. Formas del petrolero de 40.000 TPM obtenidas del TFG [2].	58
8.5. Gráfica comparativa de resistencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>MAXSURF</i> y <i>FreeCAD</i>	59
8.6. Gráfica comparativa de potencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>MAXSURF</i> y <i>FreeCAD</i>	60
8.7. Gráfica coeficientes resistivos vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>FreeCAD</i>	60
8.8. Formas del buque granelero de 35.000 TPM obtenidos de [3].	61
8.9. Gráfica comparativa de resistencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>MAXSURF</i> y <i>FreeCAD</i>	63
8.10. Gráfica comparativa de potencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>MAXSURF</i> y <i>FreeCAD</i>	63
8.11. Gráfica coeficientes resistivos vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>FreeCAD</i>	63
8.12. Formas de un buque portacontenedores de 4600 TEUS obtenida a través de [4]	64
8.13. Gráfica comparativa de resistencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>MAXSURF</i> y <i>FreeCAD</i>	66
8.14. Gráfica comparativa de potencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>MAXSURF</i> y <i>FreeCAD</i>	66
8.15. Gráfica coeficientes resistivos vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>FreeCAD</i>	67
8.16. Formas de un buque tipo ro-ro obtenidas a través de [5].	68
8.17. Gráfica comparativa de resistencia vs velocidad obtenida para un buque tipo ro-ro a través de <i>MAXSURF</i> y <i>FreeCAD</i>	69
8.18. Gráfica comparativa de potencia vs velocidad obtenida para un buque tipo ro-ro a través de <i>MAXSURF</i> y <i>FreeCAD</i>	70
8.19. Gráfica coeficientes resistivos vs velocidad obtenida para un buque tipo ro-ro a través de <i>FreeCAD</i>	70



A.1. Ventana de instalación de Python	78
A.2. Ventana de instalación de Git	79
B.1. Ventana repositorio FreeCAD Ship.	82
B.2. Ventana repositorio FreeCAD Ship para realizar pull request.	84
B.3. Ventana de comparación de archivos durante pull request.	84
C.1. Dimensiones introducidas de manera correcta e incorrecta.	87
D.1. Dimensiones del buque 1 introducido en <i>FreeCAD Ship</i> para predicción Amadeo. .	108
D.2. Gráfica comparativa de las resistencias obtenidas para el buque 1 en el método de Amadeo.	108
D.3. Gráfica comparativa de las potencias efectivas obtenidas para el buque 1 en el método de Amadeo.	109
D.4. Coeficientes resistivos del método de Amadeo obtenidos de <i>FreeCAD Ship</i> para el buque 1.	109
D.5. Dimensiones del buque 2 introducido en <i>FreeCAD Ship</i> para predicción Amadeo. .	111
D.6. Gráfica comparativa de las resistencias obtenidas para el buque 2 en el método de Amadeo.	111
D.7. Gráfica comparativa de las potencias efectivas obtenidas para el buque 2 en el método de Amadeo.	112
D.8. Coeficientes resistivos del método de Amadeo obtenidos de <i>FreeCAD Ship</i> para el buque 2.	112
D.9. Dimensiones del buque 3 introducido en <i>FreeCAD Ship</i> para predicción Amadeo. .	114
D.10. Gráfica comparativa de las resistencias obtenidas para el buque 3 en el método de Amadeo.	114
D.11. Gráfica comparativa de las potencias efectivas obtenidas para el buque 3 en el método de Amadeo.	115
D.12. Coeficientes resistivos del método de Amadeo obtenidos de <i>FreeCAD Ship</i> para el buque 3.	115
E.1. Formas escogidas para verificar la funcionalidad de la protuberancia.	118
E.2. Dimensiones obtenidas durante la verificación de la obtención de la protuberancia. .	119
E.3. Formas del buque rápido utilizado para la verificación de las funciones de obtención de dimensiones del modelo en tres dimensiones.	120
G.1. Datos iniciales del buque S60 en FreeCAD.	146
G.2. Datos iniciales del buque S60 en Maxsurf.	146
G.3. Gráfica resistencia vs velocidad obtenida para el buque S60 a través de <i>MAXSURF</i> . .	147
G.4. Gráfica resistencia vs velocidad obtenida para el buque S60 a través de <i>FreeCAD</i> . .	147
G.5. Gráfica potencia vs velocidad obtenida para el buque S60 a través de <i>MAXSURF</i> . .	148



G.6. Gráfica potencia vs velocidad obtenida para el buque S60 a través de <i>FreeCAD</i> .	148
G.7. Gráfica coeficientes resistivos vs velocidad obtenida para el buque S60 a través de <i>FreeCAD</i> .	148
G.8. Datos iniciales de un petrolero de productos en <i>FreeCAD</i> .	149
G.9. Datos iniciales de un petrolero de productos en <i>Maxsurf</i> .	149
G.10. Gráfica resistencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>MAXSURF</i> .	151
G.11. Gráfica resistencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>FreeCAD</i> .	151
G.12. Gráfica potencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>MAXSURF</i> .	151
G.13. Gráfica potencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>FreeCAD</i> .	152
G.14. Gráfica coeficientes resistivos vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de <i>FreeCAD</i> .	152
G.15. Datos iniciales un buque granelero de 35.000 TPM en <i>FreeCAD</i> .	153
G.16. Datos iniciales de un buque granelero de 35.000 TPM en <i>Maxsurf</i> .	153
G.17. Gráfica resistencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>MAXSURF</i> .	155
G.18. Gráfica resistencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>FreeCAD</i> .	155
G.19. Gráfica potencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>MAXSURF</i> .	156
G.20. Gráfica potencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>FreeCAD</i> .	156
G.21. Gráfica coeficientes resistivos vs velocidad obtenida para un buque granelero de 35.000 TPM a través de <i>FreeCAD</i> .	156
G.22. Datos iniciales un buque portacontenedores de 4600 TEUS en <i>FreeCAD</i> .	157
G.23. Datos iniciales de un buque portacontenedores de 4600 TEUS en <i>Maxsurf</i> .	157
G.24. Gráfica resistencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>MAXSURF</i> .	159
G.25. Gráfica resistencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>FreeCAD</i> .	159
G.26. Gráfica potencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>MAXSURF</i> .	160
G.27. Gráfica potencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>FreeCAD</i> .	160
G.28. Gráfica coeficientes resistivos vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de <i>FreeCAD</i> .	160
G.29. Datos iniciales de un buque ro-ro en <i>FreeCAD</i> .	161



G.30.Datos iniciales de un buque ro-ro en Maxsurf.	161
G.31.Gráfica resistencia vs velocidad obtenida para un buque tipo ro-ro a través de <i>MAXSURF</i>	163
G.32.Gráfica resistencia vs velocidad obtenida para un buque tipo ro-ro a través de <i>FreeCAD</i>	163
G.33.Gráfica potencia vs velocidad obtenida para un buque tipo ro-ro a través de <i>MAX-</i> <i>SURF</i>	163
G.34.Gráfica potencia vs velocidad obtenida para un buque tipo ro-ro a través de <i>Free-</i> <i>CAD</i>	164
G.35.Gráfica coeficientes resistivos vs velocidad obtenida para un buque tipo ro-ro a través de <i>FreeCAD</i>	164
H.1. Ventana para determinar el número de decimales deseados en <i>FreeCAD Ship</i> . . .	165
H.2. Vista de informe en <i>FreeCAD Ship</i>	166
H.3. Vista de <i>FreeCAD Ship</i> con consola de python.	167
H.4. Icono para cargar geometría en <i>FreeCAD</i>	167
H.5. Interfaz gráfica para seleccionar geometría de ejemplo.	168
H.6. Vista principal de <i>FreeCAD Ship</i> tras seleccionar un ejemplo.	168
H.7. Ventana de inicio para abrir un nuevo documento tras seleccionar el módulo “Draft” en <i>FreeCAD</i>	169
H.8. Vista principal en <i>FreeCAD</i> tras importar la geometría de un barco.	169
H.9. Modelo importado en <i>FreeCAD</i> explotado en sus caras.	170
H.10.Modelo importado en <i>FreeCAD</i> agrupado como cáscara.	170
H.11.Modelo importado en <i>FreeCAD</i> agrupado como sólido.	170
H.12.Vista de ejemplo de la ventana de <i>FreeCAD Ship</i> , previamente a crear un buque. .	170
H.13.Vista de ejemplo para crear un nuevo buque en <i>FreeCAD Ship</i>	171
H.14.Vista de ejemplo tras crear un nuevo buque en <i>FreeCAD Ship</i>	171
H.15.Vista ejemplo de la presentación de resultados obtenidos en <i>FreeCAD Ship</i>	172





Lísta de tablas

1.1. Rangos de aplicación en el método de Amadeo.	5
1.2. Rangos de aplicación en el método Holtrop-Mennen [6].	6
4.1. Datos requeridos en el método de Amadeo	18
4.2. Datos requeridos en el método de Holtrop-Mennen	19
4.3. Dimensiones y características principales del buque 1_A para validación del método de Amadeo con eslora en la flotación, L_w , y superficie mojada, S_w , obtenidas mediante fórmulas empíricas.	27
4.4. Resultados obtenidos mediante el método de Amadeo en el buque 1_A.	28
4.5. Dimensiones y características principales del buque 1_H para validación del método de Holtrop-Mennen.	29
4.6. Resultados obtenidos mediante el método de Holtrop-Mennen en el buque 1_H	30
6.1. Relación de modos de obtener las características principales del método de Amadeo.	35
6.2. Relación de modos de obtener las características principales del método de Holtrop-Mennen	40
6.3. Buque para verificación de funciones introducidas en <i>FreeCAD</i>	44
8.1. Dimensiones y características principales del buque de ejemplo que se propone en el método de Holtrop-Mennen.	54
8.2. Comparación resistencias obtenidas entre el ejemplo en el método de Holtrop-Mennen y en <i>FreeCAD Ship</i>	54
8.3. Dimensiones y características principales del buque S60.	55
8.4. Rangos de aplicación del método de Holtrop-Mennen para cargueros, y valores actuales.	55
8.5. Comparación de valores obtenidos de <i>MAXSURF</i> frente a <i>FreeCAD</i> para el buque S60.	56
8.6. Dimensiones y características principales del buque petrolero de productos.	57
8.7. Rango de aplicación del método de Holtrop-Mennen para buques petroleros y valores actuales.	58



8.8. Comparación de valores obtenidos de <i>MAXSURF</i> frente a <i>FreeCAD</i> para un buque petrolero de productos de 40.000 TPM.	59
8.9. Dimensiones y características principales de un buque granelero de 35.000 TPM.	61
8.10. Rango de aplicación del método de Holtrop-Mennen para buques graneleros, y valores actuales	62
8.11. Comparación de valores obtenidos de <i>MAXSURF</i> frente a <i>FreeCAD</i> para un buque granelero de 35.000 TPM.	62
8.12. Dimensiones y características principales de un buque portacontenedores de 4600 TEUS	64
8.13. Rango de aplicación del método de Holtrop-Mennen para portacontenedores y valores actuales	65
8.14. Comparación de valores obtenidos de <i>MAXSURF</i> frente a <i>FreeCAD</i> para un buque portacontenedores de 4600 TEUS.	65
8.15. Dimensiones y características principales de buque tipo ro-ro.	67
8.16. Rango de aplicación del método de Holtrop-Mennen para buques ferry y ro-ro y valores actuales	68
8.17. Comparación de valores obtenidos de <i>MAXSURF</i> frente a <i>FreeCAD</i> para un buque ro-ro.	69
D.1. Comparación de valores obtenidos del método de Amadeo en <i>FreeCAD Ship</i> con los valores esperados para el buque 1.	110
D.2. Comparación de valores obtenidos del método de Amadeo en <i>FreeCAD Ship</i> con los valores esperados para el buque 2.	113
D.3. Comparación de valores obtenidos del método de Amadeo en <i>FreeCAD Ship</i> con los valores esperados para el buque 2.	116
E.1. Datos requeridos en el método de Holtrop	118
G.1. Valores obtenidos de <i>FreeCAD</i> para el buque Series 60.	146
G.2. Valores obtenidos de <i>Maxsurf</i> para el buque Series 60.	147
G.3. Valores obtenidos de <i>FreeCAD</i> para un buque petrolero de productos de 40.000 TPM.	150
G.4. Valores obtenidos de <i>Maxsurf</i> para un buque granelero de 35.000 TPM.	150
G.5. Valores obtenidos de <i>FreeCAD</i> para un buque granelero de 35.000 TPM.	154
G.6. Valores obtenidos de <i>Maxsurf</i> para un buque granelero de 35.000 TPM.	154
G.7. Valores obtenidos de <i>FreeCAD</i> para un buque portacontenedores de 4600 TEUS.	158
G.8. Valores obtenidos de <i>Maxsurf</i> para un buque portacontenedores de 4600 TEUS.	158
G.9. Valores obtenidos de <i>FreeCAD</i> para un buque tipo ro-ro.	162
G.10. Valores obtenidos de <i>Maxsurf</i> para un buque tipo ro-ro.	162





Capítulo 1

Introducción.

1.1. Motivación.

En el sector naval, la predicción de resistencia al avance es una actividad ampliamente realizada. El objetivo de dicha actividad radica en conocer la potencia efectiva que el motor deberá entregar para que el buque cumpla con la misión con la que ha sido diseñado.

Uno de los métodos para obtener la predicción de resistencia al avance es mediante modelos semi-empíricos. La principal ventaja de dicho método es la rapidez y sencillez con la que se obtiene una primera aproximación. Actualmente, en el mercado existen diversos software especializados con la capacidad de realizar la predicción de resistencia al avance a través de diversos modelos semi-empíricos, optimizando así los recursos utilizados.

Los software con capacidad de predicción de resistencia al avance son una herramienta muy útil y de gran valor, sin embargo, no están al alcance de todo el mundo. La principal desventaja del software mencionado es el coste de sus licencias, ya que son software privados. De dicha desventaja nace la motivación principal de este proyecto, poder entregar al mercado una solución gratuita, de calidad, que tenga la capacidad de realizar predicciones de resistencia al avance en un amplio rango de tipos de buques, y otorgar al usuario la posibilidad de ver el código que se está utilizando para obtener los resultados, característica con la que no cuentan los software privados.

Con esta premisa principal se escoge *FreeCAD*, aprovechando la existencia de un módulo externo llamado *Ship* que cuenta con numerosas funcionalidades integradas a excepción de la predicción de resistencia al avance. De esta forma, la integración de herramientas de cálculo de predicción de resistencia al avance en *FreeCAD Ship* aumentará la cobertura del módulo en las operaciones del sector naval, incrementando el valor del programa en el mercado. Ofreciendo de esta manera, para todo tipo de usuarios, una alternativa a los software privados.

1.2. Estado del arte.

Uno de los aspectos más importantes durante el proyecto de creación de un buque nuevo o de una transformación de un buque existente, es conocer la resistencia al avance que experimentará el buque durante la navegación. Tanto al elección del motor principal, como el dimensionamiento de gran parte de los equipos y zonas del buque, como por ejemplo el dimensionamiento de la cámara de máquinas, así como las formas finales del buque, dependen altamente de la resistencia al avance obtenida. Por todo ello, es de vital importancia la obtención de un valor fiable y preciso de la resistencia al avance previamente a la construcción o modificación del buque.



Debido a la dependencia de la elección del motor principal con la resistencia al avance, la variación de resistencia tiene asociada una gran componente económica. Por ello, es importante conseguir una optimización adecuada de las formas durante el proceso de diseño del buque para lograr, una disminución, del coste asociado a la explotación del mismo.

Existen diversos métodos para la obtención de la resistencia al avance de un buque durante su concepción, entre ellos se encuentran los modelos semi-empíricos, los ensayos de remolque y el cálculo mediante simulación por CFD. El método más fiable y preciso para determinar la resistencia al avance son ensayos de remolque. La ventaja de este método es la precisión que aporta y el conocimiento que se obtiene de los ensayo. Por otro lado, las instalaciones necesarias y la creación del modelo a escala son las contrapartidas de este método.

Los modelos semi-empíricos se conforman a partir de la recopilación de datos obtenidos de diversos ensayos de remolque. A partir de dichos datos se obtienen las regresiones lineales con las que se conforman las ecuaciones que modelan el comportamiento del buque. En función del tipo de buque, sus características y su comportamiento en el agua, la resistencia asociada varía mucho. Es por ello, por lo que existen diversos modelos semi-empíricos y cada uno con rangos concretos de aplicación. Los rango de aplicación de cada método incluyen dos tipos de restricciones: en primer lugar se restringe el tipo de buques en los que se puede aplicar el método como por ejemplo veleros, buques de desplazamiento, embarcaciones de planeo...etc; en segundo lugar, se restringen en un rango específico los coeficientes de la carena, relaciones dimensionales y otros coeficientes tales como el número de Froude, las dimensiones principales específicas, entre otras. La gran ventaja de los modelos semi-empíricos es su fácil aplicación y la obtención de una primera aproximación de la resistencia al avance del buque estudiado en las primeras etapas de diseño sin la necesidad de realizar ensayos específicos ni invertir gran cantidad de tiempo y recursos.

Gracias a la simplicidad de la ecuaciones que conforman los modelos semi-empíricos y a la precisión que se obtiene, dichos modelos fueron muy bien acogidos por los usuarios desde un primer momento. Con el paso de los años se comenzaron a introducir los distintos métodos en software y hojas de cálculo, permitiendo que a día de hoy se obtenga una primera aproximación de forma muy rápida y al alcance de cualquier usuario, con introducir únicamente los datos de la embarcación. Además, en caso de tener un modelo en tres dimensiones de la embarcación se puede introducir en el software, obtener las características necesarias del buque a partir del modelo 3D, y obtener la resistencia al avance aproximada.

Por último, otro método que se ha puesto en desarrollo en los últimos años es la simulación a través de CFD. Las siglas CFD responden a "Computational Fluid Dynamics". La ventaja de este método es que simula el comportamiento del buque en el agua, por lo que se pueden optimizar las formas para mejorar la resistencia al avance. Sin embargo, debido a las simplificaciones que realiza el programa se pueden obtener errores y se necesita un amplio conocimiento en la materia para realizar una simulación correcta.

Actualmente coexisten los 3 métodos, sin embargo, en función del ámbito se utilizan más unos u otros. Por ejemplo, los ensayos de remolque son muy utilizados a lo largo de todo el sector naval, desde el ámbito académico hasta la concepción de lo buques. En el ámbito estudiantil se utiliza mucho la obtención de la resistencia al avance a través de modelos semi-empíricos gracias a la rapidez y facilidad de uso que ofrecen. En el sector profesional, en función del nicho de la empresa, de los recursos disponibles y de la formación de sus empleados varía mucho el método seleccionado. Por ello, actualizar e incrementar los recursos disponibles en el mercado, tanto en el ámbito educativo como en el profesional, en cualquiera de los métodos, es un gran aporte de valor al sector.



1.3. Selección de métodos a implementar.

En primer lugar, se realiza un estudio de los métodos semi-empíricos más utilizados actualmente por los principales software específicos del sector naval, los cuales se exponen a continuación.

- Método de Holtrop-Mennen.
- Método de Amadeo.
- Método de Van Oortmersen.
- Método de Savitsky.
- Método de Compton.

El propósito del estudio es reducir el número de modelos semi-empíricos a desarrollar, limitándolo a dos. La selección de los métodos se lleva a cabo teniendo en cuenta los siguientes criterios: capacidad para realizar predicciones de resistencia al avance en un amplio espectro de tipos de buques; métodos fiables y contrastados; mayor capacidad de implementación y utilización del método en el sector; y existencia de una gran base de datos que avale el método para realizar una correcta verificación de la funcionalidad del método una vez implementado en *FreeCAD*.

Finalmente, en base a los criterios expuestos anteriormente, se decide implementar, en el presente proyecto, el método de Amadeo y el método de Holtrop-Mennen.

- Método de Amadeo. Es un método confeccionado para pesqueros. Está basado en la siguiente división de resistencias:
 - Resistencia de fricción de placa placa equivalente, R_F .
 - Resistencia residual, R_R , en la que se incluye toda la resistencia excluida de R_F .
 - Resistencia debido a la correlación entre modelo y buque, R_A

Las regresiones fueron constituidas a partir de datos obtenidos de los ensayos de numerosos pesqueros realizados en el Canal de Experiencias Hidrodinámicas del Pardo, CEHIPAR.

Uno de los motivos principales que justifica su elección, se establece en la inexistencia del método de Amadeo en los software específicos del mercado, introduciendo de esta manera una solución muy interesante y novedosa, teniendo en cuenta que es un método con resultados muy contrastados y con gran potencial. Otros de los motivos para su elección son la sencillez de las ecuaciones y la fiabilidad del método, entre otros [7]. Cabe resaltar, además, que Amadeo García Gómez realizó sus estudios en la Escuela Técnica Superior de Ingenieros Navales de Madrid, (E.T.S.I.N). A continuación se muestra en la tabla 1.1 un resumen de los rangos de aplicación del método de Amadeo.

Tabla 1.1: Rangos de aplicación en el método de Amadeo.

Lpp min [m]	Lpp max [m]	Fn min	Fn max	Cb*B/Lpp min	Cb*B/Lpp min
25	60	0.25	0.40	0.095	0.165

- Método de Holtrop-Mennen. Fue publicado por primera vez en 1977 en la revista “International Ship Progress” por J.Holtrop, como un método estadístico para la predicción de



potencia aunque a lo largo de los años el autor fue realizando actualizaciones y mejoras del método hasta 1984 con la colaboración de G.G.J.Mennen.

El método de Holtrop-Mennen se estructura mediante la siguiente división de resistencias:

- Resistencia equivalente de placa plana.
- Resistencia debido a las formas del casco.
- Resistencia viscosa debido a los apéndices.
- Resistencia debida a la formación de olas.
- Resistencia debido al bulbo de proa no sumergido totalmente.
- Resistencia debida a la correlación entre modelo y buque.

Las regresiones utilizadas para confeccionar el método semi-empírico se llevaron a cabo a través de los datos que se recopilaron de los ensayos obtenidos del centro de Marin (Wageningen, Holanda) en un primer artículo [8]. Sin embargo, años después, se realizó una actualización del artículo, introduciendo diversas modificaciones en algunas de las ecuaciones que modelan el método de Holtrop-Mennen [9]. El rango de aplicación está constituido por buques de desplazamiento tales como: petroleros, bulkcarriers; remolcadores, costeros; portacontenedores; cargueros general; ferries. En la actualidad, es el método más utilizado, tanto a nivel educativo como a nivel profesional, integrándolo en todos los software específicos.

El amplio rango de buques en los que se puede realizar una predicción de resistencia al avance es el motivo principal de elección de este método, además, es el método más extendido y utilizado, de manera que existe una gran base de datos disponible para verificar la eficacia del método tras introducirlo en *FreeCAD*. A continuación, en la tabla 1.2, se muestra un resumen con los rangos de aplicación del método de Holtrop-Mennen.

Tabla 1.2: Rangos de aplicación en el método Holtrop-Mennen [6].

Tipo de buque	Fn máx	CP min	CP máx	L/B min	L/B máx	B/T min	B/T máx
Petroleros y graneleros	0.24	0.73	0.85	5.1	7.1	2.4	3.2
Remolcadores, costeros	0.38	0.55	0.65	3.9	6.3	2.1	3.0
Portacontenedor	0.45	0.55	0.67	6.0	9.5	3.0	4.0
Cargueros	0.30	0.56	0.75	5.3	8.0	2.4	4.0
Ro-Ro, Ferries	0.35	0.55	0.67	5.3	8.0	3.2	4.0

Los dos métodos escogidos permiten calcular la predicción de resistencia al avance en un gran abanico de buques, obteniendo así un módulo en software libre pueda ser utilizado en gran cantidad de ocasiones, introduciendo en el sector una solución gratuita, para un considerable número de buques, y robusta, debido a la eficacia de los métodos escogidos.

1.4. Software asociado.

En este apartado se presentarán las características de los principales software referidos al cálculo de la predicción de resistencia al avance existentes en el mercado, en concreto se comentarán *MAXSURF* y *NavCad - HydroComp*. A parte de los software mencionados, existen otros muy completos dentro del sector naval, como podrían ser *FORAN Ship Design Software* o *DELFTship*. Sin embargo, están orientados a otras disciplinas, y pese a que tienen la capacidad de calcular una predicción de resistencia al avance, esta no es su actividad principal o no cuentan con un módulo específico para ello.



1.4.1. NavCad - HydroComp.

NavCad es un software de ingeniería naval orientado específicamente al análisis y optimización del rendimiento de los buques a través de la simulación del movimiento del buque en el agua en diversas condiciones.

La funcionalidad principal de *NavCad* es el cálculo de la predicción de resistencia al avance por medio de una gran variedad de modelos semi-empíricos incluidos en el software. Además, tiene la capacidad de realizar análisis detallados de la eficiencia del propulsor, así como análisis de estabilidad inicial y condición de peso en rosca.

Para realizar la predicción de resistencia al avance se debe introducir inicialmente un modelo en tres dimensiones para que el software realice los cálculos y análisis pertinentes para obtener las dimensiones y datos necesarios del buque. Los datos se pueden modificar una vez obtenidos, en caso de que alguno de ellos no se haya obtenido correctamente. Por último, eligiendo el modelo semi-empírico a utilizar se obtendrá el cálculo de predicción de potencia.

En términos de coste, el software cuenta con varias opciones de licencia, incluyendo en algunos paquetes otros software de su propiedad. El precio de las licencias parte aproximadamente desde 8500 €, la más económica.

Una de las ventajas del software es que es altamente personalizable y muy visual. Todas las etapas del proceso para obtener la resistencia al avance cuentan con dicha característica, desde la visualización del método semi-empírico óptimo que propone el programa, hasta la obtención de datos del modelo en tres dimensiones, pasando por el trabajo inicial a realizar con las formas del buque. Además, el software es compatible con una amplia gama de formatos de archivo, facilitando su integración con otros programas y herramientas de diseño. Como contrapartida, es un software con un coste elevado, provocando que en ciertos casos como empresas pequeñas, no puedan utilizarlo. Además, aunque el software es muy visual, es poco intuitivo, por lo que el usuario debe contar con formación previa que le aporte un conocimiento detallado en el programa.

1.4.2. MAXSURF Resistance.

MAXSURF es un software de ingeniería naval orientado a diversas áreas dentro de la industria marítima. Cuenta con la capacidad de realizar numerosas operaciones, desde el modelado en 3D de la embarcación, análisis del casco incluyendo diversos cálculos de arquitectura naval como cálculos de estabilidad o hidrostáticas, entre otros, hasta la simulación de operaciones como la predicción de resistencias y movimientos. El software se organiza en diversos módulos, cada uno diseñado para realizar tareas específicas.

- Modeler
- Stability
- Resistance.
- Structure.
- Motions.
- VPP.

Es uno de los software más utilizado en el sector naval, tanto en el ámbito laboral como en el ámbito estudiantil, utilizado por muchas universidades. Es, por ejemplo, uno de los software que



posee la Escuela Técnica Superior de Ingenieros Navales de Madrid, (E.T.S.I.N). Convirtiéndose así en un software de gran interés.

En concreto, se comentarán las funcionalidades y capacidades del módulo “Resistance”, ya que es el responsable del cálculo de predicción de resistencia al avance. Dentro del módulo, se incluyen numerosos métodos semi-empíricos, otorgando al usuario la posibilidad de realizar predicciones de resistencia al avance en un gran abanico de tipos de buques como por ejemplo buques de desplazamiento, de planeo, yates y pesqueros. Otra de las características del método es que existe la posibilidad de realizar la predicción de dos maneras distintas, comentadas a continuación.

- Desde un modelo en 3D del buque. Desde el módulo “Modeler” se crea un proyecto y se modela el buque, existiendo también la capacidad de modelarlo en otro software e importarlo. Una vez el proyecto contiene las formas del buque, se introduce en el módulo “Resistance”, leyéndose automáticamente las dimensiones y características del buque a través del modelo en tres dimensiones, que en caso de no ser las esperadas se pueden modificar. Por último, se obtiene el cálculo de predicción de resistencia al avance según el método indicado y para las velocidades introducidas.
- Sin modelo previo. Cuenta también con la capacidad de calcular la predicción de resistencia al avance con los datos introducidos por el usuario, se debe proporcionar información detallada sobre el diseño del casco como el tamaño, la forma y la distribución de la carga. Una vez introducido los datos, se realizan los cálculos de la misma manera que el caso anterior. Es un método interesante sobretodo cuando el usuario se encuentra en una etapa inicial del proyecto y no se han obtenido aún las formas en tres dimensiones del buque.

En cuanto a las licencias de *MAXSURF*, existe una versión estudiante para las universidades, y referido a las empresas se divide en dos opciones:

- Advanced, con un precio anual de 5842 €
- Ultimate, con un precio anual de 9737 €

Uno de los beneficios del módulo *MAXSURF Resistance* es la capacidad de realizar análisis de resistencia detallados y precisos, lo que permite mejoras en el diseño del casco para un mejor rendimiento. Además, el software admite múltiples formatos de archivo y se puede integrar fácilmente con otro software y herramientas de diseño. Sin embargo, *MAXSURF* tiene algunos inconvenientes, por ejemplo, la cantidad de tiempo y recursos que necesita el proceso de modelado, así como el coste de las licencias, pudiendo desembocar en un gran problema para pequeñas empresas.



Capítulo 2

FreeCAD.

2.1. Introducción.

FreeCAD es un software de modelado paramétrico de diseño 3D, de código abierto y gratuito, utilizado en diferentes campos de la ingeniería y diseño. *FreeCAD* nace en 2002, cuando el proyecto se inició como un complemento para otro programa de diseño [10]. Hoy en día, es una herramienta de diseño independiente y completa, utilizada en diversos sectores, pero orientado internamente, sobretodo, a la ingeniería mecánica, arquitectura y diseño de productos.

FreeCAD se compone de una gran variedad de módulos que permiten a los usuarios adaptar el software a sus necesidades específicas. Cuenta con un total de 22 módulos internos, entre los que se incluyen algunos como dibujo 2D, modelado de piezas, modelado de superficies, ensamblaje y diseño de chapa metálica, entre otros. Además, *FreeCAD* admite módulos externos creados por la comunidad de usuarios, aumentando las ramas de la ingeniería y arquitectura en las que el software ofrece cobertura al mercado. Dentro de los software externos, se pueden encontrar diversos workbench como: ‘ ‘Dodo”, perteneciente a arquitectura y construcción para la construcción de cerchas, vigas y tuberías; “Rocket”, dentro de la ingeniería que proporciona las herramientas para el diseño de cohetes de aficionados; “Pyrate”, usado para diseñar lentes ópticas; o “Ship”, de la arquitectura naval con capacidad de realizar numerosos cálculos, entrando más en detalle en este último a lo largo del capítulo entre otros.

2.2. FreeCAD Ship.

El ambiente de trabajo Ship [10] se utiliza para la creación de estructuras y realización de cálculos para barcos. Cuenta con un amplio abanico de funcionalidades ya implementadas:

- Cargar una geometría de ejemplo.
- Crear un nuevo casco de una nave.
- Curva de áreas.
- Cálculo de hidrostáticas.
- Definición de cargas.
- Creación de tanques.
- Curva capacidad de tanques.



- Crear condiciones de carga.
- Cálculo de la curva GZ.

A continuación, en la imagen 2.1, se muestran las funcionalidades integradas en *FreeCAD Ship* previamente a la introducción de las nuevas herramientas correspondientes a la predicción de resistencia al avance.



Figura 2.1: Funcionalidades integradas en *FreeCAD Ship* previamente a la inclusión de las herramientas correspondientes a la predicción de la resistencia al avance.

El módulo de trabajo “Ship” funciona de forma similar a otros software del mercado. El usuario puede encontrar que a diferencia de otros software, como podría ser *Rhinoceros* o el módulo “Modeler” de *MAXSURF*, el módulo “Ship” no cuenta con un ambiente propio de modelado del buque, sin embargo, a partir de geometrías base se pueden modificar las dimensiones principales y permite la lectura de diferentes tipos de archivo, pudiendo importar archivos de buques realizados con otros software. El resto de funcionalidades trabajan de forma similar a *MAXSURF*, por ejemplo.

Una de las ventajas provenientes de ser un software libre es que todo su código se encuentra, en GitHub, a disposición de cualquier usuario, [11], permitiendo la lectura del código o comprensión del mismo, en caso de que el usuario lo desee, así como hacer modificaciones individuales en el código e incluso poder subirlo y actualizar posteriormente el repositorio.

FreeCAD Ship es una herramienta gratuita y de código abierto, convirtiéndolo en una de sus grandes ventajas y con la que compite en el mercado. Además, proporciona una gran cantidad de herramientas con capacidad para realizar numerosos cálculos, facilitando su utilización y pudiendo agrupar todas las necesidades en un mismo software. De esta forma, introducir la capacidad de realizar el cálculo de predicción de resistencia al avance aumenta el valor del software, haciéndolo más interesante de cara a los clientes. Además, el software se encuentra disponible tanto para Windows, MacOS, como para Linux. Una de las posibles desventajas es el mantenimiento del software, que será mínimo en comparación con un software de pago en el mercado.



Capítulo 3

Ecosistema de software necesario.

En este capítulo se detalla la configuración inicial necesaria para crear un proyecto en *FreeCAD*. Como consideración previa, se resalta la utilización del sistema operativo Windows a lo largo de proyecto, lo que influye en ciertos comandos, problemas relativos al propio sistema, y variaciones especiales.

3.1. Repositorio.

Uno de los puntos más importantes en los proyectos de software es el control del código. Es decir, tener un control y seguimiento de los cambios y actualizaciones realizados, así como de las versiones. Para ello, se utilizan repositorios, espacios donde se almacenan y se gestionan los archivos del código fuente y las modificaciones. Principalmente, los repositorios se utilizan en sistemas de control de versiones, permitiendo la colaboración entre distintos desarrolladores en el mismo trabajo, y un seguimiento de las versiones con la ayuda de un historial detallado.

Generalmente, los repositorios trabajan con ramas. Cada rama representa una versión distinta del código fuente al que se le han realizado cambios. Existe una jerarquía de ramas, en la rama principal se encuentra la versión del proyecto en producción, denominada rama máster o principal. A partir de la rama principal, se crean nuevas ramas con los cambios que cada desarrollador realice, permitiendo crear simultáneamente tantas ramas como se requiera. La principal ventaja radica en trabajar en un ambiente aislado de la rama principal para no comprometer la integridad de la misma.

Los cambios realizados se integran en la rama correspondiente a través de los denominados “commits”, en los que se incluye una descripción concreta y detallada de los cambios efectuados, actualizando de esta manera la rama. Posteriormente, una vez introducidos todos los cambios deseados es necesario asegurar que no existan interferencias entre la rama en la que se han incluido los cambios y la rama principal. Finalmente se pueden fusionar las dos ramas, mediante un pull request, actualizando así el proyecto en producción. ¹

En función de la herramienta de control de versiones, puede existir repositorios únicamente locales o incluso en la nube. Además, en función del proyecto, el código fuente podrá ser privado o público.

Para realizar dicho control de versiones, en este proyecto se ha utilizado Git. Por definición, “Git es un programa de código abierto para realizar el seguimiento de los cambios en los archivos de texto”, [12]. Para tener un repositorio actualizado en la nube y poder trabajar en él desde

¹En el anexo B, se incluyen explicaciones detalladas a cerca de los commits y del pull request.



cualquier parte, se ha utilizado Github, programa que ofrece un repositorio online mediante un control de versiones utilizando Git.

Se especifican a continuación los programas a instalar y pasos a seguir.

- Instalación de Git. Se puede descargar desde la propia página web [Descarga Git](#) e instalar con las opciones predeterminadas.
- Usuario Github. Accediendo a través de su propia página [Registrarse en Github](#).
- Contraseña SSH. Es un problema referido a Github que aparece en sistema operativo Windows, se detalla su solución en el anexo [A](#).

Una vez realizados los pasos especificados, es posible acceder al repositorio del módulo en el que se desee trabajar, en este caso se trata de *FreeCAD Ship*, y poder trabajar en el código fuente e introducir las novedades en el software.

3.2. Entorno *FreeCAD*.

El primer paso es instalar *FreeCAD*. Se puede descargar a través de su propia web [Descargar FreeCAD](#).

Una vez instalado, el programa contiene ciertos ambientes de trabajo o workbench que se instalan por defecto, workbench internos. El ambiente de trabajo “Ship”, ambiente en el que se introducirán las nuevas funcionalidades, no se encuentra en los ambientes internos por defecto, por lo que es necesario incluirlo. Existen dos opciones para incluir ambientes externos de trabajo en *FreeCAD*.

- Manualmente, se puede encontrar la información necesaria para realizarlo en la propia web [Instrucciones para incluir workbench externo](#). Es la opción escogida cuando se desee utilizar *FreeCAD* únicamente a nivel de usuario y no interactuar con el código.
- Si se quieren realizar actualizaciones o cambios en el código, como es el caso de este proyecto, se debe disponer del código fuente. Para ello, se realizará un clonado del repositorio específico y será de esta manera como se implementará el ambiente concreto en *FreeCAD*. Se muestran a continuación los pasos a seguir.
 1. Crear un fork del repositorio accediendo a [Repositorio workbench ship](#). De esta manera, se creará una copia del proyecto en nuestro usuario de Github.
 2. Acceder a la ruta donde se instalará el repositorio.²
 3. Crear el clonado del repositorio. Para ello, desde la ruta donde se quiera incluir la información del repositorio se escribe el siguiente comando en la consola de Windows: “git clone git@github.com:usuarioGit/freecad.ship.git” introduciendo en “usuarioGit” el nombre de usuario personal dentro de Git.

Para comprobar que se ha incluido el ambiente de trabajo correspondiente, se inicia *FreeCAD* y se comprueba en la pestaña de escenarios que se encuentra el módulo “Ship”, tal y como se muestra en la imagen [3.1](#). También se puede comprobar observando que en la ruta de clonado del repositorio se ha creado una carpeta denominada “freecad.ship”.

²Para más información de los pasos a seguir ver el Anexo [C](#).

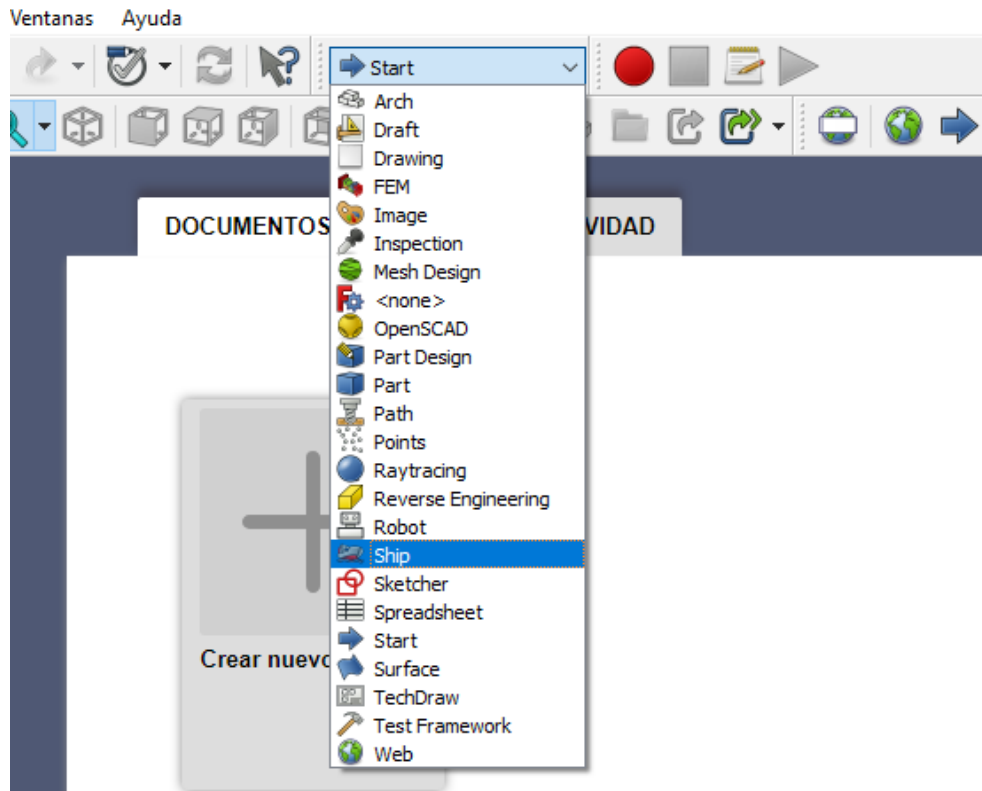


Figura 3.1: Selección del ambiente de trabajo Ship desde la pestaña de escenarios

Desde este momento, todos los cambios que se realicen en el código del repositorio que se ha clonado aparecerán en *FreeCAD* cuando se inicie la aplicación a nivel local, no aparecerán en el proyecto principal hasta que se integren las ramas. De esta forma, se podrán probar los cambios en la máquina local automáticamente.

3.3. Python.

Python es un lenguaje de programación de alto nivel utilizado globalmente para todo tipo de proyectos y software como aplicaciones, desarrollos web o pequeños desarrollos, que además cuenta con la ventaja de ser extensible en C o C++.

Python trabaja mediante la interpretación del código fuente, permitiendo una forma de escribir interactiva y muy clara. El código permite ser probado de forma instantánea sin necesidad de ser compilado. Además, una de las grandes ventajas de Python es que se puede utilizar desde diversos sistemas operativos como Windows, Mac o Linux [13].

Para la implementación del módulo y las actualizaciones en el código fuente, se utilizará como lenguaje python, por lo que será necesaria tanto la instalación de lenguaje como la instalación de un editor.

En cuanto a la instalación de python, se puede descargar desde su propia página web [Descargar Python](#) y, una vez descargada la versión adecuada ³, se instala y se comprueba que se ha realizado correctamente ⁴

³En este caso se instaló la versión 3.9.12 pese a que la última versión hasta el momento era la versión 3.11 debido a un problema de compatibilidad de versiones con el paquete pyqt5-tools.

⁴Para más información sobre la instalación, crear los path adecuados y su comprobación ver el Anexo A.



Una de las funcionalidades de python es que trabaja a partir de bibliotecas. Dichas bibliotecas están disponibles y se pueden incluir en función de las que sean necesitadas, la gran ventaja de las librerías es que abarcan una gran cantidad de tareas distintas. Para la realización del proyecto se necesitarán instalar varias librerías, para ello, desde la consola de comandos se escribirá: “Pip install nombreliberia”. Las librerías a instalar son:

- numpy. Librería para realizar distintas operaciones matemáticas.
- PyQt5. Librería para crear aplicaciones con interfaz gráfica de usuario
- matplotlib. Utilizada para la generación de gráficos en dos dimensiones.
- pyside2. Es necesario para las herramientas utilizadas para la interfaz gráfica.
- flake8. Se utiliza para verificar la correcta escritura del código y evitar errores.

En este punto, la máquina contiene el lenguaje de programación Python, por lo que será capaz de ejecutar archivos de Python desde la propia terminal de comandos de Windows. Sin embargo, para realizar cambios en el código y poder programar será necesaria la instalación de un editor. En el mercado existen muchos tipos de editores, tanto gratuitos como de pago. En este caso se ha optado por instalar Spyder, se puede obtener desde su página [Descargar Spyder](#).

El siguiente paso a realizar es crear en Spyder un proyecto asociado al módulo “ship” que se ha incluido. De esta manera se irán introduciendo los cambios realizados desde Spyder en la copia local del repositorio.

3.4. QT.

Una de las partes más importantes de las funcionalidades a introducir es el diseño de las interfaces gráficas de usuario. Para ello, se utilizará el software *Designer*, que se encuentra en PyQt5.

PyQT5 es una librería de Python utilizada para crear aplicaciones con interfaz gráfica de usuario, denominado GUI, utilizando QT, que es una biblioteca de herramientas gráficas. Dentro de PyQT5, existen distintos programas con diversas funcionalidades, y para el diseño de la interfaz gráfica de usuario se utilizará *Designer*. La ventaja de PyQT5 es su simplicidad y facilidad de uso, por ejemplo, una persona que desconoce totalmente de su existencia y funcionamiento podría aprender a utilizarla en poco tiempo sin emplear una gran cantidad de recursos.

Designer funciona como un editor virtual, permitiendo crear y diseñar interfaces gráficas de usuario de forma sencilla e intuitiva. Existen diversos elementos que se puede utilizar, como botones que pulsar, casillas de verificación, casillas para introducir texto, entre otras muchas. Además, una de las ventajas que ofrece *Designer* es la personalización. Se pueden personalizar los atributos de los elementos, tanto tamaño, color, tipo de letra...etc. Una vez creada la interfaz gráfica, se programa en lenguaje python la lógica en la que se basará la interacción entre la interfaz gráfica y el software.



Capítulo 4

Creación de una nueva plantilla de herramienta.

En el presente capítulo se realizará una explicación de los pasos a seguir en la etapa de creación de una nueva plantilla cuya finalidad es la inclusión de una nueva plantilla en *FreeCAD*.

4.1. Introducción.

Crear una nueva plantilla de herramienta engloba todos los pasos necesarios para incluir una nueva funcionalidad en *FreeCAD* y que se encuentre accesible a los usuarios, sin embargo, no incluye el código interno que realiza los cálculos pertinentes, en este caso, cálculos de predicción de resistencia al avance. Es decir, una vez creada la plantilla, se habrá integrado en *FreeCAD* la funcionalidad en el código interno, en la barra de herramientas, en la barra de menú y se podrá acceder a ella, pero todavía no realizará cálculos internos ni presentará los resultados. La realización de los cálculos internos y la presentación de resultados se incluirá en capítulos posteriores: 6; 5; y 7.

A continuación, a través de un recorrido por el flujo de trabajo en *FreeCAD*, una vez la funcionalidad esté incluida, se comentarán los trabajos que implica la creación de una nueva plantilla.

En primer lugar, se podrá realizar la predicción de potencia tanto con un modelo en tres dimensiones del buque, como sin él. En caso de contar con un modelo en tres dimensiones, es necesario seleccionarlo, a partir de este punto se podrá acceder al método semiempírico, es decir, a la herramienta, desde el menú o desde la barra de herramientas.

El primer paso es incluir una nueva funcionalidad en los diversos menús para que el usuario pueda acceder a ella. Dicha funcionalidad queda visualmente representada por su icono, tanto en la barra de herramientas como en la barra de menús, por lo que es necesario crear el icono que lo representará. A continuación, tras pulsar en la funcionalidad, se desplegará una ventana de diálogo de interacción entre la funcionalidad y el usuario para confirmar los datos con los que se van a realizar los cálculos de predicción de potencia. Dicha ventana de diálogo se consigue a través de la creación de una interfaz gráfica de usuario, también denominada GUI. Las acciones comentadas hasta el momento se incluyen en la creación de una nueva plantilla de herramienta.

A lo largo del proyecto se incluirán dos funcionalidades nuevas, la correspondiente al método de Amadeo y la correspondiente al método de Holtrop-Mennen. Se realizará una explicación de los pasos a seguir que son comunes a ambos métodos y se comentarán las particularidades de cada uno de ellos en caso de ser necesarias.



4.2. Pasos previos.

Previamente a la creación de los iconos, las interfaces gráficas...etc, es necesario crear una rama nueva para incluir todos los cambios y modificaciones.

Tras realizar los pasos comentados en los capítulos anteriores, entre los que se incluyen el clonado del repositorio en la máquina local y la creación de un proyecto nuevo en Spyder, el siguiente paso es crear una rama nueva. Para ello, se deben seguir las siguientes indicaciones:

1. Abrir una terminal de comandos en la ruta *FreeCAD Ship*.
2. Escribir: "git checkout -b nombrerama" y pulsar intro. Introduciendo en "nombrerama" el nombre específico de la rama que se está creando, siendo interesante que esta defina los cambios que se van a realizar.

Tras ejecutar dicho comando, se crea una rama nueva y se traslada al usuario a ella automáticamente. De esta forma, todos los cambios y actualizaciones que se realicen se harán en una rama aparte, asegurando la integridad del proyecto principal. Los cambios mencionados se publicarán en el repositorio más adelante.

4.3. Icono.

Visualmente, es de gran importancia que cada funcionalidad esté representada por un icono que le caracterice. De esta manera, será más fácil para el usuario distinguirla y poder acceder a ella.

Los iconos se pueden encontrar en el repositorio en: resources\icons. A la hora de crear un nuevo icono se guardará en dicha ubicación. Además, es interesante resaltar que se le debe otorgar un nombre que represente la funcionalidad. En este caso por ejemplo, se han nombrado Resistencia_amadeo y Resistencia_holtrop respectivamente

A continuación, se muestran los iconos creados para representar el cálculo de la predicción de resistencia resistencia al avance en función del método elegido.

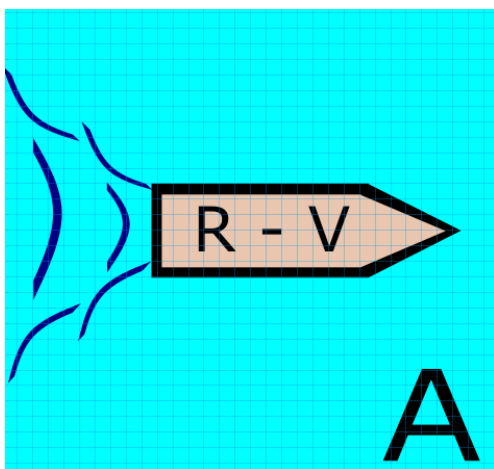


Figura 4.1: Icono método de Amadeo.

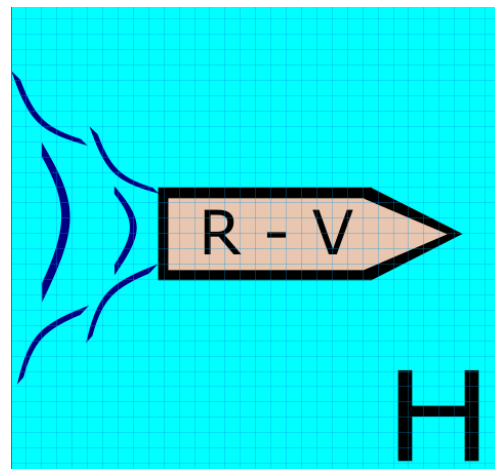


Figura 4.2: Icono método de Holtrop-Mennen.

La intención reside en representar un buque avanzando en la mar con el sistema de olas que genera por el propio avance. El texto R- V está referido a la gráfica que se define a través la



predicción de resistencia al avance, resistencia – velocidad. Por último, la letra que aparece en la esquina inferior derecha corresponde al modelo semiempírico del que se trata, Holtrop-Mennen o Amadeo.

Una de las opciones para comenzar, es copiar un icono ya existente, cambiarle el nombre y a partir de este icono realizar modificaciones. De esta manera, se parte con una plantilla que incluye, entre otras ventajas, las dimensiones pertinentes.

Tras la creación del icono, es necesario incluirlo en el código interno de iconos existentes. Dicha integración se realizará en el archivo Ship.qrc que se encuentra dentro de la carpeta de resources.¹

4.4. Interfaz gráfica de usuario.

La interfaz gráfica de usuario es la ventana a través de la cual el usuario interactúa para indicar y verificar los datos necesarios en función de cada método. En este caso, la interfaz gráfica varía en función del modelo semiempírico ya que los datos que requiere cada método son diferentes, por lo que se tratarán por separado.

Previamente a seleccionar los datos que se deberán incluir en las interfaces gráficas de usuario, es necesario entender la forma de proceder. En este caso, la interfaz gráfica se dividirá en dos columnas.

- **Etiquetas.** En la primera columna se sitúan las etiquetas que definen los datos a introducir. La función de dichas etiquetas es albergar un título y explicación a cerca de la característica con la que el usuario puede interactuar. Es decir, en el caso de la predicción de resistencia al avance un dato importante es la velocidad máxima y el usuario debe introducirlo, por lo que en la interfaz gráfica se añade una etiqueta que represente la velocidad máxima. Dichas etiquetas no tienen la capacidad de interactuar con el usuario de forma que este introduzca datos o seleccione opciones, únicamente describe la característica con la que se va a interactuar para definirla.
- **Objeto de interacción.** Por otro lado, en la columna colindante, se le asigna a cada etiqueta un elemento a través del cual el usuario puede interactuar. Existen diversos tipos de elementos como casillas de verificación, espacios editables para introducir valores, o desplegables con diversas opciones en función de la interacción necesaria. De esta forma, se le asigna un elemento a cada etiqueta para que el usuario pueda definir dicha característica.

En la denominada GUI se deben incluir todos los datos requeridos por la herramienta y que el usuario tenga que manipular. En caso de obtener la predicción de resistencia al avance desde un modelo en tres dimensiones el usuario únicamente debería introducir los valores que no sean calculables a través del modelo en tres dimensiones. Sin embargo, en caso de obtener la predicción sin un modelo el usuario se deben introducir todos los valores del buque. Por lo tanto, en este caso, para poder realizar la predicción en ambos supuestos, se introducirán todos los datos requeridos en los métodos semi-empíricos. Por lo tanto, el primer paso consiste en extraer de cada método las variables.

Por último, las interfaces gráficas se encuentran dentro del repositorio en la ruta: resources\ui. Al igual que ocurría en los iconos, se otorgará un nombre que defina la funcionalidad de la interfaz gráfica, pero en este caso el nombre tendrá la siguiente sintaxis: "TaskPanel_nombreui"

¹En el anexo C se incluyen más explicaciones a cerca de la inclusión de los iconos en Ship.qrc.



- Amadeo. A continuación, se muestran, en la Tabla 4.1, los datos requeridos para la utilización del método, extraídos de [7].

Tabla 4.1: Datos requeridos en el método de Amadeo

Datos necesarios	Dimensiones y notas.
Eslora entre perpendiculares, Lpp	m
Calado, T	m
Manga, B	m
Coefficiente de bloque, Cb	–
Superficie mojada, Sw	m^2
Eslora en la flotación, Lw	m
Volumen desplazado, ∇	m^3
Velocidades del buque, V	m/s
Longitud de tobera, l	m
Coefficiente del propulsor, η_p	–
Margen de mar	–
Diámetro propulsor, d	m
Longitud protuberancia bulbo, lb	Longitud medida a partir de la cuaderna 20, m.
Timón	Existencia o no del timón.

Finalmente, en la imagen 4.3, se muestra la interfaz gráfica del método de Holtrop-Mennen que se introducirá posteriormente en *FreeCAD*.

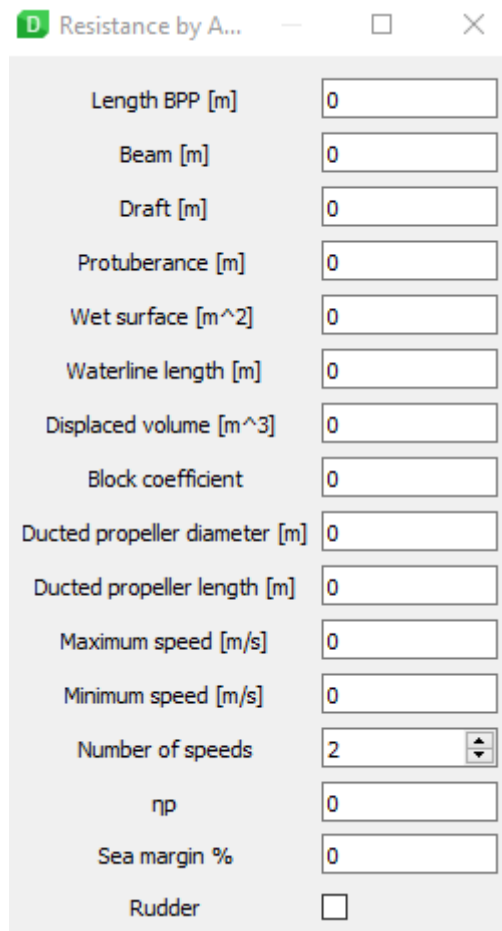


Figura 4.3: Interfaz gráfica del método de Amadeo

Para realizar el control de unidades se especificará en la etiqueta de cada uno de los



elementos la dimensión que se ha de introducir.

Se puede observar que se han asignado tres tipos de objetos de interacción distintos.

- Espacios editables. En la mayoría de casos, se ha asignado a las etiquetas un espacio en el cual el usuario introduce el valor que desee, que corresponderá a la característica definida por la etiqueta.
- “Spinbox”. En el caso del número de velocidades, el objeto que mejor lo define es una “Spinbox”. Su funcionamiento se basa en aumentar o disminuir una unidad con ayuda de las flechas asociadas. De esta manera, únicamente se permite introducir números enteros con un valor mínimo de dos velocidades.
- Casilla de verificación. En caso de querer incluir el timón en la predicción de resistencia al avance se marcará dicha casilla.

Además, *Designer* ofrece la posibilidad de introducir explicaciones adicionales. En cada uno de los parámetros se puede incluir una explicación que el usuario puede visualizar con tan solo situar el ratón encima del parámetro pertinente. Por ejemplo, en la protuberancia se ha incluido: “Medida a partir de la cuaderna 20”, y de esta forma ayudar al usuario a entender que valor exactamente debe introducir.

- Holtrop-Mennen. A continuación, siguiendo la misma dinámica que en el método de Ama-deo, se muestran en la tabla 4.2 los datos requeridos para la utilización del método, extraídos de [9].

Tabla 4.2: Datos requeridos en el método de Holtrop-Mennen

Datos necesarios	Dimensiones y notas.
Calado, T	m
Manga, B	m
Superficie mojada, Sw	m ²
Eslora en la floración, Lw	m
Volumen desplazado, ∇	m ³
Coeficiente de bloque, Cb	—
Coeficiente de la maestra, Cm	—
Coeficiente en la flotación, Cw	—
Coeficiente prismático, Cp	—
Formas del cuerpo	Formas en V / formas en U / formas normales.
Semiángulo de entrada, iE	grados, °
Posición del centro de flotación, xcb	m
Área transversal del bulbo, ABT	Medido en la cuaderna 20, m ²
Área del espejo de popa, AT	m ²
Altura del cdg de ABT, hb	m
Velocidad del buque, u	m / s
Coeficiente del propulsor, η _p	—
Margen de mar	—
Áreas de los apéndices	m ²

En este caso, no se introducirá el coeficiente prismático, C_p , en la interfaz gráfica. Esto es debido a que es un valor que se puede calcular a través del coeficiente de bloque, C_b , y el coeficiente de la maestra, C_M , mediante $C_p = C_b/C_M$. Además, en caso de que el usuario



deseo modificarlo tendrá que modificar dichos valores también, por lo que no se considera suficientemente relevante para introducirlo en la interfaz gráfica.

Para incluir las diversas formas que puede tener el buque, se incluirá un menú desplegable que recoja las tres opciones posibles. Por último, se incluirá la opción de indicar el área concreta de cada uno de los apéndices posibles. Por defecto el área de dichos apéndices se fijará en cero, de tal forma, en caso de existir el apéndice el usuario modificará dicho valor.

A continuación, en la imagen 4.4 se muestra la interfaz gráfica del método de Holtrop-Mennen que se introducirá posteriormente en *FreeCAD*.

Beam [m]	<input type="text" value="0"/>
Draft [m]	<input type="text" value="0"/>
Wet surface [m ²]	<input type="text" value="0"/>
Waterline length [m]	<input type="text" value="0"/>
Displaced volume [m ³]	<input type="text" value="0"/>
Cb	<input type="text" value="0"/>
Cm	<input type="text" value="0"/>
Cw	<input type="text" value="0"/>
Afterbody form	V-shaped sections ▾
Half angle of entrance [deg]	<input type="text" value="0"/>
Xcb [m]	<input type="text" value="0"/>
ABT [m ²]	<input type="text" value="0"/>
AT [m ²]	<input type="text" value="0"/>
hb [m]	<input type="text" value="0"/>
Maximum speed [m/s]	<input type="text" value="0"/>
Minimum speed [m/s]	<input type="text" value="0"/>
Number of speeds	<input type="text" value="2"/>
η_p	<input type="text" value="0"/>
Sea margin %	<input type="text" value="0"/>
Rudder behind skeg [m ²]	<input type="text" value="0"/>
Rudder behind stern [m ²]	<input type="text" value="0"/>
Twin-screw balance rudders [m ²]	<input type="text" value="0"/>
Shaft brackets [m ²]	<input type="text" value="0"/>
Skeg [m ²]	<input type="text" value="0"/>
Strut bossings [m ²]	<input type="text" value="0"/>
Hull bossings [m ²]	<input type="text" value="0"/>
Shafts [m ²]	<input type="text" value="0"/>
Stabilizer fins [m ²]	<input type="text" value="0"/>
Dome [m ²]	<input type="text" value="0"/>
Bilge keels [m ²]	<input type="text" value="0"/>

Figura 4.4: Interfaz gráfica del método de Holtrop-Mennen



En el método de Holtrop-Mennen existe un mayor número de variables en comparación con el método de Amadeo, desembocando en una interfaz gráfica de mayores dimensiones, como se puede observar en la imagen 4.4. Además, a diferencia del método de Amadeo, se ha incluido un desplegable para que el usuario pueda definir el tipo de formas del buque.

Al igual que con los iconos, es necesario introducir las interfaces gráficas en el archivo `Ship.qrc`².

Finalmente, para que los cambios efectuados en los iconos y en las interfaces gráficas se actualicen en *FreeCAD*, es necesario ejecutar el archivo `compile_resources`.

4.5. Creación del menú y barra de herramientas.

Tras crear el icono y la interfaz gráfica de los métodos a incluir, es necesario crear el espacio que ocuparán las herramientas nuevas en *FreeCAD*. Este espacio se refiere tanto al correspondiente a la barra de herramientas como el correspondiente al menú.

La barra de menú se encuentra en la parte superior de la ventana en *FreeCAD* y está dividido por tipos de funcionalidades. En cada tipo de funcionalidad se agrupan las herramientas que se utilizan para ello. De tal forma, en este caso se agruparán ambos métodos en "Resistance". A través de esta pestaña se podrá acceder a ambos métodos.

Por otra parte, en la barra de herramientas también se realiza una separación por funcionalidades, pero de forma más sutil. La barra de menús así como la barra de herramientas se crean en el archivo `init_gui.py` y se configuran en el archivo `ShipGui.py`, ambos archivos se encuentran en el primer nivel dentro del repositorio.

A continuación, se muestra un ejemplo de la creación de ambos espacios.

```

1 # ToolBar
2 resistancelist = ["Resistance_Amadeo",
3                 "Resistance_Holtrop"]
4
5 self.appendToolBar(
6     QT_TRANSLATE_NOOP("Workbench", "Resistance"), resistancelist)
7 self.appendMenu(
8     QT_TRANSLATE_NOOP("Workbench", "Resistance"), resistancelist)

```

En la primera parte del código, se crea una lista que incluye todas las herramientas que tengan una misma finalidad, en este caso se crea una lista con los dos modelos semiempíricos. En la segunda parte, se incluye la lista tanto en el menú como en la barra de herramientas.

A continuación, se configuran la barra de menús y la barra de herramientas de cada método en el archivo `ShipGui.py`. Se muestra un ejemplo de la manera de proceder³.

```

1 class Amadeo:
2     def IsActive(self):
3         return True
4
5     def Activated(self):
6         from . import 'nombre del archivo donde se incluyen los calculos'
7         resistanceAmadeo.load()
8
9     def GetResources(self):

```

²En el anexo C se incluyen explicaciones a cerca de la integración de las interfaces gráficas en `Ship.qrc`

³En el anexo C se incluye el código completo y explicaciones adicionales



```

10     MenuText = QT_TRANSLATE_NOOP (
11         'nombre del metodo en init_gui',
12         'nombre del menu')
13     ToolTip = QT_TRANSLATE_NOOP (
14         'nombre del metodo en init_gui',
15         'Explicacion que aparecera al colocarse'
16         'encima de la funcion en la barra de herramientas')
17     return {'Pixmap': 'nombre del icono',
18            'MenuText': MenuText,
19            'ToolTip': ToolTip}
20
21 FreeCADGui.addCommand('Resistance_Amadeo', Amadeo())

```

En primer lugar, se crea una clase con cada uno de los métodos y se especifica en que situaciones se encontrará activo, en este caso, se encontrará siempre activo, para así poder realizar predicciones sin un buque seleccionado. A continuación, se rellenan los espacios con los nombres requeridos en cada punto. Por último, para poder abrir el método y que sea funcional en el futuro, se debe incluir en la última línea el comando que se refiere al método.

En este momento, si se entra en *FreeCAD Ship*, aparecerán las nuevas funcionalidades, tanto en la barra de menús como en la barra de herramientas, con sus iconos respectivos y las explicaciones que se hayan incluido. Se muestra a continuación en la imagen 4.5 la ventana principal de *FreeCAD* tras realizar los pasos anteriores.



Figura 4.5: Imagen parcial de la ventana principal de *FreeCAD* con las nuevas funcionalidades incluidas.

Se puede observar en la imagen 4.5 que tras situar el ratón encima del icono de uno de los métodos se visualizan las explicaciones introducidas en la configuración de la barra de herramientas.

4.6. Integración de la interfaz gráfica de usuario.

Llegado este punto, se han incluido las nuevas funcionalidades en la barra de herramientas y de menús y se ha creado la interfaz gráfica anteriormente, sin embargo, si se intentan utilizar las nuevas herramientas ni siquiera se desplegará la interfaz gráfica. Esto es debido a que se debe unir, mediante código, la interfaz gráfica con *FreeCAD Ship*.

Tal y como se ha comentado en el apartado 4.4, la interfaz gráfica está formada por etiquetas a las que se le asignan individualmente un objeto de interacción. Tanto a la etiqueta como al objeto de interacción se les asignó un nombre a la hora de crearlos. Dichos nombres serán utilizados para unir la interfaz gráfica con *FreeCAD*, en concreto, se utilizará el nombre del objeto de interacción.

El siguiente paso es crear una carpeta para cada nueva funcionalidad, ubicada en el primer nivel



del repositorio, donde se encuentran las carpetas de las demás funcionalidades ya incluidas. En el interior de dicha carpeta se creará un archivo denominado TaskPanel.py.

El archivo TaskPanel está conformado por diversas partes, siendo “setupUi” la de mayor interés en este apartado, en ella se crean los enlaces entre *FreeCAD* y la Ui.

Para instanciar los elementos se sigue la siguiente norma:

“self.form.nombreui = self.widget(QtGui.Tipodelwidget, “nombreui”) ”.

Siendo:

- “nombreui”. El nombre asignado al objeto de interacción durante la creación de la interfaz gráfica.
- “Tipodelwidget”. El elemento escogido. Existen varias opciones como QSpinBox, QCheckBox, y QGroupBox, entre otros.

Se muestra a continuación, en la imagen 4.7, cómo quedan instanciada la ui en el método de Amadeo ⁴. Cabe resaltar que de todo el código que incluye el archivo TaskPanel, se está mostrando únicamente la parte del código que hace referencia al enlace de la interfaz gráfica con *FreeCAD*.

```

1 class TaskPanel:
2
3     def setupUi(self):
4         self.form.protuberance = self.widget(QtGui.QLineEdit, "protuberance")
5         self.form.Sw = self.widget(QtGui.QLineEdit, "Sw")
6         self.form.Lw = self.widget(QtGui.QLineEdit, "Lw")
7         self.form.volume = self.widget(QtGui.QLineEdit, "volume")
8         self.form.Cb = self.widget(QtGui.QLineEdit, "Cb")
9         self.form.d_length = self.widget(QtGui.QLineEdit, "d_length")
10        self.form.d_diameter = self.widget(QtGui.QLineEdit, "d_diameter")
11        self.form.max_speed = self.widget(QtGui.QLineEdit, "max_speed")
12        self.form.min_speed = self.widget(QtGui.QLineEdit, "min_speed")
13        self.form.n_speeds = self.widget(QtGui.QSpinBox, "n_speeds")
14        self.form.etap = self.widget(QtGui.QLineEdit, "etap")
15        self.form.seamargin = self.widget(QtGui.QLineEdit, "seamargin")
16        self.form.rudder = self.widget(QtGui.QCheckBox, "rudder")
17        if self.initValues():
18            return True

```

A partir de este momento, cuando se entre en *FreeCAD* y se seleccione la herramienta del método de Amadeo o del método de Holtrop-Mennen se desplegará la interfaz gráfica de usuario.

A continuación, se incorporan imágenes de las interfaces gráficas de ambos métodos integradas en *FreeCAD*.

⁴En el anexo C se incluye código adicional.

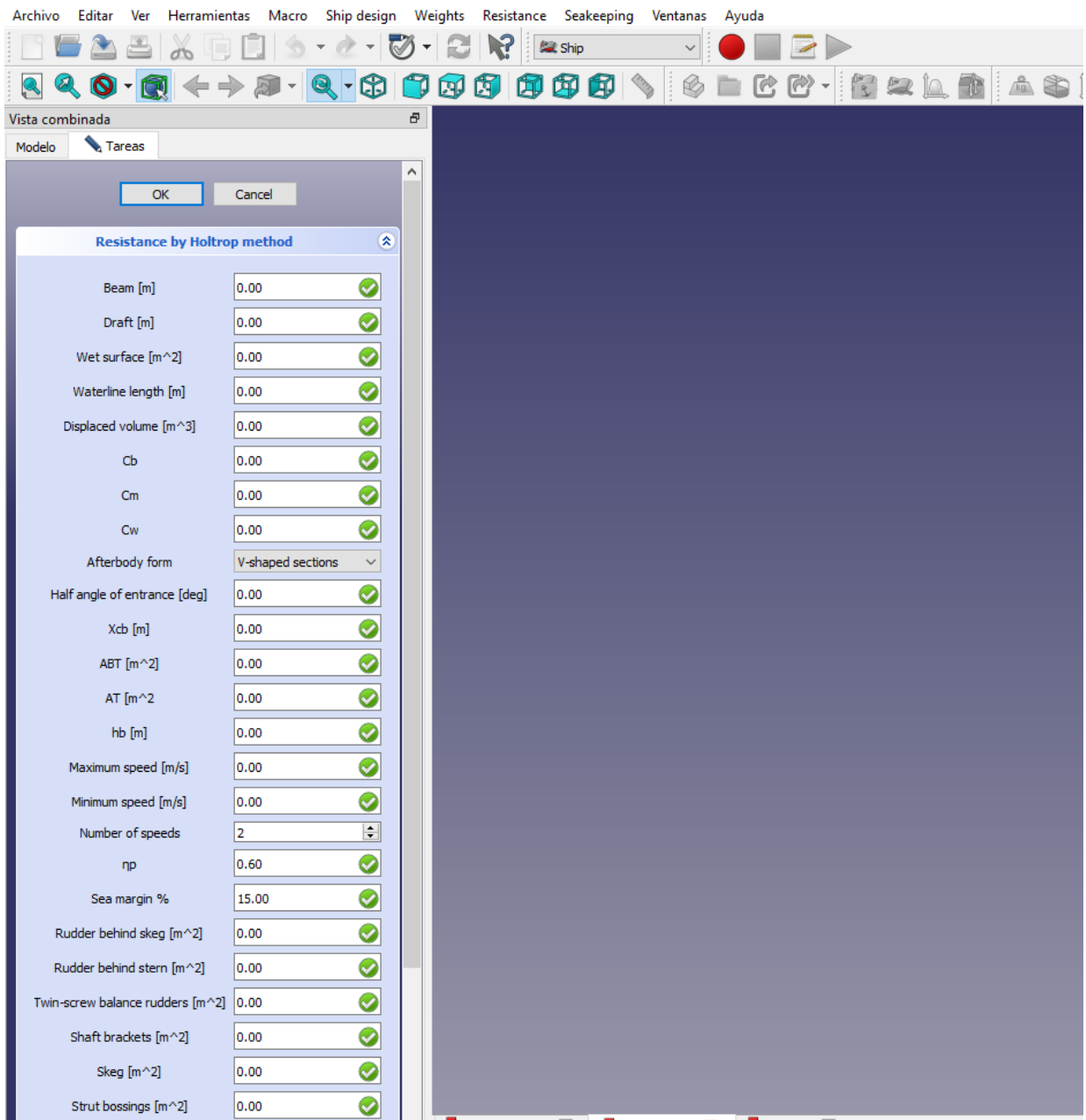


Figura 4.6: Interfaz gráfica del método de Holtrop-Mennen integrada en FreeCAD.

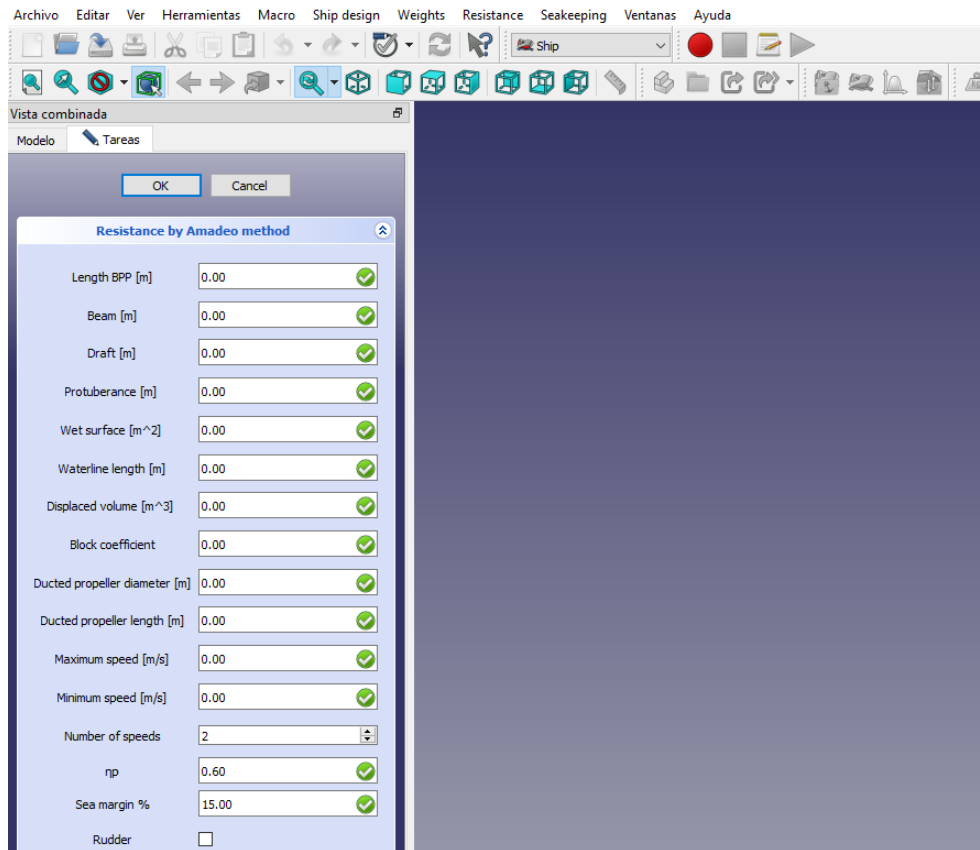


Figura 4.7: Interfaz gráfica del método de Amadeo integrada en FreeCAD.

4.7. Programación del método semiempírico.

El último paso que incluye la creación de una nueva plantilla de herramienta es programar el archivo que realiza los cálculos de predicción de potencia una vez introducidos los datos. Son los archivos en los que plasma informáticamente el método correspondiente, mediante un lenguaje específico y unas reglas concretas, para poder realizar una correcta interacción con el ordenador.

La primera actividad a realizar es leer detenidamente el método a programar y comprender su funcionamiento. Es decir, cómo se divide el método, su flujo de trabajo, las ecuaciones involucradas, los datos necesarios...etc. A partir de este punto se comenzará a programar el método.

Para ello, desde el editor de código, en este caso Spyder, se crea un archivo nuevo de Python en el interior de la carpeta del método correspondiente. A continuación, se crea una nueva función que será la que contenga el método en concreto. Las funciones están definidas por sus parámetros de entrada. En este contexto son las variables esenciales que intervienen en el método semiempírico a programar, por lo que estarán formadas por los datos que se extraen de la interfaz gráfica. Acto seguido, se programa la función siguiendo el flujo de trabajo adecuado, conformando finalmente el método.

Una de las partes más importantes es asegurar que el método funciona adecuadamente, y para ello existen dos opciones.

- Introducir manualmente los datos. Es decir, se puede ejecutar el código si, previamente a llamar a la función, se introducen el valor concreto que tomará cada una de las variables.



Esta opción, aunque puede parecer simple, al final se convierte en una opción engorrosa y poco efectiva ya que no está optimizada y se debe llamar manualmente a la función.

- Realizar un test al final del código. Es la opción escogida generalmente, se introduce al final del código un test que contiene el valor concreto de cada variable y la acción que se quiere realizar, como por ejemplo devolver ciertos datos deseados o hacer una gráfica, entre otras opciones. Es la elección de preferencia ya que siempre que se ejecute ese mismo archivo se devolverá la acción definida en el test, comprobando la eficacia del código de manera automática, sin embargo, en el momento en el que se llame a la función de manera externa, el test no se ejecutará y la función realizará los cálculos con los nuevos valores introducidos.

A continuación, se comentan ciertos detalles a tener en cuenta durante la programación de los métodos.

- Unidades. Uno de los detalles más importantes a la hora de programar el método, es conocer las unidades de los valores de entrada y de salida. La programación del método se ha realizado asumiendo que los parámetros de entrada se introducirán con las unidades del sistema internacional, acción que se incluye en el postprocesado, 7. Las resistencias se obtendrán en kN y las potencias en kW .
- Densidad y viscosidad cinemática. Tanto en el método de Amadeo como en el método de Holtrop-Mennen, no se especifica el valor que deben tomar dichos parámetros. Por ello, se han seleccionado los datos que proporciona la ITTC [14] del agua de mar a $15^{\circ}C$. Se han escogido estos datos ya que para otros procedimientos como la extrapolación de los datos obtenidos de ensayos con buque a escala a buque real son los que la ITTC proporciona, así como en otros procedimientos en los que se desconoce las propiedades exactas del agua. Por lo tanto, finalmente se han escogido: $\rho = 1025kg/m^3$ y $\nu = 1.189 * 10^6m^2/s$.
- Calidad. Es necesario asegurar la calidad de la función que se está programando. Para ello, es importante incluir un test final para observar los resultados obtenidos y compararlos con resultados esperados. Sin embargo, uno de los puntos más importantes es asegurar que la función se ejecute en todas las condiciones posibles con un flujo de trabajo adecuado en cada uno de los métodos y ninguna situación provoque errores inesperados. A continuación, se comentan algunas de las verificaciones que se han debido de tener en cuenta.
 - Velocidad nula. Existe la opción de que el usuario comience la predicción de resistencia al avance desde una velocidad igual a 0. El problema que causa esta situación es que en ocasiones la velocidad se encuentra en el denominador de una función, provocando que el denominador se anule y desemboque en un problema matemático.
 - Formulas empíricas. En ambos métodos existen formulas empíricas para la estimación de algún parámetro, como puede ser para la eslora en la flotación o la superficie mojada en el método de Amadeo o la superficie mojada en el método de Holtrop-Mennen. Es necesario comprobar que únicamente se ejecute la fórmula empírica cuando el parámetro de entrada lo determine y que el cálculo sea adecuado.
 - Ratios. Existen parámetros que varían en función de ciertos ratios. Es esencial comprobar que en función del ratio se calcula el valor del parámetro dependiente de manera adecuada.



A continuación, se muestran los valores obtenidos en el método de Holtrop-Mennen y en el método de Amadeo para distintos tipos de buques. Las gráficas mostradas no son aquellas que se obtendrán posteriormente en el capítulo 7, tras el postprocesado, son gráficas obtenidas al ejecutar los métodos desde la terminal de comandos.

- Amadeo.

En el caso del método de Amadeo, se han realizado las verificaciones del correcto funcionamiento del método a través de un ejemplo que se incluye en el propio artículo en el que Amadeo publica el método de predicción de resistencia al avance [7], y a través de diversos Trabajos de Fin de Grado, [15] y [16], en los que se ha utilizado el método de Amadeo para realizar la predicción de resistencia al avance. No se realizará un análisis comparativo de los datos obtenidos del método que se incluirá en *FreeCAD* con otro software, ya que el método de Amadeo no está incluido en otros software conocidos del mercado. A continuación se mostrará el ejemplo que propone Amadeo en el artículo, el resto de comprobaciones con variaciones del tipo de buque se incluyen en el anexo D.

En el ejemplo que propone Amadeo en el artículo, únicamente se aporta las dimensiones principales, el desplazamiento y la longitud de la protuberancia. Por lo que, tanto la eslora en la flotación como la superficie mojada deberán ser calculadas a través de las fórmulas empíricas propuestas en el propio artículo. Se ha decidido escoger aún así este ejemplo ya que de esta forma se pueden validar no solo la resistencia obtenida, sino los resultados arrojados por las fórmulas empíricas también.

Tabla 4.3: Dimensiones y características principales del buque 1_A para validación del método de Amadeo con eslora en la flotación, L_w , y superficie mojada, S_w , obtenidas mediante fórmulas empíricas.

Dimensiones	Valores
LPP	50 m
T	3.5 m
B	11.25 m
C_b	0.5034
Superficie mojada, S_w	601.35 m^2
Eslora en la flotación, L_w .	52.837 m
Volumen desplazado, ∇	991.219 m^3
Velocidades del buque, V	10.5 kN– 13.5 kN
Longitud de tobera, l	0 m
Diámetro propulsor, d	0 m
Longitud protuberancia bulbo, l_b	0.63 m
Coficiente del propulsor, η_p	0.6
Sea margin, %	0
Timón	No



Tabla 4.4: Resultados obtenidos mediante el método de Amadeo en el buque 1_A.

Speed [m/s]	CF * 10 ³	CR * 10 ³	CA * 10 ³	CT * 10 ³	Resistencia esperada [kN]	Resistance [kN]	Diferencia %
0.5	2.93	-0.14	0.99	3.78	-	0.08	-
1	2.61	-0.12	0.99	3.48	-	0.28	-
1.5	2.45	-0.10	0.99	3.34	-	0.61	-
2	2.34	-0.08	0.99	3.25	-	1.06	-
2.5	2.26	-0.05	0.99	3.20	-	1.63	-
3	2.20	-0.01	0.99	3.18	-	2.33	-
3.5	2.15	0.04	0.99	3.18	-	3.17	-
4	2.11	0.10	0.99	3.19	-	4.17	-
4.5	2.07	0.17	0.99	3.23	-	5.34	-
5	2.04	0.26	0.99	3.29	-	6.71	-
5.5	2.02	0.37	0.99	3.37	-	8.33	-
6	1.99	0.50	0.99	3.48	-	10.20	-
6.5	1.97	0.64	0.99	3.59	-	12.38	-
7	1.95	0.78	0.99	3.72	-	14.87	-
7.5	1.93	0.94	0.99	3.85	-	17.68	-
8	1.91	1.09	0.99	3.99	-	20.82	-
8.5	1.90	1.23	0.99	4.12	-	24.27	-
9	1.88	1.37	0.99	4.24	-	28.03	-
9.5	1.87	1.51	0.99	4.37	-	32.13	-
10	1.85	1.65	0.99	4.49	-	36.60	-
10.5	1.84	1.79	0.99	4.62	41.06	41.54	1.15%
11	1.83	1.95	0.99	4.77	46.89	47.08	0.40%
11.5	1.82	2.15	0.99	4.96	53.46	53.48	0.04%
12	1.81	2.41	0.99	5.21	60.87	61.15	0.47%
12.5	1.80	2.77	0.99	5.56	69.26	70.87	2.33%
13	1.79	3.33	0.99	6.11	78.78	84.20	6.88%

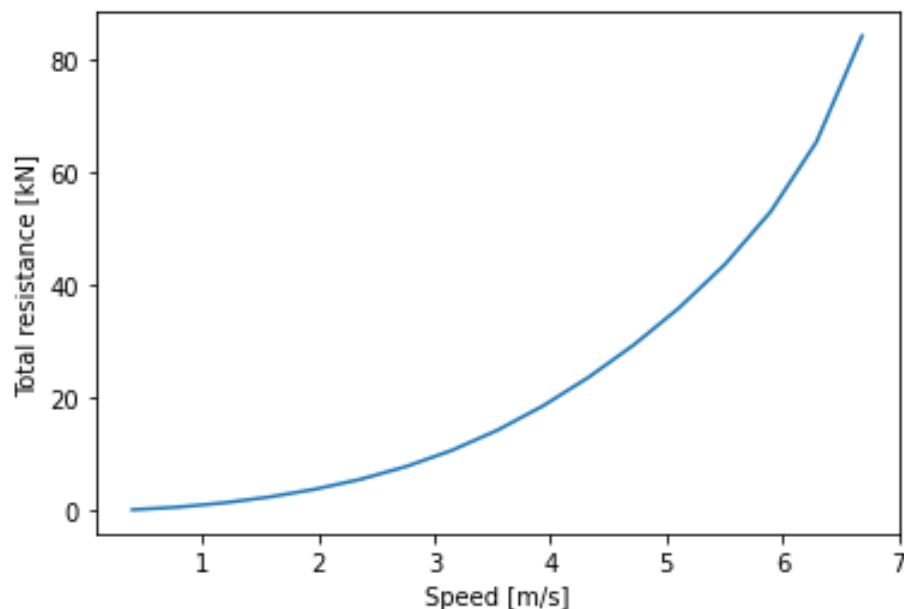


Figura 4.8: Gráfica resistencia vs velocidad obtenida mediante método de Amadeo en el buque 1_A.

Es importante recalcar que en la tabla 4.8 únicamente se puede comparar los valores de las resistencias para las velocidades entre 10.5 y 13 nudos. Esto es debido a que en [7] únicamente se presentan los resultados de las resistencias obtenidas para dichas velocidades. Además, la diferencia de resistencias obtenidas es debido a que tanto la eslora



en la flotación como la superficie mojada, tal y como se ha comentado anteriormente, han sido obtenidas a través de fórmulas empíricas. Sin embargo, se puede observar que los resultados arrojados por las fórmulas empíricas son válidos ya que las resistencias obtenidas se asemejan bastante, excepto a altas velocidades, que se obtienen diferencias mayores debido al aumento de la resistencia residual.

- Holtrop-Mennen.

En cuanto al método de Holtrop-Mennen se refiere, se realizará un análisis comparativo de los resultados obtenidos de diversos buques en *FreeCAD* comparado con *MAXSURF*. De esta manera, se mostrará a continuación una de las verificaciones realizadas y se indagará en las comprobaciones en el capítulo 8.

Tabla 4.5: Dimensiones y características principales del buque 1_H para validación del método de Holtrop-Mennen.

Dimensiones	Valores
Eslora entre perpendiculares, Lpp	44.49 m
Calado, T	4 m
Manga, B	10 m
Superficie mojada, Sw	572.547 m ²
Eslora en la flotación, Lw	44.879 m
Volumen desplazado, ∇	1007.8 m ³
Coficiente de Bloque, Cb	0.566
Coficiente de la maestra, Cm	0.934
Coficiente en la flotación, Cw	0.673
Semiángulo de entrada, iE	28 °
Posición del centro de flotación, xcb	-0.7645 m
Área transversal del bulbo, ABT	2.372 m ²
Área del espejo de popa, AT	0 m ²
Altura del cdg de ABT, hb	2.127 m
Velocidad del buque, u	3– 11 knot
Coficiente del propulsor, η _p	0.6
Sea margin, %	15
Áreas de los apéndices	0 m ²



Tabla 4.6: Resultados obtenidos mediante el método de Holtrop-Mennen en el buque 1_H

Speed [knots]	Resistance [kN]	CT * 10 ³	CF * 10 ³	CAPP	Cw * 10 ⁴	CB * 10 ³	CTR * 10 ³	CA * 10 ³
3.0	2.43	7.48	2.29	0.00	0.00	4.53	0.00	0.66
3.5	3.19	7.57	2.24	0.00	0.00	4.67	0.00	0.66
3.9	4.06	7.53	2.20	0.00	0.00	4.67	0.00	0.66
4.4	5.02	7.40	2.16	0.00	0.00	4.58	0.00	0.66
4.9	6.08	7.21	2.13	0.00	0.01	4.42	0.00	0.66
5.3	7.23	6.97	2.10	0.00	0.02	4.20	0.00	0.66
5.8	8.48	6.70	2.08	0.00	0.06	3.96	0.00	0.66
6.3	9.85	6.42	2.05	0.00	0.15	3.69	0.00	0.66
6.8	11.35	6.15	2.03	0.00	0.32	3.42	0.00	0.66
7.2	13.00	5.89	2.01	0.00	0.62	3.16	0.00	0.66
7.7	14.86	5.66	2.00	0.00	1.10	2.90	0.00	0.66
8.2	16.99	5.47	1.98	0.00	1.81	2.65	0.00	0.66
8.6	19.48	5.32	1.96	0.00	2.82	2.42	0.00	0.66
9.1	22.43	5.23	1.95	0.00	4.20	2.20	0.00	0.66
9.6	25.90	5.19	1.94	0.00	5.90	2.00	0.00	0.66
10.1	30.00	5.20	1.92	0.00	7.97	1.82	0.00	0.66
10.5	35.12	5.29	1.91	0.00	10.74	1.65	0.00	0.66
11.0	41.76	5.51	1.90	0.00	14.53	1.50	0.00	0.66

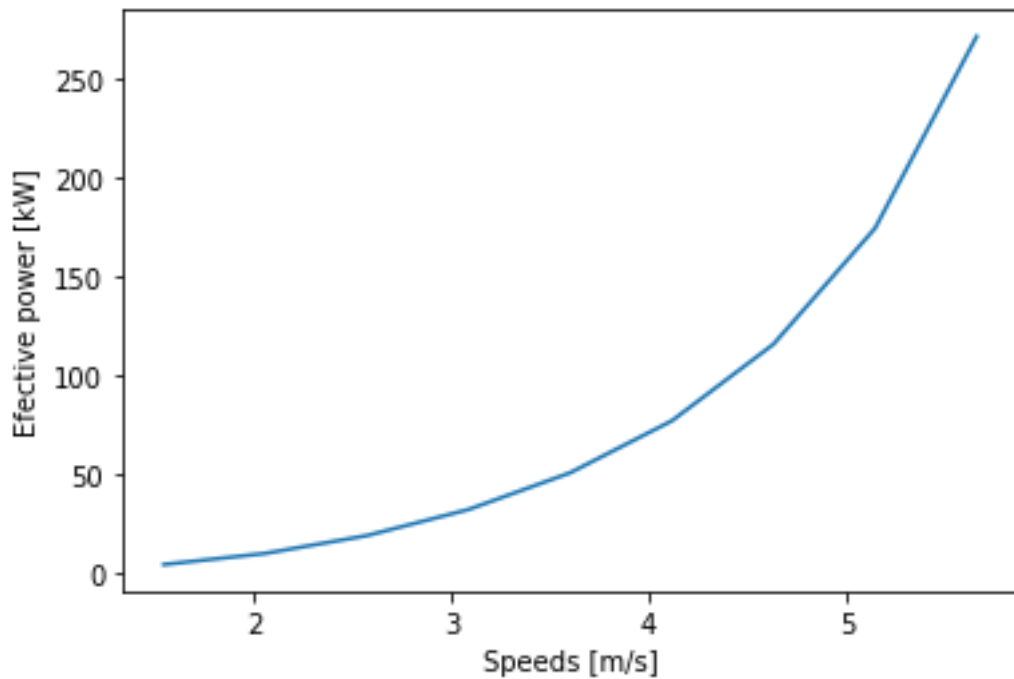


Figura 4.9: Gráfica resistencia vs velocidad obtenida mediante método de Holtrop-Mennen en el buque 1_H.



Capítulo 5

Predicción de resistencia al avance sin geometría.

En el transcurso de este capítulo se explicarán los pasos a seguir y el porqué de la implementación de la capacidad de realizar la predicción de resistencia al avance sin la necesidad de un modelo del buque en tres dimensiones.

5.1. Introducción.

Tal y como se ha comentado anteriormente, una de las grandes ventajas de utilizar un software que incluya la predicción de resistencia al avance, es la capacidad de realizar cálculos sobre un modelo del buque en tres dimensiones y que el propio programa obtenga las dimensiones y características principales que necesite para el método que se vaya a utilizar. Sin embargo, existen ocasiones en las que el usuario requiere la capacidad de obtener la resistencia al avance únicamente introduciendo los datos manualmente, por ejemplo porque se encuentra en una fase preliminar del proyecto y no cuenta con un modelo en tres dimensiones del buque o se requiere una primera y rápida estimación de la resistencia al avance.

En dichos supuestos, es de gran utilidad que el propio programa cuente con la alternativa de incluir manualmente los datos necesarios y obtener la resistencia al avance, sin la necesidad de recurrir a otros programas u hojas de cálculo para realizar esta acción.

5.2. Implicaciones de su implementación.

Siguiendo con las ventajas expuestas en la introducción, a parte de contar con la capacidad de poder introducir manualmente los datos sin la necesidad de un modelo en tres dimensiones, otra de las grandes ventajas es la comprobación y modificación de los datos obtenidos del modelo en tres dimensiones. Gracias a esta opción, una vez el usuario acceda al método que se desee utilizar podrá comprobar los valores que el programa ha obtenido del buque y en caso de que alguna función haya fallado o no haya realizado un cálculo adecuado, el usuario, con conocimiento de las dimensiones y características del buque que se está utilizando, podrá introducir los valores correctos.

Pese a haber realizado pruebas para validar y verificar las acciones introducidas que obtienen los diversos parámetros del buque, se recuerda que, por ejemplo, el semiángulo de entrada se calcula a través de una fórmula empírica, provocando que en ocasiones pudiese no entregar un valor adecuado y suficientemente aproximado. En otras ocasiones se pueden haber cometido



errores durante el modelado, originando que el programa no obtenga de manera adecuada las dimensiones requeridas. En todos estos supuestos, gracias a la opción de introducir los valores correspondientes manualmente, el proceso podrá continuar, ya que si se contase únicamente con la opción de obtener los valores de las acciones realizadas sobre el modelo en tres dimensiones, y dicho cálculo fallase o entregase un resultado incorrecto, el proceso podría llegar a paralizarse y no completar el cálculo de resistencia al avance.

Finalmente, tal y como se puede observar, la obtención de resistencia al avance mediante la inclusión de los datos de forma manual o a través de un modelo en tres dimensiones no son características individuales sin interacción entre ellas, sino que se complementan entre ellas para otorgar al usuario una mayor robustez en todos los casos y que el usuario siempre tenga la capacidad de obtener una predicción de resistencia al avance.

Además, su implementación es muy sencilla, ya que la única acción requerida es activar tanto los métodos como más adelante la presentación de resultados pese a que no haya un buque seleccionado.

Por último, no existen desventajas de su implementación per se. El único detalle, es que se permite acceder a los métodos sin un buque, por lo que el usuario podría pulsar en la herramienta sin querer o no entender del todo porqué se puede realizar la predicción de la resistencia al avance sin un buque, sin embargo, como se puede observar no es ningún detalle crítico que ponga en riesgo el funcionamiento del programa ni influya en los resultados o forma de trabajar.

5.3. Flujo de trabajo para su implementación.

Otorgar la capacidad de introducir los valores manualmente es mucho más sencillo y menos laborioso que obtener los valores a través de acciones y cálculos realizado sobre el modelo en tres dimensiones. En este caso, el requisito indispensable es que todos los valores necesarios para realizar el cálculo de predicción al avance, es decir, los parámetros de entrada de cada uno de los métodos, se encuentren instanciados en la ui del método correspondiente o almacenados en el programa. Es por este motivo, por lo que en el capítulo 4 se introdujeron todos los parámetro de entrada de los métodos, a excepción de las dimensiones principales, que se pueden modificar desde la ventana principal y se almacenan en *FreeCAD Ship*, por lo que se puede acceder a ellas posteriormente. De esta manera, el usuario podrá modificar e introducir manualmente los valores correspondientes al buque con el que desee realizar los cálculos de predicción de resistencia al avance. En las imágenes 4.7 y 4.6, se puede observar cómo verá el usuario la interfaz gráfica cuando acceda al cálculo de predicción de resistencia al avance sin modelo en tres dimensiones.



Capítulo 6

Predicción de resistencia al avance con geometría.

En el presente capítulo se explica del proceso seguido para lograr obtener las características principales de un modelo en tres dimensiones a través de *FreeCAD* para lograr realizar una predicción de resistencia al avance a través de un buque modelado en tres dimensiones, sin que el usuario tenga que introducir ninguna característica principal.

6.1. Introducción.

Dentro de las ventajas proporcionadas por un software de predicción de resistencia al avance que no puede ofrecer el cálculo manual o el cálculo a través de una hoja de cálculo, es la obtención de los datos necesarios para realizar la predicción de resistencia al avance a través de un modelo en tres dimensiones, convirtiéndose así en uno de los grandes atractivos de las nuevas funcionalidades. Para ello, el programa debe ser capaz de realizar los cálculos oportunos para obtener los datos necesarios del modelado del buque en función del método con el que se vaya a trabajar.

La obtención de las dimensiones y características necesarias del buque se realiza una vez seleccionado el método a utilizar y se muestran en la interfaz gráfica, ya que en función del método los requerimientos varían. Por lo que tras introducir el barco modelado en tres dimensiones al que se le desea realizar la predicción de resistencia al avance y seleccionar el método que se quiere utilizar, método de Amadeo o método de Holtrop-Mennen, se desplegará la ventana de interfaz de gráfica de usuario con los valores calculados a través del buque, insertados en la casilla correspondiente.

La mayoría de acciones que se deben realizar, se apoyan en herramientas implementadas anteriormente en el programa dentro de la sección del cálculo de hidrostáticas.

A continuación, se explica el tratamiento específico de los datos poder incluirlos adecuadamente en la interfaz gráfica y posteriormente las acciones y cálculos realizados para la obtención de cada característica.

6.2. Tratamiento de datos.

La mayoría de las características que se obtienen a partir de las funciones del cálculo de hidrostáticas contienen unidades. Es importante recordar que los valores que se introducen en la



interfaz gráfica deben estar en unidades del sistema internacional. Es por ello por lo que hay que realizar un tratamiento de los datos.

Existen dos opciones a la hora de obtener valores a partir del modelo en tres dimensiones.

- Contengan unidades. Tal y como se ha comentado anteriormente, prácticamente todas las características tienen dimensiones. Una vez se obtiene la característica, es importante asegurarse que se encuentra en el sistema internacional y extraer el valor. Para ello, se utiliza la siguiente sintaxis: “Nombre = nombrevariable.getValueAs(unidad deseada).Value”. En “Nombre”, se otorga el nombre que el usuario quiera y en “nombrevariable” el nombre con el que se creó el objeto de interacción en la interfaz gráfica. Además, se introduce en unidad deseada la unidad que se quiere obtener, por ejemplo:
 - Longitud = m
 - Área = m^2
 - Volumen = m^3
 - Ángulo = deg
- No contengan unidades. Se trata de características adimensionales, por ejemplo el coeficiente de bloque o el coeficiente de la maestra. Para este tipo de datos no se debe realizar ningún paso previo.

Una vez las variables tienen las unidades adecuadas para incluirlas en la interfaz gráfica y se ha extraído el valor, la sintaxis utilizada es la siguiente:
“self.form.nombrevariable.setText(str(Nombre))”.

De esta manera, se introducirá en la interfaz gráfica el valor con la dimensión correcta cuando lo obtenga del modelo en tres dimensiones.

6.3. Hidrostáticas.

Una de las ventajas de introducir las nuevas funcionalidades en *FreeCAD* y no en otros programas es que *FreeCAD* cuenta con un módulo específico para barcos en el que se incluyen numerosas herramientas, otorgando un amplio rango de cálculos posibles. En este caso, la funcionalidad más interesante es el cálculo de las hidrostáticas ya que dentro de esta herramienta se incluyen cálculos y acciones realizadas sobre el modelado en tres dimensiones del buque.

En el caso del cálculo de las hidrostáticas, todas las acciones sobre el modelado se hacen internamente, de manera que el usuario no tiene la capacidad de observar los datos obtenidos del buque, únicamente tiene la capacidad de introducir la información que no se pueden calcular a través del modelo en tres dimensiones, a diferencia del cálculo de la predicción de resistencia al avance en el que se incluye la capacidad de modificar los datos que el programa calcula a partir del modelo en tres dimensiones. Esta diferencia supone una variación en el lugar donde se incluye el código que define las acciones que el programa lleva a cabo.

Todas las herramientas internas que se utilizan en el cálculo de hidrostáticas se encuentran en la carpeta correspondiente a las hidrostáticas, en propio repositorio “Ship”. Gracias a encontrarse en el mismo repositorio, es posible utilizar estas funciones en otras funcionalidades. De esta manera, se aprovecharán ciertas herramientas de cálculo como por ejemplo el cálculo del desplazamiento del buque, ya incluidas en el programa.



6.4. Cálculos en *FreeCAD*.

Las nuevas acciones que realizará el programa se incluirán en el archivo “TaskPanel” que se encuentra en la carpeta correspondiente a cada método. El archivo “TaskPanel” se compone de numerosas funciones, sin embargo, en este apartado las funciones de mayor interés son aquellas en las que se interactúa con los datos y con la interfaz gráfica, se comentan a continuación.

- **Accept.** Tras seleccionar la herramienta que se quiere utilizar, en este caso se realizará con el método de Amadeo o al método de Holtrop-Mennen pero el flujo de trabajo es el mismo para las demás herramientas, se despliega una ventana de interacción con el usuario para que se especifiquen ciertos valores concretos necesarios. Una vez los valores se han especificado el siguiente paso es validarlos para que se desarrollen las actividades y cálculos pertinentes a través del botón “Ok” que se encuentra en la parte superior de las interfaces gráficas en 4.7 y 4.6. Esta función recopila las acciones y cálculos a ejecutar tras pulsar dicho botón. En otras funcionalidades como en el cálculo de hidrostáticas, todos los cálculos se realizan en este apartado, sin embargo, no será así en el presente módulo ya que se pretende mostrar en la interfaz de usuario inicialmente los datos obtenidos del buque.
- **SetupUi.** Esta función se ha comentado en capítulos anteriores ya que es la función en la que se deben incluir todos los elementos de interacción con datos que se han introducido en la interfaz gráfica de usuario.
- **initValues.** En esta última función, se recopilan los valores iniciales que aparecerán en la interfaz gráfica a la hora de acceder a la funcionalidad deseada y todas las acciones que se llevan a cabo para obtener dichos datos. Es decir, si para mostrar como valor inicial el desplazamiento es necesario calcularlo previamente, es en esta función donde se incluyen las acciones necesarias que debe realizar el programa. En este caso, se incluirán todas las acciones de cálculo sobre el modelo en tres dimensiones en esta función.

6.4.1. Amadeo.

A continuación, en la tabla 6.1 se incluye una recopilación de los datos que se deben introducir en el método de Amadeo y las diversas formas de obtenerlos.

Tabla 6.1: Relación de modos de obtener las características principales del método de Amadeo.

Datos necesarios	Modelado 3D	Usuario	Estimación por fórmulas
Eslora entre perpendiculares, Lpp	Sí	Sí	No
Calado, T	Sí	Sí	No
Manga, B	Sí	Sí	No
Coefficiente de bloque, Cb	Sí	Sí	No
Superficie mojada, Sw	Sí	Sí	Sí
Eslora el la flotación, Lw.	Sí	Sí	Sí
Volumen desplazado, ∇	Sí	Sí	No
Velocidades del buque, u	No	Sí	No
Coefficiente del propulsor, η_p	No	Sí	No
Margen de mar	No	Sí	No
Longitud de tobera, l	No	Sí	No
Diámetro propulsor, d	No	Sí	No
Longitud protuberancia bulbo, lb	Sí	Sí	No
Timón	No	Sí	No



A continuación, se comentarán las implicaciones de las diversas formas de obtener las dimensiones y características principales y porqué no todas las características se pueden obtener de todas las formas.

- **Usuario.** Todos los datos serán siempre obtenibles a través del usuario. Esto es debido a que se pueden incluir todas las casillas que se deseen en la interfaz gráfica y obtener a partir de ellas los datos. Es una característica interesante que se utilizará más adelante, por lo que no se detallará en este capítulo.
- **Estimación por fórmula.** La estimación por fórmulas se refiere a fórmulas empíricas que el autor proporciona en el método para la obtención de ciertas características. Estas fórmulas empíricas se incluyen en la propia programación del método semi-empírico. En el caso del método de Amadeo se proporcionan fórmulas para estimar la eslora en la flotación, L_w y la superficie mojada, S_w .
- **Modelado 3D.** Es la forma de obtención de las características y dimensiones principales de mayor interés en este capítulo. Son todas las características que se pueden obtener a través de cálculos y acciones sobre el modelo en tres dimensiones. Se puede observar como la existencia del timón, la longitud de la tobera y el diámetro del propulsor no es posible obtenerlos desde el modelado en 3D, esto es debido a que se trata de elementos que generalmente no se modelan. Por otra parte, las velocidades a las que se quieren realizar los cálculos de predicción de resistencia al avance únicamente son conocidas por el usuario, por lo que es obligatorio que este dato sea especificado.

Seguidamente, se explicarán las acciones a realizar para obtener cada una de las características principales y se incluirá el extracto del código que lo representa.

- **Dimensiones principales.** Las dimensiones principales es la característica más sencilla de obtener, esto es debido a los pasos previos que hay que realizar antes de los cálculos de predicción de resistencia al avance. En primer lugar, es caso importar o modificar las dimensiones de un buque modelo perteneciente al programa, el módulo "Ship" tiene la capacidad de leer las dimensiones principales del buque y mostrarlas por pantalla. Se muestra a continuación el código utilizado para extraer de la memoria de *FreeCAD* las dimensiones principales y poder almacenarlas en variables para utilizarlas posteriormente en el propio código.

```
1 L = self.ship.Length.getValueAs("m").Value
2 B = self.ship.Breadth.getValueAs("m").Value
3 T = self.ship.Draft.getValueAs("m").Value
```

- **Volumen desplazado, ∇ .** Para la obtención del volumen desplazado es necesario acudir a herramientas incluidas en el cálculo de hidrostáticas. En el cálculo de hidrostáticas existe una función que obtiene el desplazamiento del buque, por lo que será a través de esta función como se obtenga el volumen desplazado. Dicha función cuenta con las formas del buque, el calado, la escora y el trimado como parámetros de entrada.



```

1 disp, _, cb = Hydrostatics.displacement(self.ship,
2                                     self.ship.Draft,
3                                     Units.parseQuantity("0
4                                     deg"),
5                                     Units.parseQuantity("0
6                                     deg"))
7 vol = disp / Hydrostatics.DENS
8 V = vol.getValueAs("m^3").Value
9 self.form.volume.setText(str(V))

```

Se puede observar en los parámetros de entrada que tanto la escora como el trimado se ha fijado en cero, esto es debido a que en los métodos de predicción de resistencia al avance se supone que el buque no se encuentra escorado y se utilizará el calado medio, por lo que no será necesario utilizar el trimado.

Internamente, para obtener el desplazamiento, el programa sitúa un plano horizontal que corte al buque a una altura igual al calado, en este punto realiza un corte para guardar el sólido de la parte inferior, la obra viva del buque, y por último calcula el volumen del sólido. Finalmente, el desplazamiento obtenido se divide entre la densidad del agua del mar para adquirir el volumen desplazado.

La última línea de código, la línea 7, es en la que se guarda el valor obtenido tras realizar los cálculos en el modelo en tres dimensiones en la interfaz gráfica cuando se inicie.

- Coeficiente de bloque, Cb.

A través de la función utilizada para obtener el volumen desplazado se obtiene también el coeficiente de bloque ya que se relacionan entre sí a través de las dimensiones principales.

```

1 disp, _, cb = Hydrostatics.displacement(self.ship,
2                                     self.ship.Draft,
3                                     Units.parseQuantity("0 deg"),
4                                     Units.parseQuantity("0 deg"))
5
6 self.form.Cb.setText(str(cb))

```

- Superficie mojada, Sw.

En cuanto a la superficie mojada se refiere se utilizará una función integrada en las herramientas de cálculo de las hidrostáticas que tiene como parámetro de entrada las formas del buque, el calado, la escora y el trimado.

```

1 sw = Hydrostatics.wettedArea(self.ship.Shape.copy(), self.ship.Draft,
2                               Uits.parseQuantity("0 deg"),
3                               Units.parseQuantity("0 deg"))
4
5 Sw = sw.getValueAs("m^2").Value
6 self.form.Sw.setText(str(Sw))

```

La herramienta trabaja de manera similar a la que realiza el cálculo del volumen desplazado del buque. Inicialmente se coloca un plano horizontal que corte al buque a una altura igual al calado, se realiza un corte y se almacena la parte inferior, la obra viva. Una vez se obtiene la obra viva se realizan los cálculos pertinentes para obtener la superficie del sólido que será igual a la superficie mojada.

- Eslora en la flotación, Lw. Para la obtención de la eslora en la flotación es necesario realizar ciertos cálculos a través de valores obtenidos de una función que define el área en la



flotación. Dicha función devuelve el área en la flotación, el coeficiente en la flotación y el plano que queda definido por la línea de flotación, se trabajará a través de dicho plano. De nuevo, como en las demás funciones utilizadas hasta el momento los parámetros de entrada son las formas del buque, el calado, la escora y el trimado. Se muestra a continuación el código utilizado y posteriormente se explicarán las acciones realizadas.

```

1 area, cf, f = Hydrostatics.floatingArea(self.ship, self.ship.Draft,
2                                     Units.parseQuantity("0
3                                     Units.parseQuantity("0
4                                     deg")),
5                                     Units.parseQuantity("0
6                                     deg"))
7
8 bbox = f.BoundingBox
9 lw = Units.Quantity(bbox.XMax - bbox.XMin, Units.Length)
10 Lw = lw.getValueAs("m").Value
11
12 self.form.Lw.setText(str(Lw))

```

En primer lugar, la función sitúa un plano horizontal posicionado a una altura igual al calado del buque, se realiza un corte por un plano situado a la altura del calado, perpendicular a la superficie del buque y, a diferencia de las demás funciones, guarda el plano con el que se ha realizado el corte. El plano almacenado contiene el contorno exacto del buque en la línea de la flotación. Utilizando una herramienta denominada "BoundingBox" se encaja el contorno de la flotación del buque en un contenedor, quedando definida tanto la manga como la eslora de dicho contenedor. Por último, se obtiene la eslora del contenedor en el que se alberga el contorno de flotación del buque, dicha eslora será la eslora en la flotación.

- Longitud de la protuberancia del bulbo, lb.

De la misma manera que se ha realizado en la eslora en la flotación para la obtención de la longitud de la protuberancia del bulbo será necesario llevar a cabo ciertos cálculos y acciones con las dimensiones obtenidas. En este caso, no hay una función específica que sea capaz de realizar esta acción.

Por definición, la longitud de la protuberancia del bulbo es la longitud de protuberancia que excede de la perpendicular de proa. Para obtenerlo informáticamente la longitud de protuberancia será la diferencia entre la eslora entre perpendiculares y la longitud total del buque de la parte sumergida.

```

1 shape, _ = Hydrostatics.placeShipShape(self.ship.Shape.copy(),
2                                     self.ship.Draft,
3                                     Units.parseQuantity("0
4                                     Units.parseQuantity("0
5                                     deg")),
6                                     Units.parseQuantity("0
7                                     deg"))
8
9 shape = Hydrostatics.getUnderwaterSide(shape)
10 bbox = shape.BoundingBox
11
12 prot = Units.Quantity(bbox.XMax - bbox.XMin, Units.Length)
13 prot = prot - self.ship.Length
14
15 if prot < 0: prot = 0
16 prot = Units.Quantity(prot, Units.Length)
17 Prot = prot.getValueAs("m").Value
18 self.form.protuberance.setText(str(Prot))

```

Para ello se sitúa un plano horizontal en el buque a una altura igual al calado correspondiente, se realiza un corte y se guarda la superficie inferior, es decir, la obra viva, a través



de la herramienta “BoundingBox”, de la misma manera que en la obtención de la eslora en la flotación. Por último se compara la eslora obtenida de la obra viva con la eslora entre perpendiculares. En el supuesto en que la diferencia sea negativa la longitud de la protuberancia será cero.

En este punto, al acceder a la herramienta del método de Amadeo se mostrará en la interfaz gráfica todos los datos obtenidos del buque que se haya introducido. A continuación se muestra una imagen de la interfaz gráfica con el buque modelo “Series 60 from Iowa University”, uno de los modelos de ejemplo existentes en *FreeCAD*.

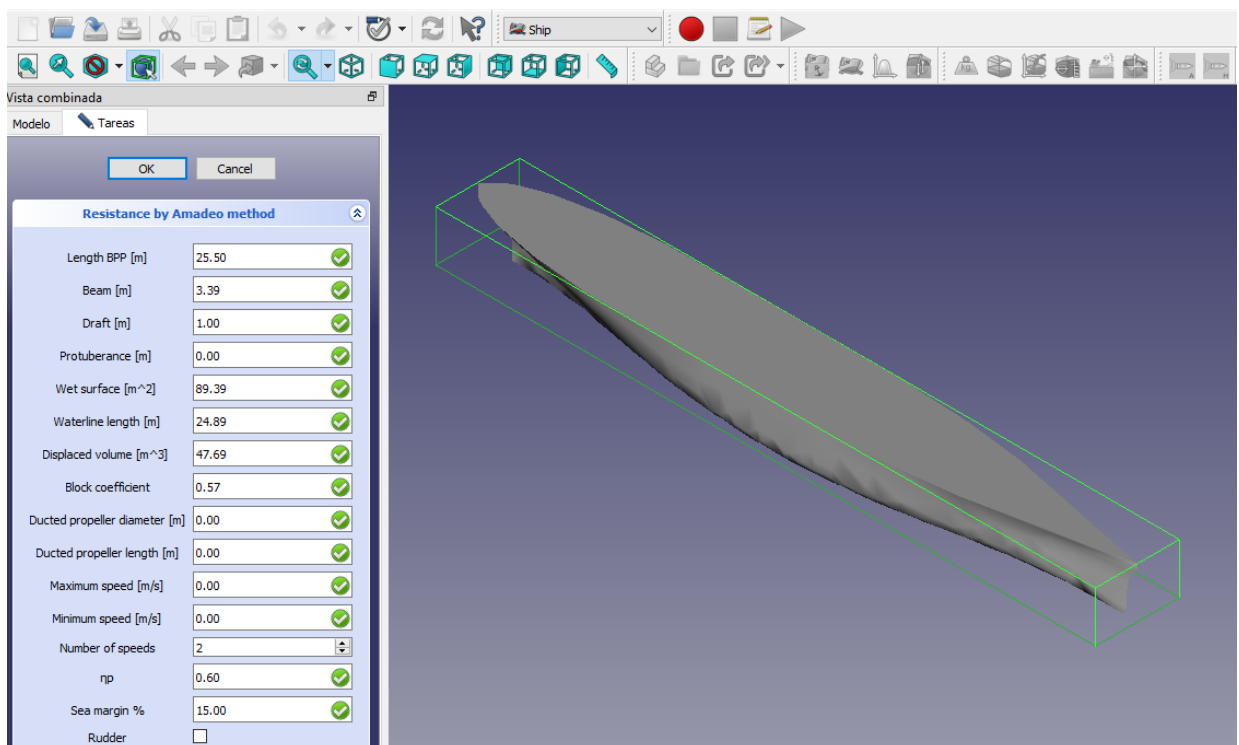


Figura 6.1: Interfaz gráfica del método de Amadeo tras la obtención de datos del modelo 3D.

Se puede observar que el coeficiente del propulsor tiene un valor distinto de cero, esto no es debido a que se haya obtenido del modelo, ya que tal y como se ha comentado al inicio no es posible, sino que se le ha otorgado un valor inicial para todos los casos. De esta manera, al acceder al método siempre aparecerá con un valor de 0.6 y el usuario, en el caso de que lo desee, podrá modificarlo.¹ Además, el margen de mar también tiene un valor predeterminado, introducido en porcentaje, que el usuario también podrá modificar. Se han otorgado estos valores ya que son los valores típicos de estas dos características, aún así, pueden variar, y es por ello por lo que se le ha otorgado al usuario la capacidad de modificarlos.

¹En el anexo F se incluyen más detalles a cerca del funcionamiento del coeficiente del propulsor en función del valor obtenido.



6.4.2. Holtrop-Mennen.

En el método de Holtrop-Mennen la manera de proceder y el flujo de trabajo será el mismo que en el método de Amadeo salvando las particularidades de cada uno de los métodos.

A continuación, en la tabla 6.1 se incluye una recopilación de los datos que se deben introducir en el método de Holtrop-Mennen y las diversas formas de obtenerlos.

Tabla 6.2: Relación de modos de obtener las características principales del método de Holtrop-Mennen

Datos necesarios	Modelado 3D	Usuario	Estimación por fórmulas
Eslora entre perpendiculares, Lpp	Sí	Sí	No
Calado, T	Sí	Sí	No
Manga, B	Sí	Sí	No
Superficie mojada, Sw	Sí	Sí	Sí
Eslora en la flotación, Lw	Sí	Sí	No
Volumen desplazado, ∇	Sí	Sí	No
Coefficiente de Bloque, Cb	Sí	Sí	No
Coefficiente de la maestra, Cm	Sí	Sí	No
Coefficiente en la flotación, Cw	Sí	Sí	No
Semiángulo de entrada, iE	No	Sí	Sí
Posición del centro de flotación, xcb	Sí	Sí	No
Área transversal del bulbo, ABT	No	Sí	No
Área del espejo de popa, AT	No	Sí	No
Altura del cdg de ABT, hb	No	Sí	No
Velocidad del buque, u	No	Sí	No
Coefficiente del propulsor, η_p	No	Sí	No
Margen de mar	No	Sí	No
Áreas de los apéndices	No	Sí	No

- Usuario. Al igual que ocurría en el método de Amadeo, es posible obtener todas las dimensiones y características principales a través del usuario.
- Fórmulas empíricas. En este caso, en el método de Holtrop-Mennen el autor propone fórmulas empíricas para estimar la superficie mojada y el semiángulo de entrada. La estimación de la superficie mojada se incluirá en la programación del método semi-empírico, sin embargo, la estimación del semiángulo de entrada es especialmente interesante que calcule inicialmente y se muestre por pantalla en la interfaz gráfica de usuario para validar que se ha obtenido de manera adecuada.
- Modelado 3D. Todas las características que se podían obtener a partir del modelo en tres dimensiones en el método de Amadeo se puede utilizar también para el método de Holtrop-Mennen así como alguna característica nueva propia del método de Holtrop-Mennen. En cuanto al área de los apéndices, generalmente no se modelan por lo que lo más sencillo es que lo introduzca el usuario. De forma similar, el cálculo del área transversal del bulbo y del espejo de popa sumergido es un cálculo bastante complicado que en caso de que el usuario aporta este datos se simplifica mucho la operación.

A continuación, de la misma forma que en el método de Amadeo se comentarán las acciones a realizar para obtener cada una de las características principales y se incluirá el extracto del código que lo representa.



- Mismas operaciones que en el método de Holtrop-Mennen. En pos de no repetir lo explicado previamente en el método de Amadeo, se indicarán las características que se obtiene de la misma forma.
 - Dimensiones principales
 - Superficie mojada, S_w .
 - Eslora en la flotación, L_w .
 - Volumen desplazado, ∇
 - Coeficiente de bloque, C_b

- Coeficiente de la maestra, C_m .

Para la obtención del coeficiente de la maestra se utilizará una función procedente del cálculo de hidrostáticas cuyos parámetro de entrada son las formas del buque y el calado. En este caso no es necesario introducir ni la escora ni el trimado ya que el coeficiente de la maestra se obtendrá con el buque adrizado.

```

1 cm = Hydrostatics.mainFrameCoeff(self.ship)
2
3 self.form.Cm.setText(str(cm))

```

De la misma forma que otras de las muchas funciones utilizadas hasta el momento se realiza un corte del buque por un plano horizontal y se almacena la obra viva del buque. En este momento se encaja el sólido en un contenedor obteniendo así todas las dimensiones principales del sólido, de esta forma se obtiene el área en la maestra y acto seguido el coeficiente de la maestra.

- Coeficiente en la flotación, C_w .

El coeficiente de flotación se obtiene a partir de la función utilizada en el método de Amadeo para obtener la eslora en la flotación. El flujo de trabajo interno de la función es el mismo que el comentado en el método de Amadeo.

```

1 area, cf, f = Hydrostatics.floatingArea(self.ship, self.ship.Draft,
2                                     Units.parseQuantity("0
3                                     deg"),
4                                     Units.parseQuantity("0
5                                     deg"))
6 self.form.Cf.setText(str(cf))

```

- Posición del centro de flotación, x_{cb} .

En cuanto a la posición del centro de flotación existe una función en el cálculo de hidrostáticas que devuelve dicho valor como un porcentaje de la eslora total medido a partir de la cuaderna maestra, siendo el eje positivo hacia proa.

```

1 disp, Vector, cb = Hydrostatics.displacement(self.ship,
2                                             self.ship.Draft,
3                                             Units.parseQuantity("0
4                                             deg"),
5                                             Units.parseQuantity("0
6                                             deg"))
7 xcb = Units.Quantity(Vector[0], Units.Length)

```



```

7 Xcb = xcb.getValueAs("m").Value
8
9 self.form.xcb.setText(str(Xcb))

```

Tras obtener el sólido que representa la obra viva del buque a través de un corte por un plano a una altura igual al calado, se obtiene el centro de gravedad del buque. En este caso la coordenada necesaria es la referida a las x.

- Semiángulo de entrada, iE .

En cuanto al semiángulo de entrada es una variable considerablemente difícil de obtener y que sea fiable en todas las ocasiones, es por ello que se ha optado por introducir en la interfaz gráfica el valor estimado que se propone en el método de Holtrop-Mennen a través de una fórmula empírica. Se ha decidido incluirlo en el inicio ya que de esta forma se propone una primera aproximación al usuario que en el supuesto en el que la aproximación no sea buena podrá modificar el valor.

En este caso, el primer paso a realizar es obtener todas las dimensiones que intervienen en la fórmula empírica con las unidades correspondientes, además habrá que hallar el valor de las variables que intervengan en la fórmula del semiángulo de entrada y también se obtengan a través de una fórmula, como es el caso del parámetro que refleja la longitud del recorrido, LR.

A continuación, en la imagen 6.2, obtenida de los apuntes de la asignatura de: “Métodos de proyecto en Arquitectura Naval”, en la E.T.S.I.N [1], el semi-ángulo de entrada.

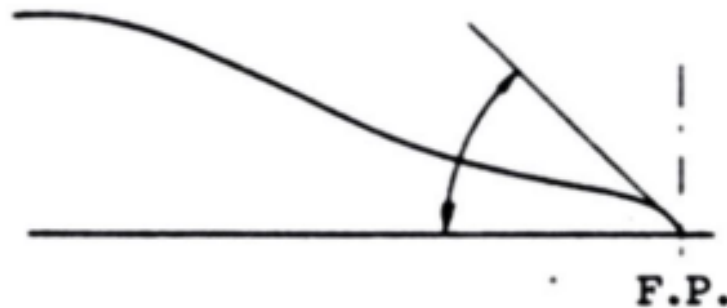


Figura 6.2: Semi-ángulo de entrada, obtenida de [1]

```

1 cp = cb / cm
2 Lw = lw.getValueAs("m").Value
3 XCB = xcb.getValueAs("m").Value
4 Bie = self.ship.Breadth.getValueAs("m").Value
5 Vie = vol.getValueAs("m^3").Value
6
7 Lr = Lw * (1 - cp + (0.06 * cp * XCB) / (4 * cp - 1))
8
9 try:
10     iE = 1 + 89 * np.exp(- (Lw / Bie) ** 0.80856 * (1 - cf) ** 0.30484
11                          * (1 - cp - 0.0225 * XCB) ** 0.6367 * (Lr /
12                          Bie) **
13                          0.34574 * ((100 * Vie) / (Lw ** 3)) **
14                          0.16302)

```



```

14 except ZeroDivisionError:
15     msg = App.Qt.translate(
16         "ship_console",
17         "ZeroDivisionError: Null ship floating area found during the"
18         " floating area computation!")
19     App.Console.PrintError(msg + '\n')
20
21     iE = 0.0
22
23 self.form.iE.setText(str(iE))

```

A modo de cortafuegos y evitar errores, se ha incluido que en caso de que se obtenga un error de división entre 0, que sería debido a un error anterior calculando la eslora en la flotación, se entregue un ángulo de entrada igual a cero pero no se paralice el proceso y el usuario pueda seguir realizando la predicción de resistencia al avance.

De la misma manera que en método de Amadeo, se utilizará el modelo “Series 60 from Iowa University” incluido en *FreeCAD* para mostrar la interfaz gráfica del método de Holtrop-Mennen tras introducir todas las acciones de cálculo sobre el modelo en tres dimensiones.

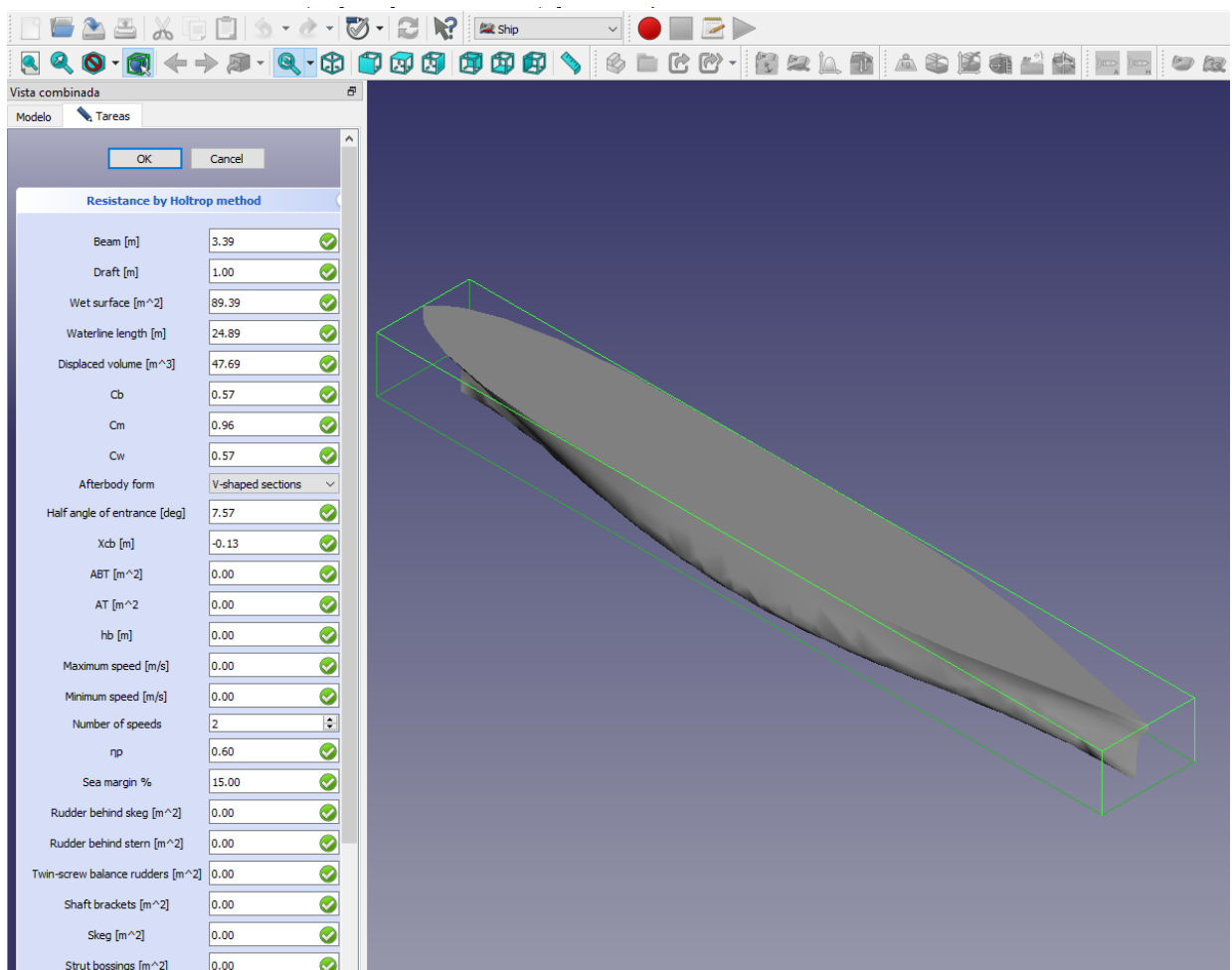


Figura 6.3: Interfaz gráfica del método de Holtrop-Mennen tras la obtención de datos del modelo 3D.

Los valores que no se alcanzan a ver en la imagen 6.3 de la interfaz gráfica del método de



Holtrop-Mennen son todos igual a cero ya que son las casillas correspondientes al área de los diferentes apéndices posibles.

6.5. Verificación de las funciones.

Las funciones introducidas anteriormente, tanto en el método de Holtrop-Mennen como en el método de Amadeo son funciones que actúan sobre sólidos, no tienen rangos de aplicación como los propios métodos. Es por ello que se puede verificar el funcionamiento de las acciones introducidas con cualquier buque en ambos métodos.

A continuación se comprueba el correcto funcionamiento de las nuevas acciones introducidas en ambos métodos. Para ello, se obtendrán, desde las nuevas funcionalidades las dimensiones del buque "Series 60 from Iowa University", modelo de ejemplo incluido en *FreeCAD*, así como de un buque rápido². Además, se realizaron verificaciones adicionales con un sólido introducido a modo de buque³.

Tabla 6.3: Buque para verificación de funciones introducidas en *FreeCAD*

Dimensiones principales	Valores
Eslora entre perpendiculares, L_{pp}	75.00 m
Manga, b	13.24 m
Calado	6 m
Volumen desplazado	3375 m^3
Área en la maestra, A_m	64 m^2
Área en la flotación, A_w	800 m^2
Longitud de protuberancia	0 m
Posición del centro de flotación, x_{cb}	-2.7 m
Coefficiente del propulsor, η_p	0.6
Margen de mar	15 %

²En el anexo E se muestran los planos del buque

³En el anexo E se incluyen explicaciones y detalles de las demás verificaciones realizadas

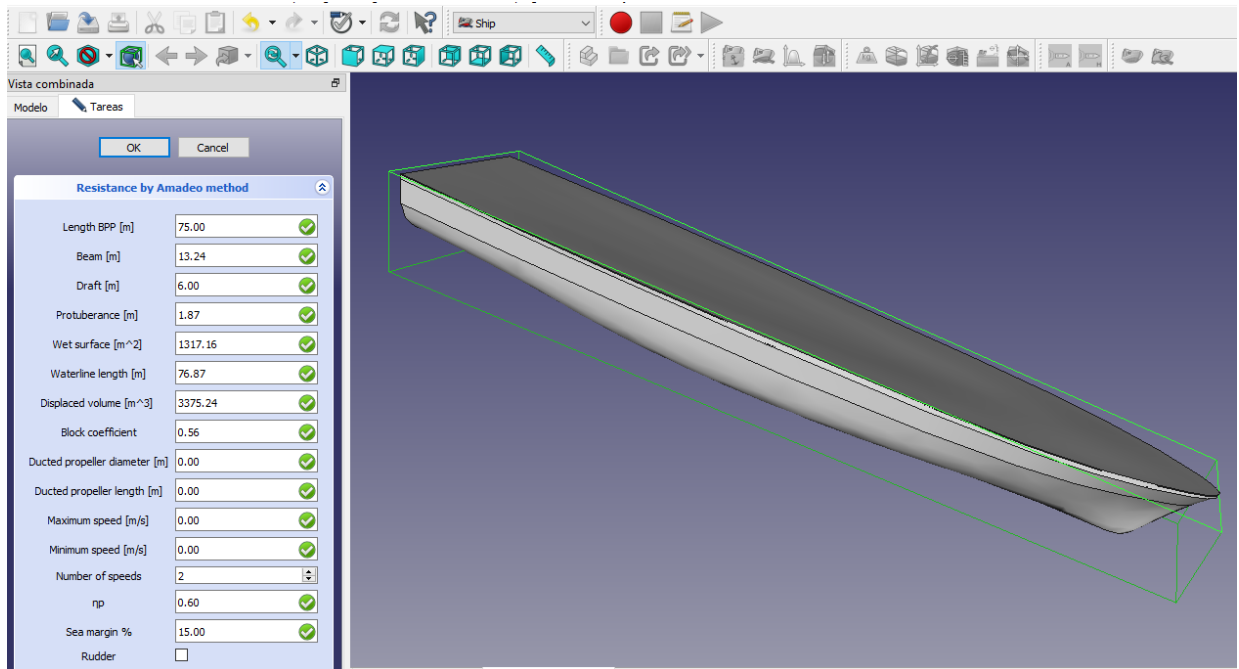


Figura 6.4: Interfaz gráfica del método de Amadeo con buque rápido.

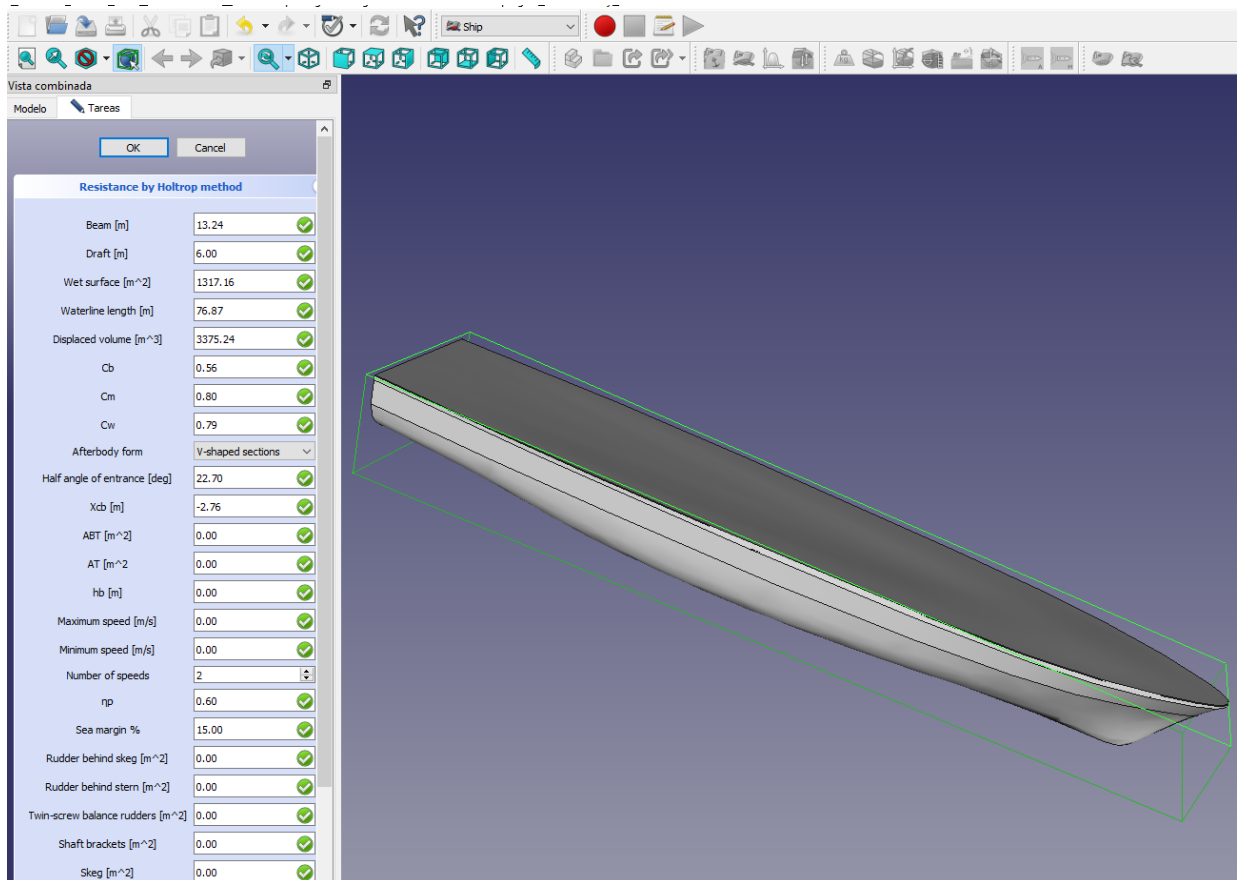


Figura 6.5: Interfaz gráfica del método de Holtrop-Mennen con buque rápido.





Capítulo 7

Postprocesado en *FreeCAD*.

En el presente capítulo se procede a explicar la última etapa necesaria para incluir al completo las nuevas funcionalidades en *FreeCAD* y que sean totalmente funcionales.

7.1. Introducción.

En los capítulos anteriores, se ha realizado un recorrido a través de los pasos necesarios a seguir para crear las nuevas funcionalidades e introducirlas en *FreeCAD* para la obtención de la predicción de resistencia al avance.

El flujo de trabajo definido hasta el momento comienza con la selección del método que se desee utilizar para obtener la predicción de resistencia al avance. Seguidamente, el programa realiza las acciones pertinentes sobre el modelo en tres dimensiones, en caso de existir, y presenta los resultados obtenidos en la interfaz gráfica del método correspondiente. En caso contrario, se presentará la interfaz gráfica limpia para que el usuario introduzca los valores deseados. A continuación, el usuario verifica los valores obtenidos, los modifica si procede, y especifica los valores de las características que no pueden ser obtenidos del modelo en tres dimensiones, como por ejemplo, la velocidad. A partir de este punto, comienzan las acciones englobadas dentro del postprocesado. El siguiente paso consta de pulsar el botón “Accept” que se encuentra en la parte superior izquierda de la interfaz gráfica y finalmente se presentan los resultados obtenidos.

Tras pulsar el botón “Accept” se realizan internamente los cálculos pertinentes para obtener la resistencia al avance y posteriormente, partiendo de los valores obtenidos, se presentan los resultados.

7.2. Obtención de la resistencia al avance.

Una vez el usuario ha validado los valores de las características del buque con los que se desea realizar la predicción de resistencia al avance se deben extraer dichos valores, y realizar una cierta manipulación de los datos para, finalmente, obtener la resistencia al avance. Para ello, se realiza un llamado a la función que describe el método semi-empírico, habiendo importado previamente el archivo del método correspondiente en el archivo TaskPanel.

Todas las acciones realizadas en el código tras pulsar el botón “Accept” se incluyen en el archivo TaskPanel correspondiente a cada método, en el espacio de definición de la función accept.



7.2.1. Extracción de los valores.

En primer lugar, se extraerán los valores previamente introducidos en la interfaz gráfica. Para ello, se utiliza la sintaxis que se muestra a continuación para cada uno de los objetos de interacción de la interfaz gráfica:

```
“nombre = Units.parseQuantity(Locale.fromString(self.form.nombrevariable.text())).Value”
```

Siendo:

- Nombre. El nombre con el que se desee almacenar el valor de la variable.
- Nombrevariable. El nombre que se le otorgó al espacio editable durante la creación de la interfaz gráfica.

Tras la utilización de dicha sintaxis se almacenará en la variable creada el valor que se haya obtenido de la característica correspondiente. Dichos valores se utilizarán a continuación para obtener la predicción de la resistencia al avance.

Los valores finales obtenidos en el paso anterior se introducen en el método correspondiente programado en 4 para obtener la resistencia al avance.

De esta manera, quedarán almacenados los distintos parámetros calculados en el modelo semi-empírico correspondiente y podrán presentarse a continuación.

7.3. Presentación de los resultados.

Finalmente, se debe programar la presentación adecuada de los resultados. En este caso, en ambos métodos, la presentación de resultados constará de varias gráficas y una hoja de cálculo que contiene los valores calculados.

7.3.1. Gráficas.

Es de vital importancia realizar una presentación clara de los resultados. En los métodos semi-empíricos, el principal valor obtenido es la resistencia al avance del buque y la potencia en función de la velocidad.

Por ello, gracias a realizar una gráfica con estos valores, el usuario podrá observar en un simple vistazo la situación de la resistencia al avance del buque estudiado y la potencia necesaria. Sin embargo, para realizar comparaciones entre buques y tener un conocimiento más exacto de la situación de las diversas resistencias que constituyen la resistencia total, es importante conocer el valor de los coeficientes resistivos. Debido a este motivo, se ha decidido crear otro gráfico que contenga la variación de los coeficientes que constituyen las diferentes resistencias en función de la velocidad.

Por lo tanto, se ha decidido crear tres gráficas, una de ellas que contenga la variación de la resistencia al avance con la velocidad, otra que contenga la variación de la potencia, tanto la potencia efectiva como la potencia al freno, y por último, otra que contenga la variación de los coeficientes de las resistencias.

En primer lugar, es necesario obtener los valores que se quieren representar. Para ello, la función programada que realiza los cálculos del método semi-empírico correspondiente debe entregar los parámetros requeridos por la gráfica.

A continuación, se programa la creación de las gráficas ¹ incluyendo en cada una de ellas los

¹En el anexo F se explica la programación tanto de las gráficas como de la hoja de cálculo y se incluyen los códigos correspondientes.



valores correspondientes que se deseen graficar.

Cabe mencionar que la programación, tanto de la creación de las gráficas como de la hoja de cálculo, se realiza en el archivo "Plotaux". Archivo que se encuentra en la carpeta correspondiente a cada uno de los métodos.

Se muestran a continuación las gráficas obtenidas a través del método de Amadeo en el buque 1_A, 7.1, 7.2, y 7.1 , cuyas dimensiones principales se muestran en la tabla 4.3.

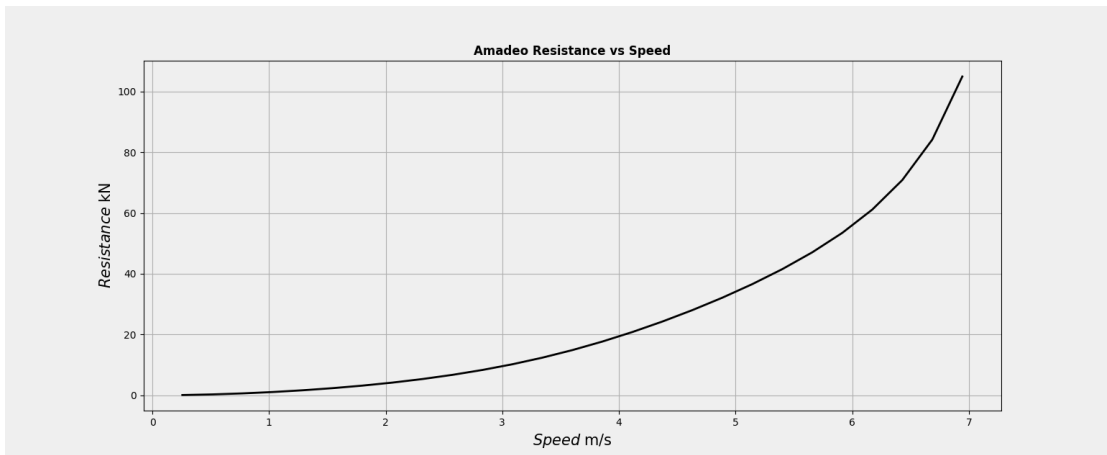


Figura 7.1: Resistencia obtenida a través del método de Amadeo en el buque 1A.

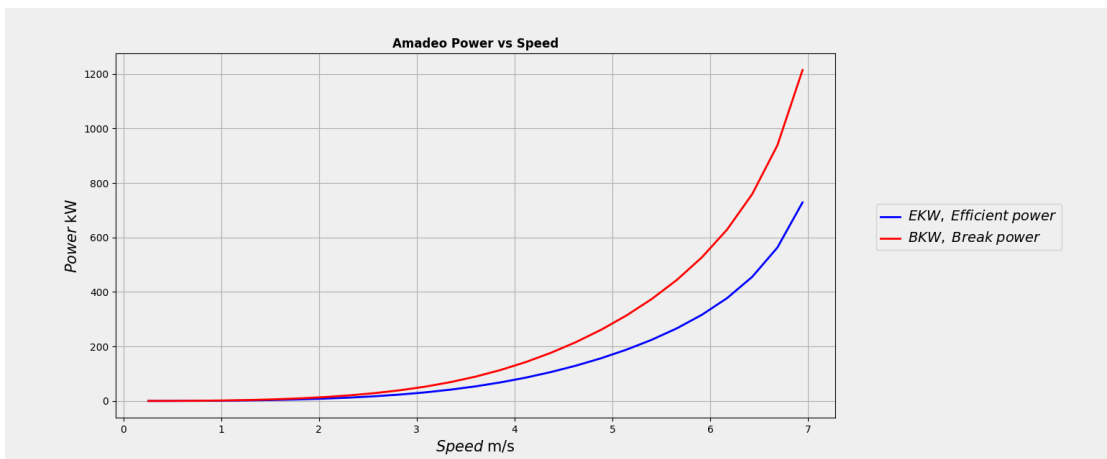


Figura 7.2: Potencias obtenidas a través del método de Amadeo en el buque 1A.



Figura 7.3: Coeficientes resistivos obtenidos a través del método de Amadeo en el buque 1A.

Se muestran a continuación las gráficas obtenidas a través del método de Amadeo en el buque 1_H, 7.4, 7.5, y 7.4, cuyas dimensiones principales se muestran en la tabla 4.5.

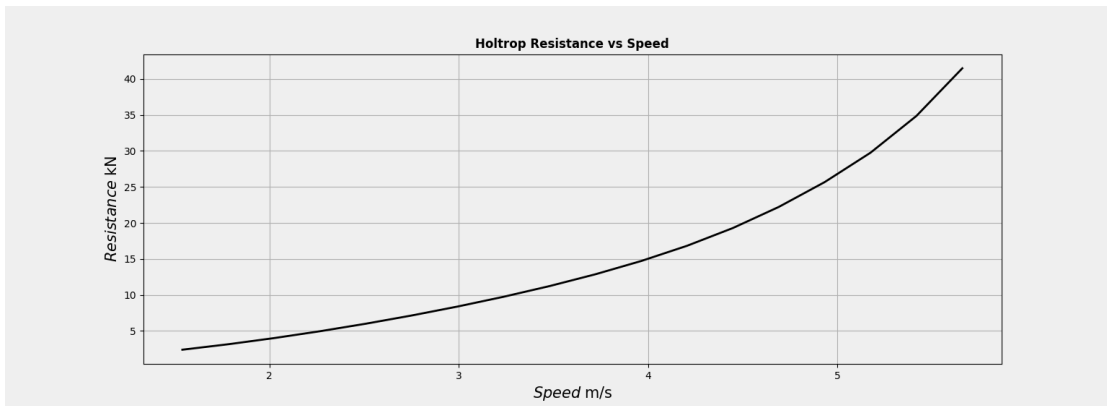


Figura 7.4: Resistencia obtenida a través del método de Holtrop-Mennen en el buque 1H.

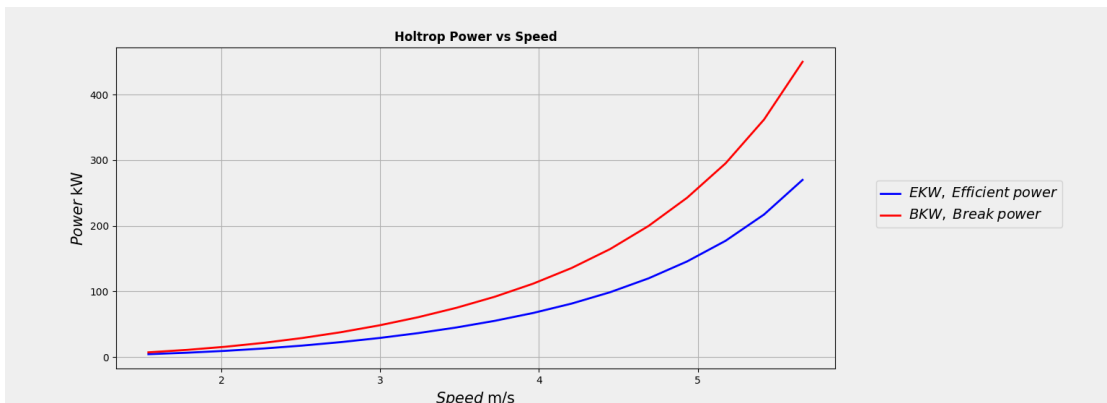


Figura 7.5: Potencias obtenidas a través del método de Holtrop-Mennen en el buque 1H.

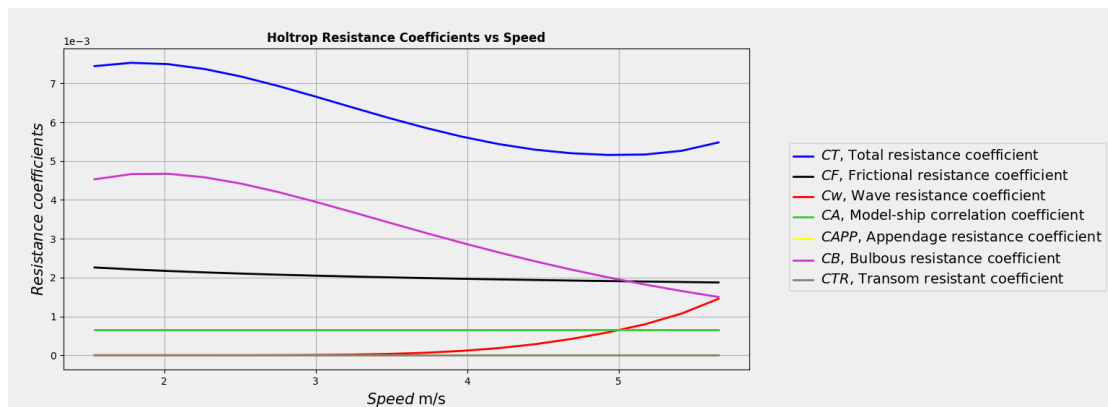


Figura 7.6: Coeficientes resistivos obtenidos a través del método de Holtrop-Mennen en el buque 1H.

En las gráficas mostradas anteriormente se puede observar que en la gráfica de los coeficientes el eje en el que se muestra el valor de los coeficientes se ha decidido mostrar en notación científica debido a los valores de los coeficientes, obteniendo una gráfica mucho más limpia. Además, en la leyenda se incluye el nombre completo de los coeficientes, facilitando la comprensión de la gráfica al usuario.

7.3.2. Hoja de cálculo.

Además de mostrar los resultados mediante gráficas se ha decidido almacenar los valores en una hoja de cálculo.

Generar una hoja de cálculo con los valores obtenidos otorga al usuario la capacidad de poder visualizar los valores exactos tanto de la resistencia total como de los diferentes coeficientes para las velocidades especificadas. Asimismo, permite pegar los valores obtenidos de otras predicciones en una de las hojas de cálculo que se generan en *FreeCAD* al realizar la predicción y poder realizar comparaciones de los valores obtenidos. También se permite realizar esta acción en otros programas, es decir, se pueden copiar los valores obtenidos de la predicción de resistencia al avance en *FreeCAD*, pegarlos en otro programa, como por ejemplo Excel, y realizar los cálculos y comparaciones que desee el usuario.

A continuación, en las imágenes 7.7 y 7.6, se muestran la ventana principal de *FreeCAD* con las hojas de cálculo que se generan al obtener la predicción de resistencia al avance.

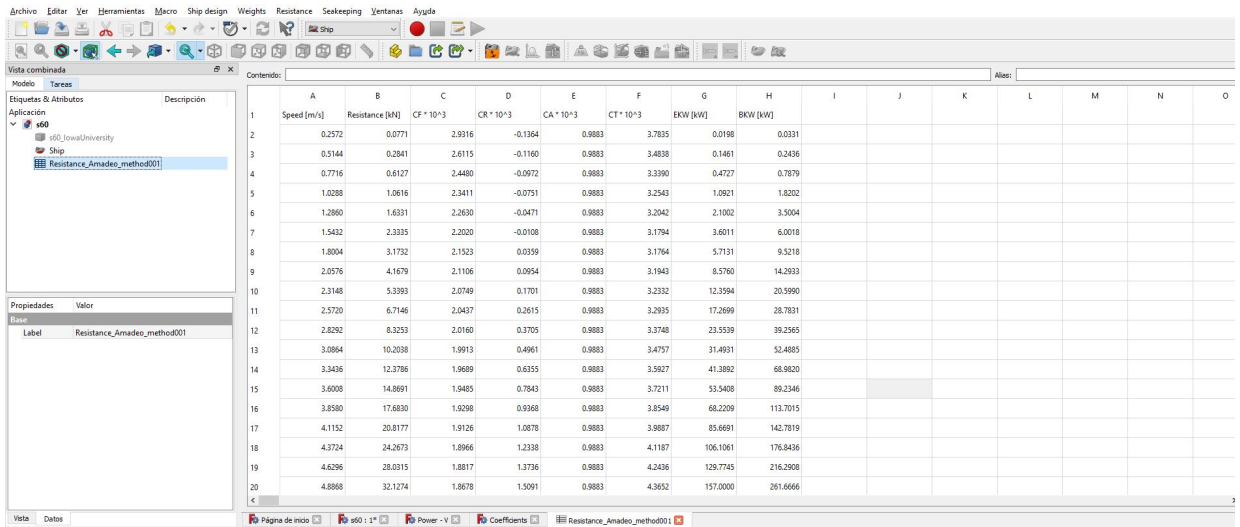


Figura 7.7: Ventana principal de *FreeCAD* con hoja de cálculo obtenidas a través del método de Amadeo en el buque 1A.

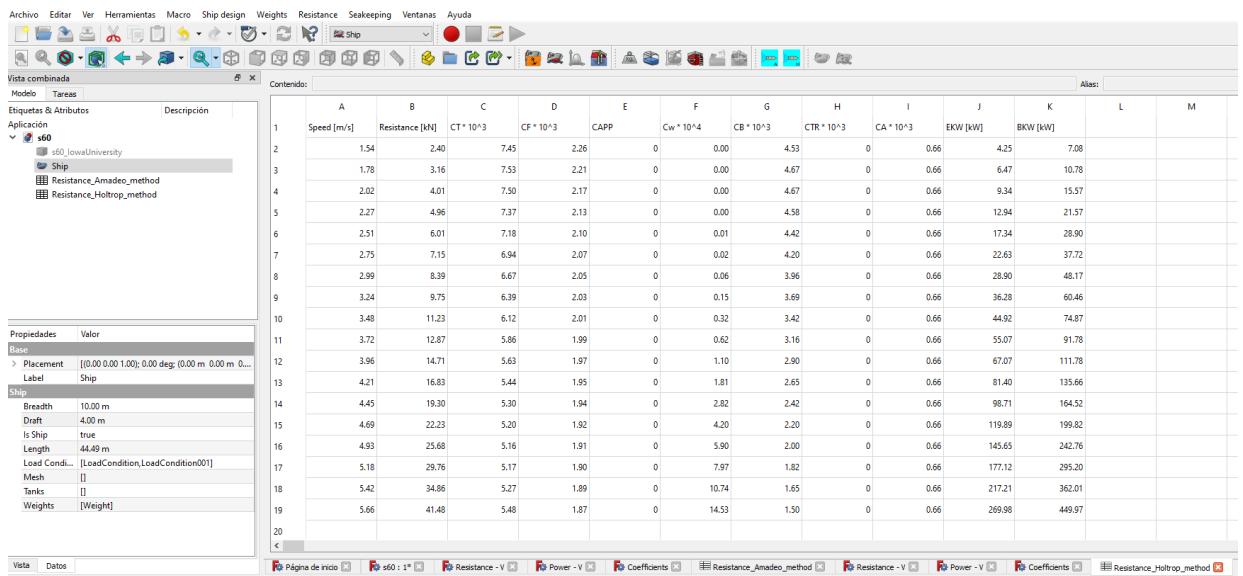


Figura 7.8: Ventana principal de *FreeCAD* con hoja de cálculo obtenidas a través del método de Holtrop-Mennen en el buque 1H.

Tras realizar el postprocesado, las nuevas funcionalidades quedan totalmente incluidas y accesibles en el software. El último paso, incluido en el anexo B, trata de la integración de la rama en la que se han incluido dichas funcionalidades en el proyecto en producción, para que cualquier usuario pueda acceder a las herramientas. Además, se realizarán las pruebas pertinentes para verificar el correcto funcionamiento de las herramientas en todos los supuestos.



Capítulo 8

Comparación con *MAXSURF* de las resistencias obtenidas.

En el presente capítulo se realiza una comparación de las resistencias obtenidas del método de Holtrop-Mennen a través de *FreeCAD* y *MAXSURF Resistance*, con la intención de validar la nueva funcionalidad integrada en *FreeCAD*.

8.1. Introducción.

Para la realizar una comprobación adecuada del método de Holtrop-Mennen se comparará la resistencia obtenida con un ejemplo numérico propuesto en [8] y, posteriormente, se estudiarán diversos tipos de buques dentro de los rangos que ofrece el método semi-empírico.

Siguiendo con lo comentado anteriormente, se propone el estudio de los buques específicos que se incluyen en los rangos de aplicación del método de Holtrop-Mennen. Para ello, se han escogido buques utilizados tanto en trabajos de fin de grado, TFG, como en trabajos de fin de máster, TFM, y proyectos fin de carrera, PFC, todos ellos de la Escuela Técnica Superior de Ingenieros Navales de Madrid, (E.T.S.I.N). Además, se realizará el estudio del buque S60 de la universidad de Iowa, buque que tiene la gran ventaja de estar muy estudiado, por lo que las formas son muy concretas y claras, así como ser uno de los buques ejemplos incluido en *FreeCAD*. La elección se debe a la cantidad de datos estudiados y comprobados que aportan estos recursos, convirtiéndose en una base de datos maravillosa.

La comparación con *Maxsurf Resistance* se realizará en varias etapas:

- Comparación mediante tabla de resultados.
- Gráficas comparativas de las resistencias y potencias efectivas, obtenidas a través de *FreeCAD* y *MAXSURF*.
- Gráfica de los coeficientes resistivos, obtenida a través de *FreeCAD*.

8.2. Ejemplo numérico.

A continuación, en la tabla 8.1, se muestran los valores propuestos en el ejemplo numérico que se encuentra en el artículo del método de Holtrop-Mennen, [8]. El único detalle a tener en cuenta es que en el artículo se obtiene únicamente la resistencia al avance para la velocidad de servicio, es decir, 25 knots.



Tabla 8.1: Dimensiones y características principales del buque de ejemplo que se propone en el método de Holtrop-Mennen.

Dimensiones	Valores	Unidades
Eslora entre perpendiculares, Lpp	200	m
Calado, T	10	m
Manga, B	32	m
Superficie mojada, Sw	7381.45	m ²
Eslora en la floración, Lw	205	m
Volumen desplazado, ∇	37500	m ³
Coeficiente de Bloque, Cb	0.571	-
Coeficiente de la maestra, Cm	0.980	-
Coeficiente en la flotación, Cf	0.75	-
Semiángulo de entrada, iE	12.08	°
Posición del centro de flotación, xcb	1.5	m
Área transversal del bulbo, ABT	20	m ²
Área del espejo de popa, AT	0	m ²
Altura del cdg de ABT, hb	4	m
Velocidad del buque, u	25	knot
Coeficiente propulsivo	0.6	-
Margen de mar	15.00	
Área de los apéndices	50.00	m ²

A diferencia de los demás casos, en esta ocasión, se compararán resistencias en vez de coeficientes. Es debido a los resultados obtenidos en el método de Holtrop-Mennen, que en este caso son resistencias.

Tabla 8.2: Comparación resistencias obtenidas entre el ejemplo en el método de Holtrop-Mennen y en *FreeCAD Ship*.

	Speed [m/s]	Resistance [kN]	RF [kN]	RAPP [kN]	RW [kN]	RB [kN]	RTR [kN]	RA [kN]
Obtenido	12.86	1809.716808	872.12	8.86	575.44	0.0026	0	220
Esperado	12.86	1793.26	869.63	8.83	557.11	0.049	0	221
Diferencia	-	0.92 %	0.29 %	0.34 %	3.29 %	-94.69 %	0.00 %	-0.45 %

Tal y como se puede observar, en la tabla 8.2, la diferencia en las resistencias totales obtenidas es mínima. Además, esta es debida a los cambios que se produjeron en las ecuaciones del método de Holtrop-Mennen en los distintos análisis. Las mayores diferencias se observan en aquellas ecuaciones con mayores cambios, por ejemplo, en la resistencia debida al bulbo se puede apreciar una gran diferencia. Por ello, debido a mínima diferencia entre resistencias totales obtenidas, se puede concluir que las ecuaciones se han introducido correctamente y corresponden al método de Holtrop-Mennen. A continuación, se realizarán comparaciones con las resistencias obtenidas del software *MAXSURF Resistance* y del software de *FreeCAD Ship* con diversos buques.

8.3. Análisis de los diversos buques.

Siguiendo con lo comentado en la introducción, dentro de los buques específicos que se incluyen en los rangos de aplicación del método de Holtrop-Mennen, se procederá a analizar un buque petrolero, un buque portacontenedores, un buque granelero y un buque ro-ro. A continuación, en



los diversos apartados se incluyen los análisis de todos ellos. Es importante recalcar que para una adecuada comparación con *MAXSURF*, en este caso referido a las potencias obtenidas, en *FreeCAD* se introducirá un margen de mar nulo, ya que *MAXSURF* no cuenta con dicha capacidad.

8.3.1. Series 60 de la universidad de Iowa.

Las dimensiones del buque son, en concreto, las obtenidas de *FreeCAD* tras validar las funciones de obtención de dimensiones y características principales de un modelo en tres dimensiones.

A continuación, en la tabla 8.3 se muestran los valores correspondientes al buque S60.

Tabla 8.3: Dimensiones y características principales del buque S60.

Dimensiones	Valores	Unidades
Eslora entre perpendiculares, Lpp	25.5	m
Calado, T	1	m
Manga, B	3.93	m
Superficie mojada, Sw	89.39	m ²
Eslora en la floración, Lw	24.89	m
Volumen desplazado, ∇	47.69	m ³
Coficiente de Bloque, Cb	0.57	-
Coficiente de la maestra, Cm	0.96	-
Coficiente en la flotación, Cw	0.57	-
Semiángulo de entrada, iE	7.57	º
Posición del centro de flotación, xcb	-0.129	m
Área transversal del bulbo, ABT	0	m ²
Área del espejo de popa, AT	0	m ²
Altura del cdg de ABT, hb	0	m
Velocidad del buque, u	0 - 8	knot
Coficiente propulsivo η_p	0.6	-
Margen de mar	0.00	
Área de los apéndices	0.00	m ²

A continuación, en la tabla 8.4, se muestran los rangos de aplicación que propone el método de Holtrop-Mennen para los cargueros, es la operación que más se puede asimilar al buque S60, y los valores actuales.

Tabla 8.4: Rangos de aplicación del método de Holtrop-Mennen para cargueros, y valores actuales.

Dimensión	Rango de aplicación	Valor actual
Cp	0.56 - 0.75	0.59
L/B	5.3 - 8	7.52
B/T	2.4 - 4	3.39
Fnmax	0.30	0.26

A continuación, se muestran las gráficas obtenidas, tanto en *FreeCAD* como en *MAXSURF*, 8.1, 8.2, y 8.3. Además, se incluye también en la tabla 8.5, los valores obtenidos en ambos programas y la diferencia en porcentaje.



Tabla 8.5: Comparación de valores obtenidos de MAXSURF frente a FreeCAD para el buque S60.

MAXSURF Holtrop			FreeCAD Holtrop				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
0	–	–	0	0	0	0	0
0.5	0.01	0.00	0.26	0.5	0.01	0.00	-4.62 %
1	0.05	0.03	0.51	1	0.05	0.03	-4.51 %
1.5	0.11	0.08	0.77	1.5	0.10	0.09	-4.45 %
2	0.19	0.19	1.03	2	0.18	0.21	-4.41 %
2.5	0.28	0.36	1.29	2.5	0.27	0.40	-4.37 %
3	0.40	0.61	1.54	3	0.38	0.68	-4.34 %
3.5	0.53	0.96	1.80	3.5	0.51	1.05	-4.29 %
4	0.68	1.41	2.06	4	0.65	1.55	-4.22 %
4.5	0.86	1.98	2.31	4.5	0.82	2.19	-4.11 %
5	1.05	2.71	2.57	5	1.01	3.00	-3.99 %
5.5	1.28	3.63	2.83	5.5	1.23	4.01	-3.89 %
6	1.55	4.77	3.09	6	1.49	5.28	-3.87 %
6.5	1.86	6.20	3.34	6.5	1.78	6.86	-3.98 %
7	2.22	7.98	3.60	7	2.12	8.80	-4.18 %
7.5	2.65	10.22	3.86	7.5	2.53	11.23	-4.55 %
8	3.18	13.08	4.12	8	3.01	14.26	-5.30 %

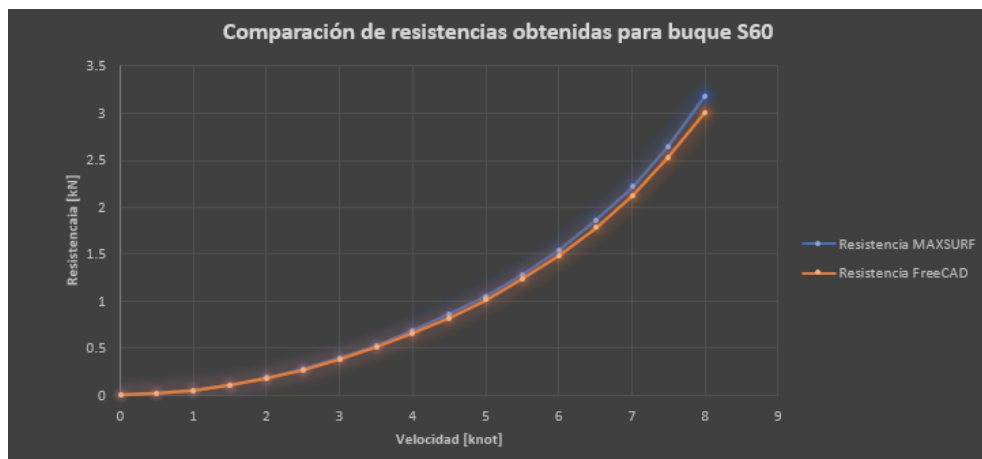


Figura 8.1: Gráfica comparativa de resistencia vs velocidad obtenida para el buque S60 a través de MAXSURF y FreeCAD.

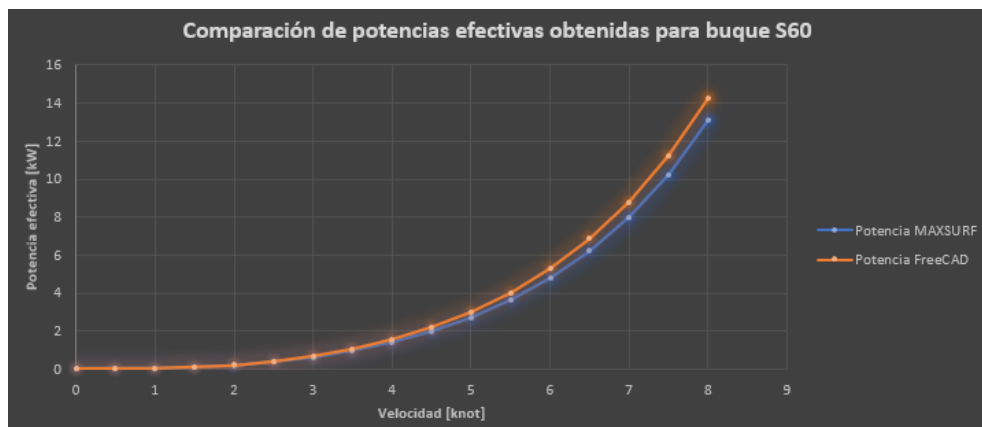


Figura 8.2: Gráfica comparativa de potencia vs velocidad obtenida para el buque S60 a través de MAXSURF y FreeCAD.

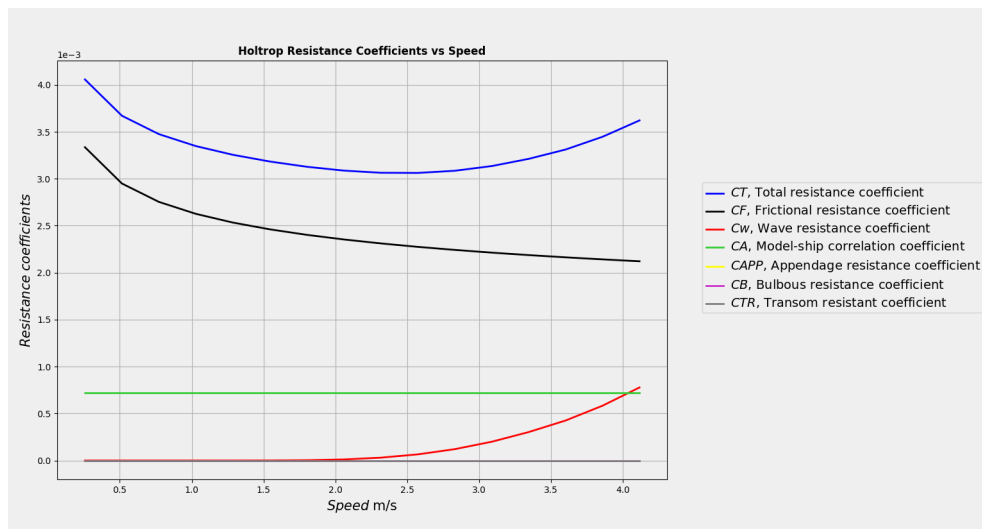


Figura 8.3: Gráfica de coeficientes resistivos vs velocidad obtenida para el buque S60 a través de *FreeCAD*.

8.3.2. Petrolero de productos de 40.000 TPM.

Los valores de las dimensiones y características básicas del petrolero que se utilizarán para obtener la resistencia al avance asociada han sido obtenidos de [2], TFG de la E.T.S.I.N.

A continuación, en la tabla 8.6, se muestran las dimensiones del buque que se introducirán tanto en *FreeCAD* como en *MAXSURF*.

Tabla 8.6: Dimensiones y características principales del buque petrolero de productos.

Dimensiones	Valores	Unidades
Eslora entre perpendiculares, Lpp	173.8	m
Calado, T	11.1	m
Manga, B	32	m
Superficie mojada, Sw	7857.049	m ²
Eslora en la floración, Lw	178.4	m
Volumen desplazado, ∇	46268	m ³
Coefficiente de Bloque, Cb	0.73	-
Coefficiente de la maestra, Cm	0.955	-
Coefficiente en la flotación, Cw	0.922	-
Semiángulo de entrada, iE	40.37	º
Posición del centro de flotación, xcb	-2.41	m
Área transversal del bulbo, ABT	8.11	m ²
Área del espejo de popa, AT	0	m ²
Altura del cdg de ABT, hb	5.37	m
Velocidad del buque, u	0 - 14	knot
Coefficiente propulsivo η_p	0.6	-
Margen de mar	0.00	-
Área de los apéndices	0.00	m ²

A continuación, en la imagen 8.4, se muestran las formas del buque petrolero de 40.000 TPM, obtenidas de [2].



Figura 8.4: Formas del petrolero de 40.000 TPM obtenidas del TFG [2].

A continuación, en la tabla 8.7, se muestran los rangos de aplicación que propone el método de Holtrop-Mennen para los buques petroleros y los valores actuales.

Tabla 8.7: Rango de aplicación del método de Holtrop-Mennen para buques petroleros y valores actuales.

Dimensión	Rango de aplicación	Valor actual
Cp	0.73 - 0.85	0.764
L/B	5.1 - 7.1	5.43
B/T	2.4 - 3.2	2.88
F _{nmax}	0.24	0.174

A continuación se muestran las gráficas obtenidas, tanto en *FreeCAD*, , como en *MAXSURF*, 8.6, 8.5, y 8.7. Además, se incluye también en la tabla 8.8, los valores obtenidos en ambos programas y la diferencia en porcentaje.



Tabla 8.8: Comparación de valores obtenidos de *MAXSURF* frente a *FreeCAD* para un buque petrolero de productos de 40.000 TPM.

MAXSURF Holtrop			FreeCAD Holtrop				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	Resistance [kN]	BKW [kW]	%
0	—	—	0	0	0	0	0
0.5	0.93	0.24	0.257	0.5	0.91	0.24	-1.82 %
1	3.40	1.75	0.514	1	3.33	1.71	-1.82 %
1.5	7.26	5.60	0.772	1.5	7.12	5.50	-1.82 %
2	12.45	12.81	1.029	2	12.22	12.58	-1.82 %
2.5	18.94	24.36	1.286	2.5	18.59	23.91	-1.81 %
3	26.69	41.19	1.543	3	26.21	40.44	-1.81 %
3.5	35.68	64.24	1.800	3.5	35.03	63.07	-1.81 %
4	45.89	94.43	2.058	4	45.06	92.72	-1.81 %
4.5	57.31	132.66	2.315	4.5	56.27	130.25	-1.81 %
5	69.91	179.83	2.572	5	68.65	176.56	-1.81 %
5.5	83.70	236.81	2.829	5.5	82.18	232.52	-1.80 %
6	98.65	304.49	3.086	6	96.87	298.97	-1.80 %
6.5	114.76	383.73	3.344	6.5	112.69	376.78	-1.80 %
7	132.02	475.41	3.601	7	129.64	466.81	-1.80 %
7.5	150.43	580.40	3.858	7.5	147.72	569.91	-1.80 %
8	169.99	699.60	4.115	8	166.93	686.96	-1.80 %
8.5	190.72	833.97	4.372	8.5	187.30	818.93	-1.80 %
9	212.65	984.58	4.630	9	208.84	966.85	-1.79 %
9.5	235.85	1152.65	4.887	9.5	231.64	1131.96	-1.79 %
10	260.41	1339.67	5.144	10	255.78	1315.74	-1.78 %
10.5	286.49	1547.52	5.401	10.5	281.43	1520.06	-1.77 %
11	314.29	1778.55	5.658	11	308.80	1747.29	-1.75 %
11.5	344.11	2035.77	5.916	11.5	338.17	2000.45	-1.73 %
12	376.29	2322.99	6.173	12	369.91	2283.39	-1.70 %
12.5	411.31	2644.96	6.430	12.5	404.49	2600.90	-1.66 %
13	449.71	3007.54	6.687	13	442.47	2958.88	-1.61 %
13.5	492.13	3417.87	6.944	13.5	484.50	3364.53	-1.55 %
14	539.34	3884.43	7.202	14	531.33	3826.42	-1.48 %

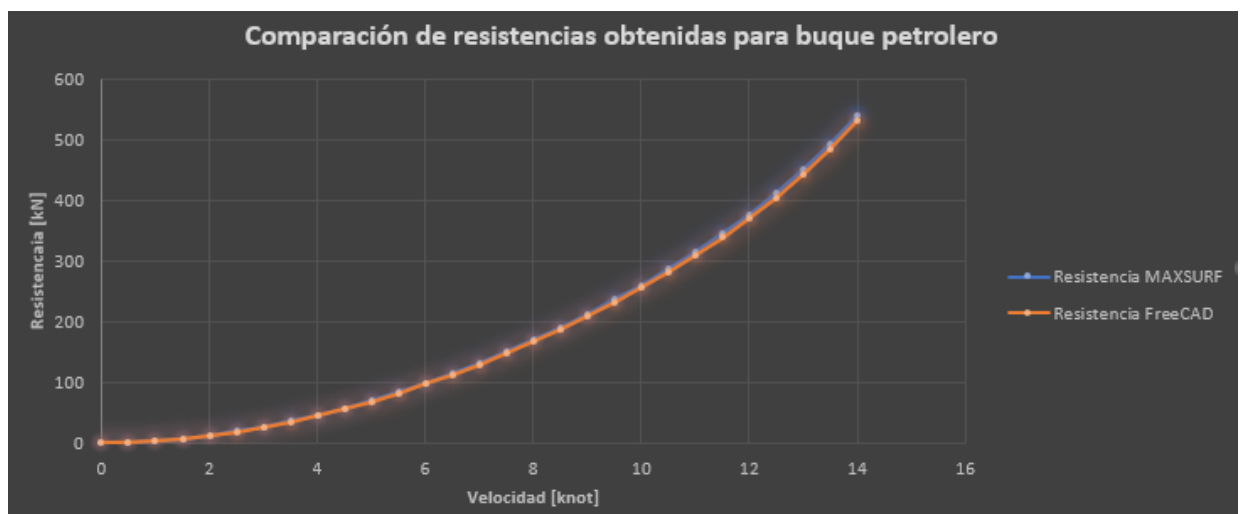


Figura 8.5: Gráfica comparativa de resistencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *MAXSURF* y *FreeCAD*.

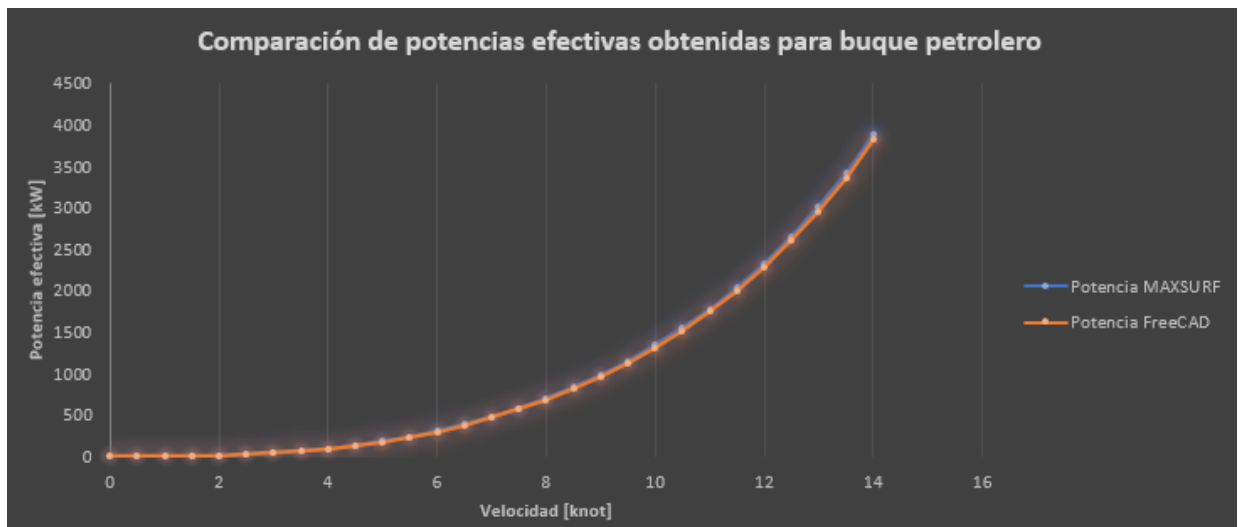


Figura 8.6: Gráfica comparativa de potencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *MAXSURF* y *FreeCAD*.



Figura 8.7: Gráfica coeficientes resistivos vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *FreeCAD*.

8.3.3. Buque granelero de 35.000TPM.

Para realizar la comparación de valores obtenidos entre *MAXSURF* y *FreeCAD* con un granelero se ha escogido [3], TFM de la E.T.S.I.N.

A continuación, en la tabla 8.9 se muestran las dimensiones del granelero.



Tabla 8.9: Dimensiones y características principales de un buque granelero de 35.000 TPM.

Dimensiones	Valores	Unidades
Eslora entre perpendiculares, L_{pp}	174.5	m
Calado, T	10.3	m
Manga, B	30	m
Superficie mojada, S_w	7561.288	m^2
Eslora en la flotación, L_w	177.71	m
Volumen desplazado, ∇	44023.41	m^3
Coefficiente de Bloque, C_b	0.802	-
Coefficiente de la maestra, C_m	0.990	-
Coefficiente en la flotación, C_w	0.903	-
Semiángulo de entrada, iE	43.27	°
Posición del centro de flotación, x_{cb}	-1.21	m
Área transversal del bulbo, ABT	0	m^2
Área del espejo de popa, AT	0	m^2
Altura del cdg de ABT , h_b	0	m
Velocidad del buque, u	0 - 14	knot
Coefficiente propulsivo, η_p	0.6	-
Margen de mar	0.00	
Área de los apéndices	0.00	m^2

A continuación, en la imagen 8.8, se muestran las formas del buque granelero de 35.000 TPM, obtenidas de [3].

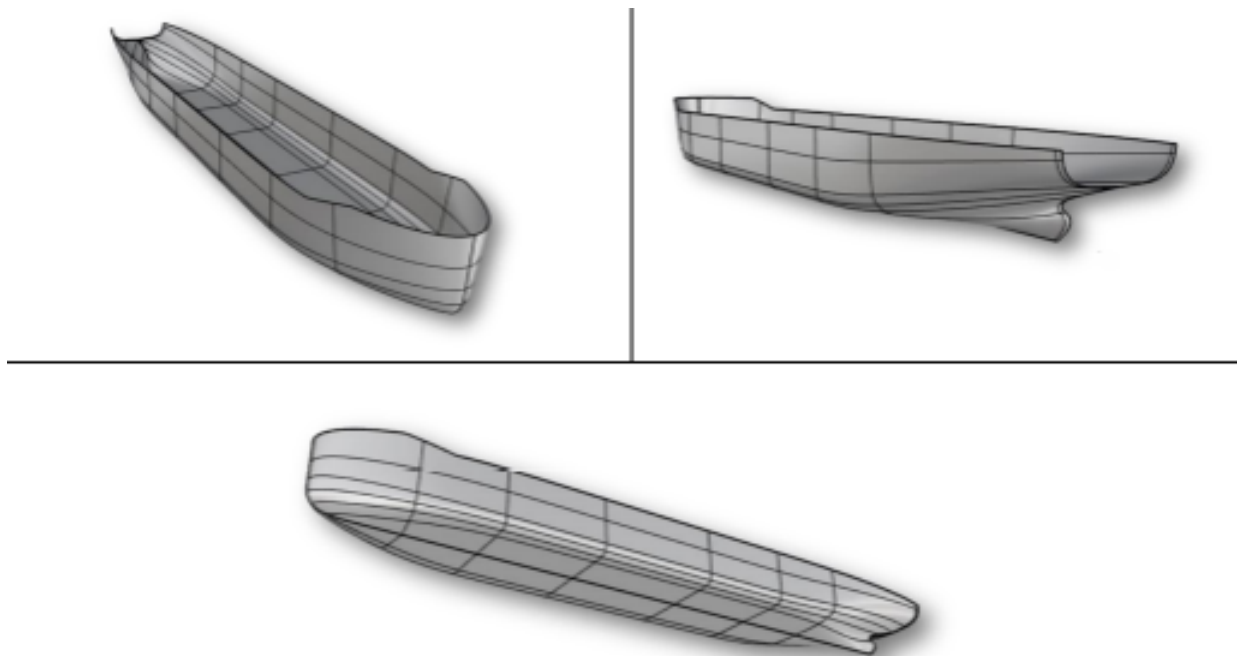


Figura 8.8: Formas del buque granelero de 35.000 TPM obtenidos de [3].

A continuación, en la tabla 8.10, se muestran los rangos de aplicación que propone el método de Holtrop-Mennen para buques graneleros, y los valores actuales.



Tabla 8.10: Rango de aplicación del método de Holtrop-Mennen para buques graneleros, y valores actuales

Dimensión	Rango de aplicación	Valor actual
Cp	0.73 - 0.85	0.81
L/B	5.1 - 7.1	5.817
B/T	2.4 - 3.2	2.913
F _{nmax}	0.24	0.174

A continuación se muestran las gráficas obtenidas, tanto en *FreeCAD*, como en *MAXSURF*, 8.9, 8.10, y 8.11. Además, se incluye también en la tabla 8.11, los valores obtenidos en ambos programas y la diferencia en porcentaje.

Tabla 8.11: Comparación de valores obtenidos de *MAXSURF* frente a *FreeCAD* para un buque granelero de 35.000 TPM.

MAXSURF Holtrop			FreeCAD Holtrop				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
0	0	0	0	0	0	0	0
0.5	0.91	0.23	0.257	0.5	0.938	0.241	2.76 %
1	3.33	1.71	0.514	1	3.417	1.758	2.73 %
1.5	7.11	5.48	0.772	1.5	7.300	5.632	2.71 %
2	12.19	12.55	1.029	2	12.522	12.883	2.69 %
2.5	18.55	23.85	1.286	2.5	19.043	24.490	2.68 %
3	26.13	40.33	1.543	3	26.833	41.408	2.67 %
3.5	34.94	62.91	1.800	3.5	35.867	64.575	2.66 %
4	44.93	92.46	2.058	4	46.126	94.909	2.66 %
4.5	56.11	129.89	2.315	4.5	57.594	133.320	2.65 %
5	68.45	176.06	2.572	5	70.257	180.702	2.64 %
5.5	81.94	231.85	2.829	5.5	84.103	237.944	2.64 %
6	96.58	298.10	3.086	6	99.120	305.925	2.63 %
6.5	112.35	375.68	3.344	6.5	115.301	385.521	2.63 %
7	129.25	465.44	3.601	7	132.641	477.615	2.62 %
7.5	147.28	568.27	3.858	7.5	151.143	583.108	2.62 %
8	166.47	685.11	4.115	8	170.821	702.962	2.61 %
8.5	186.84	817.01	4.372	8.5	191.712	838.242	2.61 %
9	208.47	965.20	4.630	9	213.885	990.200	2.60 %
9.5	231.46	1131.21	4.887	9.5	237.451	1160.373	2.59 %
10	256.00	1317.00	5.144	10	262.582	1350.721	2.57 %
10.5	282.34	1525.09	5.401	10.5	289.523	1563.769	2.55 %
11	310.80	1758.76	5.658	11	318.604	1802.790	2.51 %
11.5	341.82	2022.22	5.916	11.5	350.256	2071.975	2.47 %
12	375.94	2320.80	6.173	12	385.015	2376.622	2.41 %
12.5	413.82	2661.11	6.430	12.5	423.532	2723.313	2.35 %
13	456.24	3051.23	6.687	13	466.575	3120.079	2.27 %
13.5	504.08	3500.86	6.944	13.5	515.026	3576.549	2.17 %
14	558.36	4021.40	7.202	14	569.883	4104.070	2.06 %

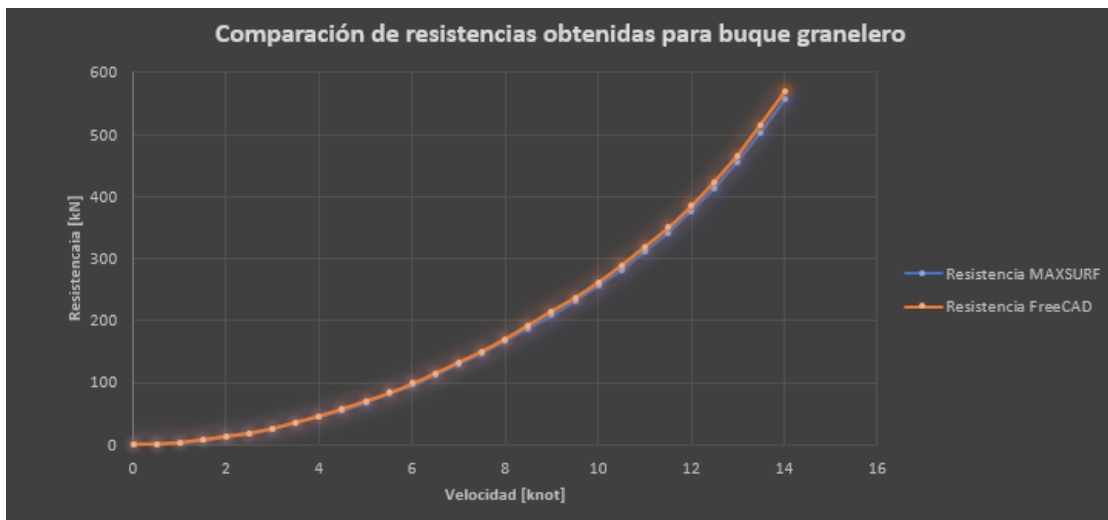


Figura 8.9: Gráfica comparativa de resistencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de MAXSURF y FreeCAD.

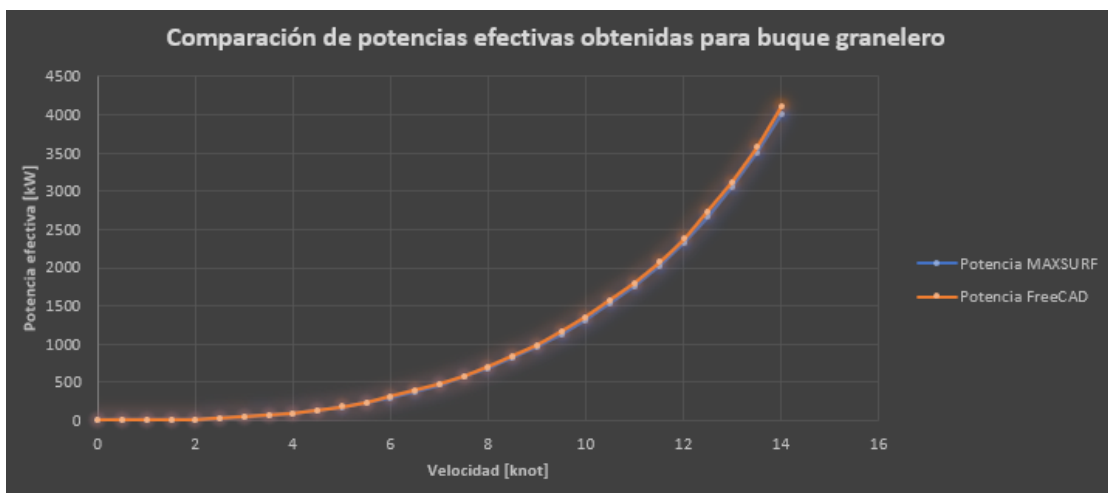


Figura 8.10: Gráfica comparativa de potencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de MAXSURF y FreeCAD.

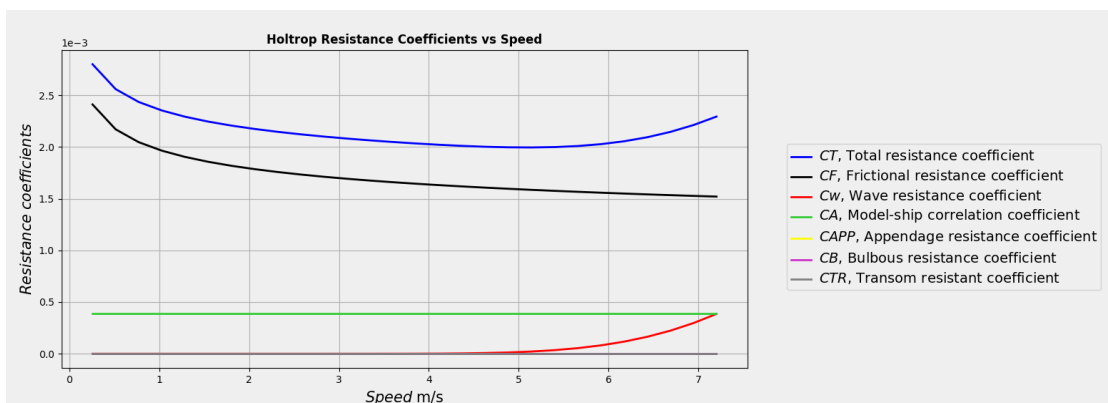


Figura 8.11: Gráfica coeficientes resistivos vs velocidad obtenida para un buque granelero de 35.000 TPM a través de FreeCAD.



8.3.4. Buque portacontenedores de 4600 TEUS.

Para la comprobación de los valores obtenidos desde *FreeCAD* y *MAXSURF* para los portacontenedores se ha escogido [4], un PFC de la E.T.S.I.N.

A continuación, en la tabla 8.12 se muestran las dimensiones del portacontenedores en cuestión.

Tabla 8.12: Dimensiones y características principales de un buque portacontenedores de 4600 TEUS

Dimensiones	Valores	Unidades
Eslora entre perpendiculares, L_{pp}	265.2	m
Calado, T	11.9	m
Manga, B	37.6	m
Superficie mojada, S_w	12485.628	m^2
Eslora en la floración, L_w	267.383	m
Volumen desplazado, ∇	76612.02	m^3
Coefficiente de Bloque, C_b	0.64	-
Coefficiente de la maestra, C_m	0.981	-
Coefficiente en la flotación, C_w	0.812	-
Semiángulo de entrada, iE	30	°
Posición del centro de flotación, x_{cb}	-1.85	m
Área transversal del bulbo, ABT	55.42	m^2
Área del espejo de popa, AT	0	m^2
Altura del cdg de ABT , h_b	5.35	m
Velocidad del buque, u	0 - 23	knot
Coefficiente propulsivo	0.6	-
Margen de mar	0.00	-
Área de los apéndices	0.00	m^2

A continuación, en la imagen 8.12, se muestran las formas de un buque portacontenedores de 4600 TEUS, obtenidas de [4].



Figura 8.12: Formas de un buque portacontenedores de 4600 TEUS obtenida a través de [4]

A continuación, en la tabla 8.13, se muestran los rangos de aplicación que propone el método



de Holtrop-Mennen para los buques portacontenedores y los valores actuales.

Tabla 8.13: Rango de aplicación del método de Holtrop-Mennen para portacontenedores y valores actuales

Dimensión	Rango de aplicación	Valor actual
Cp	0.55 - 0.67	0.652
L/B	6 - 9.5	7.053
B/T	3 - 4	3.16
F _{nmax}	0.45	0.23

A continuación se muestran las gráficas obtenidas, tanto en *FreeCAD*, ??, ??, y 8.15, como en *MAXSURF*, 8.13, y 8.14. Además, se incluye también en la tabla 8.14, los valores obtenidos en ambos programas y la diferencia en porcentaje.

Tabla 8.14: Comparación de valores obtenidos de *MAXSURF* frente a *FreeCAD* para un buque portacontenedores de 4600 TEUS.

Maxsurf Holtrop			FreeCAD Holtrop				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
0	0	0	0	0	0	0	0
1	4.5087	2.31947	0.51	1	4.51	2.32	0.11 %
2	16.60805	17.08783	1.03	2	16.62	17.10	0.07 %
3	35.74311	55.16354	1.54	3	35.73	55.15	-0.02 %
4	61.67694	126.91744	2.06	4	61.57	126.68	-0.18 %
5	94.24959	242.43089	2.57	5	93.89	241.47	-0.39 %
6	133.32803	411.53918	3.09	6	132.50	408.95	-0.62 %
7	178.79027	643.84362	3.60	7	177.25	638.24	-0.86 %
8	230.52358	948.73258	4.12	8	228.01	938.30	-1.09 %
9	288.43568	1335.45722	4.63	9	284.69	1318.01	-1.30 %
10	352.48577	1813.34348	5.14	10	347.29	1786.47	-1.47 %
11	422.74023	2392.24001	5.66	11	415.91	2353.39	-1.62 %
12	499.44882	3083.26408	6.17	12	490.84	3029.85	-1.72 %
13	583.12716	3899.82484	6.69	13	572.63	3829.32	-1.80 %
14	674.62802	4858.82092	7.20	14	662.19	4768.81	-1.84 %
15	775.1874	5981.86274	7.72	15	760.77	5870.09	-1.86 %
16	886.43803	7296.36995	8.23	16	870.04	7160.78	-1.85 %
17	1010.39008	8836.42254	8.74	17	992.04	8675.17	-1.82 %
18	1149.39074	10643.35828	9.26	18	1129.13	10454.81	-1.76 %
19	1306.02363	12765.65545	9.77	19	1283.91	12548.41	-1.69 %
20	1483.02679	15258.69783	10.29	20	1459.13	15011.49	-1.61 %
21	1684.03178	18193.15669	10.80	21	1658.42	17914.89	-1.52 %
22	1911.60185	21635.08494	11.32	22	1884.44	21325.85	-1.42 %
23	2164.15334	25606.74321	11.83	23	2135.59	25266.63	-1.32 %

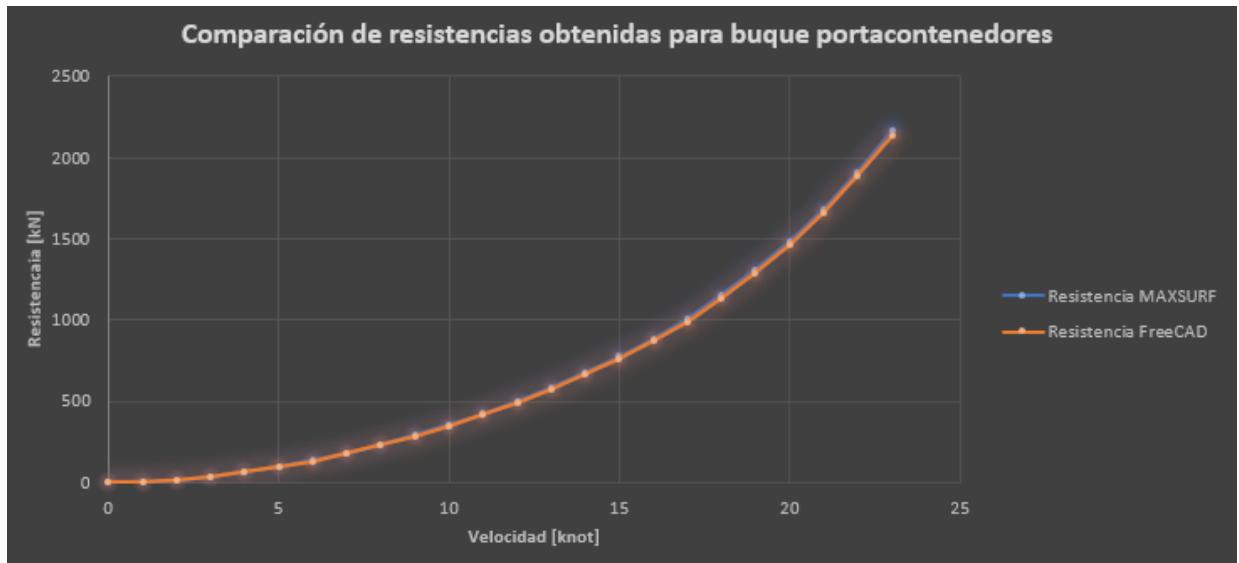


Figura 8.13: Gráfica comparativa de resistencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *MAXSURF* y *FreeCAD*.

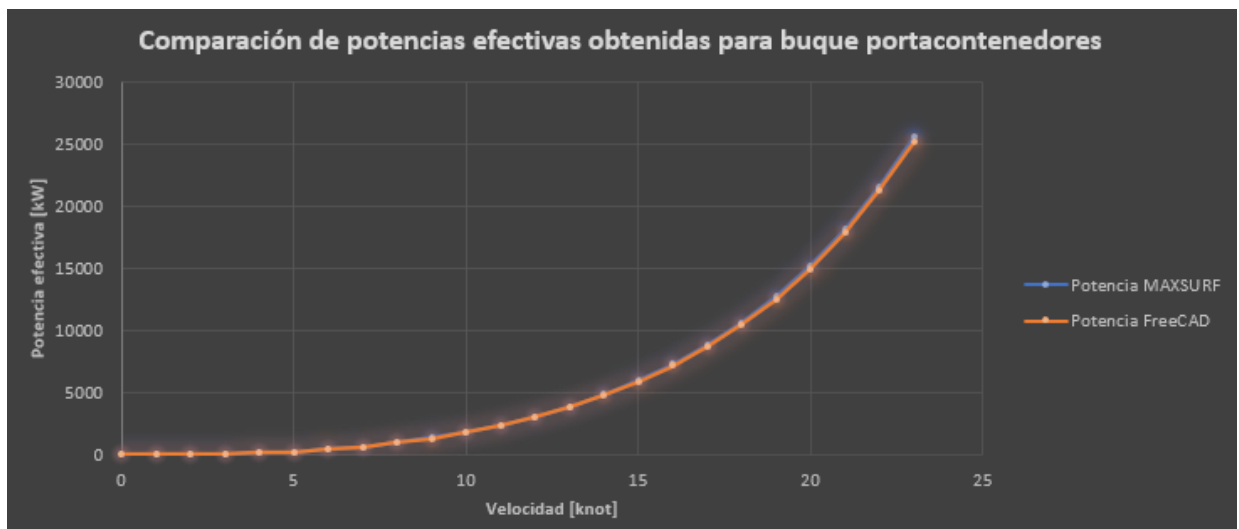


Figura 8.14: Gráfica comparativa de potencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *MAXSURF* y *FreeCAD*.

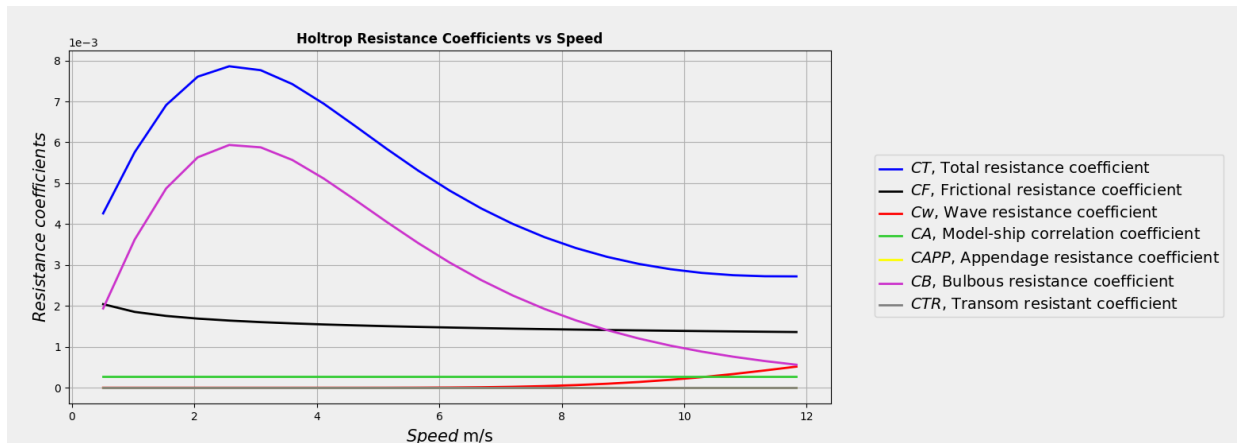


Figura 8.15: Gráfica coeficientes resistivos vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *FreeCAD*.

8.3.5. Buque tipo ro-ro.

Los valores de las dimensiones y características básicas del ferry de pasaje que se utilizarán para obtener la resistencia al avance asociada han sido obtenidos de [5], TFM de la E.T.S.I.N.

A continuación, en la tabla 8.15 se muestran las dimensiones del buque que se introducirán tanto en *FreeCAD* como en *MAXSURF*.

Tabla 8.15: Dimensiones y características principales de buque tipo ro-ro.

Dimensiones	Valores	Unidades
Eslora entre perpendiculares, L_{pp}	185	m
Calado, T	7.1	m
Manga, B	26.5	m
Superficie mojada, S_w	5784.6	m^2
Eslora en la floración, L_w	157.2	m
Volumen desplazado, ∇	20884.878	m^3
Coficiente de Bloque, C_b	0.6	-
Coficiente de la maestra, C_m	0.970	-
Coficiente en la flotación, C_w	0.72	-
Semiángulo de entrada, iE	16.63	°
Posición del centro de flotación, x_{cb}	3.1	m
Área transversal del bulbo, ABT	11.3	m^2
Área del espejo de popa, AT	0	m^2
Altura del cdg de ABT , hb	5.6	m
Velocidad del buque, u	0 - 22	knot
Coficiente propulsivo η_p	0.6	-
Margen de mar	15.00	
Área de los apéndices	0.00	m^2

A continuación, en la imagen 8.16, se muestran las formas de un buque ro-ro, obtenidas de [5].

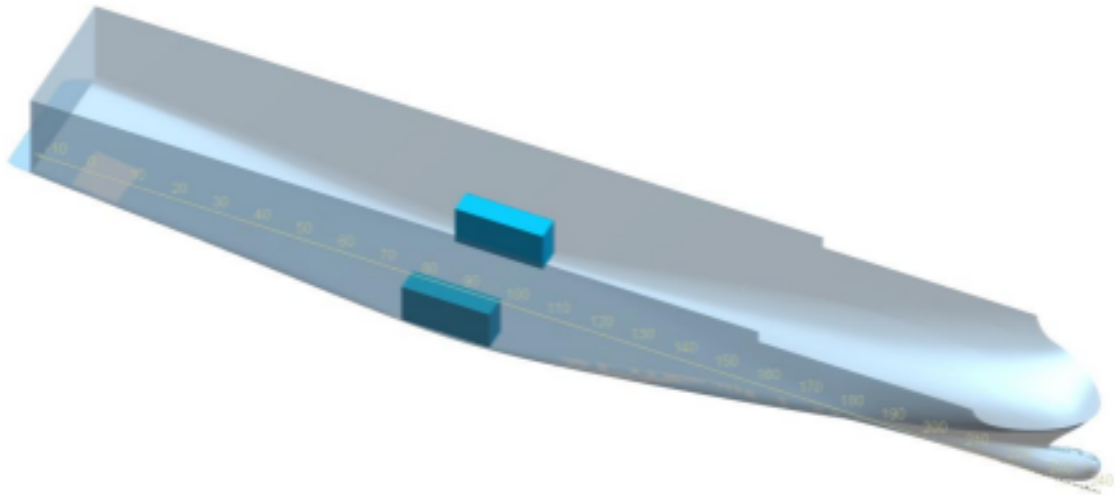


Figura 8.16: Formas de un buque tipo ro-ro obtenidas a través de [5].

A continuación, en la tabla 8.16, se muestran los rangos de aplicación que propone el método de Holtrop-Mennen para buques ferry y ro-ro y los valores actuales.

Tabla 8.16: Rango de aplicación del método de Holtrop-Mennen para buques ferry y ro-ro y valores actuales

Dimensión	Rango de aplicación	Valor actual
Cp	0.55 - 0.67	0.619
L/B	5.3 - 8.0	6.981
B/T	3.2 - 4.0	3.732
Fnmax	0.35	0.265

A continuación se muestran las gráficas obtenidas, tanto en *FreeCAD*, ??, 8.19, y ??, como en *MAXSURF*, 8.18, y 8.17. Además, se incluye también en la tabla 8.17, los valores obtenidos en ambos programas y la diferencia en porcentaje.



Tabla 8.17: Comparación de valores obtenidos de MAXSURF frente a FreeCAD para un buque ro-ro.

MAXSURF Holtrop			FreeCAD Holtrop				Diferencia
Speed (knot)	Resistance (kN)	Power(kN)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
0	–	–	0	0	0	0	0
1	2.25	1.16	0.51	1	2.31	1.19	2.37 %
2	8.54	8.79	1.03	2	8.67	8.92	1.53 %
3	18.67	28.81	1.54	3	18.73	28.91	0.34 %
4	32.38	66.63	2.06	4	32.12	66.08	-0.81 %
5	49.38	127.02	2.57	5	48.54	124.85	-1.70 %
6	69.46	214.41	3.09	6	67.88	209.50	-2.28 %
7	92.48	333.04	3.60	7	90.08	324.36	-2.60 %
8	118.35	487.09	4.12	8	115.15	473.86	-2.71 %
9	147.06	680.90	4.63	9	143.13	662.65	-2.67 %
10	178.67	919.18	5.14	10	174.13	895.75	-2.54 %
11	213.36	1207.36	5.66	11	208.35	1178.90	-2.35 %
12	251.40	1552.00	6.17	12	246.07	1518.97	-2.12 %
13	293.25	1961.20	6.69	13	287.76	1924.29	-1.87 %
14	339.46	2444.89	7.20	14	333.96	2405.02	-1.62 %
15	390.71	3015.01	7.72	15	385.34	2973.29	-1.38 %
16	447.76	3685.54	8.23	16	442.66	3643.23	-1.14 %
17	511.40	4472.49	8.74	17	506.70	4430.99	-0.92 %
18	582.54	5394.32	9.26	18	578.36	5355.16	-0.72 %
19	661.58	6466.62	9.77	19	658.05	6431.47	-0.53 %
20	748.88	7705.19	10.29	20	746.09	7675.73	-0.37 %
21	847.09	9151.43	10.80	21	845.11	9129.20	-0.23 %
22	959.90	10863.91	11.32	22	958.81	10850.68	-0.11 %

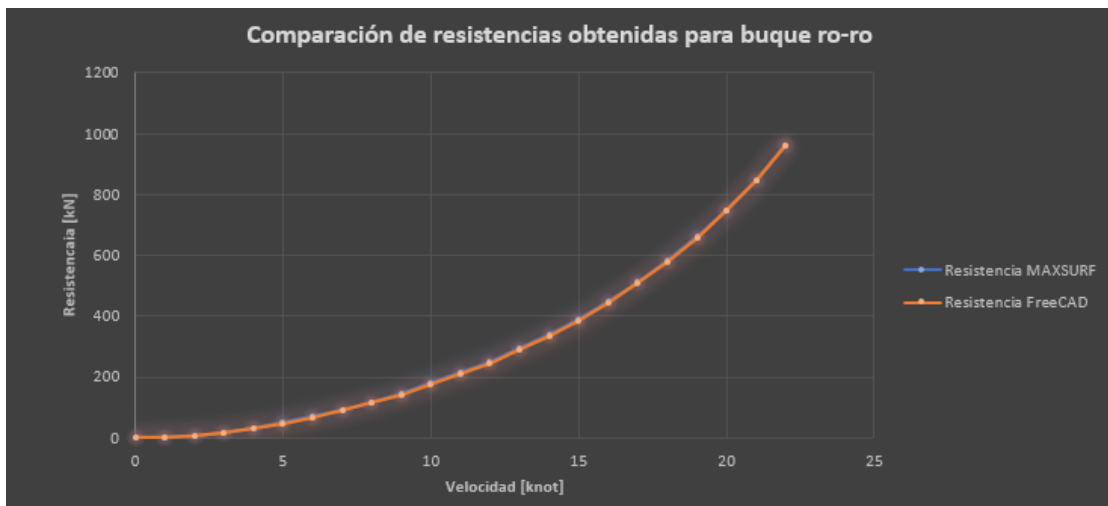


Figura 8.17: Gráfica comparativa de resistencia vs velocidad obtenida para un buque tipo ro-ro a través de MAXSURF y FreeCAD.

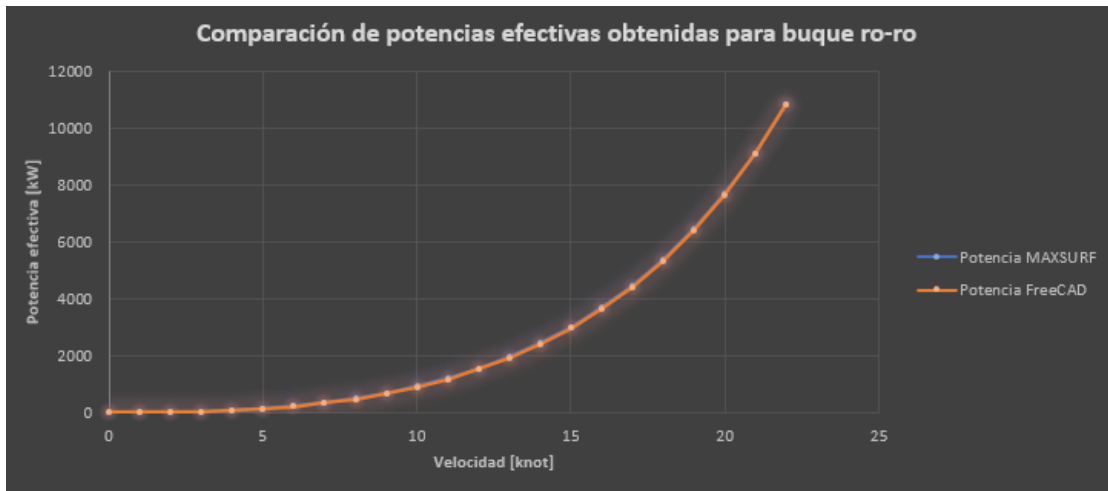


Figura 8.18: Gráfica comparativa de potencia vs velocidad obtenida para un buque tipo ro-ro a través de MAXSURF y FreeCAD.



Figura 8.19: Gráfica coeficientes resistivos vs velocidad obtenida para un buque tipo ro-ro a través de FreeCAD.

8.4. Discusión de los valores obtenidos.

A continuación, se explican y detallan las conclusiones obtenidas de los ensayos realizados en apartados anteriores con diferentes tipos de buques. Para la discusión de las diferencias obtenidas se ha utilizado el manual de usuario de MAXSURF Resistance, [17]. En él se explica la separación de resistencias utilizada y se muestra alguna de las ecuaciones empleadas. Una de las desventajas de los software privados es la dificultad para conocer el código de funcionamiento, quedando presente en este apartado.

Realizando un resumen de los resultado obtenidos es posible observar que los máximos y mínimos en las diferencias de resistencias obtenidos oscilan entre (-5.3% y 3%). Las mayores diferencias corresponden al buque Series 60 y las menores al buque portacontenedores.

A continuación, se detallan los motivos que provocan la diferencia en las resistencias obtenidas.

- Formas. FreeCAD cuenta con la capacidad de que el usuario introduzca las formas del buque, ya que siguiendo con lo mencionado en el artículo del método de Holtrop-Mennen, [9],



la resistencia debida a la formación de olas varía en función de las formas. Sin embargo, *MAXSURF* no cuenta con dicha capacidad. Una de las opciones posibles es que internamente, en función de ciertos rangos y parámetros, *MAXSURF* seleccione unas formas u otras, automáticamente. A pesar de ello, no existe evidencia de que dicha selección ocurra. Por el contrario, en *FreeCAD* si que es observable que la resistencia total varía en función de las formas introducidas por el usuario.

- Ecuaciones que modelan el método. Una de las diferencias observables es en la ecuación de la resistencia por correlación, siendo distinta ambos software. Además, se pueden hacer ciertas hipótesis sobre el resto de las ecuaciones. Se tratan de hipótesis ya que al ser *MAXSURF* un producto privado, no es posible saber exactamente las ecuaciones.

Para conocer si las ecuaciones son iguales se han realizado varias predicciones, modificando diversos datos, y lo que se ha observado es que los valores obtenidos de los dos software varían de forma distinta. Es decir, las resistencias obtenidas no varían de la misma manera, mientras que la resistencia debida a la fricción varía de una forma muy similar, la resistencia debida a las olas, aún teniendo en cuenta que en *MAXSURF* se incluye también la resistencia debida al bulbo, varía de forma distinta, por lo que se puede concluir que internamente, las ecuaciones que modelan las diversas resistencias no son iguales.

- Formas de trabajar. Otra de las diferencias a comentar es la forma de trabajar de cada uno de los software al introducir un modelo en tres dimensiones. *MAXSURF* trabaja por rebanadas, es decir, itera en las rebanadas por las que está compuesto el buque, obteniendo así las dimensiones necesarias. Es un método bastante rápido y que utiliza pocos recursos, sin embargo, existen geometrías en las que pueden existir errores. *FreeCAD*, por otra parte, trabaja por volúmenes, es decir, obtiene los resultados a través de un volumen completo. Se trata de un método más exacto y seguro, pero por otra partes es un método más costoso a nivel de recursos.
- Densidad y viscosidad cinemática. Por último, y pese a que la diferencia es mínima, es interesante mencionar que existe diferencia en la viscosidad cinemática utilizada en los diferentes software. Esta puede ser debida a una actualización de los valores entregados por la ITTC para el agua de mar a 15°C, [14].

Se puede observar que al ser *MAXSURF* un software privado, es complicado asegurar a que son debidas las diferencias. Sin embargo, es posible que estas sean a causa del mantenimiento y actualizaciones, pudiendo haber variado la obtención de las resistencias a lo largo del tiempo, en base a posibles peticiones de clientes. Además, cabe la posibilidad de que existan ciertos coeficientes de seguridad en *MAXSURF*, convirtiéndolo en un método más conservador, ya que las resistencias obtenidas son mayores, casi siempre, que aquellas calculadas en *FreeCAD*.

Pese a todas diferencias mencionadas, las diferencias de resistencias obtenidas no son notables. Además, la resistencia obtenida en *FreeCAD* en el ejemplo numérico, concuerda con la obtenida en [8], teniendo en cuenta que las diferencias en las resistencias calculas son debidas a que las ecuaciones introducidas en el método de Holtrop-Mennen en *FreeCAD* corresponden a la actualización de las ecuaciones del método en [9]. Con todo ello, queda validado el método de Holtrop-Mennen introducido en *FreeCAD Ship*.





Capítulo 9

Conclusiones y líneas de trabajo futuras.

9.1. Conclusiones.

Al inicio de este proyecto se propuso el reto de implementar varios modelos semi-empíricos de predicción de resistencia al avance en software libre.

Como figura en las especificaciones, la intención radicaba en introducir, en particular, dos métodos, el método de Amadeo y el método de Holtrop-Mennen. Dicha premisa se cumple a lo largo de todo el documento, comenzando en el capítulo 4, en el apartado 4.7. En él, se programan ambos métodos de predicción de resistencia al avance para introducirlos en *FreeCAD Ship*, además, durante el resto del documento, se realizan numerosas pruebas para verificar que ambos métodos funcionan adecuadamente. En concreto, en el capítulo 8 se comparan los valores obtenidos en *FreeCAD Ship* con *MAXSURF*, validando el método de Holtrop-Mennen, y en el anexo D se valida el método de Amadeo.

La característica remarcable de los métodos introducidos es la capacidad de obtener la predicción de resistencia al avance a través de un modelo del buque en tres dimensiones, detallada en el capítulo 6, y sin él, detallada en el capítulo 5. En la primera variante, además, se introducen numerosas funciones que extraen del modelo del buque las dimensiones y características necesarias para realizar la predicción de resistencia al avance, para que de esta forma, el usuario no deba introducir manualmente estos datos, cumpliendo de esta manera dos de las distintas fases del trabajo expuestas en las especificaciones.

Posteriormente, la presentación de los resultados obtenidos se ha conseguido a través del post-procesado realizado en el capítulo 7. En él, se muestran los pasos a seguir para lograr presentar los resultados en tres gráficas distintas e incluirlos en una hoja de cálculo, para que el usuario pueda utilizarlos posteriormente.

Por último, tras validar ambos métodos gracias a las pocas diferencias encontradas en los resultados obtenidos, observable en diversas gráficas en el capítulo 8, se realizó el proceso de “pull request”, detallado en el anexo B. Tras aceptarse el proceso, el código referido a las nuevas funcionalidades queda integrado en *FreeCAD Ship*. De esta manera, todo usuario es capaz de acceder a él descargando *FreeCAD* desde: “https://www.freecad.org/downloads.php?lang=es_ES” y obteniendo posteriormente el módulo externo Ship.

Finalmente, tras lograr implementar ambos métodos semi-empíricos en software libre y validarlos con otros software, se alcanzará una de las metas iniciales en el proyecto, incluir *FreeCAD*



Ship en la Escuela Técnica Superior de Ingenieros Navales y poder utilizarse en la asignatura: "Hidrodinámica del buque I".

Con todo ello, se concluye que la implementación de las nuevas herramientas en *FreeCAD Ship* ha sido un éxito.

9.2. Líneas futuras de trabajo.

Se establecen como líneas futuras de trabajo:

- La verificación del cumplimiento de los rangos de aplicación del método escogido. Una de las características implementadas en otros software es la capacidad de mostrar al usuario si se cumplen los requerimientos del método escogido. Es decir, si los rangos de aplicación se cumplen adecuadamente, o existe algún o algunos parámetros que se escapan de dicho rango de aplicación. La introducción de esta característica se puede abordar por varios caminos. En el primero de ellos, tras aceptar los valores y entrar al postprocesado se muestra al usuario una la ventana de información de la situación de cumplimiento los rangos. De esta forma, el usuario sería conocedor de que el buque no cumple con los requerimientos y la resistencia al avance obtenida no será suficientemente precisa, o podrá cambiar de método, a uno que se ajuste más. La otra opción, es mostrar por pantalla, mientras el usuario introduce los datos, si las dimensiones son adecuadas para el método escogido o deberían ser mayores o menores. Personalmente, debido al flujo de trabajo implementado en *FreeCAD Ship*, sería más conveniente mostrar una ventana de información final en el que se comparasen los parámetros obtenidos con los requeridos, e incluso, mostrar que método, dentro de los accesibles, se ajusta más a dicho buque.
- La inclusión del número de Froude y la velocidad en nudos, tanto en las gráficas, como en los valores mostrados en la hoja de cálculo. En primer lugar, la inclusión del número de Froude en las gráficas se basa en la creación de un nuevo eje x, bajo el eje en que se muestra la velocidad. En dicho eje se representaría el número de Froude, incluyendo al usuario la capacidad de observar la situación de la resistencia, los coeficientes resistivos y las potencias en función del número de Froude. En cuanto a la introducción del mismo en los valores mostrados en la hoja de cálculo sería necesario incluir una columna más en la creación de la hoja de cálculo. Por otra parte, referido a la velocidad en nudos, el trabajo principal es crear una nueva unidad en *FreeCAD* que sean los nudos, de esta forma, en el código interno se podrían hacer cambios de manera adecuada en las unidades de las velocidades. Por último, modificar la unidad de las velocidades en la interfaz gráfica, en las gráficas obtenidas y en las hojas de cálculos.
- El estudio de la influencia en el cambio en la densidad y viscosidad cinemática. Un estudio interesante sería un análisis en profundidad de la implicación e influencia de la variación de la densidad y viscosidad cinemática en las resistencias obtenidas, así como la obtención del valor óptimo de ambas propiedades para la obtención de la resistencia al avance.



Bibliografía

- [1] “Métodos de proyecto en arquitectura naval,” June 2023. [Antonio Villalaba Herreros, Jaime Pancorbo Crespo].
- [2] A. Antequera Vega, “Dimensionamiento y definición de formas de un petrolero de productos de 40.000 tpm,” 2021.
- [3] L. Teterea, “Buque granelero de 35.000 tpm con medios propios de carga y descarga y autonomía de 15000 millas,” 2021.
- [4] L. M. Antonio y Pablo, “Portacontenedores 4600 teus,”
- [5] M. P. Hermosilla Pérez, “Diseño inicial y conceptual de un buque tipo ro-ro de capacidad de carga equivalente a 3.500 m.l de rodadura,” 2021.
- [6] R. A. Castro, J. J. A. Azpíroz, and M. M. Fernández, *El proyecto básico del buque mercante*. Fondo Editorial de Ingeniería Naval, 1997.
- [7] A. García Gómez, “Predicción de potencia y optimización del bulbo de proa en buques pesqueros,” *Canal de Experiencias Hidrodinámicas de El Pardo, Publicación*, no. 131, 1991.
- [8] J. Holtrop and G. Mennen, “A statistical power prediction method,” *International shipbuilding progress*, vol. 25, no. 290, 1978.
- [9] J. Holtrop, “A statistical re-analysis of resistance and propulsion data,” *Published in International Shipbuilding Progress, ISP, Volume 31, Number 363*, 1984.
- [10] “Acerca de FreeCAD - FreeCAD Documentation,” Feb. 2023. <https://wiki.freecad.org/About-FreeCAD/es>.
- [11] Freecad, “freecad.ship,” June 2023. <https://github.com/FreeCAD/freecad.ship>.
- [12] “Git - Reference,” Mar. 2023. <https://git-scm.com/docs>.
- [13] “Documentation Python,” Mar. 2023. <https://docs.python.org/3>.
- [14] D. ITTC *et al.*, “Fresh water and seawater properties—7.5-02-01-03,” 2011.
- [15] J. A. Ortiz Contreras, “Bonitero del cantábrico,” 2014.
- [16] F. J. Navarro Muñoz, “Estudio de optimización hidrodinámica de la influencia de diferentes tipos de bulbos de proa para un pesquero arrastrero por medio de simulaciones con cfd,” 2021.
- [17] “MAXSURF Resistance manual,” June 2023. <https://communities.bentley.com/products/offshore/m/mediagallery/276122>.



- [18] F. J. Navarro Muñoz *et al.*, “Estudio de optimización hidrodinámica de la influencia de diferentes tipos de bulbos de proa para un pesquero arrastrero por medio de simulaciones con cfd,” 2021.



Anexo A

Instalación detallada del ecosistema necesario.

En el presente anexo se realiza una descripción detallada a cerca de la instalación de los programas necesario, la resolución de problemas que aparece en alguno de ellos y detalles a tener en cuenta.

Instalación de Python.

El primer lugar, tal y como se describe en el capítulo 3, en el apartado 3.3, es necesario descargar el propio software de Python desde su página web.

Una decisión importante a tomar es la versión del lenguaje de programación que se va a descargar e instalar. Existen numerosas versiones, en el momento de realización de este TFG se encuentra disponible hasta la versión 11.2. Sin embargo, dependiendo de las librerías que se deseen instalar se escogerá una versión de Python u otra, conociendo previamente hasta la versión en la que las librerías tienen mantenimiento, ya sea porque es una librería que va a dejar de funcionar y se introducirá una distinta o bien porque aún no se ha actualizado a las últimas versiones de Python. Cabe resaltar que se pueden instalar varias versiones de Python en la misma terminal y utilizarlas por separado.

En este caso, se instaló la versión 3.9.12 ya que era necesario instalar la librería PyQt5-tools para el desarrollo de las interfaces gráficas. La versión 3.9.12 era la última hasta la que toda la librería PyQt5-tools ofrecía soporte y mantenimiento. Sin embargo, en las versiones posteriores de python se ofrece la posibilidad de instalar el software específico para el desarrollo de las interfaces gráficas, “Designer”, que hasta la versión 3.9.12 se incluían en la propia librería. Finalmente, se decidió que era preferible instalar la librería completa, por lo que se optó por la versión 3.9.12.

Una vez descargado, se procede a la instalación. Durante el proceso, aparece la siguiente ventana.



Figura A.1: Ventana de instalación de Python

Es recomendable pulsar en “Install Now” e instalar el programa por defecto. En caso de querer modificar los archivos que se van a instalar se puede realizar desde “Customize installation”.

Por otra parte, es muy importante marcar la casilla de la parte baja de la ventana “Add python.exe to PATH”. Al marcar dicha casilla, se crearán automáticamente las variables de entorno necesarias en el sistema para que se detecte Python. En caso de no marcar la casilla se deberá incluir las variables de entorno manualmente. Para ello, se deben seguir los pasos mostrados a continuación.

1. En el explorador de archivos, click derecho en: “Este equipo”. A continuación, seleccionar “Propiedades”.
2. En la ventana de ajustes que se despliega, seleccionar: “configuración avanzada del sistema”.
3. En la nueva ventana desplegada, seleccionar: “Variables de entorno” y dentro de ella: “Variables de entorno del sistema”.
4. Por último, en esta última ventana, incluir la ruta de instalación de Python y la ruta de los scripts de Python, los cuales se encuentran en un carpeta dentro de la ruta de instalación.

Por último, para comprobar que se ha realizado correctamente la instalación de Python se abrirá una terminal de comandos y se escribirá “python”. Si se ha instalado correctamente se mostrará la versión instalada, en caso contrario se redirigirá a la tienda de Windows.

Si no aparece la versión instalada y se marcó la casilla de añadir las variables de entorno automáticamente, se deberán seguir los pasos para añadir manualmente las variables de entorno y comprobar de nuevo que se han incluido las rutas adecuadas. Si, por el contrario, se realizó



manualmente, será necesario comprobar que las rutas incluidas en las variables de entorno son las correctas.

Git.

Tras descargar Git se procederá a la instalación. Durante el proceso, aparece la siguiente ventana.

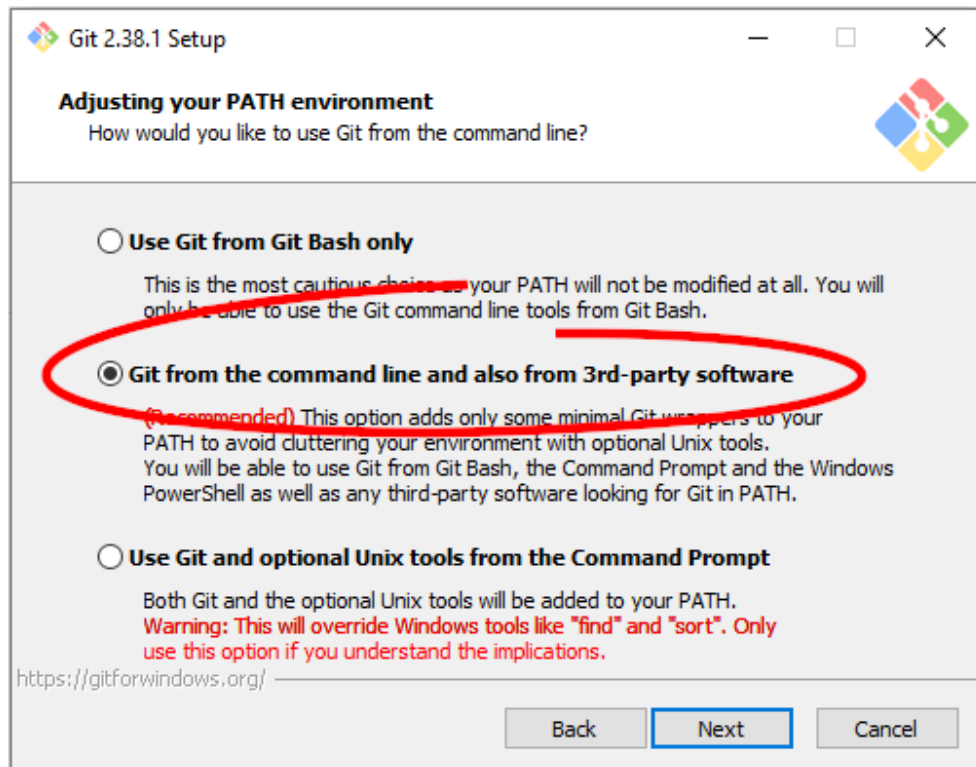


Figura A.2: Ventana de instalación de Git

Se recomienda mantener las opciones por defecto, tan solo es necesario asegurarse que se encuentra en el PATH.

Spyder

Por último, es necesario instalar un editor de código para poder interactuar y crear código. Existen muchos editores de código distintos, en esta ocasión se utilizó Spyder, sin embargo, se pueden utilizar otros editores de código a gusto del usuario.

Tras la instalación, que se realiza con las opciones por defecto, se crea un proyecto asociado al clonado del repositorio FreeCAD ¹.

Para crear un proyecto, desde Spyder, en la barra de menús se selecciona "Projects\New Project". A continuación, en la ventana que se despliega, se marca la opción de: "directorio existente" y se introduce la ruta del repositorio de FreeCAD Ship. A partir de este momento, todos los cambios que se realicen en los archivos del repositorio desde Spyder se guardarán en la ruta del repositorio y se incluirán en FreeCAD automáticamente.

¹Los detalles para realizar el clonado del repositorio se encuentran en el anexo B.





Anexo B

Repositorio FreeCAD Ship.

El presente anexo se divide en dos secciones principales, la primera de ellas contiene la información y explicaciones relativas al clonado del repositorio en la máquina local, la segunda, contiene la información relacionada con la actualización del repositorio con los cambios y novedades introducidas.

Clonado del repositorio.

En el apartado actual se procede a realizar una explicación detallada de los pasos a seguir, necesarios para realizar el clonado del repositorio FreeCAD Ship.

Creación de contraseña SSH.

En primer lugar, es necesario contar con una contraseña SSH en Git. La contraseña SSH es una credencial para acceder al protocolo de red segura. Se utiliza para crear un enlace seguro entre terminales remotas, en este caso, será necesario para poder realizar un clonado del repositorio en Github y para subir los cambios incluidos en el código, ya que todo ello es una transferencia de datos remota.

La forma de crear la contraseña SSH varía en función del sistema operativo, en este caso se explicará la forma de hacerlo en Windows. Para ello, se deben seguir los pasos mostrados a continuación.

1. En primer lugar, se abre una terminal de comandos y asegurándose de que se encuentra en la carpeta de inicio, `C:\Users \usuariowindows`, se escribe "ssh-keygen" y se pulsa intro.
2. A continuación, se muestra un mensaje en la terminal de comandos para indicar la ruta donde se guardará los pares de claves públicas y privadas generadas, es posible dejar este espacio en blanco y pulsar intro para acceder al siguiente paso. En caso de dejarlo en blanco se guardarán en una carpeta denominada `.ssh` en la carpeta de inicio.
3. Posteriormente, se muestra otro mensaje, esta vez para crear una frase de contraseña y cifrar las claves generadas, de nuevo, es posible pulsar intro y dejar este espacio en blanco.
4. Por último, se muestra un mensaje para confirmar la fase de contraseña indicada en el paso anterior, en caso de haberlo dejado en blanco, volver a pulsar intro.



5. En este punto, ya se ha creado la contraseña ssh y se puede comprobar introduciendo en la ventana de comandos “cd .ssh”, se pulsa intro y una vez en esa ruta se escribe “dir”. Aparecerán por pantalla dos archivos, la contraseña privada asociada no se deberá compartir jamás.

Una vez generada la contraseña se debe incluir en Github. Para ello:

1. Desde la ruta anterior, C:\Users \usuariowindows\.ssh, se escribe en la terminal de comandos “type id_rsa.pub | clip” y se pulsa intro. Tras esta acción se copia la contraseña en el portapapeles.
2. Ingresar a Github y acceder a los ajustes pulsando en el icono de usuario.
3. A continuación, pulsando en “Nueva contraseña SSH”, se abre una nueva ventana donde se deben rellenar los campos que aparecen. En el nombre, se indica un nombre referido a la máquina en la que se han generado las contraseñas, puede ser cualquier nombre, y en contraseña se pegará la contraseña desde el portapapeles.
4. Por último, se pulsa en “Añadir contraseña SSH”. De esta forma, se habrá generado una contraseña SSH e incluido en Github.

Clonado del repositorio

Una vez se cumplen los requisitos mencionados en el capítulo 3, en el apartado 3.1, se procede al clonado del repositorio en la máquina local.

Tras acceder al [Repositorio workbench ship](#) se abre una ventana similar a la que se muestra a continuación:

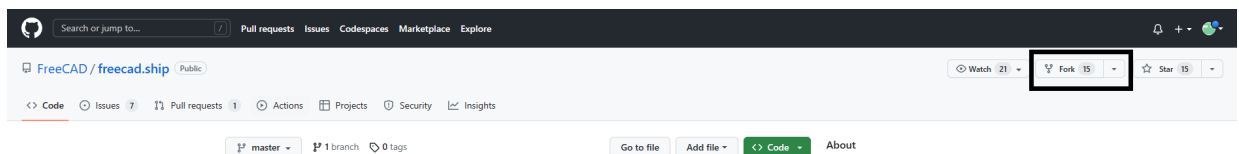


Figura B.1: Ventana repositorio FreeCAD Ship.

Pulsando en el botón que aparece arriba a la derecha denominado “fork”, se realiza una copia personal del repositorio, será esta copia la que se clonará en la máquina local y será modificada. Una vez realizadas las modificaciones pertinentes se unirá la copia con el repositorio mediante una “pull request”.

Una vez creado el fork, en el explorador de archivos se siguen los siguientes pasos:

1. Acceder a la ruta: “C:\Users\usuariowindows\AppData\Roaming\FreeCAD”.
2. En la ruta anterior, crear una carpeta llamada: “Mod”, en caso de no existir.
3. En la carpeta “Mod”, escribir: “cmd” en la barra de direcciones para que se abra una terminal de comandos en esta ruta.
4. Por último, para terminar el clonado del repositorio seguir el paso 3 del apartado 3.2.



Actualización del repositorio.

La actualización del repositorio se divide en dos etapas. En la primera etapa se actualiza la rama del repositorio que se creó para introducir los cambios y actualizaciones, esta acción se realiza a través de “commits”. Finalmente, una vez se han introducido en la rama todos los cambios y actualizaciones deseadas se debe integrar dicha rama en el proyecto en producción, acción denominada “pull request”.

Commits.

Los “commits” son la actualización del repositorio con los archivos creados o modificados en el repositorio local, es decir, en el repositorio clonado. Tras realizar este paso, la rama del repositorio quedará actualizada.

El primer paso para realizar un “commit” es conocer los archivos que han sido modificados. Para ello, en la ruta del repositorio, en este caso, “C:\Users\usuariowindows\AppData\Roaming\FreeCAD”, se escribe “cmd” en la barra de direcciones y para abrir una terminal de comandos en la ruta mencionada. Tras abrirse la terminal se escribe: “git status” y se pulsa “intro”. A continuación, se mostrará por pantalla la rama del repositorio en la que se encuentra actualmente y los archivos que han sido modificados. Dentro de los archivos modificados se encuentran aquellos que han sido eliminados, los archivos nuevos, y los que se han modificado.

Seguidamente, en la misma ruta dentro de la terminal, se procede a realizar la actualización de los archivos deseados. Para ello se sigue el siguiente proceso.

1. Subir el archivo. Para ello se sigue la siguiente sintaxis: “git add nombrearchivo”. Introduciendo en “nombrearchivo” el nombre del archivo modificado, que se suele copiar de la lista de archivos obtenida al introducir anteriormente el comando “git status”. De esta manera, se subirá un único archivo, sin embargo, existe la opción de subir a la vez todos los archivos modificados si en vez de introducir: “git add nombrearchivo”, se introduce: “git add .”.
2. Explicación. A continuación, tras subir el archivo, es importante añadir una breve explicación de los cambios introducidos. Para ello, se introduce la siguiente sintaxis: “git commit -m “explicación””. Es importante mencionar que dicha explicación se asocia al archivo que se ha subido. Por lo tanto, si se han subido todos los archivos modificados, todos tendrán la misma explicación. Es por ello, por lo que únicamente se deben subir en cada tanda los archivos que comparten explicación, si no lo hiciesen, se deberá realizar tantos procesos como archivos se quieran actualizar en el repositorio.
3. Actualización. Por último, para actualizar el repositorio con los archivos subidos anteriormente y las explicaciones asociadas, se debe introducir la siguiente sintaxis: “git push origin “nomberama””

Las etapas mencionadas anteriormente, las tres en su conjunto, constituyen una actualización del repositorio. Si se deseara actualizar varias veces el repositorio, o fuese necesario debido a las distintas explicaciones, se debería realizar el proceso entero tantas veces como fuese necesario.

Un detalle a mencionar, es que en ocasiones no se quieren subir todos los archivos que han sido modificados. Por ejemplo, al crear un proyecto en Spyder asociado al repositorio, se crean copias de seguridad y otros archivos que no se quieren subir al repositorio. Para evitar que se suban estos archivos se abre el archivo .gitignore, podría ser que se encuentre oculto, y en la



sección “Spyder project settings” se añade una línea que ponga “.spyproject”. De esta forma, no se subirán los archivos procedentes del proyecto de Spyder al repositorio al hacer los commits pertinentes.

Tras realizar el proceso completo, se habrá actualizado la rama del repositorio.

Pull request.

Tal y como se ha comentado en la introducción, esta es la última acción a realizar para actualizar finalmente el repositorio e introducir los cambios en el proyecto en producción. Es una acción delicada, ya que hasta el momento, los cambios introducidos se encontraban en una rama paralela y no afectaba al proyecto en producción, evitando ponerlo en peligro en caso de existir incompatibilidades o errores. Es por ello, por lo que esta acción debe ser aprobada por un usuario con permisos de escritura en el proyecto, asegurando siempre la estabilidad del mismo.

Una vez se confirma que las nuevas herramientas funcionan adecuadamente, los valores arrojados por las mismas son los adecuados y las funcionalidades están terminadas, se procede a realizar el pull request.

El primer paso es acceder a la rama creada donde se han incluido todas las funcionalidades. A continuación, se debe pulsar en el “contribute”, botón observable en la imagen B.2. Posteriormente, se pulsa en el botón que se despliega “Open Pull request”.

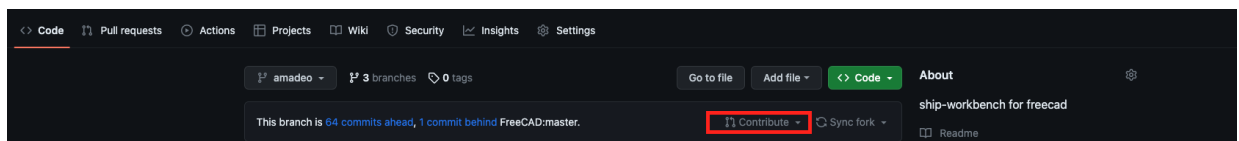


Figura B.2: Ventana repositorio FreeCAD Ship para realizar pull request.

A continuación, se redirigirá a una nueva ventana en la cual se podrán observar todos los “commits” realizados en la rama, así como la modificación de todos los archivos y el estado de los mismos. En la imagen B.3 se muestra la ventana en cuestión.

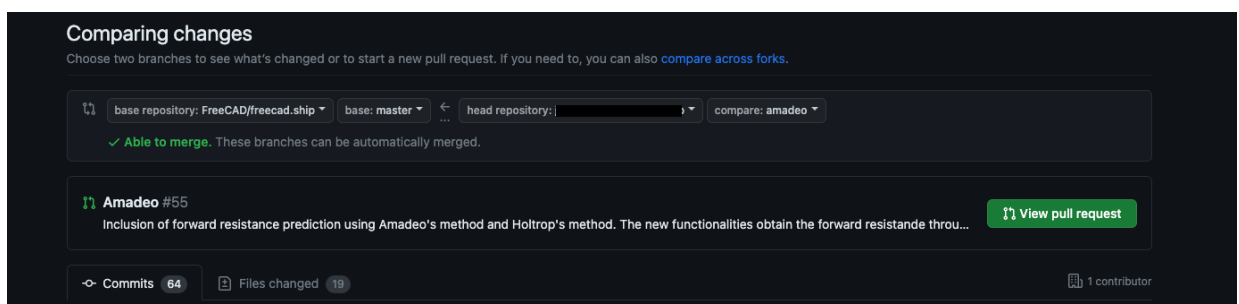


Figura B.3: Ventana de comparación de archivos durante pull request.

El siguiente paso es abrir un pull request e introducir una descripción detallada, pudiendo incluso adjuntar imágenes, de los cambios y actualizaciones que se han introducido, porqué se ha realizado, cuáles son las ventajas...etc.

Finalmente, tras realizar el pull request, se abre una ventana de interacción en la que el usuario con capacidad de escritura en el repositorio podrá comunicarse a través de una conversación si se requieren cambiar ciertas partes del código o cualquier inconveniente. Además, Github muestra la situación del código que se pretende introducir con el código actual. Es decir, se



muestra si existen o no, conflictos entre códigos.

Tras la aceptación del código por parte de los administradores, la rama pasará a integrarse en el proyecto en producción, por lo tanto, las funcionalidades quedarán disponibles para cualquier usuario que descargue el workbench, Ship.





Anexo C

Creación de una nueva plantilla de herramienta.

Interfaz gráfica de usuario.

La interfaz gráfica de usuario es uno de los puntos más importantes de la funcionalidad, ya que es a través de la cual el usuario interactúa con el programa y especifica los datos correspondientes.

A continuación, se expande la información entregada en el capítulo 4, en el apartado 4.4.

El primer punto importante a seguir son los nombres escogidos. Tanto las etiquetas como los elementos de interacción para definir las características están definidos a través de un nombre en concreto, dicho nombre se deberá utilizar posteriormente. Se recomienda siempre introducir el mismo nombre al elemento y a la etiqueta, pero añadiendo “_label” al nombre de la etiqueta, de esta forma siempre irán en bloque.

Además, existen numerosos elementos de interacción, se explicarán a continuación los más comunes y utilizados en la interfaz gráfica del método de Amadeo o de Holtrop.

- Espacios editables, QLineEdit. Son espacios en los que el usuario puede introducir información. Al ser editables, en primera instancia permite introducir cualquier dimensión, sin embargo, tiene la capacidad de asignar una dimensión concreta, y con la ayuda de FreeCAD, que es capaz de interpretar dimensiones, se mostrará una señal de prohibido para indicar que la dimensión introducida es incorrecta. Tiene numerosas funcionalidades a parte, como permitir configurar la cantidad de decimales, el número mínimo introducible, el mayor número que se puede introducir, entre otras funcionalidades que le aporta mucha flexibilidad. Se muestra a continuación, en la imagen 4.1, la visualización de los espacios editables si se han introducido las dimensiones de manera correcta o incorrecta.

Wet surface	0.00 mm	✘
Waterline length	24.89 m	✔

Figura C.1: Dimensiones introducidas de manera correcta e incorrecta.

La superficie mojada está diseñada con dimensiones de área por lo que si se introduce un



valor con dimensiones de longitud el programa marcará un error. En cambio, la eslora en la flotación tiene las unidades adecuadas y lo marca como tal.

Sin embargo, pese a esta gran funcionalidad, en el presente proyecto no se ha podido utilizar. Esto es debido a que en el momento de acceder a las funcionalidades, tanto el método de Amadeo, como el método de Holtrop, si existe algún error en el modelo y no se pueden calcular bien, todas las dimensiones de la interfaz gráfica se colocan automáticamente en milímetros. Es decir, mm , mm^2 o mm^3 , dificultando la labor al usuario. Es por ello, por lo que finalmente la dimensión de cada una de las características se muestra en el nombre de la misma y no se realiza un control de unidades.

- Casillas de verificación, QCheckBox. Son casillas que tienen la capacidad de marcarse y desmarcarse. Es una utilidad bastante interesante en casos en lo que se desee contar capacidad de decidir si visualizar o no una ventana, o marcar si existe un elemento o no, entre otros muchos ejemplos. En este caso, se ha utilizado para que el usuario marque si se desea introducir el timón en los cálculos pertinentes de resistencia al avance.
- Desplegables, QComboBox. Son utilizados para que el usuario pueda elegir entre varias opciones. En el caso del método de Holtrop se ha utilizado para que el usuario seleccione el tipo de formas del barco. Es una opción muy interesante en casos en lo que hay que elegir una única opción entre varias. Se pueden configurar insertando tantas opciones como sean necesarias y en el orden deseado
- Spinbox. Son espacios numéricos configurables en el que se puede seleccionar el número mínimo que contendrá el espacio, el número máximo y el salto de número.

Por último, una de las grandes ventajas del software Designer es que te permite asociar una descripción y ayuda. Para acceder a ello se debe hacer click derecho sobre el elemento o etiqueta en el que se quiera introducir la ayuda y se selecciona la opción "Tooltip". La ayuda que se introduzca se visualizará cuando el usuario mantenga el ratón encima de la etiqueta o elemento correspondiente.

Inclusión de los iconos e interfaces gráficas en FreeCAD.

Tal y como se ha comentado en el capítulo 4, tras crear los iconos y las interfaces gráficas en la carpeta correspondiente, es necesario incluirlo en Ship.qrc. Es en dicho archivo, en el que se incluyen todos los iconos y las interfaces gráficas.

Los archivos .qrc son archivos relativos a Qt que se utilizan para almacenar archivos de configuración.

Cada objeto nuevo se incluirá el último de la lista del tipo correspondiente. La lista de objetos se encuentra separada en iconos e interfaces gráficas, por lo que cada objeto nuevo se incluirá en su lugar respectivamente. A continuación se muestra el código:

```

1 <RCC>
2   <qresource>
3     <file>icons/Ship_AreaCurve.svg</file>
4     <file>icons/Ship_CapacityCurve.svg</file>
5     <file>icons/Ship_Instance.svg</file>
6     <file>icons/Ship_GZ.svg</file>
7     <file>icons/Ship_Hydrostatics.svg</file>
8     <file>icons/Ship_Load.svg</file>
9     <file>icons/Ship_LoadCondition.svg</file>

```



```
10 <file>icons/Ship_Logo.svg</file>
11 <file>icons/Ship_Module.svg</file>
12 <file>icons/Ship_SinkAndTrim.svg</file>
13 <file>icons/Ship_Tank.svg</file>
14 <file>icons/Ship_Weight.svg</file>
15 <file>icons/Ship_Workbench.svg</file>
16 <file>icons/Seakeeping_SetMesh.svg</file>
17 <file>icons/Seakeeping_RAOs.svg</file>
18 <file>icons/Resistance_Amadeo.svg</file>
19 <file>icons/Resistance_Holtrop.svg</file>
20 <file>translations/Ship.qm</file>
21 <file>translations/Ship_af.qm</file>
22 <file>translations/Ship_cs.qm</file>
23 <file>translations/Ship_de.qm</file>
24 <file>translations/Ship_es-ES.qm</file>
25 <file>translations/Ship_fi.qm</file>
26 <file>translations/Ship_fr.qm</file>
27 <file>translations/Ship_hr.qm</file>
28 <file>translations/Ship_hu.qm</file>
29 <file>translations/Ship_it.qm</file>
30 <file>translations/Ship_ja.qm</file>
31 <file>translations/Ship_nl.qm</file>
32 <file>translations/Ship_no.qm</file>
33 <file>translations/Ship_pl.qm</file>
34 <file>translations/Ship_pt-BR.qm</file>
35 <file>translations/Ship_ro.qm</file>
36 <file>translations/Ship_ru.qm</file>
37 <file>translations/Ship_sk.qm</file>
38 <file>translations/Ship_sv-SE.qm</file>
39 <file>translations/Ship_tr.qm</file>
40 <file>translations/Ship_uk.qm</file>
41 <file>translations/Ship_zh-CN.qm</file>
42 <file>translations/Ship_zh-TW.qm</file>
43 <file>translations/Ship_pt-PT.qm</file>
44 <file>translations/Ship_sr.qm</file>
45 <file>translations/Ship_el.qm</file>
46 <file>translations/Ship_sl.qm</file>
47 <file>translations/Ship_eu.qm</file>
48 <file>translations/Ship_ca.qm</file>
49 <file>translations/Ship_gl.qm</file>
50 <file>translations/Ship_kab.qm</file>
51 <file>translations/Ship_ko.qm</file>
52 <file>translations/Ship_fil.qm</file>
53 <file>translations/Ship_id.qm</file>
54 <file>translations/Ship_lt.qm</file>
55 <file>translations/Ship_val-ES.qm</file>
56 <file>ui/TaskPanel_shipAreasCurve.ui</file>
57 <file>ui/TaskPanel_shipCapacityCurve.ui</file>
58 <file>ui/TaskPanel_shipCreateShip.ui</file>
59 <file>ui/TaskPanel_shipCreateTank.ui</file>
60 <file>ui/TaskPanel_shipCreateWeight.ui</file>
61 <file>ui/TaskPanel_shipGZ.ui</file>
62 <file>ui/TaskPanel_shipHydrostatics.ui</file>
63 <file>ui/TaskPanel_shipLoadExample.ui</file>
64 <file>ui/TaskPanel_shipSinkAndTrim.ui</file>
65 <file>ui/TaskPanel_seakeepingSetMesh.ui</file>
66 <file>ui/TaskPanel_seakeepingRAOs.ui</file>
```



```

67     <file>ui/TaskPanel_resistanceAmadeo.ui</file>
68     <file>ui/TaskPanel_resistanceHoltrop.ui</file>
69     </qresource>
70 </RCC>
    
```

Los iconos, tanto del método de Amadeo como del método de Holtrop, se han incluido en las líneas 18 y 19. En cuanto a las interfaces gráficas, se han incluido en las líneas 67 y 68. Se puede observar cómo se han incluido los últimos de la lista respectivamente, respetando el lugar de los iconos y de las interfaces gráficas. En cuanto al nombre con el que se incluye, es el mismo nombre con el que se guarda el archivo al crearlo.

Por último, es importante ejecutar el archivo compile_resources.py. Al ejecutar este archivo se actualizará el archivo Ship.qrc y lo incluirá en FreeCAD. Cada vez que se realicen cambios en el icono o en la interfaz gráfica y se desee visualizarlo en FreeCAD es necesario ejecutar el archivo.

Para ejecutar el archivo, se abre una terminal de comandos en la ruta: C:\Users\usuariowindows\AppData\Roaming\FreeCAD\Mod\freecad.ship\freecad\ship, se escribe: "python compile_resources.py" y se pulsa "intro".

Creación del menú y de la barra de herramientas.

A continuación, se incluye el código completo tanto del archivo init_gui como del archivo ShipGui.

```

1  #*****
2  #*
3  #*   Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com>
4  #*
5  #*   This program is free software; you can redistribute it and/or modify
6  #*   it under the terms of the GNU Lesser General Public License (LGPL)
7  #*   as published by the Free Software Foundation; either version 2 of
8  #*   the License, or (at your option) any later version.
9  #*   for detail see the LICENCE text file.
10 #*
11 #*   This program is distributed in the hope that it will be useful,
12 #*   but WITHOUT ANY WARRANTY; without even the implied warranty of
13 #*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 #*   GNU Library General Public License for more details.
15 #*
16 #*   You should have received a copy of the GNU Library General Public
17 #*   License along with this program; if not, write to the Free Software
18 #*   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
19 #*   USA
20 #*
21 #*****
22
23 import FreeCADGui as Gui
24 import FreeCAD as App
25 import os
26
27
28 def QT_TRANSLATE_NOOP(context, text):
29     return text
30
31
32 class ShipWorkbench(Gui.Workbench):
    
```



```

33     """Ships design workbench."""
34     def __init__(self):
35         _dir = os.path.dirname(__file__)
36         self.__class__.Icon = os.path.join(_dir, "resources", "icons", "
            Ship_Workbench.svg")
37         self.__class__.MenuText = "Ship"
38         self.__class__.ToolTip = "Ship module provides some of the commonly
            used tool to design ship forms"
39
40     from . import ShipGui
41
42     def Initialize(self):
43         from PySide import QtCore, QtGui
44
45         Gui.addLanguagePath(os.path.join(os.path.dirname(__file__),
46                                         "resources", "translations"))
47
48         try:
49             import FreeCAD.Plot
50         except ImportError:
51             try:
52                 import freecad.plot
53             except ImportError:
54                 msg = App.Qt.translate(
55                     "ship_console",
56                     "freecad.plot is disabled, tools cannot graph output
                    curves, install freecad.plot with addon-manager",
57                     None)
58                 App.Console.PrintWarning(msg + '\n')
59
60         # ToolBar
61         shiplist = ["Ship_LoadExample",
62                   "Ship_CreateShip",
63                   "Ship_AreasCurve",
64                   "Ship_Hydrostatics"]
65         weightslist = ["Ship_Weight",
66                       "Ship_Tank",
67                       "Ship_Capacity",
68                       "Ship_LoadCondition",
69                       "Ship_SinkAndTrim",
70                       "Ship_GZ"]
71         resistancelist = ["Resistance_Amadeo",
72                          "Resistance_Holtrop"]
73         seakeepinglist = ["Seakeeping_SetMesh",
74                          "Seakeeping_RAOs"]
75
76         self.appendToolBar(
77             QT_TRANSLATE_NOOP("Workbench", "Ship design"), shiplist)
78         self.appendToolBar(
79             QT_TRANSLATE_NOOP("Workbench", "Weights"), weightslist)
80         self.appendToolBar(
81             QT_TRANSLATE_NOOP("Workbench", "Resistance"), resistancelist)
82         self.appendToolBar(
83             QT_TRANSLATE_NOOP("Workbench", "Seakeeping"), seakeepinglist)
84         self.appendMenu(
85             QT_TRANSLATE_NOOP("Workbench", "Ship design"), shiplist)
86         self.appendMenu(
            QT_TRANSLATE_NOOP("Workbench", "Weights"), weightslist)

```



```

87     self.appendMenu (
88         QT_TRANSLATE_NOOP ("Workbench", "Resistance"), resistancelist)
89     self.appendMenu (
90         QT_TRANSLATE_NOOP ("Workbench", "Seakeeping"), seakeepinglist)
91
92 Gui.addWorkbench (ShipWorkbench ())
    
```

Es importante recalcar que a la hora de crear la lista, se puede nombrar a las funcionalidades que se encuentren en ella con el nombre que se desee, pero es crucial recordar y utilizar ese mismo nombre en el archivo ShipGui. Por otro lado, es fundamental seguir el mismo orden respecto a las demás listas al incluir la lista nueva en la barra de herramientas y en la barra de menús.

```

1  #*****
2  #*
3  #*  Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com>
4  #*
5  #*  This program is free software; you can redistribute it and/or modify
6  #*  it under the terms of the GNU Lesser General Public License (LGPL)
7  #*  as published by the Free Software Foundation; either version 2 of
8  #*  the License, or (at your option) any later version.
9  #*  for detail see the LICENCE text file.
10 #*
11 #*  This program is distributed in the hope that it will be useful,
12 #*  but WITHOUT ANY WARRANTY; without even the implied warranty of
13 #*  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 #*  GNU Library General Public License for more details.
15 #*
16 #*  You should have received a copy of the GNU Library General Public
17 #*  License along with this program; if not, write to the Free Software
18 #*  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
19 #*  USA
20 #*
21 #*****
22
23 import FreeCAD
24 import FreeCADGui
25 import os
26
27 from . import Ship_rc
28 from .shipUtils import Selection
29
30
31 FreeCADGui.addLanguagePath(":/Ship/translations")
32 FreeCADGui.addIconPath(":/Ship/icons")
33
34
35 def QT_TRANSLATE_NOOP (context, text):
36     return text
37
38
39 class LoadExample:
40     def Activated(self):
41         from . import shipLoadExample
42         shipLoadExample.load()
43
44     def GetResources (self):
    
```



```

45     MenuText = QT_TRANSLATE_NOOP (
46         'Ship_LoadExample',
47         'Load an example ship geometry')
48     ToolTip = QT_TRANSLATE_NOOP (
49         'Ship_LoadExample',
50         'Load an example ship hull geometry.')
51     return {'Pixmap': 'Ship_Load',
52            'MenuText': MenuText,
53            'ToolTip': ToolTip}
54
55
56 class CreateShip:
57     def IsActive(self):
58         return bool(Selection.get_solids())
59
60     def Activated(self):
61         from . import shipCreateShip
62         shipCreateShip.load()
63
64     def GetResources(self):
65         MenuText = QT_TRANSLATE_NOOP (
66             'Ship_CreateShip',
67             'Create a new ship')
68         ToolTip = QT_TRANSLATE_NOOP (
69             'Ship_CreateShip',
70             'Create a new ship instance on top of the hull geometry')
71         return {'Pixmap': 'Ship_Module',
72                'MenuText': MenuText,
73                'ToolTip': ToolTip}
74
75
76 class AreasCurve:
77     def IsActive(self):
78         return bool(Selection.get_ships())
79
80     def Activated(self):
81         from . import shipAreasCurve
82         shipAreasCurve.load()
83
84     def GetResources(self):
85         MenuText = QT_TRANSLATE_NOOP (
86             'Ship_AreasCurve',
87             'Areas curve')
88         ToolTip = QT_TRANSLATE_NOOP (
89             'Ship_AreasCurve',
90             'Plot the transversal areas curve')
91         return {'Pixmap': 'Ship_AreaCurve',
92                'MenuText': MenuText,
93                'ToolTip': ToolTip}
94
95
96 class Hydrostatics:
97     def IsActive(self):
98         return bool(Selection.get_ships())
99
100     def Activated(self):
101         from . import shipHydrostatics

```



```

102     shipHydrostatics.load()
103
104     def GetResources(self):
105         MenuText = QT_TRANSLATE_NOOP(
106             'Ship_Hydrostatics',
107             'Hydrostatics')
108         ToolTip = QT_TRANSLATE_NOOP(
109             'Ship_Hydrostatics',
110             'Plot the ship hydrostatics')
111         return {'Pixmap': 'Ship_Hydrostatics',
112                'MenuText': MenuText,
113                'ToolTip': ToolTip}
114
115
116     class CreateWeight:
117         def IsActive(self):
118             return bool(Selection.get_shapes()) and bool(Selection.get_doc_ships
119                 ())
120
121         def Activated(self):
122             from . import shipCreateWeight
123             shipCreateWeight.load()
124
125         def GetResources(self):
126             MenuText = QT_TRANSLATE_NOOP(
127                 'Ship_Weight',
128                 'Create a new ship weight')
129             ToolTip = QT_TRANSLATE_NOOP(
130                 'Ship_Weight',
131                 'Create a new ship weight')
132             return {'Pixmap': 'Ship_Weight',
133                    'MenuText': MenuText,
134                    'ToolTip': ToolTip}
135
136     class CreateTank:
137         def IsActive(self):
138             return bool(Selection.get_solids()) and bool(Selection.get_doc_ships
139                 ())
140
141         def Activated(self):
142             from . import shipCreateTank
143             shipCreateTank.load()
144
145         def GetResources(self):
146             MenuText = QT_TRANSLATE_NOOP(
147                 'Ship_Tank',
148                 'Create a new tank')
149             ToolTip = QT_TRANSLATE_NOOP(
150                 'Ship_Tank',
151                 'Create a new tank')
152             return {'Pixmap': 'Ship_Tank',
153                    'MenuText': MenuText,
154                    'ToolTip': ToolTip}
155
156     class TankCapacity:

```



```

157     def IsActive(self):
158         return bool(Selection.get_tanks())
159
160     def Activated(self):
161         from . import shipCapacityCurve
162         shipCapacityCurve.load()
163
164     def GetResources(self):
165         MenuText = QT_TRANSLATE_NOOP(
166             'Ship_Capacity',
167             'Tank capacity curve')
168         ToolTip = QT_TRANSLATE_NOOP(
169             'Ship_Capacity',
170             'Plot the tank capacity curve (level-volume curve)')
171         return {'Pixmap': 'Ship_CapacityCurve',
172                 'MenuText': MenuText,
173                 'ToolTip': ToolTip}
174
175
176 class LoadCondition:
177     def IsActive(self):
178         return bool(Selection.get_ships())
179
180     def Activated(self):
181         from . import shipCreateLoadCondition
182         shipCreateLoadCondition.load()
183
184     def GetResources(self):
185         MenuText = QT_TRANSLATE_NOOP(
186             'Ship_LoadCondition',
187             'Create a new loading condition')
188         ToolTip = QT_TRANSLATE_NOOP(
189             'Ship_LoadCondition',
190             'Create a new load condition spreadsheet')
191         return {'Pixmap': 'Ship_LoadCondition',
192                 'MenuText': MenuText,
193                 'ToolTip': ToolTip}
194
195
196 class SinkAndTrim:
197     def IsActive(self):
198         return bool(Selection.get_lcs())
199
200     def Activated(self):
201         from . import shipSinkAndTrim
202         shipSinkAndTrim.load()
203
204     def GetResources(self):
205         MenuText = QT_TRANSLATE_NOOP(
206             'Ship_SinkAndTrim',
207             'Equilibrium draft and angle')
208         ToolTip = QT_TRANSLATE_NOOP(
209             'Ship_SinkAndTrim',
210             'Create a eschematic view of the ship equilibrium state')
211         return {'Pixmap': 'Ship_SinkAndTrim',
212                 'MenuText': MenuText,
213                 'ToolTip': ToolTip}

```



```

214
215
216 class GZ:
217     def IsActive(self):
218         return bool(Selection.get_lcs())
219
220     def Activated(self):
221         from . import shipGZ
222         shipGZ.load()
223
224     def GetResources(self):
225         MenuText = QT_TRANSLATE_NOOP(
226             'Ship_GZ',
227             'GZ curve computation')
228         Tooltip = QT_TRANSLATE_NOOP(
229             'Ship_GZ',
230             'Plot the GZ curve')
231         return {'Pixmap': 'Ship_GZ',
232                 'MenuText': MenuText,
233                 'Tooltip': Tooltip}
234
235 class Amadeo:
236     def IsActive(self):
237         return bool(Selection.get_ships())
238
239     def Activated(self):
240         from . import resistanceAmadeo
241         resistanceAmadeo.load()
242
243     def GetResources(self):
244         MenuText = QT_TRANSLATE_NOOP(
245             'Resistance_Amadeo',
246             'Resistance Amadeo prediction')
247         Tooltip = QT_TRANSLATE_NOOP(
248             'Resistance_Amadeo',
249             'Compute the resistance by Amadeo method')
250         return {'Pixmap': 'Resistance_Amadeo',
251                 'MenuText': MenuText,
252                 'Tooltip': Tooltip}
253
254 class Holtrop:
255     def IsActive(self):
256         return bool(Selection.get_ships())
257
258     def Activated(self):
259         from . import resistanceHoltrop
260         resistanceHoltrop.load()
261
262     def GetResources(self):
263         MenuText = QT_TRANSLATE_NOOP(
264             'Resistance_Holtrop',
265             'Resistance Holtrop prediction')
266         Tooltip = QT_TRANSLATE_NOOP(
267             'Resistance_Holtrop',
268             'Compute the resistance by Holtrop method')
269         return {'Pixmap': 'Resistance_Holtrop',
270                 'MenuText': MenuText,

```



```

271         'ToolTip': ToolTip}
272
273 class SetMesh:
274     def IsActive(self):
275         return bool(Selection.get_meshes()) and bool(Selection.get_doc_ships
                ())
276
277     def Activated(self):
278         from . import seakeepingSetMesh
279         seakeepingSetMesh.load()
280
281     def GetResources(self):
282         MenuText = QT_TRANSLATE_NOOP(
283             'Seakeeping_SetMesh',
284             'Set ship surface mesh')
285         ToolTip = QT_TRANSLATE_NOOP(
286             'Seakeeping_SetMesh',
287             'Associate the surface mesh to the ship')
288         return {'Pixmap': 'Seakeeping_SetMesh',
289                'MenuText': MenuText,
290                'ToolTip': ToolTip}
291
292
293 class RAOs:
294     def IsActive(self):
295         return bool(Selection.get_lcs_with_mesh())
296
297     def Activated(self):
298         from . import seakeepingRAOs
299         seakeepingRAOs.load()
300
301     def GetResources(self):
302         MenuText = QT_TRANSLATE_NOOP(
303             'Seakeeping_RAOs',
304             'Plot RAOs')
305         ToolTip = QT_TRANSLATE_NOOP(
306             'Seakeeping_RAOs',
307             'Compute and plot the RAOs')
308         return {'Pixmap': 'Seakeeping_RAOs',
309                'MenuText': MenuText,
310                'ToolTip': ToolTip}
311
312
313 FreeCADGui.addCommand('Ship_LoadExample', LoadExample())
314 FreeCADGui.addCommand('Ship_CreateShip', CreateShip())
315 FreeCADGui.addCommand('Ship_AreasCurve', AreasCurve())
316 FreeCADGui.addCommand('Ship_Hydrostatics', Hydrostatics())
317 FreeCADGui.addCommand('Ship_Weight', CreateWeight())
318 FreeCADGui.addCommand('Ship_Tank', CreateTank())
319 FreeCADGui.addCommand('Ship_Capacity', TankCapacity())
320 FreeCADGui.addCommand('Ship_LoadCondition', LoadCondition())
321 FreeCADGui.addCommand('Ship_SinkAndTrim', SinkAndTrim())
322 FreeCADGui.addCommand('Ship_GZ', GZ())
323 FreeCADGui.addCommand('Resistance_Amadeo', Amadeo())
324 FreeCADGui.addCommand('Resistance_Holtrop', Holtrop())
325 FreeCADGui.addCommand('Seakeeping_SetMesh', SetMesh())
326 FreeCADGui.addCommand('Seakeeping_RAOs', RAOs())

```



Las primeras líneas de código de las diferentes clases, en el método de Amadeo 236 - 237, definen cuándo se encuentra activa la herramienta. En este caso, se ha decidido que la herramienta estará activa siempre, se seleccione un barco o no. La decisión radica en poder utilizar la herramienta en las dos variantes mencionadas, a través de un modelo del buque en tres dimensiones o sin él. Las siguientes líneas de código, antes de la configuración de la barra de herramientas y menú, importan el archivo en el que se encuentran los cálculos que realizará la funcionalidad. Por último, es importante incluir en parte final del código los nuevos comandos que se refieren a las funcionalidades nuevas introducidas.

Programación del método semi-empírico

A continuación, se incluyen los códigos correspondientes a la programación de los métodos semi-empíricos.

Código de Amadeo.

```

1 # *****
2 # *
3 # *   Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com>
4 # *
5 # *   This program is free software; you can redistribute it and/or modify
6 # *   it under the terms of the GNU Lesser General Public License (LGPL)
7 # *   as published by the Free Software Foundation; either version 2 of
8 # *   the License, or (at your option) any later version.
9 # *   for detail see the LICENCE text file.
10 # *
11 # *   This program is distributed in the hope that it will be useful,
12 # *   but WITHOUT ANY WARRANTY; without even the implied warranty of
13 # *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 # *   GNU Library General Public License for more details.
15 # *
16 # *   You should have received a copy of the GNU Library General Public
17 # *   License along with this program; if not, write to the Free Software
18 # *   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
19 # *   USA
20 # *
21 # *****
22
23 import numpy as np
24
25
26 def Lw_auto(L, V, prot):
27     assert prot >= 0
28     Lw = 1.11 * V ** (1/3) + 0.874 * L - 2.56
29     if prot > 0:
30         Lw = Lw + prot
31     return Lw
32
33
34 def Sw_auto(L, V, prot):
35     assert prot >= 0
36     if prot == 0:
37         Sw = 3.019 * V ** (2/3) + 0.602 * V ** (1/3) * L - 1.734
38     elif prot > 0:
39         Sw = 4.420 * V ** (2/3) + 0.378 * V ** (1/3) * L - 26.5

```



```

40     return Sw
41
42
43 def Amadeo(L, B, T, Cb, V, u, etap, seamargin, prot=0, Sw='auto', Lw='auto',
44           d=None, l=None, has_rudder=False):
45     """
46     Amadeo resistance prediction method.
47
48     Positional parameters:
49     L -- Length between perpendiculars.
50     B -- Beam.
51     T -- draft.
52     Cb -- Block coefficient.
53     V -- Displaced volume.
54     u -- Speed.
55     etap -- Propulsive coefficient.
56     seamargin -- sea margin to calculate EKW and BKW
57
58     keyword arguments:
59
60     Prot -- bow bulb length.
61     Sw -- Wet surface.
62     Lw -- Waterline length.
63     D -- Ducted propeller diameter.
64     l -- Ducted propeller length.
65
66     Returns:
67
68     Rt -- total resistance in kN.
69     uu -- speeds higher than 0 m/s.
70     CF -- Friction coefficient.
71     CA -- Roughness coefficient.
72     CR -- residual resistance coefficient.
73     CT -- Total resistance coefficient.
74     EKW -- Effective power in kW.
75     BKW -- Break power in kw.
76     """
77     rho = 1025 #kg/m3
78     g = 9.81 #m/s2
79     nu = 1.1892*10**-6 #m2/s
80
81     assert np.all(u >= 0)
82     valid_u_mask = u > 0
83     uu = u[valid_u_mask]
84     Rt = np.zeros(len(uu), dtype=float)
85     CF = np.zeros(len(uu), dtype=float)
86     CA = np.zeros(len(uu), dtype=float)
87     CR = np.zeros(len(uu), dtype=float)
88     CT = np.zeros(len(uu), dtype=float)
89
90     Lw = Lw or 'auto'
91     Sw = Sw or 'auto'
92
93     if Lw == 'auto':
94         Lw = Lw_auto(L, V, prot)
95
96     if Sw == 'auto':

```



```

97     Sw = Sw_auto(L, V, prot)
98
99     """ Calculation of total wetted surface """
100
101     STCC = 0.1 * L * T if has_rudder else 0.0
102
103     if d is not None and l is not None:
104
105         STB = 1.13 * np.pi * d ** 2 * (l / d) / 0.5
106
107     else:
108         STB = 0
109
110     STW = STB + Sw + STCC
111
112     """ Calculation of resistance coefficients. """
113
114     Ca = (69 + 200 * Cb * B / L - 0.26 * L + 1300 / L - 29.5 * np.log10(L)
115           + 17 * B / T - (B / T) ** 2) * 10 ** -5
116     Fn = uu / (np.sqrt(g * L))
117     RR = 1.24 * Cb * B / L + 0.265 * Fn ** 2 + 2.151 * Fn - 0.298
118     CA = np.linspace(Ca, Ca, len(uu))
119
120     if prot == 0:
121
122         Rn = Lw * uu / nu
123         CF = 0.075 / (np.log10(Rn) - 2) ** 2
124         CT = (CF + Ca) / (1 - RR)
125         CR = CT * RR
126         Rt = 0.5 * rho * CT * STW * uu ** 2 / 1000 #kN
127
128     else:
129
130         Rn = Lw * uu / nu
131         CF = 0.075 / (np.log10(Rn) - 2) ** 2
132         Fnbb = uu / (np.sqrt(g * prot))
133         a = -47.3 * Fnbb**3 + 292.7 * Fnbb**2 - 579.7 * Fnbb + 351.7
134         b = 166.7 * Fnbb**3 - 1037.6 * Fnbb**2 + 2062.8 * Fnbb - 1244.8
135         DES = a * L / B + b
136         RRcb = RR / (1 + DES / 100)
137         CT = (CF + Ca) / (1 - RRcb)
138         CR = CT * RRcb
139         Rt = 0.5 * rho * CT * STW * uu ** 2 / 1000 #kN
140
141     EKW = Rt * uu * (1 + seamargin)
142     BKW = EKW / etap
143
144     return Rt, uu, CF, CA, CR, CT, EKW, BKW, Lw, Sw
145
146 if __name__ == '__main__':
147
148     import matplotlib.pyplot as plt
149
150     L = 21.42
151     Lw = 22.498
152     B = 6.34
153     T = 2.52

```



```

154     Cb = 0.233
155     prot = 0.98
156     V = 103.369
157     etap = 0.6
158     seamargin = 0.15
159     Sw = ()
160
161     speeds = np.linspace(0, 6.1728, num = 13)
162
163     Resistance, uu, Cfric, Croughness, Cresidual, CTotal, EKW, BKW, Lw, Sw =
164     (
165     Amadeo(L, B, T, Cb, V, speeds, etap, seamargin, prot, Sw, Lw))
166
167     plt.plot(uu, Resistance)
168     plt.xlabel("Speed [m/s]")
169     plt.ylabel("Total resistance [kN]")
170     plt.show()
171
172     plt.plot(uu, EKW)
173     plt.xlabel("Speed [m/s]")
174     plt.ylabel("Effective power [kW]")
175     plt.show()

```

Código de Holtrop.

```

1 # *****
2 # *
3 # *   Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com>
4 # *
5 # *   This program is free software; you can redistribute it and/or modify
6 # *   it under the terms of the GNU Lesser General Public License (LGPL)
7 # *   as published by the Free Software Foundation; either version 2 of
8 # *   the License, or (at your option) any later version.
9 # *   for detail see the LICENCE text file.
10 # *
11 # *   This program is distributed in the hope that it will be useful,
12 # *   but WITHOUT ANY WARRANTY; without even the implied warranty of
13 # *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 # *   GNU Library General Public License for more details.
15 # *
16 # *   You should have received a copy of the GNU Library General Public
17 # *   License along with this program; if not, write to the Free Software
18 # *   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
19 # *   USA
20 # *
21 # *****
22
23 import numpy as np
24
25
26 def Sw_auto(B, T, Lw, Cb, Cw, Cm, ABT):
27     Sw = (Lw * (2 * T + B) * np.sqrt(Cm) * (0.453 + 0.4425 * Cb
28         - 0.2862 * Cm - 0.003467 * (B / T) + 0.3696 * Cw
29         + 2.38 * (ABT / Cb))
30     return Sw
31
32

```



```

33 def Holtrop(B, T, Lw, V, Cb, Cm, Cw, cstern, iE, xcb, u, hb, etap,
34         seamargin, Sapplist, ABT, AT, Sw='auto'):
35
36     """
37     Holtrop resistance prediction method.
38     Positional parameters:
39
40     B -- Beam.
41     T -- draft.
42     Lw -- Waterline length.
43     V -- Displaced volume.
44     Cb -- Block coefficient.
45     Cm -- Midship-section coefficient.
46     Cw -- Waterplane coefficient.
47     Cstern -- Coefficient related to the specic shape of the afterbody.
48     iE -- Half angle of entrance.
49     xcb -- Longitudinal position of the center of buoyancy.
50     u -- speed.
51     hb -- position of the center of the transverse area ABT above the keel
52         line
53     etap -- Propulsive coefficient.
54     seamargin -- sea margin to calculate EKW and BKW
55     Sapplist -- List of appendage' areas.
56     ABT -- Transverlse bulb area.
57     AT -- Inmersed part of the transverse area of the transom.
58
59     keyword arguments:
60
61     Sw -- Wet surface
62
63     Returns:
64
65     Rt -- Total resistance in kN.
66     uu -- Speeds higher than 0 m/s.
67     CT -- Total resistance coefficient.
68     CF -- Frictional resistance coefficient.
69     CAPP -- Appendage resistance coefficient.
70     CW -- Wave resistance coefficient.
71     CB -- Additional resistance coefficient due to the presence of a bulbous
72         bow near the surface.
73     CTR -- Additional resistance coefficient due to the immersed transom.
74     CA -- Model-ship correlation resistance coefficient.
75     """
76
77     rho = 1025 # kg/m3
78     g = 9.81 # m/s2
79     nu = 1.1892*10**-6 # m2/s
80     Cp = Cb / Cm
81
82     assert T > 0
83     assert Lw > 0
84     assert Cm > 0
85     assert B > 0
86     assert V > 0
87     assert np.all(u >= 0)
88     valid_u_mask = u > 0
89     uu = u[valid_u_mask]

```



```

89 Rt = np.zeros(len(uu), dtype=float)
90 RF = np.zeros(len(uu), dtype=float)
91 RAPP = np.zeros(len(uu), dtype=float)
92 RW = np.zeros(len(uu), dtype=float)
93 RB = np.zeros(len(uu), dtype=float)
94 RTR = np.zeros(len(uu), dtype=float)
95 RA = np.zeros(len(uu), dtype=float)
96 CT = np.zeros(len(uu), dtype=float)
97 CAPP = np.zeros(len(uu), dtype=float)
98 CB = np.zeros(len(uu), dtype=float)
99 CTR = np.zeros(len(uu), dtype=float)
100
101 Sw = Sw or 'auto'
102
103 if Sw == 'auto':
104     Sw = Sw_auto(B, T, Lw, Cb, Cw, Cm, ABT)
105
106 #Frictional resistance
107
108 Rn = Lw * uu / nu
109 CF = 0.075 / (np.log10(Rn) - 2) ** 2
110 RF = 1 / 2 * rho * uu ** 2 * CF * Sw / 1000 #kN
111
112 #Form factor
113
114 Lr = Lw * (1 - Cp + (0.06 * Cp * xcb) / (4 * Cp - 1))
115
116 if cstern == 0: c13 = 0.89
117 elif cstern == 1: c13 = 1
118 elif cstern == 2: c13 = 1.11
119
120 k1_1 = (0.93 + 0.487118 * c13 * (B / Lw) ** 1.06806 * (T / Lw) ** 0.46106
121         *
122         (Lw / Lr) ** 0.121563 * (Lw ** 3 / V) ** 0.36486 * (1 - Cp) **
123         (-0.604247))
124
125 #Appendance resistance
126
127 k2values = [1.7, 1.5, 2.8, 3.0, 1.7, 3.0, 2.0, 3.0, 2.8, 2.7, 1.4]
128 sumk2 = 0
129 sumsapp = 0
130 for i in range(0, 10):
131     k2 = 0 if Sapplist[i] == 0 else k2values[i] * Sapplist[i]
132     sumk2 = sumk2 + k2
133     sumsapp = sumsapp + Sapplist[i]
134
135 try:
136     k2eq = sumk2 / sumsapp
137
138 except ZeroDivisionError:
139     k2eq = 0.0
140
141 RAPP = 1 / 2 * rho * uu ** 2 * k2eq * sumsapp * CF / 1000 #kN
142
143 if k2eq != 0:
144     CAPP = RAPP * 1000 / (1 / 2 * rho * sumsapp * uu ** 2)

```



```

144
145 #Wave resistance
146
147 Fn = uu / (np.sqrt(g * Lw))
148
149 ratioB = B / Lw
150
151 if ratioB <= 0.11: c7 = 0.229577 * ratioB ** 0.33333
152 elif ratioB >= 0.25: c7 = 0.5 - 0.0625 * (1 / ratioB)
153 elif 0.11 < ratioB < 0.25: c7 = ratioB
154
155 c1 = 2223105 * c7 ** 3.78613 * (T / B) ** 1.07961 * (90 - iE) **
      (-1.37565)
156
157 c3 = 0.56 * ABT ** 1.5 / (B * T * (0.31 * np.sqrt(ABT) + T - hb))
158
159 c2 = np.exp(-1.89 * np.sqrt(c3))
160
161 c5 = 1 - (0.8 * AT) / (B * T * Cm)
162
163 c17 = (6919.3 * Cm ** (- 1.3346) * (V / Lw ** 3) ** 2.00977 *
164       ((Lw / B) - 2) ** 1.40692)
165
166 c16 = 1.73014 - 0.7067 * Cp if Cp >= 0.8 else (8.07981 * Cp - 13.8673 *
167       Cp ** 2 + 6.984388 * Cp **
168       3)
169
170 ratioLw = Lw ** 3 / V
171 if ratioLw <= 512: c15 = -1.69385
172 elif ratioLw >= 1727: c15 = 0.0
173 elif 512 < ratioLw < 1727: c15 = (-1.69385 + ((Lw / (V ** (1 / 3))) -
174       8.0)
175       / 2.36)
176
177 m1 = (0.0140407 * (Lw / T) - (1.75254 * V ** (1 / 3)) / Lw - 4.79323 *
178       (B / Lw) - c16)
179
180 m3 = -7.2035 * (B / Lw) ** 0.326869 * (T / B) ** 0.605375
181
182 m4 = c15 * 0.4 * np.exp(- 0.034 * Fn ** (- 3.29))
183
184 if (1 / ratioB) >= 12: lamda = 1.446 * Cp - 0.36
185 if (1 / ratioB) < 12: lamda = 1.446 * Cp - 0.03 * (Lw / B)
186
187 l = len(Fn) - 1
188 for i in (0, l):
189     if Fn[i] <= 0.4:
190         RW = (c1 * c2 * c5 * V * rho * g * np.exp(m1 *
191             Fn ** (- 0.9) + m4 * np.cos(lamda * Fn ** (- 2)))) / 1000
192     if Fn[i] >= 0.5: RW = (c17 * c2 * c5 * V * rho * g * np.exp(m3 *
193             Fn ** (- 0.9) + m4 * np.cos(lamda * Fn ** (- 2)))) /
194             1000
195     if 0.4 < Fn[i] < 0.55:
196         Rwa = (c1 * c2 * c5 * V * rho * g * np.exp(m1 * 0.4 ** (-0.9)

```



```

197                                     + m4 * np.cos(lamda * 6.25))) / 1000
198
199     Rwb = (c17 * c2 * c5 * V * rho * g * np.exp(m3 * 0.55 ** (- 0.9)
200             + m4 * np.cos(lamda * 0.55 ** (- 2)))) /
201             1000
202
203     RW = Rwa + (10 * Fn - 4) * ((Rwb - Rwa) / 1.5) #kN
204
205     Cw = RW * 1000 / (1 / 2 * rho * Sw * uu ** 2)
206     #Additional resistance due to the presence of a bulbous
207
208     if ABT != 0:
209
210         Pb = 0.56 * np.sqrt(ABT) / (T - 1.5 * hb)
211
212         Fni = (uu / (np.sqrt(g * (T - hb - 0.25 * np.sqrt(ABT)))) + 0.15 *
213             uu **
214             2))
215
216         RB = ((0.11 * np.exp(-3 * Pb ** (-2)) * Fni ** 3 * ABT ** 1.5 * g *
217             rho) / (1 + Fni ** 2)) * 1 / 1000 #kN
218
219         CB = RB * 1000 / (1 / 2 * rho * ABT * uu ** 2)
220
221     #Additional resistance due to the immersed transom
222
223     if AT != 0:
224
225         Fnt = uu / (np.sqrt((2 * g * AT) / (B + B * Cw)))
226         l2 = len(Fnt) - 1
227         for i in (0,l2):
228
229             c6 = 0 if Fnt[i] >= 5 else 0.2 * (1 - 0.2 * Fnt[i])
230
231         RTR = 1 / 2 * rho * uu ** 2 * AT * c6 / 1000 #kN
232         CTR = RTR * 1000 / (1 / 2 * rho * AT * uu ** 2)
233
234     #Correlation resistance
235
236     ratioT = T / Lw
237     c4 = 0.04 if ratioT >= 0.04 else ratioT
238
239     Ca = (0.006 * (Lw + 100) ** (-0.16) - 0.00205 + 0.003 * np.sqrt(Lw / 7.5)
240         * Cb ** 4 * c2 * (0.04 - c4))
241
242     RA = 1 / 2 * rho * uu ** 2 * Sw * Ca / 1000 #kN
243     CA = np.linspace(Ca, Ca, len(uu))
244     #Total resistance
245
246     CT = CF + CAPP + Cw + CB + CTR + CA
247     Rt = RF * k1_1 + RAPP + RW + RB + RTR + RA
248
249     EKW = Rt * uu * (1 + seamargin)
250     BKW = EKW / etap
251
252     return Rt, uu, CT, CF, CAPP, Cw, CB, CTR, CA, EKW, BKW, Sw

```



```
252
253 if __name__ == '__main__':
254
255     import matplotlib.pyplot as plt
256
257     Lw = 44.879
258     B = 10
259     T = 4
260     V = 1007.8
261     Cb = 0.566
262     Cm = 0.9345
263     Cw = 0.6733
264     cstern = 1
265     iE = 28
266     xcb = -0.7645
267     hb = 2.127
268     etap = 0.6
269     seamargin = 0.15
270     Sapplist = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
271     ABT = 2.372
272     AT = 0
273     Sw = 572.547
274
275     speeds = np.linspace(1.5432, 5.6584, num=9)
276
277     Rtotal, uu, CT, CF, CAPP, Cw, CB, CTR, CA, EKW, BKW, Sw = Holtrop(B,
278         T, Lw, V, Cb, Cm, Cw, cstern, iE, xcb, speeds, hb, etap, seamargin,
279         Sapplist, ABT, AT, Sw)
280
281     plt.plot(uu, Rtotal)
282     plt.xlabel("Speeds [m/s]")
283     plt.ylabel("Total resistance [kN]")
284     plt.show()
285
286     plt.plot(uu, EKW)
287     plt.xlabel("Speeds [m/s]")
288     plt.ylabel("Effective power [kW]")
289     plt.show()
```



Anexo D

Verificación del método de Amadeo.

En el presente anexo se continúan las pruebas realizadas para la verificación del método de Amadeo. Las gráficas que se muestran a continuación han sido obtenidas a través del método de Amadeo en *FreeCAD Ship*, tras incluirlo al completo.

A continuación, se realizan pruebas con un total de 3 variantes de buques y se comentan los resultados obtenidos. Todas las variantes de buques han sido escogidas del TFG de la Universidad Politécnica de Cartagena, [18].

Como apunte, para cada buque, se mostrarán los datos introducidos en *FreeCAD Ship*, una gráfica comparativa de las resistencias y potencias obtenidas y una tabla con la comparación de los valores obtenidos.

Buque 1.

En la imagen [D.1](#) se muestran las dimensiones del buque 1 utilizado. A continuación en la tabla [D.1](#) se muestra la comparación de los valores obtenidos y por último, en las gráficas [D.2](#) y [D.3](#) se muestran las gráficas comparativas.



Resistance by Amadeo method

Length BPP [m]	21.420	✓
Beam [m]	6.340	✓
Draft [m]	2.550	✓
Protuberance [m]	0.000	✓
Wet surface [m ²]	154.434	✓
Waterline length [m]	21.781	✓
Displaced volume [m ³]	99.450	✓
Block coefficient	0.231	✓
Ducted propeller diameter [m]	0.000	✓
Ducted propeller length [m]	0.000	✓
Maximum speed [m/s]	6.944	✓
Minimum speed [m/s]	1.543	✓
Number of speeds	21	⌵
np	0.600	✓
Sea margin %	0.000	✓
Rudder	<input type="checkbox"/>	

Figura D.1: Dimensiones del buque 1 introducido en *FreeCAD Ship* para predicción Amadeo.

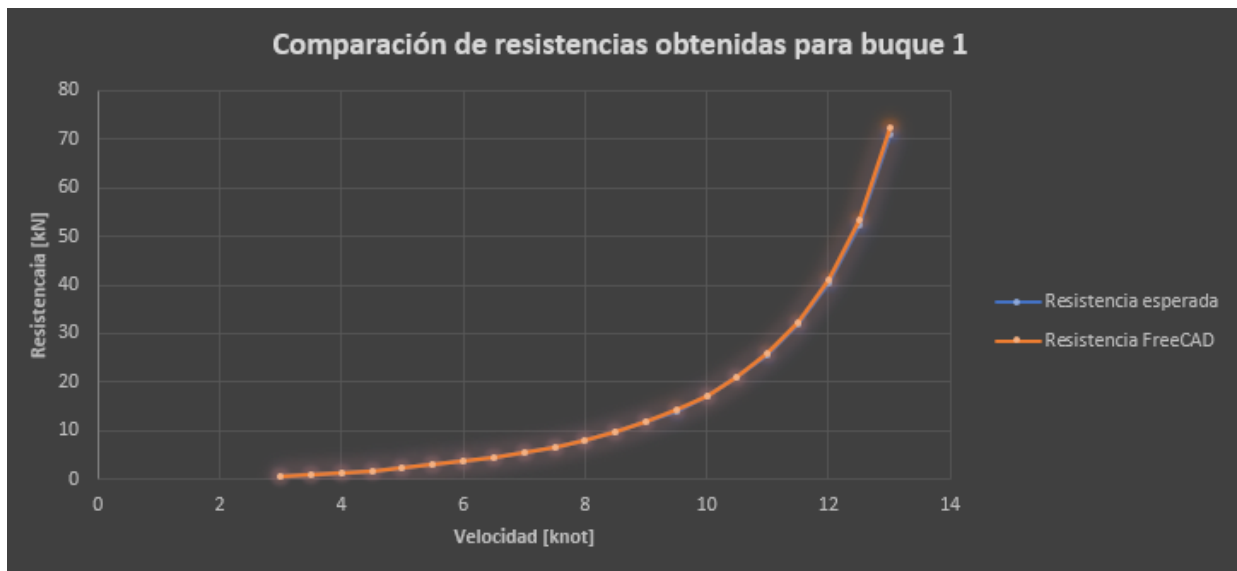


Figura D.2: Gráfica comparativa de las resistencias obtenidas para el buque 1 en el método de Amadeo.

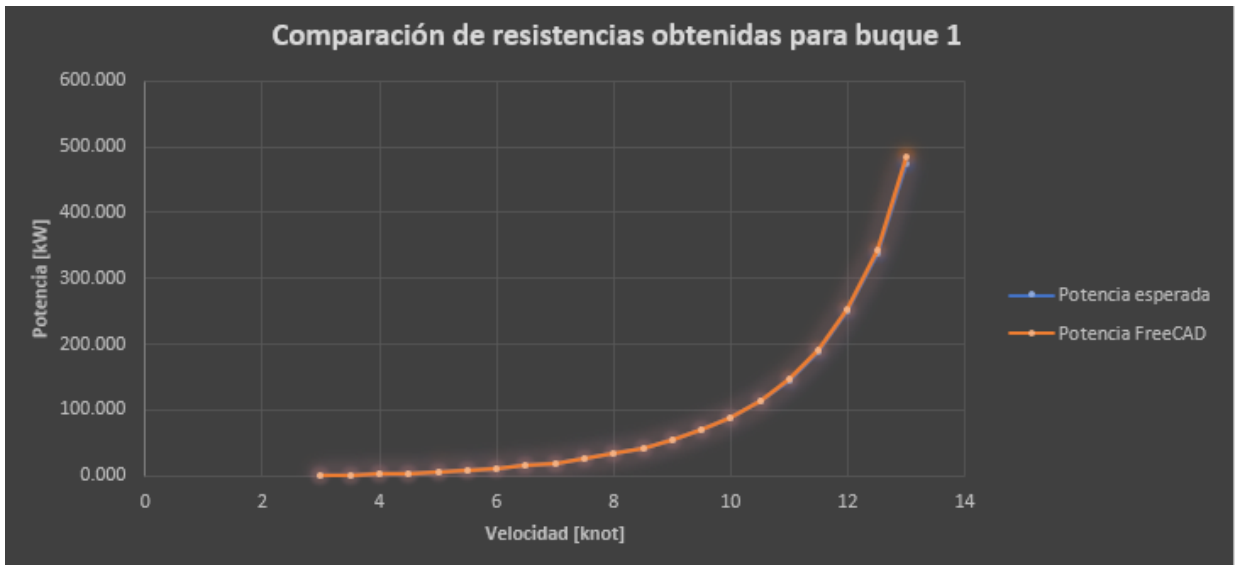


Figura D.3: Gráfica comparativa de las potencias efectivas obtenidas para el buque 1 en el método de Amadeo.



Figura D.4: Coeficientes resistivos del método de Amadeo obtenidos de *FreeCAD Ship* para el buque 1.



Tabla D.1: Comparación de valores obtenidos del método de Amadeo en *FreeCAD Ship* con los valores esperados para el buque 1.

Amadeo esperado			FreeCAD Amadeo				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
3	0.74	1.142	1.543	0	0.743	1.147	0.43 %
3.5	1.033	1.860	1.800	3.5	1.037	1.867	0.41 %
4	1.388	2.856	2.058	4	1.395	2.870	0.50 %
4.5	1.816	4.204	2.315	4.5	1.825	4.224	0.50 %
5	2.326	5.982	2.572	5	2.338	6.012	0.51 %
5.5	2.93	8.290	2.829	5.5	2.946	8.333	0.54 %
6	3.644	11.247	3.086	6	3.664	11.308	0.55 %
6.5	4.486	14.999	3.344	6.5	4.512	15.087	0.59 %
7	5.48	19.732	3.601	7	5.514	19.855	0.63 %
7.5	6.657	25.683	3.858	7.5	6.700	25.848	0.65 %
8	8.054	33.144	4.115	8	8.110	33.372	0.70 %
8.5	9.722	42.508	4.372	8.5	9.795	42.823	0.75 %
9	11.729	54.301	4.630	9	11.823	54.734	0.80 %
9.5	14.167	69.231	4.887	9.5	14.291	69.831	0.87 %
10	17.166	88.302	5.144	10	17.330	89.138	0.95 %
10.5	20.914	112.961	5.401	10.5	21.133	114.138	1.05 %
11	25.692	145.376	5.658	11	25.993	147.071	1.17 %
11.5	31.946	188.980	5.916	11.5	32.372	191.486	1.33 %
12	40.422	249.517	6.173	12	41.047	253.360	1.55 %
12.5	52.472	337.395	6.430	12.5	53.441	343.604	1.85 %
13	70.822	473.601	6.687	13	72.454	484.484	2.30 %

Buque 2.

En la imagen [D.5](#) se muestran las dimensiones del buque 2 utilizado. A continuación en la tabla [D.2](#) se muestra la comparación de los valores obtenidos y por último, en las gráficas [D.6](#) y [D.7](#) se muestran las gráficas comparativas.



Resistance by Amadeo method

Length BPP [m]	21.420	✓
Beam [m]	6.340	✓
Draft [m]	2.550	✓
Protuberance [m]	0.980	✓
Wet surface [m ²]	161.263	✓
Waterline length [m]	23.246	✓
Displaced volume [m ³]	103.549	✓
Block coefficient	0.226	✓
Ducted propeller diameter [m]	0.000	✓
Ducted propeller length [m]	0.000	✓
Maximum speed [m/s]	6.944	✓
Minimum speed [m/s]	1.543	✓
Number of speeds	22	
np	0.600	✓
Sea margin %	0.000	✓
Rudder	<input type="checkbox"/>	

Figura D.5: Dimensiones del buque 2 introducido en *FreeCAD Ship* para predicción Amadeo.

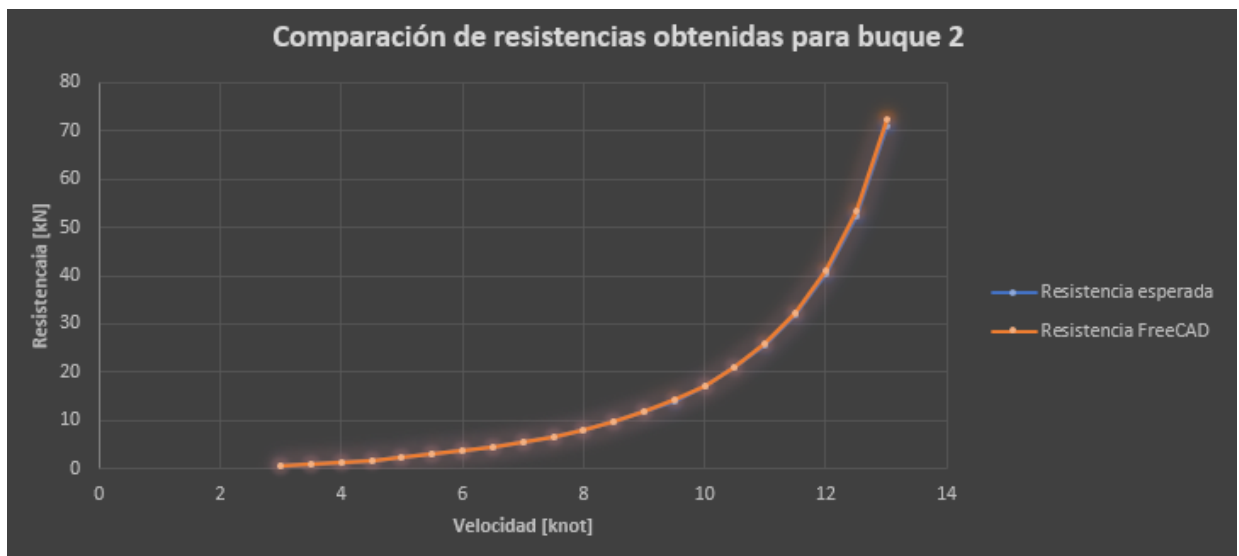


Figura D.6: Gráfica comparativa de las resistencias obtenidas para el buque 2 en el método de Amadeo.



Figura D.7: Gráfica comparativa de las potencias efectivas obtenidas para el buque 2 en el método de Amadeo.

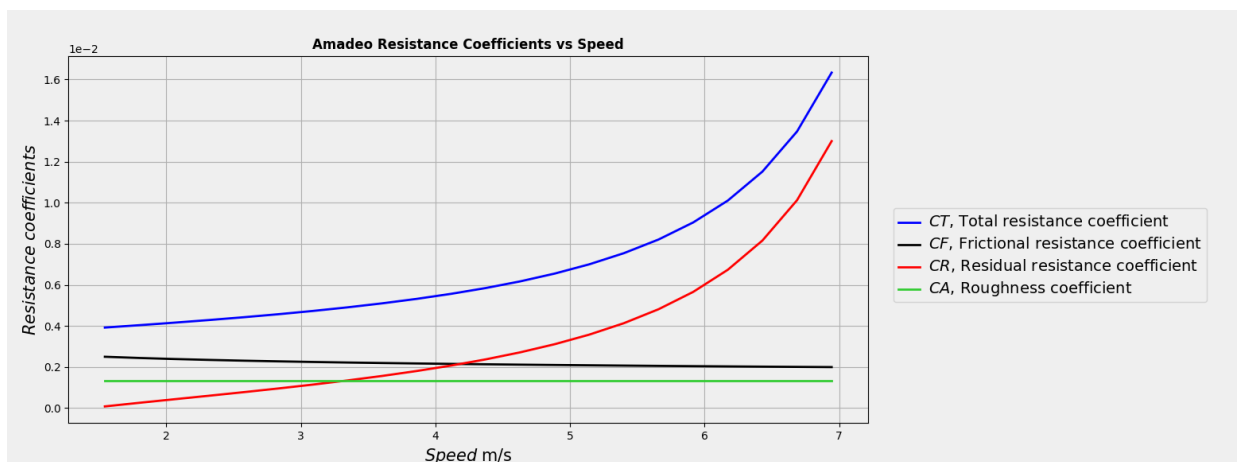


Figura D.8: Coeficientes resistivos del método de Amadeo obtenidos de *FreeCAD Ship* para el buque 2.



Tabla D.2: Comparación de valores obtenidos del método de Amadeo en *FreeCAD Ship* con los valores esperados para el buque 2.

Amadeo esperado			FreeCAD Amadeo				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
3	0.765	1.181	1.543	0.000	0.772	1.191	0
3.5	1.07	1.917	1.800	3.500	1.081	1.947	1.53 %
4	1.43	2.938	2.058	4.000	1.455	2.993	1.87 %
4.5	1.86	4.308	2.315	4.500	1.898	4.393	1.97 %
5	2.37	6.101	2.572	5.000	2.417	6.217	1.91 %
5.5	2.97	8.406	2.829	5.500	3.022	8.549	1.71 %
6	3.67	11.330	3.086	6.000	3.722	11.487	1.38 %
6.5	4.49	14.999	3.344	6.500	4.530	15.146	0.98 %
7	5.44	19.570	3.601	7.000	5.462	19.667	0.50 %
7.5	6.54	25.224	3.858	7.500	6.538	25.224	0.00 %
8	7.82	32.193	4.115	8.000	7.784	32.034	-0.50 %
8.5	9.33	40.781	4.372	8.500	9.233	40.372	-1.00 %
9	11.09	51.361	4.630	9.000	10.929	50.596	-1.49 %
9.5	13.19	64.437	4.887	9.500	12.929	63.183	-1.95 %
10	15.68	80.678	5.144	10.000	15.314	78.774	-2.36 %
10.5	18.70	101.019	5.401	10.500	18.193	98.264	-2.73 %
11	22.41	126.793	5.658	11.000	21.726	122.935	-3.04 %
11.5	27.04	159.958	5.916	11.500	26.149	154.689	-3.29 %
12	32.98	203.567	6.173	12.000	31.829	196.471	-3.49 %
12.5	40.84	262.627	6.430	12.500	39.361	253.094	-3.63 %
13	51.74	346.002	6.687	13.000	49.793	332.975	-3.77 %

Buque 3.

En la imagen [D.9](#) se muestran las dimensiones del buque 3 utilizado. A continuación en la tabla [D.3](#) se muestra la comparación de los valores obtenidos y por último, en las gráficas [D.10](#) y [D.11](#) se muestran las gráficas comparativas.



Resistance by Amadeo method	
Length BPP [m]	21.4200 ✓
Beam [m]	6.3400 ✓
Draft [m]	2.4000 ✓
Protuberance [m]	0.9800 ✓
Wet surface [m ²]	161.3730 ✓
Waterline length [m]	23.0440 ✓
Displaced volume [m ³]	103.9290 ✓
Block coefficient	0.228 ✓
Ducted propeller diameter [m]	0.0000 ✓
Ducted propeller length [m]	0.0000 ✓
Maximum speed [m/s]	6.9444 ✓
Minimum speed [m/s]	1.5432 ✓
Number of speeds	22
np	0.6000 ✓
Sea margin %	0.0000 ✓
Rudder	<input type="checkbox"/>

Figura D.9: Dimensiones del buque 3 introducido en *FreeCAD Ship* para predicción Amadeo.

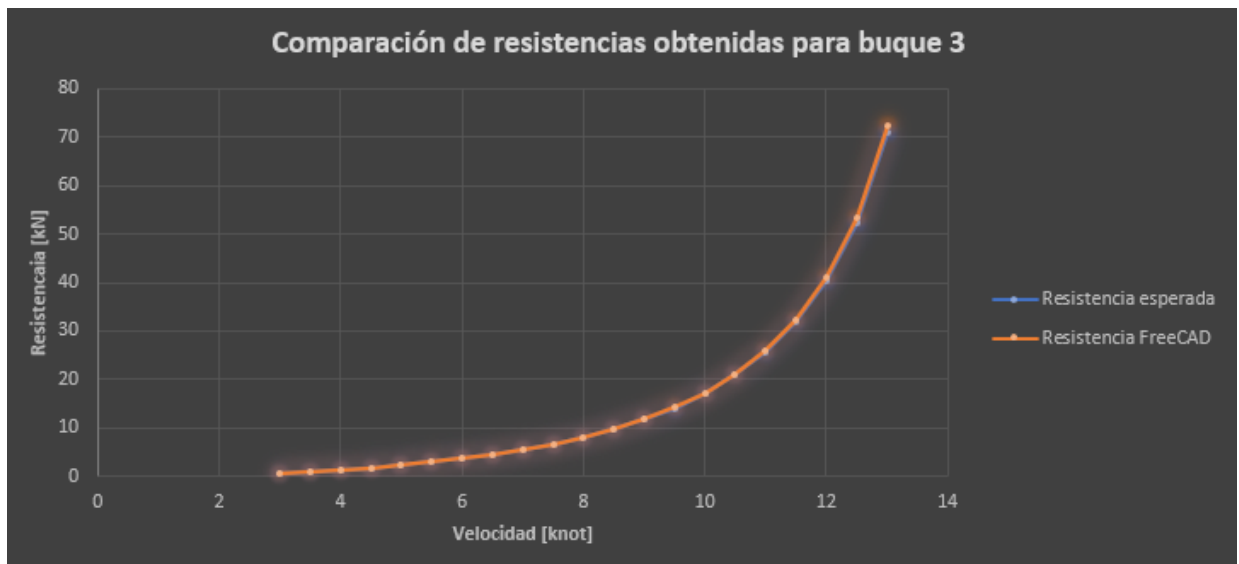


Figura D.10: Gráfica comparativa de las resistencias obtenidas para el buque 3 en el método de Amadeo.

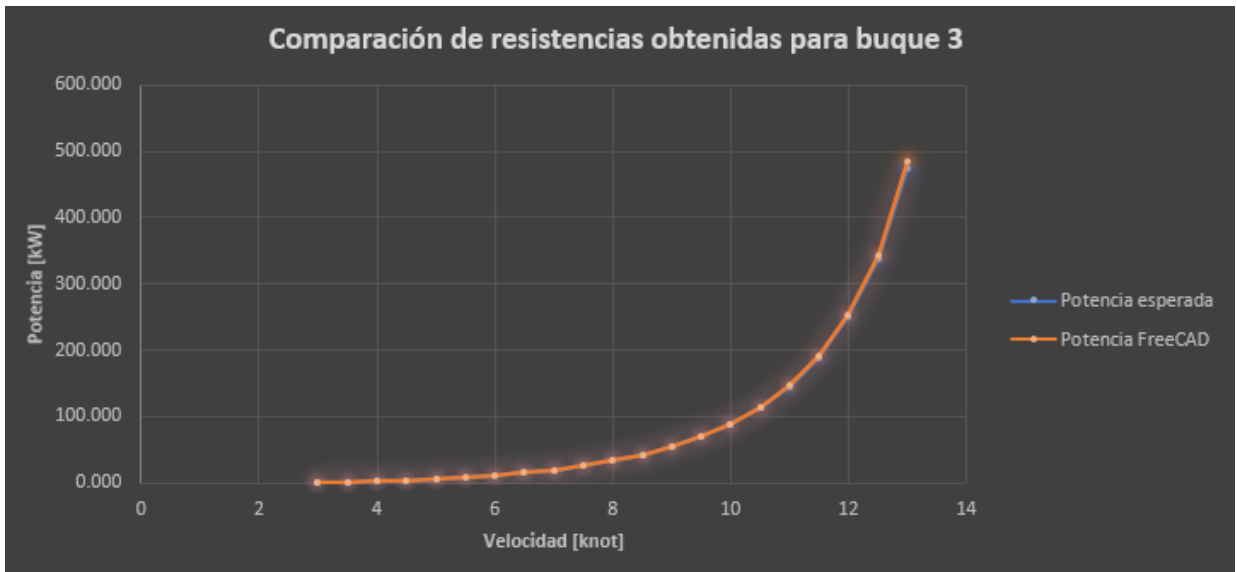


Figura D.11: Gráfica comparativa de las potencias efectivas obtenidas para el buque 3 en el método de Amadeo.



Figura D.12: Coeficientes resistivos del método de Amadeo obtenidos de *FreeCAD Ship* para el buque 3.



Tabla D.3: Comparación de valores obtenidos del método de Amadeo en *FreeCAD Ship* con los valores esperados para el buque 2.

Amadeo esperado			FreeCAD Amadeo				Diferencia
Speed (knot)	Resistance (kN)	Power(kW)	speed [m/s]	speed[knot]	resistance [kN]	EKW [kW]	%
3	0.773	1.193	1.543	3.000	0.777	1.200	0.57 %
3.5	1.076	1.937	1.800	3.500	1.090	1.962	1.26 %
4	1.443	2.969	2.058	4.000	1.466	3.016	1.59 %
4.5	1.88	4.352	2.315	4.500	1.912	4.427	1.73 %
5	2.395	6.160	2.572	5.000	2.436	6.266	1.72 %
5.5	3.001	8.490	2.829	5.500	3.046	8.617	1.49 %
6	3.707	11.441	3.086	6.000	3.751	11.578	1.19 %
6.5	4.531	15.150	3.344	6.500	4.566	15.266	0.77 %
7	5.488	19.761	3.601	7.000	5.506	19.825	0.32 %
7.5	6.601	25.467	3.858	7.500	6.591	25.428	-0.15 %
8	7.899	32.506	4.115	8.000	7.848	32.295	-0.65 %
8.5	9.414	41.162	4.372	8.500	9.309	40.704	-1.11 %
9	11.201	51.856	4.630	9.000	11.020	51.017	-1.62 %
9.5	13.314	65.063	4.887	9.500	13.038	63.714	-2.07 %
10	15.838	81.471	5.144	10.000	15.444	79.445	-2.49 %
10.5	18.889	102.023	5.401	10.500	18.350	99.114	-2.85 %
11	22.631	128.055	5.658	11.000	21.917	124.017	-3.15 %
11.5	27.313	161.573	5.916	11.500	26.385	156.080	-3.40 %
12	33.317	205.659	6.173	12.000	32.123	198.286	-3.59 %
12.5	41.276	265.405	6.430	12.500	39.737	255.511	-3.73 %
13	52.307	349.787	6.687	13.000	50.290	336.299	-3.86 %

Conclusiones.

En los tres tipos de buque se puede observar un aumento de la diferencia de resistencias obtenidas, con el aumento de las velocidades. Pero por lo general, las diferencias obtenidas son mínimas y, por lo tanto, asumibles.

Las diferencias encontradas pueden ser debidas a varios factores.

- Densidad y viscosidad cinemática. Los valores de ambas propiedades son distintos, en ello puede radicar cierta parte de la diferencia obtenida.
- Resistencia debida a la correlación modelo buque. Dicha resistencia varía considerablemente, ello puede ser debido a una actualización de las ecuaciones utilizadas.

Finalmente, además de los resultados obtenidos en este mismo anexo, se ha observado que los resultados obtenidos por el método de Holtrop-Mennen, pese a que varían, por motivos obvios, no se obtienen resistencias muy dispares, lo cual es otro indicativo de que el método de Amadeo introducido en *FreeCAD Ship* es válido. Con todo ello, se da por validado el método de Amadeo.



Anexo E

Verificación de las características principales obtenidas.

En el apartado 6.5 del capítulo 6, se realiza una verificación de las funciones introducidas para obtener las características principales del buque a través de un modelo en tres dimensiones en uno de los buques de ejemplo en FreeCAD, concretamente el modelo Series 60 de la Universidad de Iowa, y en un buque rápido. Sin embargo, la función que realiza el cálculo de la protuberancia de los buques no se ha podido verificar completamente, se ha comprobado que muestra un valor nulo en los casos probados, pero se desconoce el valor entregado en los casos de existencia de protuberancia.

A continuación, se explica el procedimiento seguido para verificar el cálculo de la protuberancia y se muestra una imagen de las formas del buque rápido utilizado en el capítulo 6, en el apartado 5.5.

Cálculo de la protuberancia

Al igual que el resto de funciones utilizadas, es necesario verificar el funcionamiento del cálculo de la protuberancia en FreeCAD. Para ello, deberá mostrar el resultado adecuado en todos los casos.

Hasta ahora, en las pruebas realizadas, los buques no contaban con protuberancia y los resultados obtenidos han sido correctos. Sin embargo, es de vital importancia comprobar que funciona adecuadamente en buques con bulbo.

Dentro de los límites del proyecto no se encontraba contemplado el modelado de un buque con bulbo para poder verificar dicha función, además, tampoco se contaba con un modelo en tres dimensiones con bulbo modelado de manera adecuada para poder verificar la función. Es por ello, por lo que finalmente se ha optado por realizar el cálculo con un sólido que esté conformado por un cuerpo rectangular, simulando el cuerpo del buque, y un cuerpo cilíndrico unido en la proa del cuerpo rectangular, simulando el bulbo. De esta manera, con un modelo mucho más sencillo se puede comprobar el funcionamiento de la herramienta. Para poder convertir el sólido en un buque es necesario realizar ciertas acciones que se especifican en H.

En la imagen E.1, se muestran las formas del sólido escogido para realizar las comprobaciones pertinentes para verificar el funcionamiento de la herramienta.

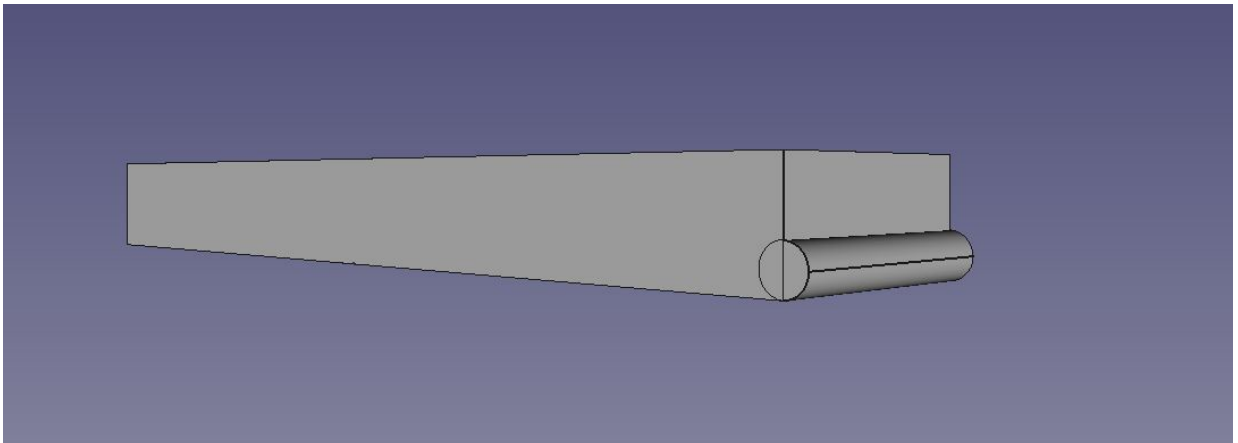


Figura E.1: Formas escogidas para verificar la funcionalidad de la protuberancia.

Tal y como se puede observar, es un sólido muy simple formado por dos objetos. La ventaja de crear este sólido, además de la simplicidad del mismo y verificar el funcionamiento del cálculo de la protuberancia, es que a la vez se verifican también el resto de las funcionalidades.

A continuación, en la tabla E.1, se incluyen las características principales del sólido utilizado.

Tabla E.1: Datos requeridos en el método de Holtrop

Dimensiones	Valores .
Lpp	50 m
T	3.5 m
B	11.25 m
Superficie mojada, Sw	1016.6 m ²
Eslora en la floración, Lw	50 m
Volumen desplazado, ∇	1968.75 m ³
Velocidades del buque, V	0 - 8 kN
Longitud de tobera, l	0 m
Diámetro propulsor, d	0 m
Longitud protuberancia bulbo, lb	1 m
Coefficiente del propulsor, η_p	0.6
Sea margin, %	15
Timón	No

Por último, se muestra, en la imagen E.2, las dimensiones obtenidas.

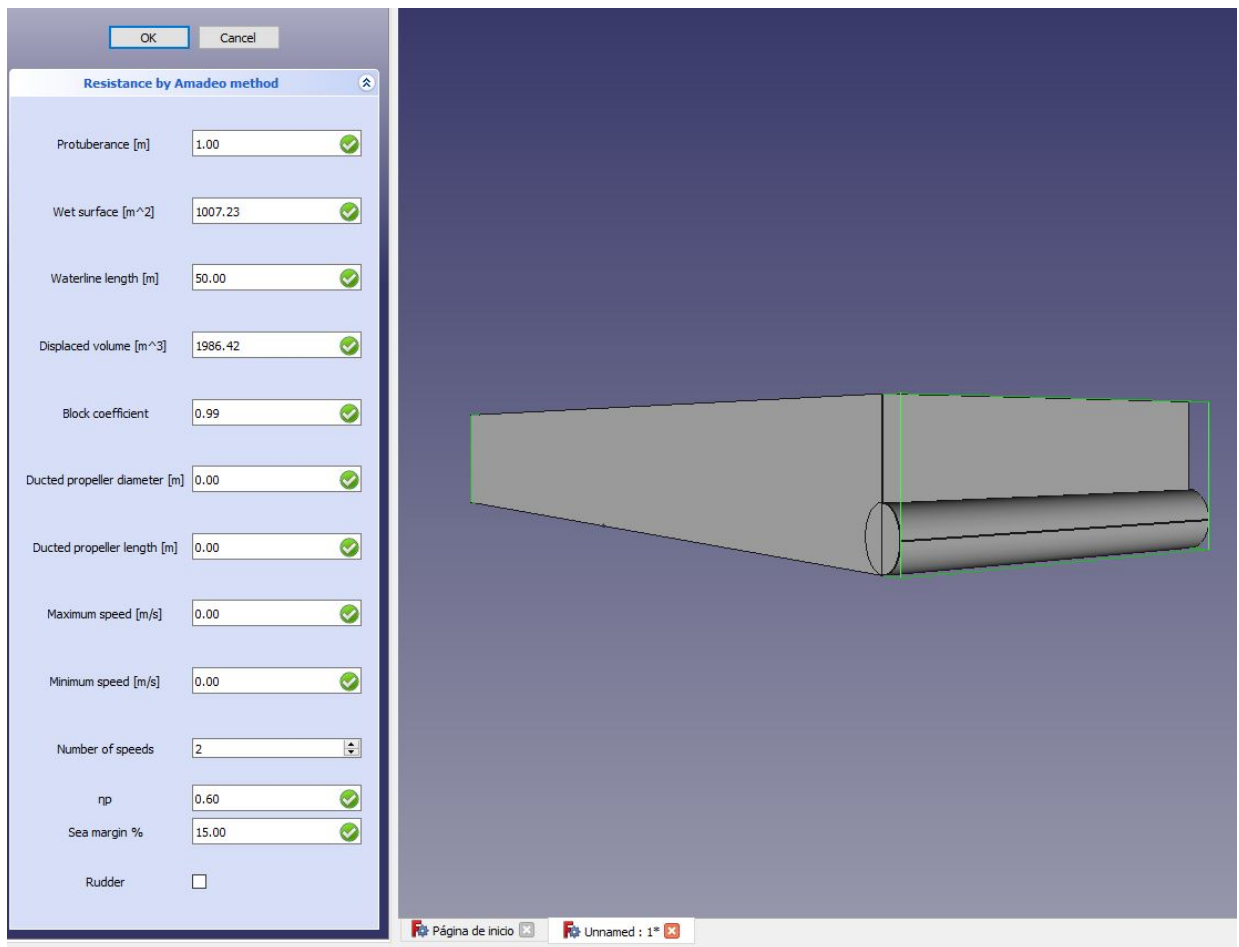


Figura E.2: Dimensiones obtenidas durante la verificación de la obtención de la protuberancia.

Se puede observar que las funciones trabajan de forma adecuada, únicamente se puede observar diferencia entre los valores obtenidos y los iniciales, en la superficie mojada y en el volumen desplazado. Sin embargo, la mayor diferencia se da en la superficie mojada y es una diferencia de un 0.9% únicamente. Por lo tanto, quedan validadas las funciones que obtienen las características principales del modelo en tres dimensiones.

Formas del buque rápido.

A continuación, en la imagen E.3, se muestran las formas del buque rápido utilizado para realizar la verificación de las funciones en el capítulo 6, en el apartado 6.5.



Figura E.3: Formas del buque rápido utilizado para la verificación de las funciones de obtención de dimensiones del modelo en tres dimensiones.



Anexo F

Postprocesado.

En el presente anexo se incluyen las explicaciones necesarias para realizar el postprocesado, así como el código relativo tanto al archivo TaskPanel, ya que la definición de las acciones a realizar en el postprocesado se incluye en este apartado, tal y como se explica en el capítulo 7, como el archivo PlotAux, donde se define la presentación de resultados.

Planificación de la presentación de resultados.

En primer lugar, es necesario planificar la presentación de resultados, ya que a través de los métodos semi-empíricos se obtienen los datos que se utilizarán para realizar el postprocesado.

En este caso, tal y como se comenta en el capítulo 7, la presentación de resultados se divide tanto en gráficas como en una hoja de cálculo que recoge los valores obtenidos.

El código que recoge los valores obtenidos del método semi-empírico y establece las acciones a realizar en el postprocesado, se recoge en el archivo TaskPanel, y todo el código de la presentación de resultados, tanto las gráficas como la hoja de cálculo, se realizará en el archivo PlotAux, propio de cada uno de los métodos.

Gráficas.

La creación de las gráficas se realiza con el paquete FreeCAD.Plot. Es un paquete cuyas funciones son muy similares a las funciones existentes en la librería matplotlib.pyplot. A continuación, se incluyen los pasos necesarios a seguir para la creación de una gráfica.

1. Definición de las variables que se van a procesar. Para ello se definirá una nueva gráfica siguiendo la siguiente sintaxis: “def plot(variables):”. Introduciendo en “variables” el nombre de las variables que se desee graficar.
2. Crear la gráfica. El siguiente paso es crear la ventana en FreeCAD donde se situará la gráfica. La sintaxis utilizada en este caso es: “plt = Plot.figure(‘título’)” Introduciendo en “Título”, el nombre de la ventana en la que se establecerá la gráfica.
3. Rellenar la gráfica. Se siguen varias sintaxis.
 - Gráfica. Para realizar la gráfica es necesario introducir: “nombre = Plot.plot(variables, r‘\$etiqueta\$’)”. En “nombre”, se introduce el nombre que se desee, servirá para per-



sonalizar la gráfica. Por otra parte, en “etiqueta”, se introduce la etiqueta que posteriormente se podrá observar en la leyenda.

- Personalización de la línea. Por último, en este apartado, se puede personalizar la línea por la que está formada la gráfica. Existen diversas sintaxis en función de la característica a modificar. En las sintaxis que se muestran a continuación, en “nombre” se debe introducir el nombre que se ha entregado a la gráfica.
 - Estilo. “nombre”.line.set_linestyle(‘propiedad’). Existen diversas propiedades, por ejemplo. ‘-’ para una línea sólida, ‘.’ para una línea de puntos, o ‘-’ para una línea discontinua, entre otras propiedades posibles.
 - Ancho de línea. “nombre”.line.set_linewidth(ancho). Introduciendo en “ancho” el valor del ancho de línea deseado, por ejemplo, 2.0.
 - Color. “nombre”.line.set_color((código de color)). En “código de color” es necesario introducir el código de color.
4. Configuración. Además de la configuración anterior existen más tipos de configuración, por ejemplo los ejes de la gráfica, el título...etc. En este caso, previamente a la configuración es necesario introducir la siguiente sintaxis: “ax = Plot.axes()”
- Título de la gráfica. “ax.set_title(“Título”, propiedades)”. En “propiedades” se pueden introducir distintas propiedades, como por ejemplo, el título subrayado, o en negrita, entre otras opciones.
 - Título de los ejes. “Plot.xlabel(r’\$título\$’)” o “Plot.ylabel(r’\$título\$’)”
 - Formato de los ejes. Para modificar el formato de los ejes e incluir los ejes en notación científica, por ejemplo, se utiliza la siguiente sintaxis: “plt.axes.ticklabel_format(axis = ‘y’, scilimits = (0,0))”. Se puede incluir también el eje x, o solo el eje x si se desee.
 - Posición de la gráfica. “ax.set_position([left, bottom, width, height])”
 - Leyenda. Es la última configuración, “Plot.legend(True, pos= (x,y))”. Siendo “x,y” las coordenadas de la leyenda. Por defecto, la leyenda se sitúa en la mejor zona posible, sin embargo, a través del comando “pos” se otorga la capacidad al usuario de desplazar la gráfica.

Además de las funcionalidades básicas comentadas anteriormente, existen otras muchas funcionalidades, como por ejemplo crear etiquetas de texto que se muestren en la gráfica, crear numerosos ejes en cada dirección...etc. En los apartados posteriores se integra el código PlotAux, código utilizado para la creación de las gráficas, de forma que se podrá observar la aplicación de las funciones mencionadas anteriormente.

Hoja de cálculo.

La creación de la hoja de cálculo es más sencilla que la creación de las gráficas. En esencia, consta de asignar a cada una de las celdas el valor que debe tomar, extrayendo dicho valor de los resultados obtenidos de los métodos semi-empíricos. A continuación, al igual que con la creación de las gráficas, se muestran los pasos a seguir para la creación de la hoja de cálculo que recoge los valores obtenidos.

1. Definir las variables. De la misma forma que ocurría en las gráficas, es necesario definir las variables que se mostrarán en la hoja de cálculo. Para ello, se sigue la siguiente sintaxis: “def spreadsheet(variables)”



2. Activar la hoja de cálculo. Se debe seguir la sintaxis mostrada a continuación: “s = FreeCAD.activeDocument().addObject(‘Spreadsheet::Sheet’,‘nombre deseado’)”. Introduciendo en “nombre deseado” el título de la hoja de cálculo.
3. Título de los encabezados. El primer paso es titular los encabezados bajo los que posteriormente se introducirán los valores correspondientes. Se muestra a continuación la sintaxis para una columna en concreto. “s.set(“A1”, “Título”)” .La sintaxis utilizada para el resto de columnas se realiza de la misma manera pero con las siguientes letras del abecedario.
4. Resto de valores. El resto de valores se rellenan a través de un bucle “for”. Es más sencillo entender el funcionamiento a través del código PlotAux incluido en los siguientes apartados.

De nuevo, se puede observar la aplicación de los pasos mencionados anteriormente en los apartados posteriores, en los archivos PlotAux.

Código TaskPanel método de Amadeo.

Se puede observar las acciones definidas que se deben realizar en el postprocesado del método de Amadeo en las líneas a , en el código que se muestra a continuación.

```

1 #*****
2 #*
3 #* Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com> *
4 #*
5 #* This program is free software; you can redistribute it and/or modify *
6 #* it under the terms of the GNU Lesser General Public License (LGPL) *
7 #* as published by the Free Software Foundation; either version 2 of *
8 #* the License, or (at your option) any later version. *
9 #* for detail see the LICENCE text file. *
10 #*
11 #* This program is distributed in the hope that it will be useful, *
12 #* but WITHOUT ANY WARRANTY; without even the implied warranty of *
13 #* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
14 #* GNU Library General Public License for more details. *
15 #*
16 #* You should have received a copy of the GNU Library General Public *
17 #* License along with this program; if not, write to the Free Software *
18 #* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 *
19 #* USA *
20 #*
21 #*****
22
23 import numpy as np
24 import FreeCAD as App
25 import FreeCADGui as Gui
26 from FreeCAD import Units
27 from PySide import QtGui, QtCore
28 from . import PlotAux
29 from . import Amadeo
30 from .. import Ship_rc
31 from ..import Instance
32 from ..shipUtils import Locale
33 from ..shipUtils import Selection

```



```

34 from ..shipHydrostatics import Tools as Hydrostatics
35 from ..init_gui import QT_TRANSLATE_NOOP
36
37
38 class TaskPanel:
39     def __init__(self):
40         self.name = "Compute resistance prediction Amadeo method"
41         self.ui = ":/ui/TaskPanel_resistanceAmadeo.ui"
42         self.form = Gui.PySideUic.loadUi(self.ui)
43         self.ship = None
44
45     def accept(self):
46         has_rudder = self.form.rudder.isChecked()
47         L = Units.parseQuantity(Locale.fromString(self.form.Lpp.text())).
            Value
48         B = Units.parseQuantity(Locale.fromString(self.form.Beam.text())).
            Value
49         T = Units.parseQuantity(Locale.fromString(self.form.Draft.text())).
            Value
50         prot = Units.parseQuantity(Locale.fromString(self.form.protuberance.
            text())).Value
51         Sw = Units.parseQuantity(Locale.fromString(self.form.Sw.text())).
            Value
52         Lw = Units.parseQuantity(Locale.fromString(self.form.Lw.text())).
            Value
53         V = Units.parseQuantity(Locale.fromString(self.form.volume.text())).
            Value
54         Cb = Units.parseQuantity(Locale.fromString(self.form.Cb.text())).
            Value
55         d = Units.parseQuantity(Locale.fromString(self.form.d_diameter.text()
            )).Value
56         l = Units.parseQuantity(Locale.fromString(self.form.d_length.text()
            )).Value
57         umax = Units.parseQuantity(Locale.fromString(self.form.max_speed.text
            ())).Value
58         umin = Units.parseQuantity(Locale.fromString(self.form.min_speed.text
            ())).Value
59         eta_p = Units.parseQuantity(Locale.fromString(self.form.eta_p.text()
            )).Value
60         seamargin = Units.parseQuantity(Locale.fromString(self.form.seamargin
            .text())).Value
61         n = self.form.n_speeds.value()
62
63
64         if 1 >= eta_p >= 0:
65
66             eta_p = eta_p
67
68         elif eta_p > 1:
69             msg = App.Qt.translate(
70                 "ship_console",
71                 "The propulsive coefficient cannot be higher than 1")
72             App.Console.PrintError(msg + '\n')
73
74         if Cb > 1:
75             msg = App.Qt.translate(
76                 "ship_console",

```



```

77         "The block coefficient cannot be higher than 1")
78         App.Console.PrintError(msg + '\n')
79
80     if Lw == 0:
81         Lw = ()
82         L_w = 1
83     else: L_w = 0
84     if Sw == 0:
85         Sw = ()
86         S_w = 1
87     else: S_w = 0
88     if d == 0: d = None
89     if l == 0: l = None
90     seamargin = seamargin / 100
91
92     speeds = np.linspace(umin, umax, num = n)
93     resis, speed, CF, CA, CR, CT, EKW, BKW, Lw, Sw = Amadeo.Amadeo(L, B, T
94     ,
95     Cb, V, speeds, etap, seamargin, prot, Sw, Lw, d, l, has_rudder =
96     has_rudder)
97
98     if L_w == 1: App.Console.PrintMessage("Lw = " + str("{:.3f}".format(
99     Lw)) + " m^2" + '\n')
100     if S_w == 1: App.Console.PrintMessage("Sw = " + str("{:.3f}".format(
101     Sw)) + " m^2" + '\n')
102
103     PlotAux.Plot(speed, resis, CF, CR, CA, CT, EKW, BKW)
104
105     return True
106
107 def reject(self):
108     return True
109
110 def clicked(self, index):
111     pass
112
113 def open(self):
114     pass
115
116 def needsFullSpace(self):
117     return True
118
119 def isAllowedAlterSelection(self):
120     return False
121
122 def isAllowedAlterView(self):
123     return True
124
125 def isAllowedAlterDocument(self):
126     return False
127
128 def helpRequested(self):
129     pass
130
131 def setupUi(self):
132     self.form.Lpp = self.widget(QtGui.QLineEdit, "Lpp")
133     self.form.Beam = self.widget(QtGui.QLineEdit, "Beam")

```



```

130 self.form.Draft = self.widget(QtGui.QLineEdit, "Draft")
131 self.form.protuberance = self.widget(QtGui.QLineEdit, "protuberance")
132 self.form.Sw = self.widget(QtGui.QLineEdit, "Sw")
133 self.form.Lw = self.widget(QtGui.QLineEdit, "Lw")
134 self.form.volume = self.widget(QtGui.QLineEdit, "volume")
135 self.form.Cb = self.widget(QtGui.QLineEdit, "Cb")
136 self.form.d_length = self.widget(QtGui.QLineEdit, "d_length")
137 self.form.d_diameter = self.widget(QtGui.QLineEdit, "d_diameter")
138 self.form.max_speed = self.widget(QtGui.QLineEdit, "max_speed")
139 self.form.min_speed = self.widget(QtGui.QLineEdit, "min_speed")
140 self.form.n_speeds = self.widget(QtGui.QSpinBox, "n_speeds")
141 self.form.etap = self.widget(QtGui.QLineEdit, "etap")
142 self.form.seamargin = self.widget(QtGui.QLineEdit, "seamargin")
143 self.form.rudder = self.widget(QtGui.QCheckBox, "rudder")
144 if self.initValues():
145     return True
146
147 def getMainWindow(self):
148     toplevel = QtGui.QApplication.topLevelWidgets()
149     for i in toplevel:
150         if i.metaObject().className() == "Gui::MainWindow":
151             return i
152     raise RuntimeError("No main window found")
153
154 def widget(self, class_id, name):
155     """Return the selected widget.
156
157     Keyword arguments:
158     class_id -- Class identifier
159     name -- Name of the widget
160     """
161     mw = self.getMainWindow()
162     form = mw.findChild(QtGui.QWidget, "AmadeoTaskPanel")
163     return form.findChild(class_id, name)
164
165 def initValues(self):
166     """ Set initial values for fields
167     """
168     sel_ships = Selection.get_ships()
169     try:
170         self.ship = sel_ships[0]
171     except:
172         pass
173     if len(sel_ships) > 1:
174         msg = App.Qt.translate(
175             "ship_console",
176             "More than one ship have been selected (just the one labelled
177             "
178             "'{}' is considered)".format(self.ship.Label))
179         App.Console.PrintWarning(msg + '\n')
180
181     etap = 0.6
182     seamargin = 15
183     self.form.etap.setText(str(etap))
184     self.form.seamargin.setText(str(seamargin))
185
186     try:

```



```

186     disp,_, cb = Hydrostatics.displacement(self.ship,
187                                         self.ship.Draft,
188                                         Units.parseQuantity("0 deg
189                                         "),
190                                         Units.parseQuantity("0 deg
191                                         "))
192     vol = disp / Hydrostatics.DENS
193
194     sw = Hydrostatics.wettedArea(self.ship.Shape.copy(), self.ship.
195     Draft,
196                                     Units.parseQuantity("0 deg"),
197                                     Units.parseQuantity("0 deg"))
198
199     shape, _ = Hydrostatics.placeShipShape(self.ship.Shape.copy(),
200     self.ship.Draft,
201     Units.parseQuantity("0 deg
202     "),
203     Units.parseQuantity("0 deg
204     "))
205     shape = Hydrostatics.getUnderwaterSide(shape)
206     bbox = shape.BoundingBox
207
208     prot = Units.Quantity(bbox.XMax - bbox.XMin, Units.Length)
209     prot = prot - self.ship.Length
210     if prot < 0: prot = 0
211     prot = Units.Quantity(prot, Units.Length)
212
213     area, cf, f = Hydrostatics.floatingArea(self.ship, self.ship.
214     Draft,
215     Units.parseQuantity("0
216     deg"),
217     Units.parseQuantity("0
218     deg"))
219
220     bbox = f.BoundingBox
221     lw = Units.Quantity(bbox.XMax - bbox.XMin, Units.Length)
222
223     Prot = prot.getValueAs("m").Value
224     Sw = sw.getValueAs("m^2").Value
225     Lw = lw.getValueAs("m").Value
226     V = vol.getValueAs("m^3").Value
227
228     except:
229         Prot = 0.0
230         Sw = 0.0
231         Lw = 0.0
232         V = 0.0
233         cb = 0.0
234
235     try:
236         lpp = self.ship.Length.getValueAs("m").Value
237         B = self.ship.Breadth.getValueAs("m").Value
238         T = self.ship.Draft.getValueAs("m").Value
239     except:
240         lpp = 0.0
241         B = 0.0

```



```

235         T = 0.0
236
237         self.form.Lpp.setText(str(lpp))
238         self.form.Beam.setText(str(B))
239         self.form.Draft.setText(str(T))
240         self.form.protuberance.setText(str(Prot))
241         self.form.Lw.setText(str(Lw))
242         self.form.Sw.setText(str(Sw))
243         self.form.volume.setText(str(V))
244         self.form.Cb.setText(str(cb))
245
246         return False
247
248 def createTask():
249     panel = TaskPanel()
250     Gui.Control.showDialog(panel)
251     if panel.setupUi():
252         Gui.Control.closeDialog()
253         return None
254     return panel

```

Código TaskPanel método de Holtrop.

Se puede observar las acciones definidas que se deben realizar en el postprocesado del método de Holtrop en las líneas a , en el código que se muestra a continuación.

```

1  #*****
2  #*
3  #*   Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com>
4  #*
5  #*   This program is free software; you can redistribute it and/or modify
6  #*   it under the terms of the GNU Lesser General Public License (LGPL)
7  #*   as published by the Free Software Foundation; either version 2 of
8  #*   the License, or (at your option) any later version.
9  #*   for detail see the LICENCE text file.
10 #*
11 #*   This program is distributed in the hope that it will be useful,
12 #*   but WITHOUT ANY WARRANTY; without even the implied warranty of
13 #*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 #*   GNU Library General Public License for more details.
15 #*
16 #*   You should have received a copy of the GNU Library General Public
17 #*   License along with this program; if not, write to the Free Software
18 #*   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
19 #*   USA
20 #*
21 #*****
22
23 import numpy as np
24 import FreeCAD as App
25 import FreeCADGui as Gui
26 from FreeCAD import Units
27 from PySide import QtGui, QtCore
28 from . import PlotAux
29 from . import Holtrop
30 from .. import Ship_rc

```



```

31 from ..import Instance
32 from ..shipUtils import Locale
33 from ..shipUtils import Selection
34 from ..shipHydrostatics import Tools as Hydrostatics
35 from ..init_gui import QT_TRANSLATE_NOOP
36
37
38 class TaskPanel:
39     def __init__(self):
40         self.name = "Compute resistance prediction Holtrop method"
41         self.ui = ":/ui/TaskPanel_resistanceHoltrop.ui"
42         self.form = Gui.PySideUic.loadUi(self.ui)
43         self.ship = None
44
45     def accept(self):
46         B = Units.parseQuantity(Locale.fromString(self.form.Beam.text())).
47             Value
48         T = Units.parseQuantity(Locale.fromString(self.form.Draft.text())).
49             Value
50         Sw = Units.parseQuantity(Locale.fromString(self.form.Sw.text())).
51             Value
52         Lw = Units.parseQuantity(Locale.fromString(self.form.Lw.text())).
53             Value
54         V = Units.parseQuantity(Locale.fromString(self.form.volume.text())).
55             Value
56         Cb = Units.parseQuantity(Locale.fromString(self.form.Cb.text())).
57             Value
58         Cm = Units.parseQuantity(Locale.fromString(self.form.Cm.text())).
59             Value
60         Cw = Units.parseQuantity(Locale.fromString(self.form.Cf.text())).
61             Value
62         cstern = self.form.form.currentIndex()
63         iE = Units.parseQuantity(Locale.fromString(self.form.iE.text())).
64             Value
65         xcb = Units.parseQuantity(Locale.fromString(self.form.xcb.text())).
66             Value
67         ABT = Units.parseQuantity(Locale.fromString(self.form.ABT.text())).
68             Value
69         AT = Units.parseQuantity(Locale.fromString(self.form.AT.text())).
70             Value
71         hb = Units.parseQuantity(Locale.fromString(self.form.hb.text())).
72             Value
73         umax = Units.parseQuantity(Locale.fromString(self.form.max_speed.text
74             ())).Value
75         umin = Units.parseQuantity(Locale.fromString(self.form.min_speed.text
76             ())).Value
77         eta_p = Units.parseQuantity(Locale.fromString(self.form.eta_p.text ()))
78             .Value
79         seamargin = Units.parseQuantity(Locale.fromString(self.form.seamargin
80             .text ())).Value
81         rbskeg = Units.parseQuantity(Locale.fromString(self.form.rbskeg.text
82             ())).Value
83         rbstern = Units.parseQuantity(Locale.fromString(self.form.rbstern.
84             text ())).Value
85         twsbr = Units.parseQuantity(Locale.fromString(self.form.twsbr.text ()))
86             .Value
87         sbr = Units.parseQuantity(Locale.fromString(self.form.sbr.text ())).

```



```

68     Value
        skeg = Units.parseQuantity(Locale.fromString(self.form.skeg.text())).
        Value
69     strut_bossing = Units.parseQuantity(Locale.fromString(self.form.
        strut_bossing.text())).Value
70     hull_bossings = Units.parseQuantity(Locale.fromString(self.form.
        hull_bossings.text())).Value
71     shafts = Units.parseQuantity(Locale.fromString(self.form.shafts.text
        ())).Value
72     stab_fins = Units.parseQuantity(Locale.fromString(self.form.stab_fins
        .text())).Value
73     dome = Units.parseQuantity(Locale.fromString(self.form.dome.text())).
        Value
74     bkl = Units.parseQuantity(Locale.fromString(self.form.bkl.text())).
        Value
75     n = self.form.n_speeds.value()
76
77
78     if 1 >= eta_p >= 0:
79
80         etap = eta_p
81
82     elif eta_p > 1:
83         msg = App.Qt.translate(
84             "ship_console",
85             "The propulsive coefficient cannot be higher than 1")
86         App.Console.PrintError(msg + '\n')
87
88     if Cb > 1:
89         msg = App.Qt.translate(
90             "ship_console",
91             "The block coefficient cannot be higher than 1")
92         App.Console.PrintError(msg + '\n')
93
94     if Cw > 1:
95         msg = App.Qt.translate(
96             "ship_console",
97             "The waterplane coefficient cannot be higher than 1")
98         App.Console.PrintError(msg + '\n')
99
100    if Sw == 0:
101        Sw = ()
102        S_w = 1
103    else: S_w = 0
104    seamargin = seamargin / 100
105
106    Sapplist = [rbskeg, rbstern, twsbr, sbr, skeg, strut_bossing,
107                hull_bossings, shafts, stab_fins, dome, bkl]
108
109
110    speeds = np.linspace(umin, umax, num = n)
111    Rtotal, speed, CT, CF, CAPP, Cw, CB, CTR, CA, EKW, BKW, Sw = Holtrop.
        Holtrop( B,
112                T, Lw, V, Cb, Cm, Cw, cstern, iE, xcb, speeds, hb, etap,
        seamargin,
113
        Sapplist, ABT, AT, Sw
        )

```



```

114
115     if S_w == 1: App.Console.PrintMessage("Sw = " + str("{:.3f}".format(
116         Sw)) + " m^2" + '\n')
117
118     PlotAux.Plot(speed, Rtotal, CT, CF, CAPP, Cw, CB, CTR, CA, EKW, BKW)
119     return True
120
121 def reject(self):
122     return True
123
124 def clicked(self, index):
125     pass
126
127 def open(self):
128     pass
129
130 def needsFullSpace(self):
131     return True
132
133 def isAllowedAlterSelection(self):
134     return False
135
136 def isAllowedAlterView(self):
137     return True
138
139 def isAllowedAlterDocument(self):
140     return False
141
142 def helpRequested(self):
143     pass
144
145 def setupUi(self):
146
147     self.form.Beam = self.widget(QtGui.QLineEdit, "Beam")
148     self.form.Draft = self.widget(QtGui.QLineEdit, "Draft")
149     self.form.Sw = self.widget(QtGui.QLineEdit, "Sw")
150     self.form.Lw = self.widget(QtGui.QLineEdit, "Lw")
151     self.form.volume = self.widget(QtGui.QLineEdit, "volume")
152     self.form.Cb = self.widget(QtGui.QLineEdit, "Cb")
153     self.form.Cm = self.widget(QtGui.QLineEdit, "Cm")
154     self.form.Cf = self.widget(QtGui.QLineEdit, "Cf")
155     self.form.form = self.widget(QtGui.QComboBox, "form")
156     self.form.iE = self.widget(QtGui.QLineEdit, "iE")
157     self.form.xcb = self.widget(QtGui.QLineEdit, "xcb")
158     self.form.ABT = self.widget(QtGui.QLineEdit, "ABT")
159     self.form.AT = self.widget(QtGui.QLineEdit, "AT")
160     self.form.hb = self.widget(QtGui.QLineEdit, "hb")
161     self.form.max_speed = self.widget(QtGui.QLineEdit, "max_speed")
162     self.form.min_speed = self.widget(QtGui.QLineEdit, "min_speed")
163     self.form.n_speeds = self.widget(QtGui.QSpinBox, "n_speeds")
164     self.form.etap = self.widget(QtGui.QLineEdit, "etap")
165     self.form.seamargin = self.widget(QtGui.QLineEdit, "seamargin")
166     self.form.rbskeg = self.widget(QtGui.QLineEdit, "rbskeg")
167     self.form.rbstern = self.widget(QtGui.QLineEdit, "rbstern")
168     self.form.twsbr = self.widget(QtGui.QLineEdit, "twsbr")
169     self.form.sbr = self.widget(QtGui.QLineEdit, "sbr")
170     self.form.skeg = self.widget(QtGui.QLineEdit, "skeg")

```



```

170     self.form.strut_bossing = self.widget(QtGui.QLineEdit, "strut_bossing
171         ")
172     self.form.hull_bossings = self.widget(QtGui.QLineEdit, "hull_bossings
173         ")
174     self.form.shafts = self.widget(QtGui.QLineEdit, "shafts")
175     self.form.stab_fins = self.widget(QtGui.QLineEdit, "stab_fins")
176     self.form.dome = self.widget(QtGui.QLineEdit, "dome")
177     self.form.bkl = self.widget(QtGui.QLineEdit, "bkl")
178     if self.initValues():
179         return True
180
181     def getMainWindow(self):
182         toplevel = QtGui.QApplication.topLevelWidgets()
183         for i in toplevel:
184             if i.metaObject().className() == "Gui::MainWindow":
185                 return i
186         raise RuntimeError("No main window found")
187
188     def widget(self, class_id, name):
189         """Return the selected widget.
190
191         Keyword arguments:
192         class_id -- Class identifier
193         name -- Name of the widget
194         """
195         mw = self.getMainWindow()
196         form = mw.findChild(QtGui.QWidget, "HoltropTaskPanel")
197         return form.findChild(class_id, name)
198
199     def initValues(self):
200         """ Set initial values for fields
201         """
202         sel_ships = Selection.get_ships()
203         try:
204             self.ship = sel_ships[0]
205         except:
206             pass
207         if len(sel_ships) > 1:
208             msg = App.Qt.translate(
209                 "ship_console",
210                 "More than one ship have been selected (just the one labelled
211                 "
212                 "'{}' is considered)".format(self.ship.Label))
213             App.Console.PrintWarning(msg + '\n')
214
215         etap = 0.6
216         seamargin = 15
217         self.form.etap.setText(str(etap))
218         self.form.seamargin.setText(str(seamargin))
219
220         try:
221             disp, Vector, cb = Hydrostatics.displacement(self.ship,
222                 self.ship.Draft,
223                 Units.parseQuantity("0 deg

```



```

223         vol = disp / Hydrostatics.DENS
224
225         sw = Hydrostatics.wettedArea(self.ship.Shape.copy(), self.ship.
226             Draft,
227             Units.parseQuantity("0 deg"),
228             Units.parseQuantity("0 deg"))
229
230         area, cf, f = Hydrostatics.floatingArea(self.ship, self.ship.
231             Draft,
232             Units.parseQuantity("0
233                 deg"),
234             Units.parseQuantity("0
235                 deg"))
236
237         bbox = f.BoundingBox
238         lw = Units.Quantity(bbox.XMax - bbox.XMin, Units.Length)
239
240         cm = Hydrostatics.mainFrameCoeff(self.ship)
241
242         xcb = Units.Quantity(Vector[0], Units.Length)
243
244         cp = cb / cm
245         Lw = lw.getValueAs("m").Value
246         Xcb = xcb.getValueAs("m").Value
247         Bie = self.ship.Breadth.getValueAs("m").Value
248         V = vol.getValueAs("m^3").Value
249         Sw = sw.getValueAs("m^2").Value
250
251         Lr = Lw * (1 - cp + (0.06 * cp * Xcb) / (4 * cp - 1))
252
253         try:
254             iE = 1 + 89 * np.exp(- (Lw / Bie) ** 0.80856 * (1 - cf) **
255                 0.30484
256                 * (1 - cp - 0.0225 * Xcb) ** 0.6367 * (Lr /
257                     Bie) **
258                     0.34574 * ((100 * V) / (Lw ** 3)) **
259                     0.16302)
260
261         except ZeroDivisionError:
262             msg = App.Qt.translate(
263                 "ship_console",
264                 "ZeroDivisionError: Null ship floating area found during the"
265                 " floating area computation!")
266             App.Console.PrintError(msg + '\n')
267
268         iE = 0.0
269
270     except:
271         Lw = 0.0
272         Sw = 0.0
273         V = 0.0
274         Xcb = 0.0
275         cb = 0.0
276         cm = 0.0
277         cf = 0.0
278         iE = 0.0

```



```

272     try:
273         B = self.ship.Breadth.getValueAs("m").Value
274         T = self.ship.Draft.getValueAs("m").Value
275     except:
276         B = 0.0
277         T = 0.0
278
279     self.form.Beam.setText(str(B))
280     self.form.Draft.setText(str(T))
281     self.form.Lw.setText(str(Lw))
282     self.form.Sw.setText(str(Sw))
283     self.form.volume.setText(str(V))
284     self.form.Cb.setText(str(cb))
285     self.form.Cm.setText(str(cm))
286     self.form.Cf.setText(str(cf))
287     self.form.iE.setText(str(iE))
288     self.form.xcb.setText(str(Xcb))
289     return False
290
291 def createTask():
292     panel = TaskPanel()
293     Gui.Control.showDialog(panel)
294     if panel.setupUi():
295         Gui.Control.closeDialog()
296         return None
297     return panel

```

Código de postprocesado, PlotAux, método de Amadeo.

A continuación, se muestra el código utilizado para el postprocesado relativo al método de Amadeo.

```

1  #*****
2  #*
3  #*   Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com>
4  #*
5  #*   This program is free software; you can redistribute it and/or modify
6  #*   it under the terms of the GNU Lesser General Public License (LGPL)
7  #*   as published by the Free Software Foundation; either version 2 of
8  #*   the License, or (at your option) any later version.
9  #*   for detail see the LICENCE text file.
10 #*
11 #*   This program is distributed in the hope that it will be useful,
12 #*   but WITHOUT ANY WARRANTY; without even the implied warranty of
13 #*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14 #*   GNU Library General Public License for more details.
15 #*
16 #*   You should have received a copy of the GNU Library General Public
17 #*   License along with this program; if not, write to the Free Software
18 #*   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
19 #*   USA
20 #*
21 #*****
22
23 import os
24 import sys

```



```

25 import numpy as np
26 import FreeCAD
27 from FreeCAD import Base
28 import Spreadsheet
29
30 class Plot(object):
31     def __init__(self, speed, resis, CF, CR, CA, CT, EKW, BKW):
32         """ Constructor. performs the plot and shows it.
33         @param Speed, Ship speed.
34         @param Resis, Resistance computed.
35         @param CF, Frictional resistance coefficient.
36         @param CR, residual resistance coefficient.
37         @param CA, Roughness coefficient
38         @param CT, Total resistance coefficient.
39         @param EKW, Efficient power.
40         @param BKW, Break power.
41         """
42         self.plot(speed, resis)
43         self.plotPower(speed, EKW, BKW)
44         self.plotCoeff(speed, CF, CR, CT, CA)
45         self.spreadSheet(speed, resis, CF, CR, CA, CT, EKW, BKW)
46
47     def plot(self, speed, resis):
48         """ Perform Amadeo resistance pplot.
49         @param Speed, Ship speed.
50         @param Resis, Resistance computed.
51         @return True if error happens.
52         """
53         try:
54             from FreeCAD.Plot import Plot
55         except ImportError:
56             try:
57                 from freecad.plot import Plot
58             except ImportError:
59                 msg = FreeCAD.Qt.translate(
60                     "ship_console",
61                     "Plot module is disabled")
62                 FreeCAD.Console.PrintWarning(msg + '\n')
63                 return True
64
65         plt = Plot.figure('Resistance')
66         self.plt = plt
67
68         # Write axes titles
69         ax = Plot.axes()
70         Plot.xlabel(r'$Speed \; \mathrm{m/s}$')
71         Plot.ylabel(r'$Resistance \; \mathrm{kN}$')
72         ax.spines['right'].set_color((0.0, 0.0, 0.0))
73         ax.spines['top'].set_color((0.0, 0.0, 0.0))
74         ax.xaxis.label.set_fontsize(15)
75         ax.yaxis.label.set_fontsize(15)
76         ax.set_title('Amadeo Resistance vs Speed', fontweight='bold')
77
78         resis = Plot.plot(speed, resis, r'$Resistance \; \mathrm{kN}$')
79         resis.line.set_linestyle('-')
80         resis.line.set_linewidth(2.0)
81         resis.line.set_color((0.0, 0.0, 0.0))

```



```

82
83     # Show grid
84     Plot.grid(True)
85     # End
86     return False
87
88 def plotPower(self, speed, EKW, BKW):
89     """ Perform Amadeo power plot.
90     @param Speed, Ship speed.
91     @param EKW, Efficient power.
92     @param BKW, Break power.
93     @return True if error happens.
94     """
95     try:
96         from FreeCAD.Plot import Plot
97     except ImportError:
98         try:
99             from freecad.plot import Plot
100        except ImportError:
101            msg = FreeCAD.Qt.translate(
102                "ship_console",
103                "Plot module is disabled")
104            FreeCAD.Console.PrintWarning(msg + '\n')
105            return True
106
107     plt = Plot.figure('Power')
108     self.plt3 = plt
109
110     # Write axes titles
111     ax = Plot.axes()
112     ax.set_position([0.1, 0.1, 0.65, 0.8])
113     Plot.xlabel(r'$Speed \; \mathrm{m/s}$')
114     Plot.ylabel(r'$Power \; \mathrm{kW}$')
115     ax.spines['right'].set_color((0.0, 0.0, 0.0))
116     ax.spines['top'].set_color((0.0, 0.0, 0.0))
117     ax.xaxis.label.set_fontsize(15)
118     ax.yaxis.label.set_fontsize(15)
119     ax.set_title('Amadeo Power vs Speed', fontweight='bold')
120
121     EKW = Plot.plot(speed, EKW, r'$EKW, \; Efficient \; power$')
122     EKW.line.set_linestyle('-')
123     EKW.line.set_linewidth(2.0)
124     EKW.line.set_color((0.0, 0.0, 1.0))
125     self.EKW = EKW
126
127     BKW = Plot.plot(speed, BKW, r'$BKW, \; Break \; power$')
128     BKW.line.set_linestyle('-')
129     BKW.line.set_linewidth(2.0)
130     BKW.line.set_color((1.0, 0.0, 0.0))
131     self.BKW = BKW
132
133
134     # Show grid
135     Plot.grid(True)
136     Plot.legend(True, pos =(1.05,0.6))
137     # End
138     return False

```



```

139
140 def plotCoeff(self, speed, CF, CR ,CT, CA):
141     """ Perform Amadeo coefficients plot.
142     @param speed, Ship speed.
143     @param CF, Frictional resistance coefficient.
144     @param CR, Residual resistance coefficient.
145     @param CA, Roughness coefficient
146     @param CT, Total resistance coefficient.
147     @return True if error happens.
148     """
149     try:
150         from FreeCAD.Plot import Plot
151     except ImportError:
152         try:
153             from freecad.plot import Plot
154         except ImportError:
155             msg = FreeCAD.Qt.translate(
156                 "ship_console",
157                 "Plot module is disabled")
158             FreeCAD.Console.PrintWarning(msg + '\n')
159             return True
160
161
162     plt = Plot.figure('Coefficients')
163     self.plt2 = plt
164
165     ax = Plot.axes()
166     ax.set_position([0.05, 0.1, 0.65, 0.8])
167     Plot.xlabel(r'$Speed \; \mathrm{m/s}$')
168     Plot.ylabel(r'$Resistance \; coefficients$')
169     ax.spines['right'].set_color((0.0, 0.0, 0.0))
170     ax.spines['top'].set_color((0.0, 0.0, 0.0))
171     ax.xaxis.label.set_fontsize(15)
172     ax.yaxis.label.set_fontsize(15)
173     ax.set_title('Amadeo Resistance Coefficients vs Speed', fontweight='
174         bold')
175
176     plt.axes.ticklabel_format(axis = 'y', scilimits = (0,0))
177
178     CT = Plot.plot(speed, CT, r'$CT$, Total resistance coefficient')
179     CT.line.set_linestyle('-')
180     CT.line.set_linewidth(2.0)
181     CT.line.set_color((0.0, 0.0, 1.0))
182     self.CT = CT
183
184     CF = Plot.plot(speed, CF, r'$CF$, Frictional resistance coefficient')
185     CF.line.set_linestyle('-')
186     CF.line.set_linewidth(2.0)
187     CF.line.set_color((0.0, 0.0, 0.0))
188     self.CF = CF
189
190     CR = Plot.plot(speed, CR, r'$CR$, Residual resistance coefficient')
191     CR.line.set_linestyle('-')
192     CR.line.set_linewidth(2.0)
193     CR.line.set_color((1.0, 0.0, 0.0))
194     self.CR = CR

```



```

195 CA = Plot.plot(speed, CA, r'$CA$, Roughness coefficient')
196 CA.line.set_linestyle('-')
197 CA.line.set_linewidth(2.0)
198 CA.line.set_color((0.2, 0.8, 0.2))
199 self.CA = CA
200
201
202 # Show grid
203 Plot.grid(True)
204 Plot.legend(True, pos =(1.025,0.6))
205 # End
206 return False
207
208 def spreadSheet(self, speed, resis, CF, CR, CA, CT, EKW, BKW):
209     """ Write the output data file.
210     @param Speed, Ship speed.
211     @param Resis, Resistance computed.
212     @param CF, Frictional resistance coefficient.
213     @param CR, residual resistance coefficient.
214     @param CA, Roughness coefficient
215     @param CT, Total resistance coefficient.
216     @param EKW, Efficient power.
217     @param BKW, Break power.
218     """
219     s = FreeCAD.activeDocument().addObject('Spreadsheet::Sheet',
220                                             'Resistance Amadeo method')
221
222     # Print the header
223     s.set("A1", "Speed [m/s]")
224     s.set("B1", "Resistance [kN]")
225     s.set("C1", "CF * 10^3")
226     s.set("D1", "CR * 10^3")
227     s.set("E1", "CA * 10^3")
228     s.set("F1", "CT * 10^3")
229     s.set("G1", "EKW [kW]")
230     s.set("H1", "BKW [kW]")
231
232
233     # Print the data
234     for i in range(len(resis)):
235         s.set("A{}".format(i + 2), str(speed[i]))
236         s.set("B{}".format(i + 2), str(resis[i]))
237         s.set("C{}".format(i + 2), str(CF[i] * 1000))
238         s.set("D{}".format(i + 2), str(CR[i] * 1000))
239         s.set("E{}".format(i + 2), str(CA[i] * 1000))
240         s.set("F{}".format(i + 2), str(CT[i] * 1000))
241         s.set("G{}".format(i + 2), str(EKW[i]))
242         s.set("H{}".format(i + 2), str(BKW[i]))
243
244     # Recompute
245     FreeCAD.activeDocument().recompute()

```



Código de postprocesado, PlotAux, método de Holtrop.

A continuación, se muestra el código utilizado para el postprocesado relativo al método de Holtrop.

```

1 #*****
2 #*
3 #* Copyright (c) 2011, 2016 Jose Luis Cercos Pita <jlcercos@gmail.com> *
4 #*
5 #* This program is free software; you can redistribute it and/or modify *
6 #* it under the terms of the GNU Lesser General Public License (LGPL) *
7 #* as published by the Free Software Foundation; either version 2 of *
8 #* the License, or (at your option) any later version. *
9 #* for detail see the LICENCE text file. *
10 #*
11 #* This program is distributed in the hope that it will be useful, *
12 #* but WITHOUT ANY WARRANTY; without even the implied warranty of *
13 #* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
14 #* GNU Library General Public License for more details. *
15 #*
16 #* You should have received a copy of the GNU Library General Public *
17 #* License along with this program; if not, write to the Free Software *
18 #* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 *
19 #* USA *
20 #*
21 #*****
22
23 import os
24 import sys
25 import math
26 import FreeCAD
27 from FreeCAD import Base
28 import Spreadsheet
29
30
31 class Plot(object):
32     def __init__(self, speed, Rtotal, CT, CF, CAPP, Cw, CB, CTR, CA, EKW, BKW
33 ):
34         """ Constructor. performs the plot and shows it.
35         @param Speed, Ship speed.
36         @param Rtotal, Resistance computed.
37         @param CT, Total resistance coefficient
38         @param CF, Frictional resistance coefficient.
39         @param CAPP, Appendage resistance coefficient.
40         @param Cw, Wave resistance coefficient.
41         @param CB, Additional resistance coefficient due to the presence of a
42                 bulbous bow near the surface
43                 transom.
44         @param CTR, Additional resistance coefficient due to the immersed.
45         @param CA, Model-ship correlation resistance coefficient.
46         @param EKW, Efficient power.
47         @param BKW, Break power.
48         """
49         self.plot(speed, Rtotal)
50         self.plotPower(speed, EKW, BKW)
51         self.plotCoeff(speed, CT, CF, CAPP, Cw, CB, CTR, CA)
52         self.spreadSheet(speed, Rtotal, CT, CF, CAPP, Cw, CB, CTR, CA, EKW,

```



```

    BKW)
52
53 def plot(self, speed, Rtotal):
54     """ Perform Holtrop resistance plot.
55     @param Speed, Ship speed.
56     @param Rtotal, Resistance computed.
57     @return True if error happens.
58     """
59     try:
60         from FreeCAD.Plot import Plot
61     except ImportError:
62         try:
63             from freecad.plot import Plot
64         except ImportError:
65             msg = FreeCAD.Qt.translate(
66                 "ship_console",
67                 "Plot module is disabled")
68             FreeCAD.Console.PrintWarning(msg + '\n')
69             return True
70
71     plt = Plot.figure('Resistance')
72     self.plt = plt
73
74     ax = Plot.axes()
75     Plot.xlabel(r'$Speed \; \mathrm{m/s}$')
76     Plot.ylabel(r'$Resistance \; \mathrm{kN}$')
77     ax.spines['right'].set_color((0.0, 0.0, 0.0))
78     ax.spines['top'].set_color((0.0, 0.0, 0.0))
79     ax.xaxis.label.set_fontsize(15)
80     ax.yaxis.label.set_fontsize(15)
81     ax.set_title('Holtrop Resistance vs Speed', fontweight='bold')
82
83     areas = Plot.plot(speed, Rtotal, 'Resistance Holtrop method')
84     areas.line.set_linestyle('--')
85     areas.line.set_linewidth(2.0)
86     areas.line.set_color((0.0, 0.0, 0.0))
87
88     # Show grid
89     Plot.grid(True)
90     # End
91     return False
92
93 def plotPower(self, speed, EKW, BKW):
94     """ Perform Holtrop power plot.
95     @param Speed, Ship speed.
96     @param EKW, Efficient power.
97     @param BKW, Break power.
98     @param ship Active ship instance.
99     @return True if error happens.
100     """
101     try:
102         from FreeCAD.Plot import Plot
103     except ImportError:
104         try:
105             from freecad.plot import Plot
106         except ImportError:
107             msg = FreeCAD.Qt.translate(

```



```

108         "ship_console",
109         "Plot module is disabled")
110     FreeCAD.Console.PrintWarning(msg + '\n')
111     return True
112
113     plt = Plot.figure('Power')
114     self.plt3 = plt
115
116     # Write axes titles
117     ax = Plot.axes()
118     ax.set_position([0.1, 0.1, 0.65, 0.8])
119     Plot.xlabel(r'$Speed \; \mathrm{m/s}$')
120     Plot.ylabel(r'$Power \; \mathrm{kW}$')
121     ax.spines['right'].set_color((0.0, 0.0, 0.0))
122     ax.spines['top'].set_color((0.0, 0.0, 0.0))
123     ax.xaxis.label.set_fontsize(15)
124     ax.yaxis.label.set_fontsize(15)
125     ax.set_title('Holtrop Power vs Speed', fontweight='bold')
126
127     EKW = Plot.plot(speed, EKW, r'$E_{KW} \; \; \text{Efficient} \; \text{power}$')
128     EKW.line.set_linestyle('-')
129     EKW.line.set_linewidth(2.0)
130     EKW.line.set_color((0.0, 0.0, 1.0))
131     self.EKW = EKW
132
133     BKW = Plot.plot(speed, BKW, r'$B_{KW} \; \; \text{Break} \; \text{power}$')
134     BKW.line.set_linestyle('-')
135     BKW.line.set_linewidth(2.0)
136     BKW.line.set_color((1.0, 0.0, 0.0))
137     self.BKW = BKW
138
139
140     # Show grid
141     Plot.grid(True)
142     Plot.legend(True, pos=(1.05,0.6))
143     # End
144     return False
145
146
147     def plotCoeff(self, speed, CT, CF, CAPP, Cw, CB, CTR, CA):
148         """ Perform Holtrop coefficients plot.
149         @param speed, Ship speed.
150         @param CT, Total resistance coefficient
151         @param CF, Frictional resistance coefficient.
152         @param CAPP, Appendage resistance coefficient.
153         @param Cw, Wave resistance coefficient.
154         @param CB, Additional resistance coefficient due to the presence of a
155             bulbous bow near the surface
156         @param CTR, Additional resistance coefficient due to the immersed.
157             transom.
158         @param CA, Model-ship correlation resistance coefficient.
159         """
160         try:
161             from FreeCAD.Plot import Plot
162         except ImportError:
163             try:
164                 from freecad.plot import Plot

```



```

165         except ImportError:
166             msg = FreeCAD.Qt.translate(
167                 "ship_console",
168                 "Plot module is disabled")
169             FreeCAD.Console.PrintWarning(msg + '\n')
170             return True
171
172     plt = Plot.figure('Coefficients')
173     self.plt2 = plt
174
175     plt.axes.ticklabel_format(axis = 'y', scilimits = (0,0))
176
177     ax = Plot.axes()
178     ax.set_position([0.05, 0.1, 0.62, 0.8])
179     Plot.xlabel(r'$Speed \; \mathrm{m/s}$')
180     Plot.ylabel(r'$Resistance \; coefficients$')
181     ax.spines['right'].set_color((0.0, 0.0, 0.0))
182     ax.spines['top'].set_color((0.0, 0.0, 0.0))
183     ax.xaxis.label.set_fontsize(15)
184     ax.yaxis.label.set_fontsize(15)
185     ax.set_title('Holtrop Resistance Coefficients vs Speed', fontweight='
186         bold')
187
188     CT = Plot.plot(speed, CT, r'$CT$, Total resistance coefficient')
189     CT.line.set_linestyle('--')
190     CT.line.set_linewidth(2.0)
191     CT.line.set_color((0.0, 0.0, 1.0))
192     self.CT = CT
193
194     CF = Plot.plot(speed, CF, r'$CF$, Frictional resistance coefficient')
195     CF.line.set_linestyle('--')
196     CF.line.set_linewidth(2.0)
197     CF.line.set_color((0.0, 0.0, 0.0))
198     self.CF = CF
199
200     Cw = Plot.plot(speed, Cw, r'$Cw$, Wave resistance coefficient')
201     Cw.line.set_linestyle('--')
202     Cw.line.set_linewidth(2.0)
203     Cw.line.set_color((1.0, 0.0, 0.0))
204     self.Cw = Cw
205
206     CA = Plot.plot(speed, CA, r'$CA$, Model-ship correlation coefficient
207         ')
208     CA.line.set_linestyle('--')
209     CA.line.set_linewidth(2.0)
210     CA.line.set_color((0.2, 0.8, 0.2))
211     self.CA = CA
212
213     CAPP = Plot.plot(speed, CAPP, r'$CAPP$, Appendage resistance
214         coefficient')
215     CAPP.line.set_linestyle('--')
216     CAPP.line.set_linewidth(2.0)
217     CAPP.line.set_color((1.0, 1.0, 0.0))
218     self.CAPP = CAPP
219
220     CB = Plot.plot(speed, CB, r'$CB$, Bulbous resistance coefficient')
221     CB.line.set_linestyle('--')

```



```

219     CB.line.set_linewidth(2.0)
220     CB.line.set_color((0.8, 0.2, 0.8))
221
222
223     CTR = Plot.plot(speed, CTR, r'$CTR$, Transom resistant coefficient')
224     CTR.line.set_linestyle('--')
225     CTR.line.set_linewidth(2.0)
226     CTR.line.set_color((0.5, 0.5, 0.5))
227     self.CTR = CTR
228
229     # Show grid
230     Plot.grid(True)
231     Plot.legend(True, pos =(1.05,0.72))
232     # End
233     return False
234
235 def spreadSheet(self, speed, Rtotal, CT, CF, CAPP, Cw, CB, CTR, CA, EKW,
236 BKW):
237     """ Write the output data file.
238     @param speed, Ship speed.
239     @param CT, Total resistance coefficient
240     @param CF, Frictional resistance coefficient.
241     @param CAPP, Appendage resistance coefficient.
242     @param Cw, Wave resistance coefficient.
243     @param CB, Additional resistance coefficient due to the presence of a
244     bulbous bow near the surface
245     transom.
246     @param CA, Model-ship correlation resistance coefficient.
247     @param EKW, Efficient power.
248     @param BKW, Break power.
249     """
250     s = FreeCAD.activeDocument().addObject('Spreadsheet::Sheet',
251                                             'Resistance Holtrop method')
252
253     # Print the header
254     s.set("A1", "Speed [m/s]")
255     s.set("B1", "Resistance [kN]")
256     s.set("C1", "CT * 10^3")
257     s.set("D1", "CF * 10^3")
258     s.set("E1", "CAPP * 10^3")
259     s.set("F1", "CW * 10^4")
260     s.set("G1", "CB * 10^3")
261     s.set("H1", "CTR * 10^3")
262     s.set("I1", "CA * 10^3")
263     s.set("J1", "EKW [kW]")
264     s.set("K1", "BKW [kW]")
265
266
267     # Print the data
268     for i in range(len(Rtotal)):
269         s.set("A{}".format(i + 2), str(speed[i]))
270         s.set("B{}".format(i + 2), str(Rtotal[i]))
271         s.set("C{}".format(i + 2), str(CT[i] * 1000))
272         s.set("D{}".format(i + 2), str(CF[i] * 1000))
273         s.set("E{}".format(i + 2), str(CAPP[i] * 1000))
274         s.set("F{}".format(i + 2), str(Cw[i] * 10000))

```



```
275         s.set("G{}".format(i + 2), str(CB[i] * 1000))
276         s.set("H{}".format(i + 2), str(CTR[i] * 1000))
277         s.set("I{}".format(i + 2), str(CA[i] * 1000))
278         s.set("J{}".format(i + 2), str(EKW[i]))
279         s.set("K{}".format(i + 2), str(BKW[i]))
280
281     # Recompute
282     FreeCAD.activeDocument().recompute()
```



Anexo G

Comparación de valores obtenidos en FreeCAD con Maxsurf

En el presente anexo se procede a complementar el capítulo 8. Para ello, se muestran imágenes de los valores introducidos en ambos software, tanto FreeCAD como Maxsurf, y se introducen las tablas directamente obtenidas de ambos software. Además, se muestran las gráficas tanto de resistencia como de potencia obtenidas en ambos programas. En el capítulo 8, se introdujo una tabla comparativa de los valores obtenidos para cada uno de los métodos, en este caso se mostrarán las tablas completas obtenidas, pero por separado.

Series 60 de la universidad de Iowa.

A continuación, en las imágenes G.1 y G.2, se muestran las interfaces gráficas, tanto de FreeCAD como de Maxsurf, con los datos iniciales del buque Series 60. En las tablas G.1 y G.2 se incluyen los resultados obtenidos de cada uno de los software.



Resistance by Holtrop method

Wet surface [m ²]	89.39	✓
Waterline length [m]	24.89	✓
Displaced volume [m ³]	47.69	✓
Cb	0.57	✓
Cm	0.96	✓
Cw	0.57	✓
Afterbody form	V-shaped sections	▼
Half angle of entrance [deg]	7.57	✓
Xcb [m]	-0.13	✓
ABT [m ²]	0.00	✓
AT [m ²]	0.00	✓
hb [m]	0.00	✓
Maximum speed [m/s]	4.12	✓
Minimum speed [m/s]	0.00	✓
Number of speeds	17	▼
ηp	0.60	✓
Sea margin %	15.00	✓
Rudder behind skeg [m ²]	0.00	✓
Rudder behind stern [m ²]	0.00	✓
Twin-screw balance rudders [m ²]	0.00	✓
Shaft brackets [m ²]	0.00	✓
Skeg [m ²]	0.00	✓
Strut bossings [m ²]	0.00	✓
Hull bossings [m ²]	0.00	✓
Shafts [m ²]	0.00	✓
Stabilizer fins [m ²]	0.00	✓

	Item	Value	Units	Holtrop
1	LWL	24.29	m	24.29
2	Beam	3.93	m	3.93
3	Draft	1	m	1
4	Displaced volume	47.69	m ³	47.69
5	Wetted area	89.39	m ²	89.39
6	Prismatic coeff. (Cp)	0.59		0.59
7	Waterpl. area coeff. (Cwp)	0.57		0.57
8	1/2 angle of entrance	7.6	deg.	7.6
9	LCG from midships(+ve for'd)	-0.129	m	-0.129
10	Transom area	0	m ²	0
11	Transom wl beam	0	m	--
12	Transom draft	0	m	--
13	Max sectional area	3.328	m ²	--
14	Bulb transverse area	0	m ²	0
15	Bulb height from keel	0	m	0
16	Draft at FP	1	m	1
17	Deadrise at 50% LWL	0	deg.	--
18	Hard chine or Round bilge	Hard chine		--
19				
20	Frontal Area	0	m ²	
21	Headwind	0	kn	
22	Drag Coefficient	0		
23	Air density	0.001	tonne/	
24	Appendage Area	0	m ²	
25	Nominal App. length	0	m	
26	Appendage Factor	1		
27				
28	Correlation allowance	use 19th ITT		Calculated
29	Kinematic viscosity	0.00000118	m ² /s	
30	Water Density	1.0259	tonne/	

Figura G.2: Datos iniciales del buque S60 en Maxsurf.

Figura G.1: Datos iniciales del buque S60 en FreeCAD.

Tabla G.1: Valores obtenidos de FreeCAD para el buque Series 60.

Speed [m/s]	Resistance [kN]	CT * 10 ³	CF * 10 ³	CAPP* 10 ³	Cw * 10 ⁴	CB * 10 ³	CTR * 10 ³	CA * 10 ³	EKW [kW]	BKW [kW]
0.26	0.01	4.06	3.34	0.00	0.00	0.00	0.00	0.72	0.00	0.01
0.51	0.05	3.67	2.95	0.00	0.00	0.00	0.00	0.72	0.03	0.04
0.77	0.10	3.48	2.75	0.00	0.00	0.00	0.00	0.72	0.08	0.13
1.03	0.18	3.35	2.63	0.00	0.00	0.00	0.00	0.72	0.18	0.30
1.29	0.27	3.26	2.53	0.00	0.00	0.00	0.00	0.72	0.35	0.58
1.54	0.38	3.18	2.46	0.00	0.01	0.00	0.00	0.72	0.58	0.97
1.80	0.51	3.13	2.40	0.00	0.03	0.00	0.00	0.72	0.91	1.52
2.06	0.65	3.09	2.35	0.00	0.12	0.00	0.00	0.72	1.34	2.24
2.31	0.82	3.06	2.31	0.00	0.31	0.00	0.00	0.72	1.89	3.16
2.57	1.01	3.06	2.28	0.00	0.66	0.00	0.00	0.72	2.59	4.32
2.83	1.23	3.09	2.24	0.00	1.21	0.00	0.00	0.72	3.47	5.79
3.09	1.48	3.14	2.21	0.00	2.00	0.00	0.00	0.72	4.57	7.62
3.34	1.78	3.21	2.19	0.00	3.03	0.00	0.00	0.72	5.94	9.90
3.60	2.12	3.31	2.16	0.00	4.26	0.00	0.00	0.72	7.62	12.70
3.86	2.52	3.45	2.14	0.00	5.84	0.00	0.00	0.72	9.73	16.21
4.12	3.00	3.62	2.12	0.00	7.78	0.00	0.00	0.72	12.35	20.59



Tabla G.2: Valores obtenidos de Maxsurf para el buque Series 60.

Speed (knot)	Froude N. LWL	Resistance (kN)	Power(kN)
0	0	—	—
0.5	0.016	0.01	0.00
1	0.033	0.05	0.03
1.5	0.049	0.11	0.08
2	0.066	0.19	0.19
2.5	0.082	0.28	0.36
3	0.099	0.40	0.61
3.5	0.115	0.53	0.96
4	0.132	0.68	1.41
4.5	0.148	0.86	1.98
5	0.165	1.05	2.71
5.5	0.181	1.28	3.63
6	0.198	1.55	4.77
6.5	0.214	1.86	6.20
7	0.23	2.22	7.98
7.5	0.247	2.65	10.22
8	0.263	3.18	13.08

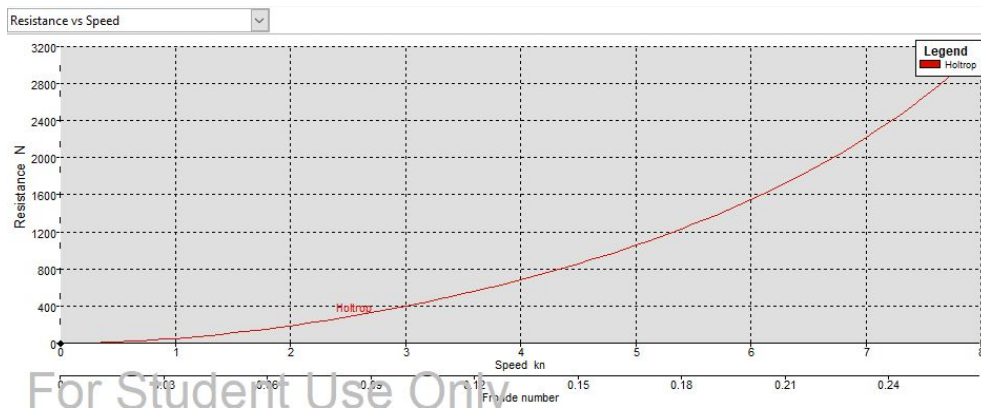


Figura G.3: Gráfica resistencia vs velocidad obtenida para el buque S60 a través de MAXSURF.

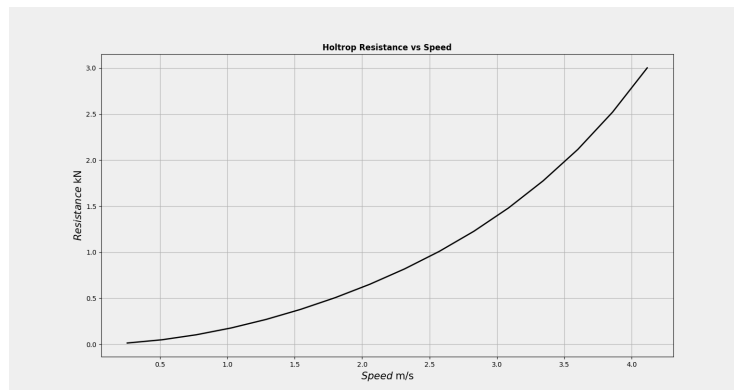


Figura G.4: Gráfica resistencia vs velocidad obtenida para el buque S60 a través de FreeCAD.

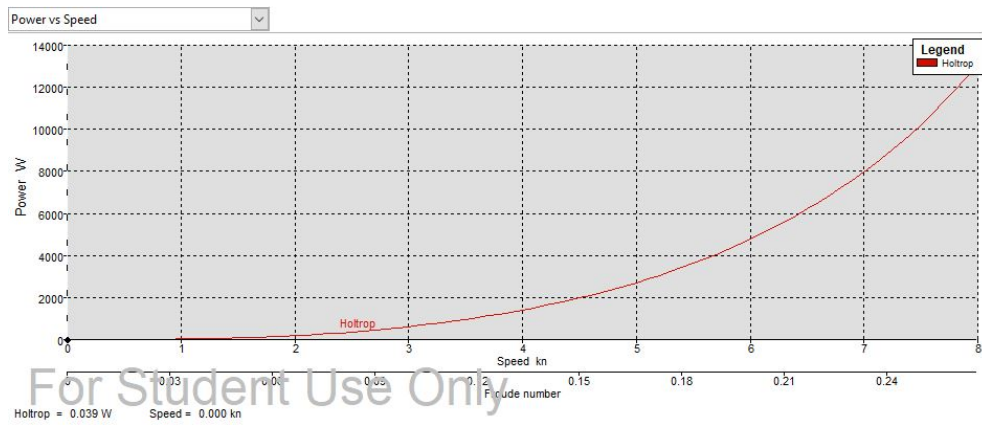


Figura G.5: Gráfica potencia vs velocidad obtenida para el buque S60 a través de *MAXSURF*.

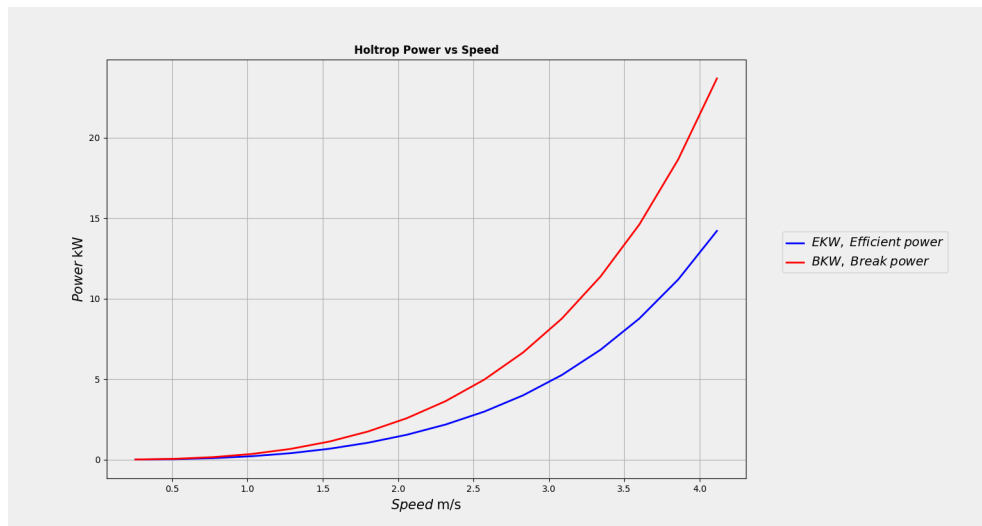


Figura G.6: Gráfica potencia vs velocidad obtenida para el buque S60 a través de *FreeCAD*.

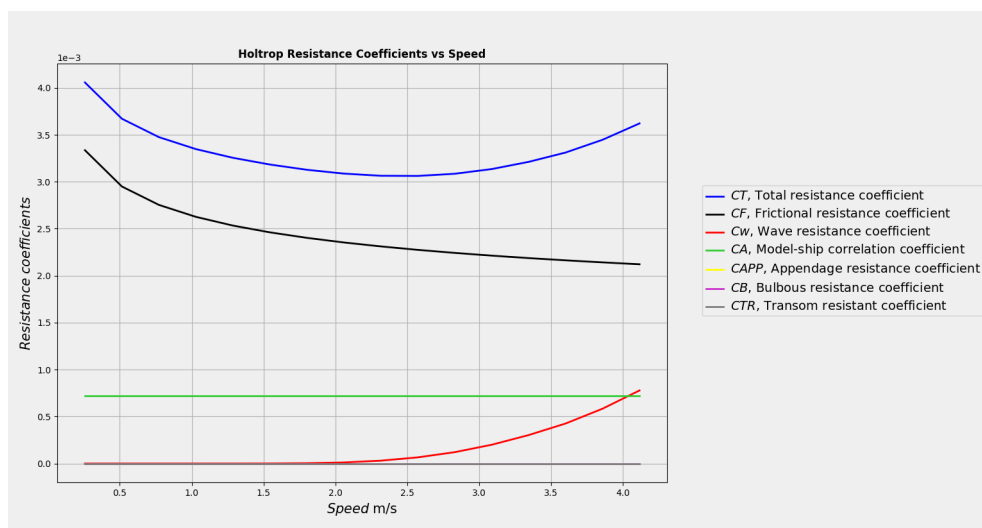


Figura G.7: Gráfica coeficientes resistivos vs velocidad obtenida para el buque S60 a través de *FreeCAD*.



Petrolero de productos de 40.000 TPM.

A continuación, en las imágenes G.8 y G.9, se muestran las interfaces gráficas, tanto de FreeCAD como de Maxsurf, con los datos iniciales del buque petrolero de productos de 40000TPM. En las tablas G.3 y G.4 se incluyen los resultados obtenidos de cada uno de los software.

Resistance by Holtrop method

Wet surface [m ²]	785704.90	✓
Waterline length [m]	178.40	✓
Displaced volume [m ³]	46268.00	✓
Cb	0.73	✓
Cm	0.95	✓
Cw	0.92	✓
Afterbody form	U_shaped sections	▼
Half angle of entrance [deg]	40.37	✓
Xcb [m]	-2.41	✓
ABT [m ²]	8.11	✓
AT [m ²]	0.00	✓
hb [m]	5.37	✓
Maximum speed [m/s]	7.20	✓
Minimum speed [m/s]	0.00	✓
Number of speeds	29	↕
np	0.60	✓
Sea margin %	15.00	✓
Rudder behind skeg [m ²]	0.00	✓
Rudder behind stern [m ²]	0.00	✓
Twin-screw balance rudders [m ²]	0.00	✓
Shaft brackets [m ²]	0.00	✓
Skeg [m ²]	0.00	✓
Strut bossings [m ²]	0.00	✓
Hull bossings [m ²]	0.00	✓
Shafts [m ²]	0.00	✓

	Item	Value	Units	Holtrop
1	LWL	178.4	m	178.4
2	Beam	32	m	32
3	Draft	11.1	m	11.1
4	Displaced volume	46268	m ³	46268
5	Wetted area	7857.049	m ²	7857.049
6	Prismatic coeff. (Cp)	0.764		0.764
7	Waterpl. area coeff. (Cwp)	0.922		0.922
8	1/2 angle of entrance	40.4	deg.	40.4
9	LCG from midships(+ve for'd)	-2.41	m	-2.41
10	Transom area	0	m ²	0
11	Transom wl beam	0	m	--
12	Transom draft	0	m	--
13	Max sectional area	339.463	m ²	--
14	Bulb transverse area	8.11	m ²	8.11
15	Bulb height from keel	5.37	m	5.37
16	Draft at FP	11.1	m	11.1
17	Deadrise at 50% LWL	0	deg.	--
18	Hard chine or Round bilge	Hard chine		--
19				
20	Frontal Area	0	m ²	
21	Headwind	0	kn	
22	Drag Coefficient	0		
23	Air density	0.001	tonne/	
24	Appendage Area	0	m ²	
25	Nominal App. length	0	m	
26	Appendage Factor	1		
27				
28	Correlation allowance	use 19th ITT		Calculated
29	Kinematic viscosity	0.00000118	m ² /s	
30	Water Density	1.0259	tonne/	

Figura G.9: Datos iniciales de un petrolero de productos en Maxsurf.

Figura G.8: Datos iniciales de un petrolero de productos en FreeCAD.



Tabla G.3: Valores obtenidos de FreeCAD para un buque petrolero de productos de 40.000 TPM.

Speed [m/s]	Resistance [kN]	CT * 10 ³	CF * 10 ³	CAPP* 10 ³	Cw * 10 ⁴	CB * 10 ³	CTR * 10 ³	CA * 10 ³	EKW [kW]	BKW [kW]
0.257	0.914	2.801	2.413	0.0	0.00	8.10E-05	0.0	0.388	0.235	0.392
0.514	3.334	2.560	2.172	0.0	0.00	1.59E-04	0.0	0.388	1.715	2.858
0.772	7.124	2.436	2.048	0.0	0.00	2.32E-04	0.0	0.388	5.497	9.162
1.029	12.225	2.354	1.966	0.0	0.00	2.98E-04	0.0	0.388	12.577	20.961
1.286	18.595	2.294	1.905	0.0	0.00	3.56E-04	0.0	0.388	23.913	39.855
1.543	26.205	2.246	1.858	0.0	0.00	4.03E-04	0.0	0.388	40.440	67.400
1.800	35.033	2.208	1.820	0.0	0.00	4.41E-04	0.0	0.388	63.074	105.123
2.058	45.060	2.175	1.787	0.0	0.00	4.68E-04	0.0	0.388	92.716	154.526
2.315	56.270	2.148	1.759	0.0	0.00	4.87E-04	0.0	0.388	130.253	217.089
2.572	68.649	2.123	1.735	0.0	0.00	4.96E-04	0.0	0.388	176.564	294.274
2.829	82.185	2.102	1.713	0.0	0.00	4.98E-04	0.0	0.388	232.517	387.528
3.086	96.868	2.082	1.694	0.0	0.00	4.94E-04	0.0	0.388	298.972	498.287
3.344	112.688	2.065	1.676	0.0	0.00	4.85E-04	0.0	0.388	376.785	627.974
3.601	129.640	2.049	1.660	0.0	0.00	4.71E-04	0.0	0.388	466.808	778.014
3.858	147.720	2.034	1.646	0.0	0.00	4.54E-04	0.0	0.388	569.905	949.842
4.115	166.933	2.021	1.632	0.0	0.01	4.35E-04	0.0	0.388	686.963	1144.939
4.372	187.295	2.009	1.619	0.0	0.02	4.14E-04	0.0	0.388	818.929	1364.881
4.630	208.841	1.999	1.608	0.0	0.03	3.92E-04	0.0	0.388	966.852	1611.420
4.887	231.636	1.991	1.597	0.0	0.06	3.70E-04	0.0	0.388	1131.958	1886.597
5.144	255.781	1.985	1.586	0.0	0.109	3.48E-04	0.0	0.388	1315.737	2192.895
5.401	281.429	1.983	1.577	0.0	0.184	3.26E-04	0.0	0.388	1520.055	2533.426
5.658	308.795	1.985	1.567	0.0	0.294	3.05E-04	0.0	0.388	1747.286	2912.143
5.916	338.166	1.992	1.559	0.0	0.451	2.84E-04	0.0	0.388	2000.454	3334.090
6.173	369.912	2.005	1.550	0.0	0.666	2.65E-04	0.0	0.388	2283.393	3805.655
6.430	404.494	2.026	1.542	0.0	0.952	2.46E-04	0.0	0.388	2600.898	4334.830
6.687	442.470	2.055	1.535	0.0	1.323	2.29E-04	0.0	0.388	2958.885	4931.475
6.944	484.496	2.095	1.528	0.0	1.791	2.12E-04	0.0	0.388	3364.535	5607.558
7.202	531.329	2.146	1.521	0.0	2.370	1.97E-04	0.0	0.388	3826.420	6377.367

Tabla G.4: Valores obtenidos de Maxsurf para un buque granelero de 35.000 TPM.

Speed (knot)	Froude N. LWL	Resistance (kN)	Power(kN)
0	0	-	-
0.5	0.006	0.93	0.24
1	0.012	3.40	1.75
1.5	0.018	7.26	5.60
2	0.025	12.45	12.81
2.5	0.031	18.94	24.36
3	0.037	26.69	41.19
3.5	0.043	35.68	64.24
4	0.049	45.89	94.43
4.5	0.055	57.31	132.66
5	0.062	69.91	179.83
5.5	0.068	83.70	236.81
6	0.074	98.65	304.49
6.5	0.08	114.76	383.73
7	0.086	132.02	475.41
7.5	0.092	150.43	580.40
8	0.099	169.99	699.60
8.5	0.105	190.72	833.97
9	0.111	212.65	984.58
9.5	0.117	235.85	1152.65
10	0.123	260.41	1339.67
10.5	0.129	286.49	1547.52
11	0.136	314.29	1778.55
11.5	0.142	344.11	2035.77
12	0.148	376.29	2322.99
12.5	0.154	411.31	2644.96
13	0.16	449.71	3007.54
13.5	0.166	492.13	3417.87
14	0.173	539.34	3884.43

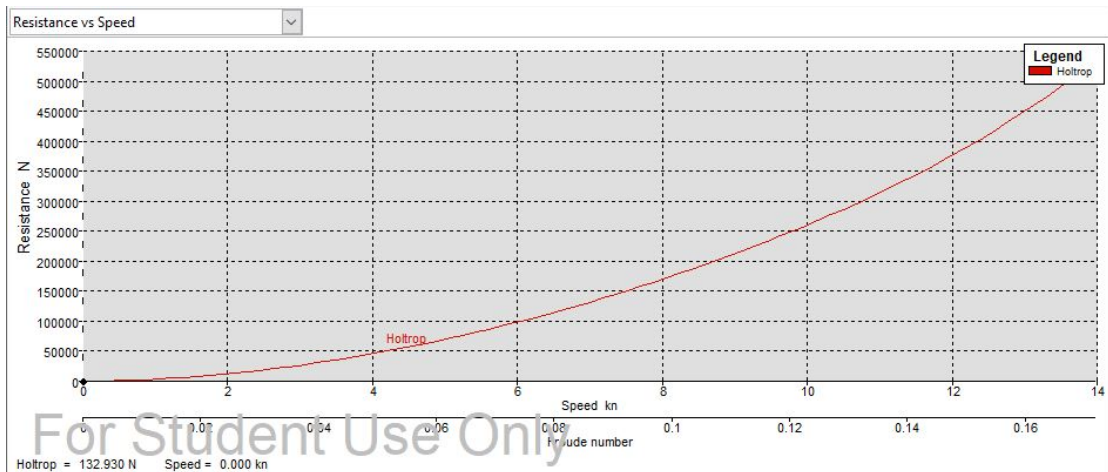


Figura G.10: Gráfica resistencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *MAXSURF*.

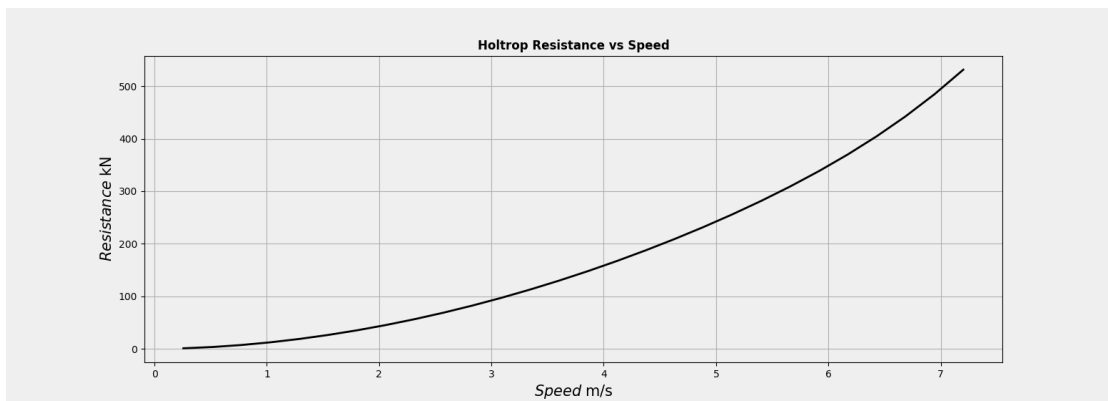


Figura G.11: Gráfica resistencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *FreeCAD*.

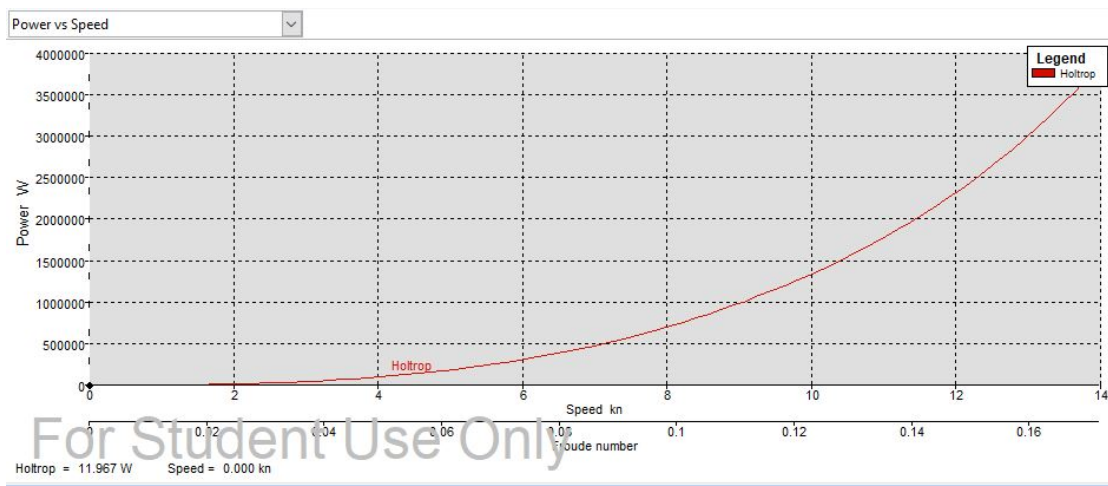


Figura G.12: Gráfica potencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *MAXSURF*.

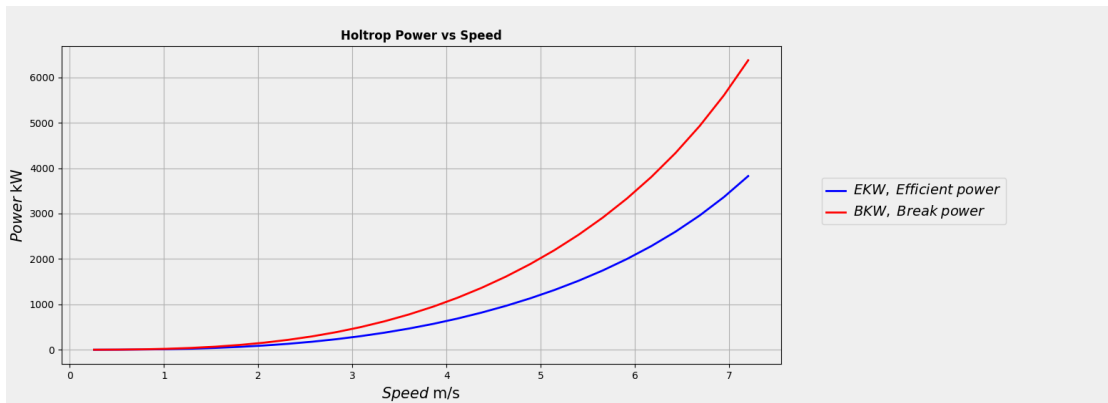


Figura G.13: Gráfica potencia vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *FreeCAD*.

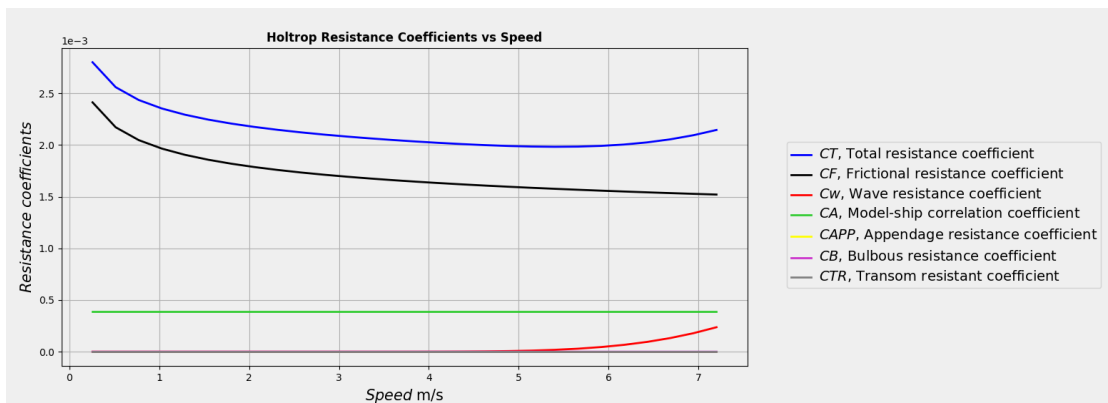


Figura G.14: Gráfica coeficientes resistivos vs velocidad obtenida para un buque petrolero de productos de 40.000 TPM a través de *FreeCAD*.

Buque granelero de 35.000 TPM.

A continuación, en las imágenes [G.15](#) y [G.16](#), se muestran las interfaces gráficas, tanto de *FreeCAD* como de *Maxsurf*, con los datos iniciales de un buque granelero de 35.000 TPM. En las tablas [G.5](#) y [G.6](#) se incluyen los resultados obtenidos de cada uno de los software.



Resistance by Holtrop method

Wet surface [m ²]	7561.29	✓
Waterline length [m]	177.71	✓
Displaced volume [m ³]	44023.41	✓
Cb	0.80	✓
Cm	0.99	✓
Cw	0.90	✓
Afterbody form	U_shaped sections	▼
Half angle of entrance [deg]	43.27	✓
Xcb [m]	-1.21	✓
ABT [m ²]	0.00	✓
AT [m ²]	0.00	✓
hb [m]	0.00	✓
Maximum speed [m/s]	7.20	✓
Minimum speed [m/s]	0.00	✓
Number of speeds	29	▼
np	0.60	✓
Sea margin %	15.00	✓
Rudder behind skeg [m ²]	0.00	✓
Rudder behind stern [m ²]	0.00	✓
Twin-screw balance rudders [m ²]	0.00	✓
Shaft brackets [m ²]	0.00	✓
Skeg [m ²]	0.00	✓
Strut bossings [m ²]	0.00	✓
Hull bossings [m ²]	0.00	✓
Shafts [m ²]	0.00	✓

Figura G.15: Datos iniciales un buque granelero de 35.000 TPM en FreeCAD.

	Item	Value	Units
1	LWL	177.71	m
2	Beam	30	m
3	Draft	10.3	m
4	Displaced volume	44023.41	m ³
5	Wetted area	7561.29	m ²
6	Prismatic coeff. (Cp)	0.81	
7	Waterpl. area coeff. (Cwp)	0.9	
8	1/2 angle of entrance	43.3	deg.
9	LCG from midships(+ve for'd)	-1.21	m
10	Transom area	0	m ²
11	Transom wl beam	0	m
12	Transom draft	0	m
13	Max sectional area	305.835	m ²
14	Bulb transverse area	0	m ²
15	Bulb height from keel	0	m
16	Draft at FP	10.3	m
17	Deadrise at 50% LWL	0	deg.
18	Hard chine or Round bilge	Hard chine	
19			
20	Frontal Area	0	m ²
21	Headwind	0	kn
22	Drag Coefficient	0	
23	Air density	0.001	tonne/
24	Appendage Area	0	m ²
25	Nominal App. length	0	m
26	Appendage Factor	1	
27			
28	Correlation allowance	use 19th ITT	
29	Kinematic viscosity	0.00000118	m ² /s
30	Water Density	1.0259	tonne/

Figura G.16: Datos iniciales de un buque granelero de 35.000 TPM en Maxsurf.



Tabla G.5: Valores obtenidos de FreeCAD para un buque granelero de 35.000 TPM.

Speed [m/s]	Resistance [kN]	CT * 10 ³	CF * 10 ³	CAPP* 10 ³	Cw * 10 ⁴	CB * 10 ³	CTR * 10 ³	CA * 10 ³	EKW [kW]	BKW [kW]
0.257	0.941	2.800	2.412	0	0.00	0	0	0.389	0.242	0.403
0.514	3.427	2.560	2.171	0	0.00	0	0	0.389	1.763	2.938
0.772	7.320	2.435	2.046	0	0.00	0	0	0.389	5.648	9.413
1.029	12.555	2.353	1.964	0	0.00	0	0	0.389	12.917	21.528
1.286	19.093	2.293	1.904	0	0.00	0	0	0.389	24.554	40.923
1.543	26.902	2.246	1.857	0	0.00	0	0	0.389	41.515	69.192
1.800	35.958	2.207	1.819	0	0.00	0	0	0.389	64.739	107.899
2.058	46.243	2.175	1.786	0	0.00	0	0	0.389	95.149	158.582
2.315	57.739	2.147	1.758	0	0.00	0	0	0.389	133.654	222.756
2.572	70.432	2.123	1.734	0	0.00	0	0	0.389	181.152	301.919
2.829	84.311	2.101	1.712	0	0.00	0	0	0.389	238.532	397.553
3.086	99.364	2.082	1.693	0	0.00	0	0	0.389	306.676	511.127
3.344	115.583	2.064	1.675	0	0.00	0	0	0.389	386.463	644.105
3.601	132.964	2.048	1.659	0	0.00	0	0	0.389	478.775	797.959
3.858	151.508	2.034	1.645	0	0.01	0	0	0.389	584.519	974.199
4.115	171.232	2.022	1.631	0	0.02	0	0	0.389	704.656	1174.426
4.372	192.172	2.011	1.619	0	0.03	0	0	0.389	840.254	1400.423
4.630	214.396	2.002	1.607	0	0.07	0	0	0.389	992.567	1654.279
4.887	238.016	1.997	1.596	0	0.12	0	0	0.389	1163.139	1938.564
5.144	263.205	1.996	1.585	0	0.21	0	0	0.389	1353.929	2256.548
5.401	290.208	1.999	1.576	0	0.35	0	0	0.389	1567.473	2612.454
5.658	319.356	2.009	1.567	0	0.54	0	0	0.389	1807.046	3011.743
5.916	351.080	2.028	1.558	0	0.81	0	0	0.389	2076.849	3461.414
6.173	385.917	2.055	1.550	0	1.169	0	0	0.389	2382.190	3970.316
6.430	424.520	2.094	1.542	0	1.636	0	0	0.389	2729.662	4549.436
6.687	467.656	2.146	1.534	0	2.228	0	0	0.389	3127.309	5212.182
6.944	516.211	2.212	1.527	0	2.962	0	0	0.389	3584.775	5974.625
7.202	571.182	2.294	1.520	0	3.853	0	0	0.389	4113.424	6855.707

Tabla G.6: Valores obtenidos de Maxsurf para un buque granelero de 35.000 TPM.

Speed (knot)	Froude N. LWL	Resistance (kN)	Power(kN)
0	0	-	-
0.5	0.006	0.91	0.23
1	0.012	3.33	1.71
1.5	0.018	7.11	5.48
2	0.025	12.19	12.55
2.5	0.031	18.55	23.85
3	0.037	26.13	40.33
3.5	0.043	34.94	62.91
4	0.049	44.93	92.46
4.5	0.055	56.11	129.89
5	0.062	68.45	176.06
5.5	0.068	81.94	231.85
6	0.074	96.58	298.10
6.5	0.08	112.35	375.68
7	0.086	129.25	465.44
7.5	0.092	147.28	568.27
8	0.099	166.47	685.11
8.5	0.105	186.84	817.01
9	0.111	208.47	965.20
9.5	0.117	231.46	1131.21
10	0.123	256.00	1317.00
10.5	0.129	282.34	1525.09
11	0.136	310.80	1758.76
11.5	0.142	341.82	2022.22
12	0.148	375.94	2320.80
12.5	0.154	413.82	2661.11
13	0.16	456.24	3051.23
13.5	0.166	504.08	3500.86
14	0.173	558.36	4021.40

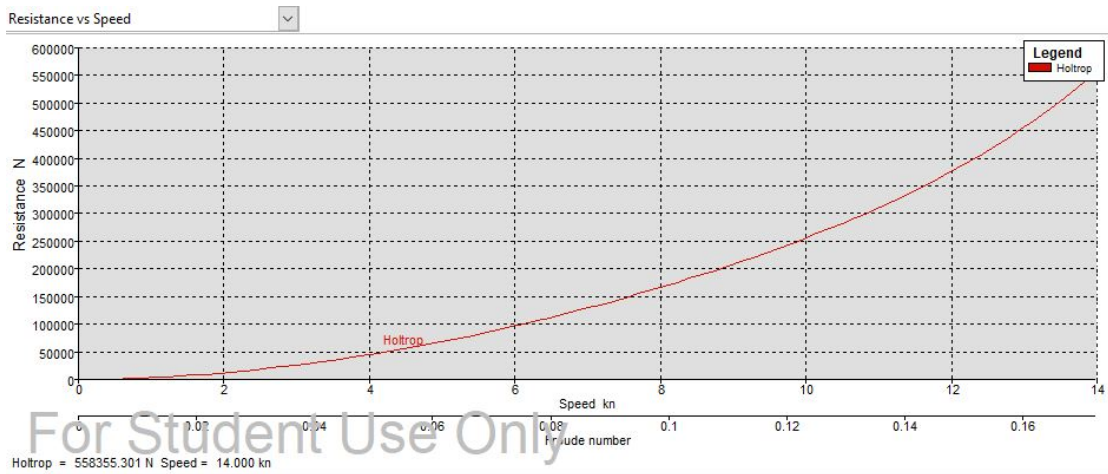


Figura G.17: Gráfica resistencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de *MAXSURF*.



Figura G.18: Gráfica resistencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de *FreeCAD*.

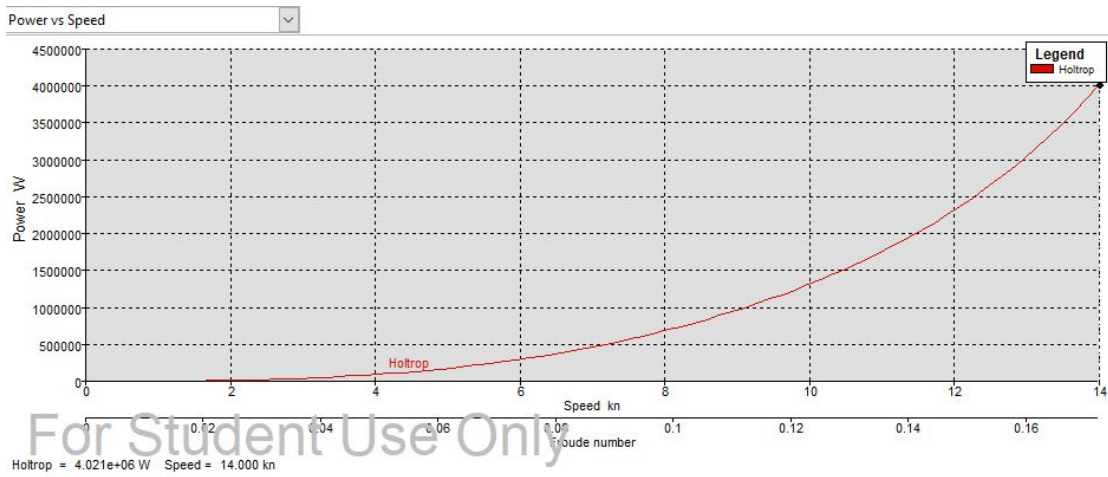


Figura G.19: Gráfica potencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de *MAXSURF*.

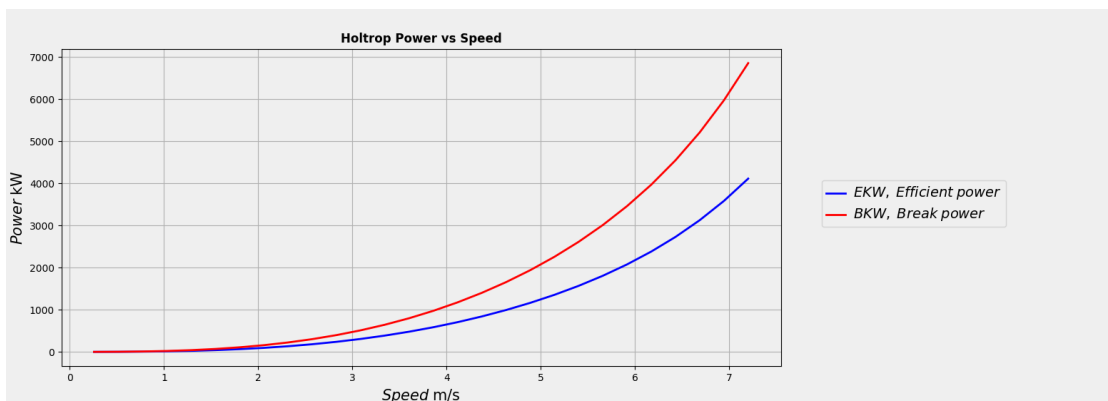


Figura G.20: Gráfica potencia vs velocidad obtenida para un buque granelero de 35.000 TPM a través de *FreeCAD*.

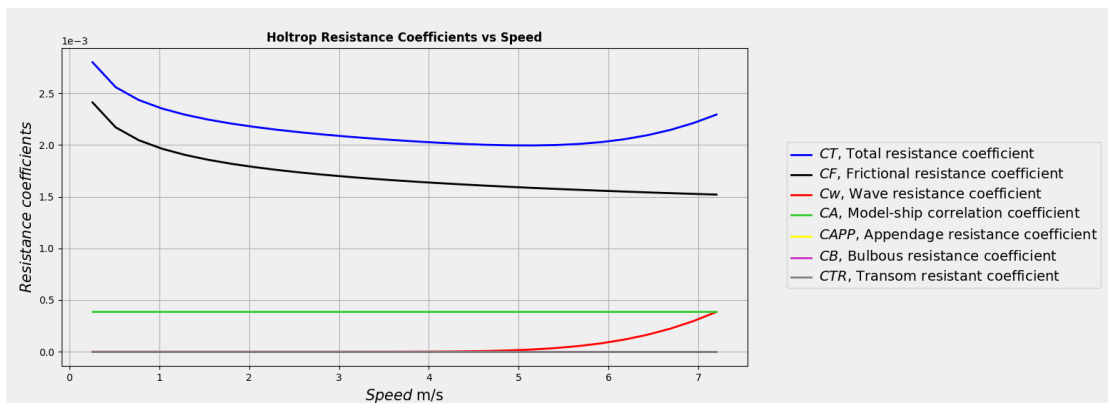


Figura G.21: Gráfica coeficientes resistivos vs velocidad obtenida para un buque granelero de 35.000 TPM a través de *FreeCAD*.



Buque portacontenedores de 4600 TEUS.

A continuación, en las imágenes G.22 y G.23, se muestran las interfaces gráficas, tanto de FreeCAD como de Maxsurf, con los datos iniciales de un buque portacontenedores de 4600 TEUS. En las tablas G.7 y G.8 se incluyen los resultados obtenidos de cada uno de los software.

Resistance by Holtrop method

Wet surface [m ²]	12485.63	✓
Waterline length [m]	267.38	✓
Displaced volume [m ³]	76612.02	✓
Cb	0.64	✓
Cm	0.98	✓
Cw	0.91	✓
Afterbody form	Normal section shape	▼
Half angle of entrance [deg]	30.00	✓
Xcb [m]	-1.85	✓
ABT [m ²]	55.42	✓
AT [m ²]	0.00	✓
hb [m]	5.35	✓
Maximum speed [m/s]	11.83	✓
Minimum speed [m/s]	0.00	✓
Number of speeds	24	▼
ηp	0.60	✓
Sea margin %	15.00	✓
Rudder behind skeg [m ²]	0.00	✓
Rudder behind stern [m ²]	0.00	✓
Twin-screw balance rudders [m ²]	0.00	✓
Shaft brackets [m ²]	0.00	✓
Skeg [m ²]	0.00	✓
Strut bossings [m ²]	0.00	✓
Hull bossings [m ²]	0.00	✓
Shafts [m ²]	0.00	✓

Figura G.22: Datos iniciales un buque portacontenedores de 4600 TEUS en FreeCAD.

	Item	Value	Units	Holtrop
1	LWL	267.383	m	267.383
2	Beam	37.6	m	37.6
3	Draft	11.9	m	11.9
4	Displaced volume	76612.02	m ³	76612.02
5	Wetted area	12485.628	m ²	12485.62
6	Prismatic coeff. (Cp)	0.652		0.652
7	Waterpl. area coeff. (Cwp)	0.812		0.812
8	1/2 angle of entrance	30	deg.	30
9	LCG from midships(+ve for'd)	-1.85	m	-1.85
10	Transom area	0	m ²	0
11	Transom wl beam	0	m	--
12	Transom draft	0	m	--
13	Max sectional area	439.456	m ²	--
14	Bulb transverse area	55.42	m ²	55.42
15	Bulb height from keel	5.35	m	5.35
16	Draft at FP	11.9	m	11.9
17	Deadrise at 50% LWL	0	deg.	--
18	Hard chine or Round bilge	Hard chine		--
19				
20	Frontal Area	0	m ²	
21	Headwind	0	kn	
22	Drag Coefficient	0		
23	Air density	0.001	tonne/	
24	Appendage Area	0	m ²	
25	Nominal App. length	0	m	
26	Appendage Factor	1		
27				
28	Correlation allowance	use 19th ITT		Calculated
29	Kinematic viscosity	0.00000118	m ² /s	
30	Water Density	1.0259	tonne/	

Figura G.23: Datos iniciales de un buque portacontenedores de 4600 TEUS en Maxsurf.



Tabla G.7: Valores obtenidos de FreeCAD para un buque portacontenedores de 4600 TEUS.

Speed [m/s]	Resistance [kN]	CT * 10 ³	CF * 10 ³	CAPP* 10 ³	Cw * 10 ⁴	CB * 10 ³	CTR * 10 ³	CA * 10 ³	EKW [kW]	BKW [kW]
0.51	4.43	4.26	2.04	0	0.00	1.94	0	0.28	2.28	3.80
1.03	16.32	5.76	1.85	0	0.00	3.62	0	0.28	16.79	27.98
1.54	35.09	6.91	1.76	0	0.00	4.87	0	0.28	54.14	90.24
2.06	60.45	7.60	1.69	0	0.00	5.63	0	0.28	124.39	207.31
2.57	92.20	7.86	1.64	0	0.00	5.94	0	0.28	237.13	395.21
3.09	130.12	7.76	1.60	0	0.00	5.88	0	0.28	401.61	669.35
3.60	174.07	7.42	1.57	0	0.00	5.57	0	0.28	626.80	1044.67
4.12	223.93	6.94	1.55	0	0.00	5.11	0	0.28	921.50	1535.83
4.63	279.60	6.40	1.52	0	0.00	4.59	0	0.28	1294.44	2157.40
5.14	341.08	5.84	1.50	0	0.01	4.05	0	0.28	1754.54	2924.23
5.66	408.49	5.31	1.49	0	0.02	3.53	0	0.28	2311.37	3852.29
6.17	482.09	4.82	1.47	0	0.06	3.06	0	0.28	2975.87	4959.78
6.69	562.47	4.38	1.46	0	0.12	2.63	0	0.28	3761.35	6268.91
7.20	650.50	4.00	1.44	0	0.23	2.25	0	0.28	4684.68	7807.79
7.72	747.47	3.68	1.43	0	0.40	1.93	0	0.28	5767.49	9612.49
8.23	855.03	3.42	1.42	0	0.65	1.65	0	0.28	7037.28	11728.80
8.74	975.23	3.20	1.41	0	0.98	1.41	0	0.28	8528.19	14213.65
9.26	1110.43	3.03	1.40	0	1.41	1.20	0	0.28	10281.65	17136.09
9.77	1263.23	2.90	1.39	0	1.94	1.03	0	0.28	12346.27	20577.11
10.29	1436.37	2.81	1.38	0	2.59	0.88	0	0.28	14777.41	24629.01
10.80	1633.51	2.75	1.38	0	3.35	0.76	0	0.28	17645.84	29409.73
11.32	1857.29	2.73	1.37	0	4.23	0.65	0	0.28	21018.59	35030.98
11.83	2106.10	2.72	1.36	0	5.16	0.56	0	0.28	24917.73	41529.55

Tabla G.8: Valores obtenidos de Maxsurf para un buque portacontenedores de 4600 TEUS.

Speed (knot)	Froude N. LWL	Resistance (kN)	Power(kN)
0	0	—	—
1	0.01	4.51	2.32
2	0.02	16.61	17.09
3	0.03	35.74	55.16
4	0.04	61.68	126.92
5	0.05	94.25	242.43
6	0.06	133.33	411.54
7	0.07	178.79	643.84
8	0.08	230.52	948.73
9	0.09	288.44	1335.46
10	0.1	352.49	1813.34
11	0.111	422.74	2392.24
12	0.121	499.45	3083.26
13	0.131	583.13	3899.82
14	0.141	674.63	4858.82
15	0.151	775.19	5981.86
16	0.161	886.44	7296.37
17	0.171	1010.39	8836.42
18	0.181	1149.39	10643.36
19	0.191	1306.02	12765.66
20	0.201	1483.03	15258.70
21	0.211	1684.03	18193.16
22	0.221	1911.60	21635.08
23	0.231	2164.15	25606.74

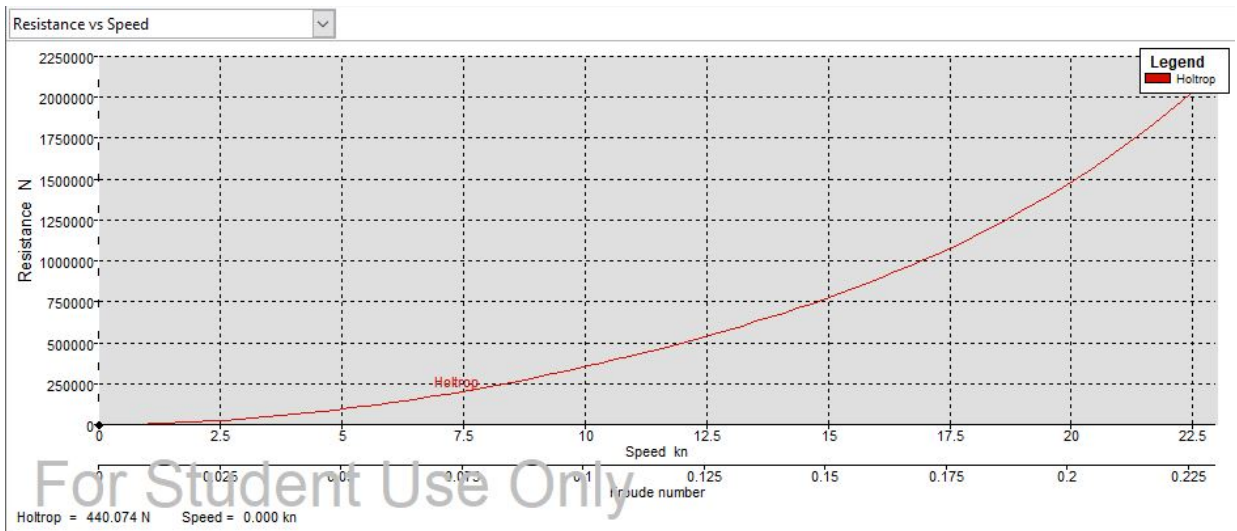


Figura G.24: Gráfica resistencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *MAXSURF*.

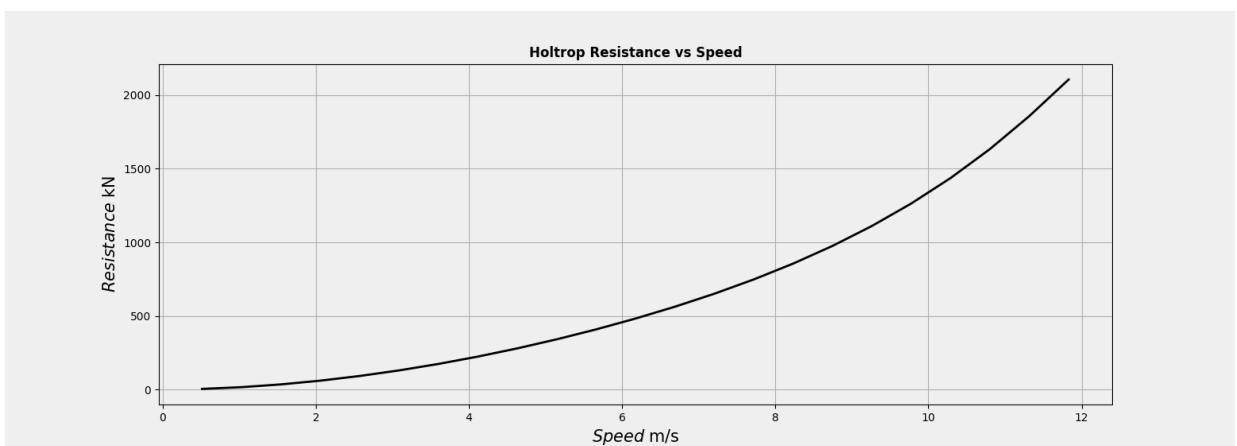


Figura G.25: Gráfica resistencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *FreeCAD*.



Figura G.26: Gráfica potencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *MAXSURF*.

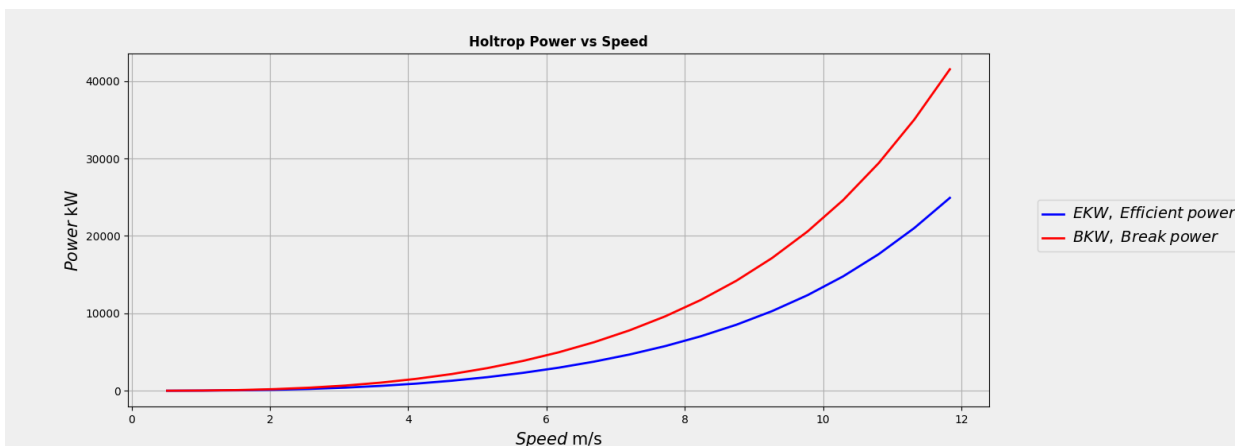


Figura G.27: Gráfica potencia vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *FreeCAD*.

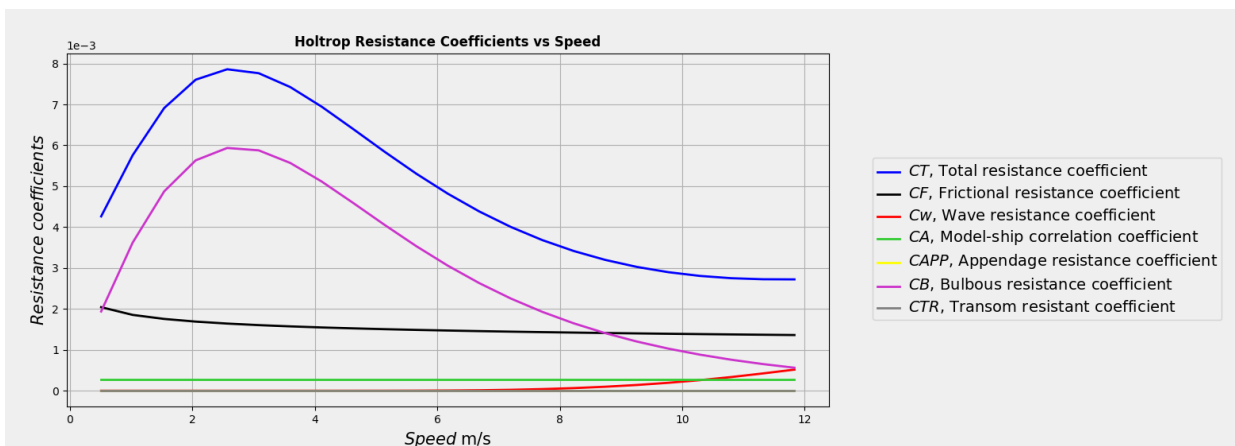


Figura G.28: Gráfica coeficientes resistivos vs velocidad obtenida para un buque portacontenedores de 4600 TEUS a través de *FreeCAD*.



Buque tipo ro-ro.

A continuación, en las imágenes G.29 y G.30, se muestran las interfaces gráficas, tanto de FreeCAD como de Maxsurf, con los datos iniciales del buque Ro-ro. En las tablas G.9 y G.10 se incluyen los resultados obtenidos de cada uno de los software.

Resistance by Holtrop method

Wet surface [m ²]	5784.6000	✓
Waterline length [m]	157.2000	✓
Displaced volume [m ³]	20884.8780	✓
Cb	0.600	✓
Cm	0.970	✓
Cw	0.720	✓
Afterbody form	V-shaped sections	▼
Half angle of entrance [deg]	14.0000	✓
Xcb [m]	3.1000	✓
ABT [m ²]	11.3000	✓
AT [m ²]	0.0000	✓
hb [m]	5.6000	✓
Maximum speed [m/s]	11.3168	✓
Minimum speed [m/s]	0.0000	✓
Number of speeds	22	▼
ηp	0.6000	✓
Sea margin %	15.0000	✓
Rudder behind skeg [m ²]	0.0000	✓
Rudder behind stern [m ²]	0.0000	✓
Twin-screw balance rudders [m ²]	0.0000	✓
Shaft brackets [m ²]	0.0000	✓
Skeg [m ²]	0.0000	✓
Strut bossings [m ²]	0.0000	✓
Hull bossings [m ²]	0.0000	✓
Shafts [m ²]	0.0000	✓
Stabilizer fins [m ²]	0.0000	✓
Dome [m ²]	0.0000	✓
Bilge keels [m ²]	0.0000	✓

Figura G.29: Datos iniciales de un buque ro-ro en FreeCAD.

	Item	Value	Units
1	LWL	202.921	m
2	Beam	26.5	m
3	Draft	7.1	m
4	Displaced volume	21217.561	m ³
5	Wetted area	5784.6	m ²
6	Prismatic coeff. (Cp)	0.616	
7	Waterpl. area coeff. (Cwp)	0.85	
8	1/2 angle of entrance	18.2	deg.
9	LCG from midships(+ve for'd)	6.1	m
10	Transom area	0	m ²
11	Transom wl beam	0	m
12	Transom draft	0	m
13	Max sectional area	169.741	m ²
14	Bulb transverse area	11.3	m ²
15	Bulb height from keel	5.6	m
16	Draft at FP	7.1	m
17	Deadrise at 50% LWL	0	deg.
18	Hard chine or Round bilge	Hard chine	
19			
20	Frontal Area	0	m ²
21	Headwind	0	kn
22	Drag Coefficient	0	
23	Air density	0.001	tonne/
24	Appendage Area	0	m ²
25	Nominal App. length	0	m
26	Appendage Factor	1	
27			
28	Correlation allowance	use 19th ITT	
29	Kinematic viscosity	0.0000118	m ² /s
30	Water Density	1.0259	tonne/

Figura G.30: Datos iniciales de un buque ro-ro en Maxsurf.



Tabla G.9: Valores obtenidos de FreeCAD para un buque tipo ro-ro.

Speed [m/s]	Resistance [kN]	CT * 10 ³	CF * 10 ³	CAPP* 10 ³	Cw * 10 ⁴	CB * 10 ³	CTR * 10 ³	CA * 10 ³	EKW [kW]	BKW [kW]
0.51	2.31	52.30	2.15	0.00	0.00	49.78	0.00	0.36	1.19	1.98
1.03	8.67	81.29	1.95	0.00	0.00	78.98	0.00	0.36	8.92	14.87
1.54	18.73	87.68	1.84	0.00	0.00	85.48	0.00	0.36	28.91	48.19
2.06	32.12	80.45	1.77	0.00	0.00	78.32	0.00	0.36	66.08	110.14
2.57	48.54	68.09	1.72	0.00	0.00	66.00	0.00	0.36	124.85	208.09
3.09	67.88	55.30	1.68	0.00	0.00	53.26	0.00	0.36	209.50	349.17
3.60	90.08	44.04	1.65	0.00	0.00	42.03	0.00	0.36	324.36	540.59
4.12	115.15	34.81	1.62	0.00	0.00	32.82	0.00	0.36	473.86	789.77
4.63	143.13	27.51	1.59	0.00	0.01	25.55	0.00	0.36	662.65	1104.41
5.14	174.13	21.83	1.57	0.00	0.04	19.89	0.00	0.36	895.75	1492.91
5.66	208.35	17.47	1.56	0.00	0.09	15.54	0.00	0.36	1178.90	1964.84
6.17	246.07	14.11	1.54	0.00	0.18	12.20	0.00	0.36	1518.97	2531.61
6.69	287.76	11.54	1.52	0.00	0.33	9.62	0.00	0.36	1924.29	3207.15
7.20	333.96	9.57	1.51	0.00	0.55	7.64	0.00	0.36	2405.02	4008.37
7.72	385.34	8.04	1.50	0.00	0.83	6.10	0.00	0.36	2973.29	4955.49
8.23	442.66	6.87	1.48	0.00	1.20	4.90	0.00	0.36	3643.23	6072.05
8.74	506.70	5.97	1.47	0.00	1.66	3.97	0.00	0.36	4430.99	7384.99
9.26	578.36	5.27	1.46	0.00	2.19	3.23	0.00	0.36	5355.16	8925.26
9.77	658.05	4.74	1.45	0.00	2.80	2.64	0.00	0.36	6431.47	10719.12
10.29	746.09	4.33	1.45	0.00	3.45	2.18	0.00	0.36	7675.73	12792.88
10.80	845.11	4.02	1.44	0.00	4.21	1.80	0.00	0.36	9129.20	15215.34
11.32	958.81	3.80	1.43	0.00	5.13	1.50	0.00	0.36	10850.68	18084.47

Tabla G.10: Valores obtenidos de Maxsurf para un buque tipo ro-ro.

Speed (knot)	Froude N. LWL	Resistance (kN)	Power(kN)
0	0	–	–
1	0	2.25372	1.15941
2	0.012	8.53947	8.78617
3	0.023	18.67049	28.8148
4	0.035	32.37906	66.62892
5	0.046	49.38303	127.02414
6	0.058	69.46488	214.41493
7	0.069	92.48156	333.03639
8	0.081	118.35336	487.08984
9	0.092	147.06344	680.90374
10	0.104	178.67474	919.18226
11	0.115	213.35584	1207.35697
12	0.127	251.40323	1551.99595
13	0.138	293.25069	1961.19547
14	0.15	339.46267	2444.88558
15	0.161	390.71398	3015.00955
16	0.173	447.7571	3685.53842
17	0.185	511.40174	4472.49234
18	0.196	582.53975	5394.31806
19	0.208	661.58408	6466.61684
20	0.219	748.88412	7705.18547
21	0.231	847.0933	9151.43123
22	0.242	959.89769	10863.9087

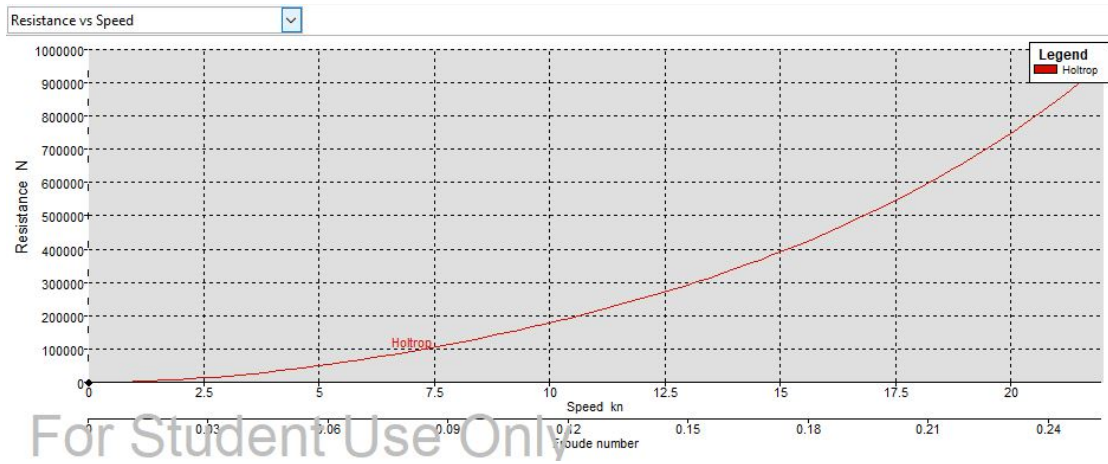


Figura G.31: Gráfica resistencia vs velocidad obtenida para un buque tipo ro-ro a través de MAXSURF.

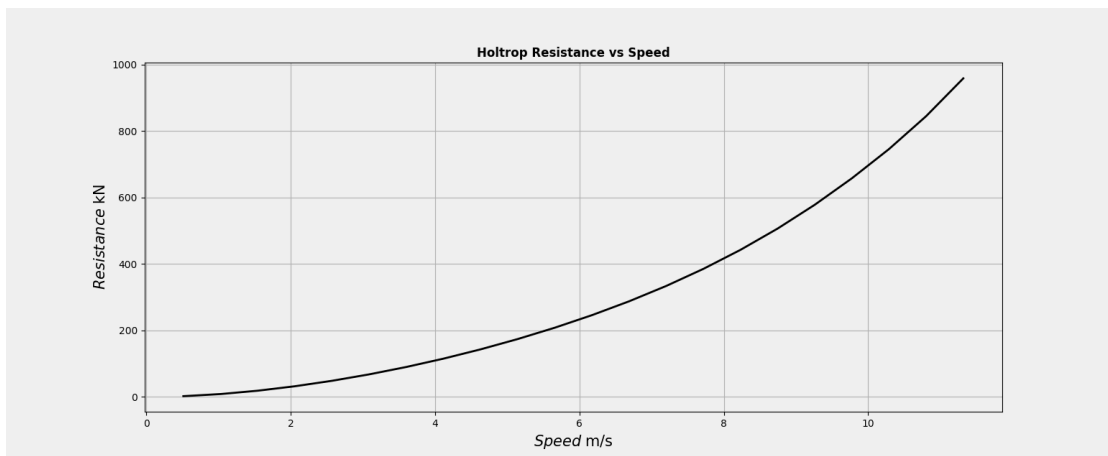


Figura G.32: Gráfica resistencia vs velocidad obtenida para un buque tipo ro-ro a través de FreeCAD.

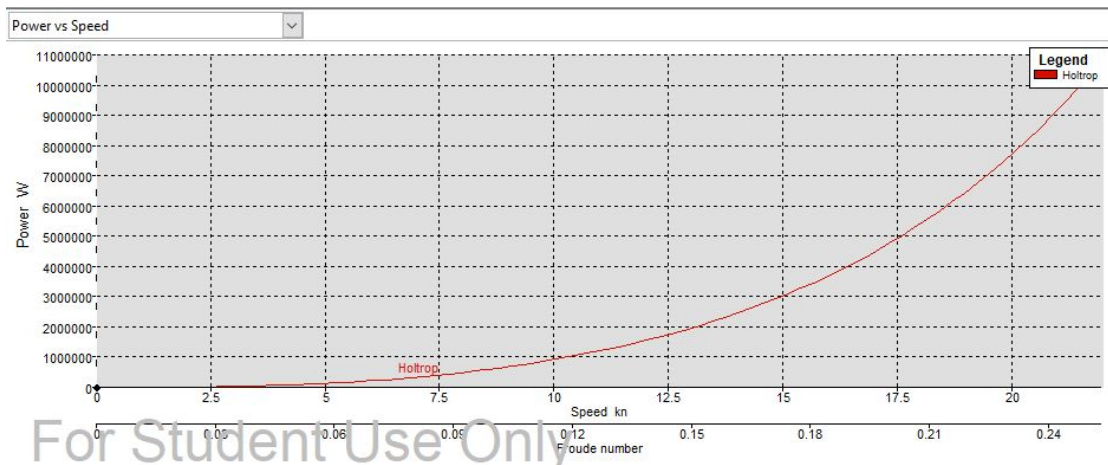


Figura G.33: Gráfica potencia vs velocidad obtenida para un buque tipo ro-ro a través de MAXSURF.

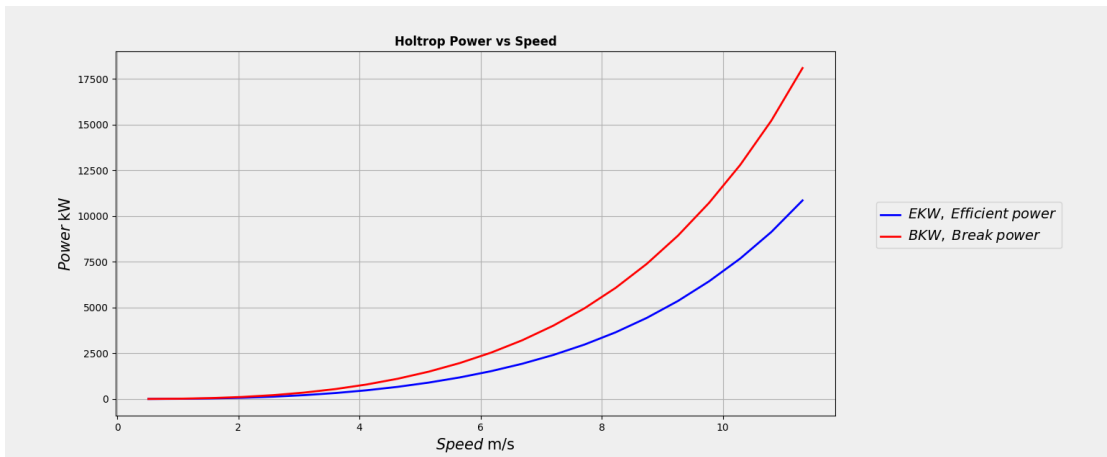


Figura G.34: Gráfica potencia vs velocidad obtenida para un buque tipo ro-ro a través de *FreeCAD*.



Figura G.35: Gráfica coeficientes resistivos vs velocidad obtenida para un buque tipo ro-ro a través de *FreeCAD*.



Anexo H

Manual de usuario.

En este anexo se incluye un manual de usuario con el fin de explicar detalladamente el flujo de trabajo a realizar para obtener la predicción de resistencia en las diversas variantes posibles.

Configuración inicial.

En primer lugar, hay que configurar adecuadamente el módulo de *FreeCAD Ship*.

Número de decimales

Una de las configuraciones más importantes visualmente, es el número de decimales. Normalmente, lo adecuado es entre dos y tres decimales, pero la elección final queda a gusto del usuario.

Para modificar los decimales hay que pulsar en: “Editar” y a continuación en “Preferencias”, finalmente, los decimales quedan definidos en la pestaña: “Unidades”. Una vez seleccionados el número de decimales deseados, pulsar en “Apply”.

A continuación, en la imagen [H.1](#), se muestra la interfaz gráfica para modificar la cantidad de decimales.

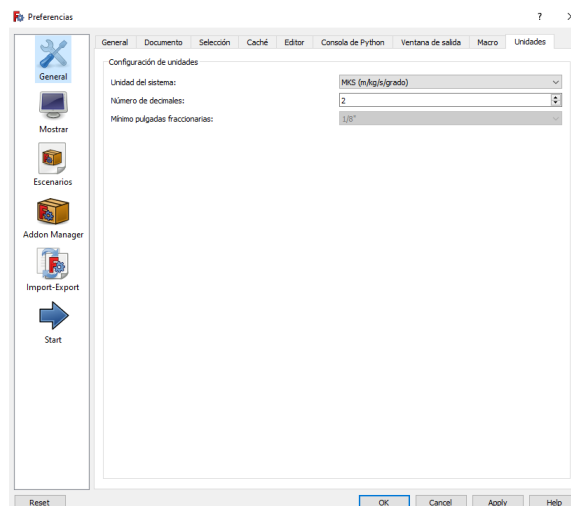


Figura H.1: Ventana para determinar el número de decimales deseados en *FreeCAD Ship*.



Separación decimal

Por defecto, la separación decimal es mediante el punto, “.”, sin embargo, hay que tener en cuenta la configuración local del dispositivo. Es decir, en la configuración del sistema operativo, el usuario debe tener que la separación decimal se establezca mediante el punto, ya que si se establece mediante la coma, “,”, pese a que existe la opción de cambiar en *FreeCAD* la separación decimal, habrá incompatibilidades en el código y puede dar resistencias erróneas, o incluso, no obtener resultados. Por ello, es necesario comprobar y fijar la separación decimal mediante el punto, “.”, en el sistema operativo en el que se esté utilizando *FreeCAD*.

Vista de informe

Una de las vistas interesantes a utilizar por el usuario es la vista de informe. La vista de informe consta de un espacio que se sitúa por defecto en la zona inferior de la pantalla, en ella se mostrarán mensajes que estén incluidos en el código, a lo largo del proceso. En este caso, se muestran mensajes en varios supuestos.

- Errores. Siempre que exista algún error se desplegará automáticamente la vista de informe, en caso de no estar previamente desplegada y se mostrará la descripción del error, así como la parte del código en el que falla.
- Eslora en la flotación y superficie mojada desconocida. El método de Amadeo y de Holtrop-Mennen tienen la capacidad de calcular algunas dimensiones mediante fórmulas empíricas, en caso de ser desconocidas por el usuario. En estos casos, se ha introducido la característica de que se muestre en la vista de informe el valor calculado por las formulas empíricas.

Para desplegar la vista de informe hay que pulsar en: “Ver”, “Paneles” y a continuación en “Vista de informe”. Se muestra, a continuación, en la imagen H.2, la interfaz gráfica de *FreeCAD Ship* con la vista de informe desplegada.

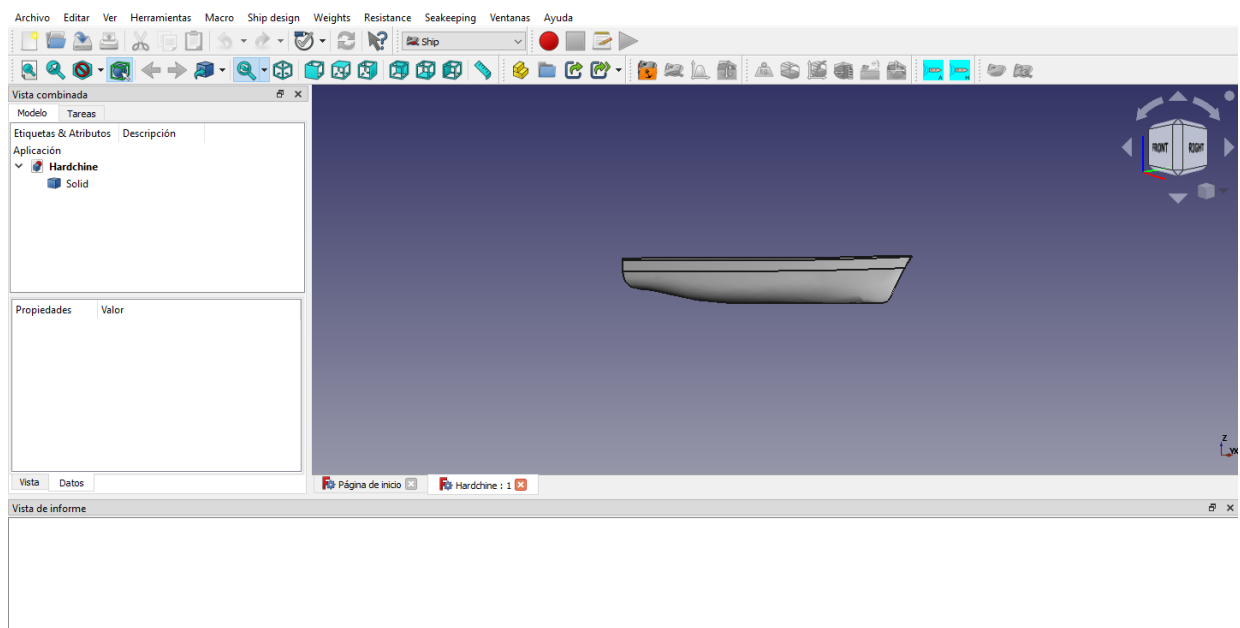


Figura H.2: Vista de informe en *FreeCAD Ship*.



Consola de python

Otra opción a parte de la vista de informe es la consola de python. Se puede desplegar una consola de python en la que se van mostrando todos los pasos que se estén realizando en *FreeCAD*. La principal ventaja de la consola de python se da cuando se está desarrollando código ya que permite introducir comandos y realizar pruebas con las funciones deseadas, comprobar los valores requeridos por cada función...etc.

La consola de python se puede desplegar sola, se despliega por defecto en la zona inferior de la vista, o con la vista de informe, no son excluyentes.

Para desplegar la consola de python hay que pulsar en: “Ver”, “Paneles” y a continuación en “Consola de python”. Se muestra, a continuación, en la imagen H.3 , la interfaz gráfica de *FreeCAD Ship* con la consola de python desplegada.

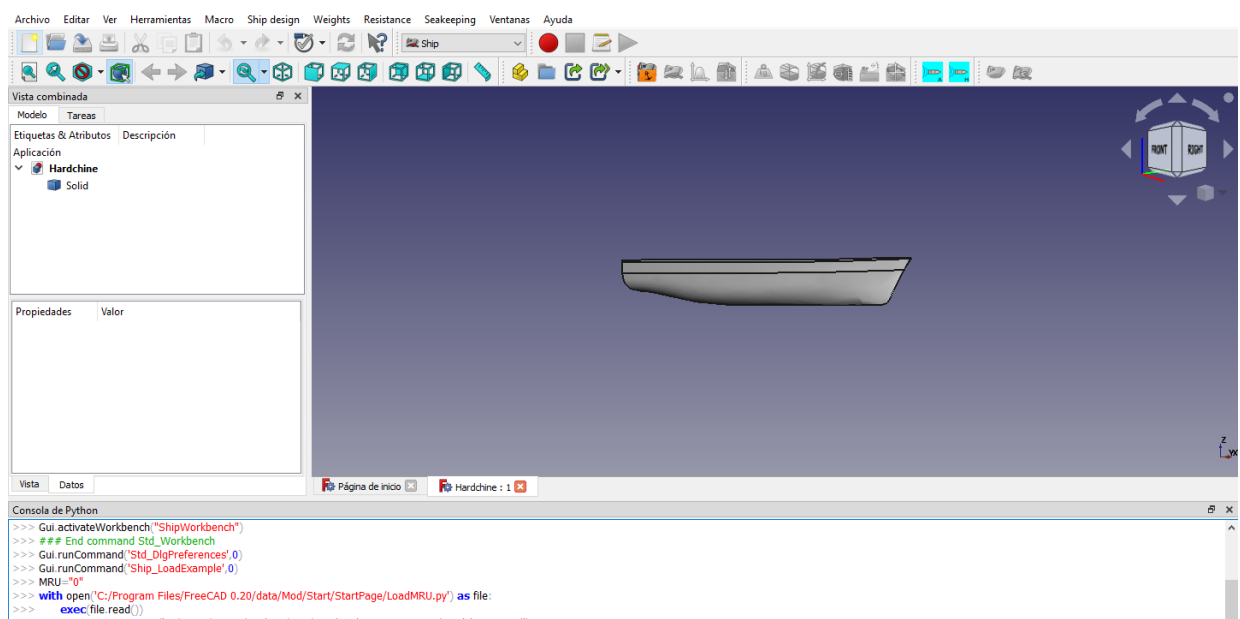


Figura H.3: Vista de *FreeCAD Ship* con consola de python.

Predicción de resistencia al avance a través de un modelo en tres dimensiones.

En primer lugar, se explicará el flujo de trabajo a seguir para obtener la predicción de resistencia al avance con un modelo en tres dimensiones. En este caso, existen dos supuestos, que se quiera obtener la predicción a través de uno de los modelos de los ejemplos o a través de una geometría que ha sido creada en otro software y se va a exportar a *FreeCAD Ship*.

Modelo de ejemplo.

En el supuesto de querer realizar la predicción de resistencia al avance a un modelo en tres dimensiones incluido en *FreeCAD Ship* el primer paso, tras seleccionar el módulo “Ship”, se pulsa en el icono de cargar geometría, que se muestra a continuación en la imagen H.4.



Figura H.4: Icono para cargar geometría en FreeCAD.



Tras seleccionar dicho icono, se despliega una ventana de interacción para seleccionar el ejemplo deseado, se muestra a continuación, en la imagen [H.5](#).

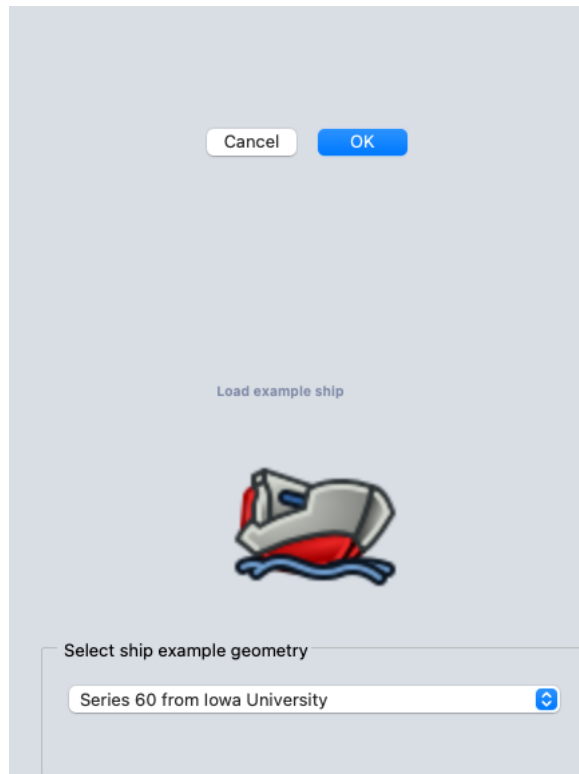


Figura H.5: Interfaz gráfica para seleccionar geometría de ejemplo.

Finalmente, tras aceptar el modelo a cargar, aparece la geometría del buque seleccionado y en el lateral izquierdo un árbol de buques abiertos. Se muestra la vista en la imagen [H.6](#)

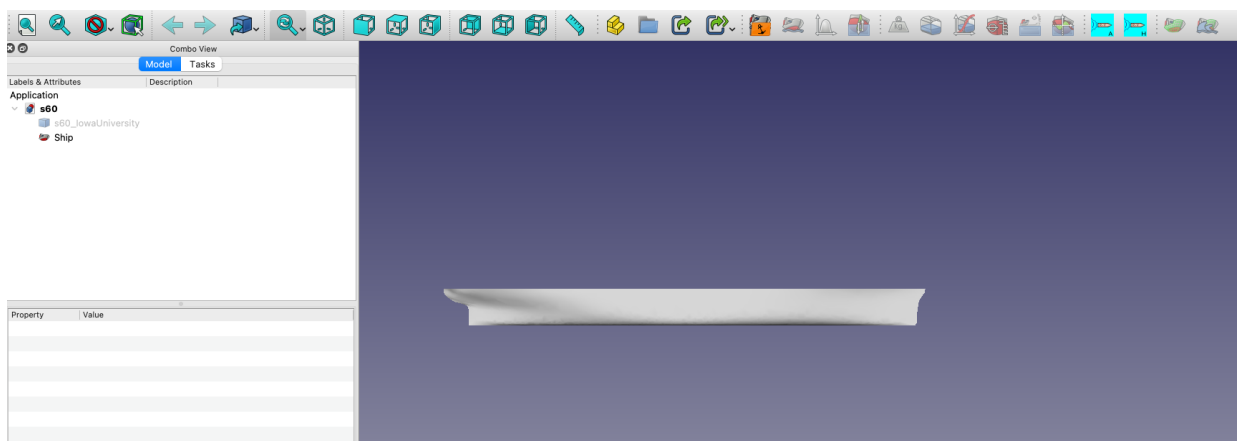


Figura H.6: Vista principal de FreeCAD Ship tras seleccionar un ejemplo.

Una vez cargado el modelo de ejemplo, para realizar la predicción de resistencia al avance hay que seleccionar el buque y posteriormente, seleccionar el método escogido. Tras desplegarse la interfaz de usuario del método seleccionado, se rellenan los campos que no pueden calcularse a través del modelo, como las velocidades, por ejemplo y se pulsa en aceptar. A continuación, se presentarán los resultados obtenidos, detallado en apartados posteriores.

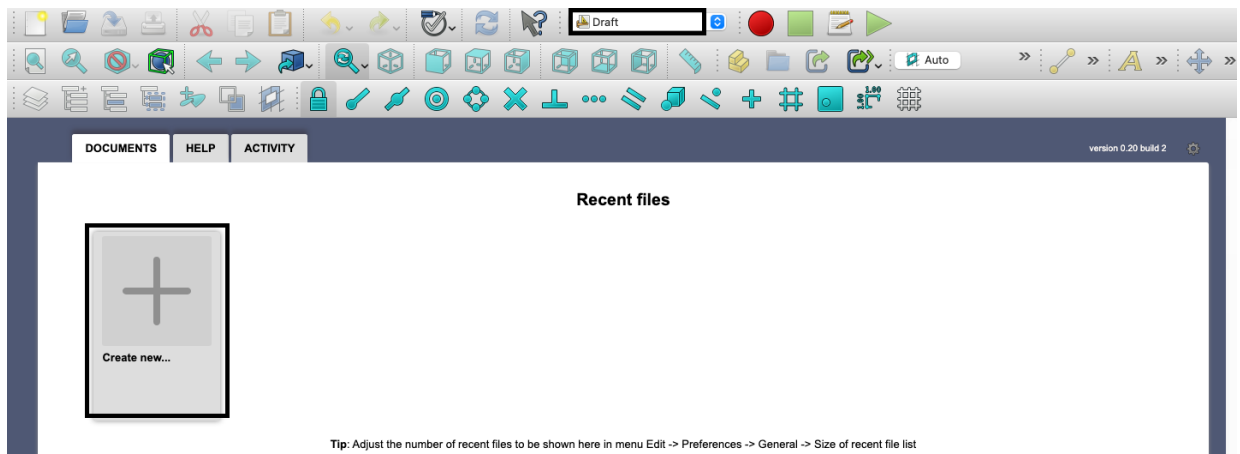


Figura H.7: Ventana de inicio para abrir un nuevo documento tras seleccionar el módulo ‘ ‘Draft’ ’ en *FreeCAD*.

Importando la geometría.

Otra opción, es importar la geometría modelada en otro software. Para ello, el primer paso es exportar la geometría en el software de modelado como “.step”. Es importante recalcar que el modelo debe estar completamente cerrado para lograr una importación correcta y adecuada.

A continuación, en *FreeCAD* se accede al módulo interno: “Draft”, viene instalado por defecto, y se abre un nuevo Documento. En la imagen H.7, se muestra la ventana principal para abrir un nuevo documento de trabajo.

Posteriormente, se accede a: “Archivo” y a continuación: “Importar...”. Se desplegará una ventana en la que se debe seleccionar el modelo que se desea importar.

Tras importar el modelo, se observará en la ventana principal la geometría, en la imagen H.8, se muestra un ejemplo de la vista principal.

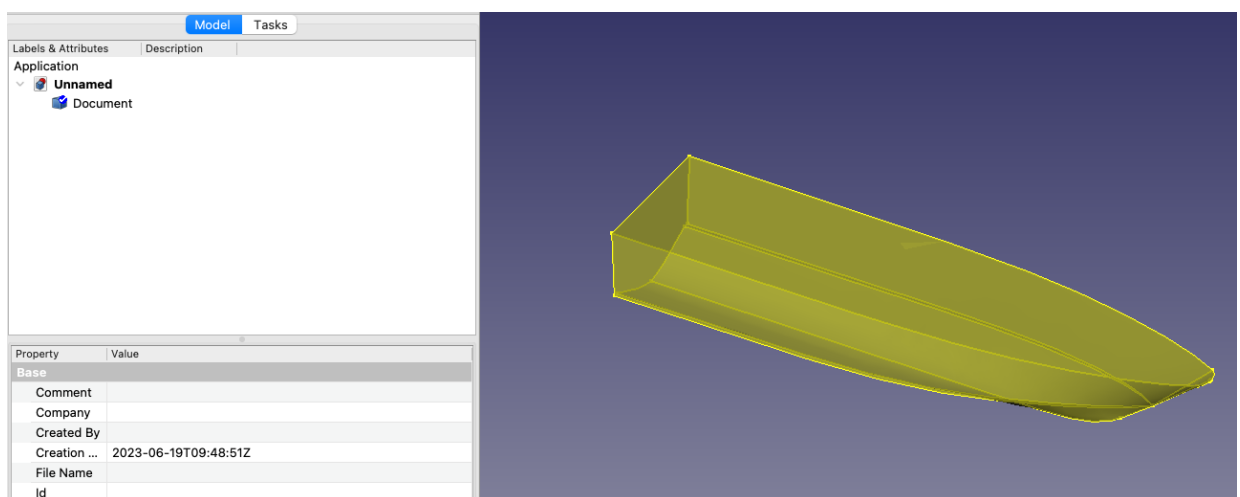


Figura H.8: Vista principal en *FreeCAD* tras importar la geometría de un barco.

Tras importar el modelo hay que crear con él una superficie. Para ello, con el modelo seleccionado en el árbol que aparece en el lateral izquierdo, se accede a: “Modificación” y a continuación “Degradar”. Ahora mismo, se ha explotado el modelo seleccionado en diversas caras. A con-



tinuación, se vuelve a acceder a “Modificación” y se pulsa, en este caso, “Actualización”. Se puede observar que ha cambiado el árbol lateral, se ha creado una cascara. De nuevo, con la cáscara seleccionada, se actualiza la geometría, es decir, se realiza el paso anterior para actualizar las caras a una cáscara. En las imágenes [H.9](#), [H.10](#), y [H.11](#), se observa la evolución del árbol lateral en los diversos pasos mostrados anteriormente.

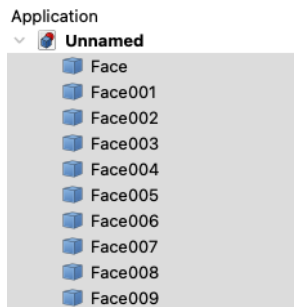


Figura H.9: Modelo importado en *FreeCAD* explotado en sus caras.

Figura H.10: Modelo importado en *FreeCAD* agrupado como cáscara.

Figura H.11: Modelo importado en *FreeCAD* agrupado como sólido.

Actualmente, se ha conseguido una superficie cerrada para poder crear un buque. A continuación, se accede al módulo :“Ship”. Con el buque seleccionado se accede al menú: “Crear un buque nuevo”, en la imagen [H.12](#) se muestra un ejemplo de la visualización de la ventana. Posteriormente, tras acceder al menú mencionado anteriormente, se desplegará una ventana para crear el buque con la geometría seleccionada, se muestra una imagen ejemplo en [H.13](#). Es importante recalcar que a la hora de modelar la geometría del buque, se debe colocar la cuaderna maestra en el punto “cero” para que posteriormente en *FreeCAD* se coloquen bien las dimensiones, sino, en el módulo “Draft” habrá que realizar un movimiento del sólido para colocarlo adecuadamente.

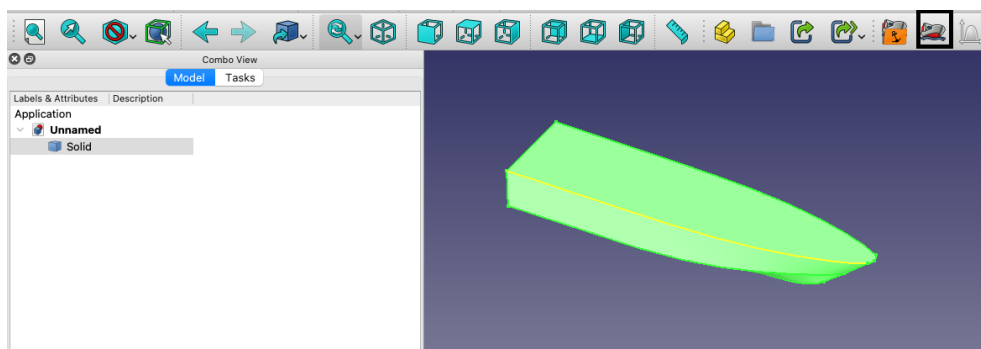


Figura H.12: Vista de ejemplo de la ventana de *FreeCAD Ship*, previamente a crear un buque.

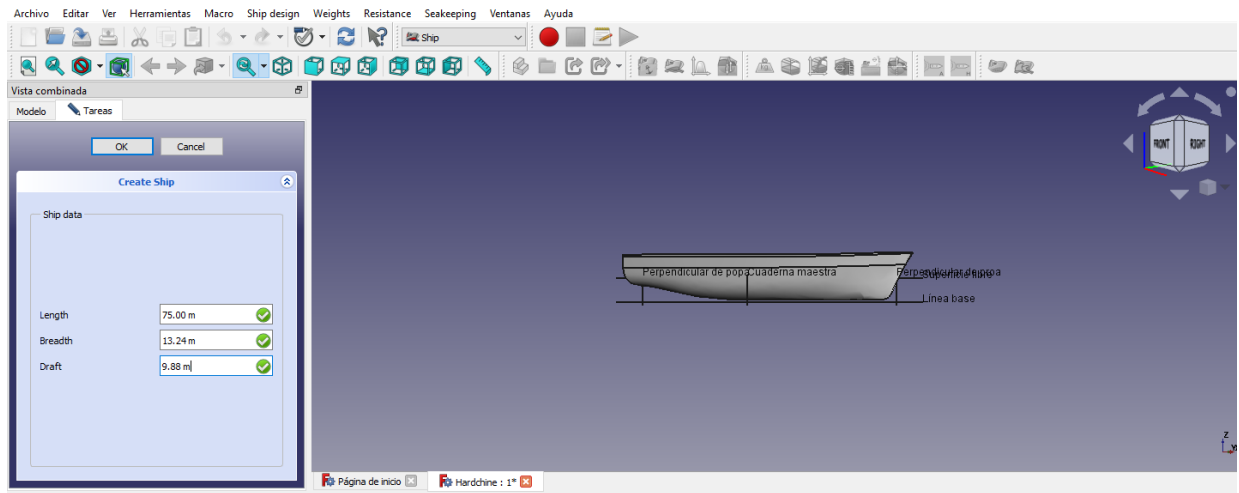


Figura H.13: Vista de ejemplo para crear un nuevo buque en *FreeCAD Ship*.

A continuación, tras validar las dimensiones, se crea un buque nuevo y aparecerá como tal en el árbol situado en el lateral izquierdo. Se muestra a continuación, en la imagen H.14, una ejemplo de la vista que tendrá el usuario tras crear un nuevo buque.

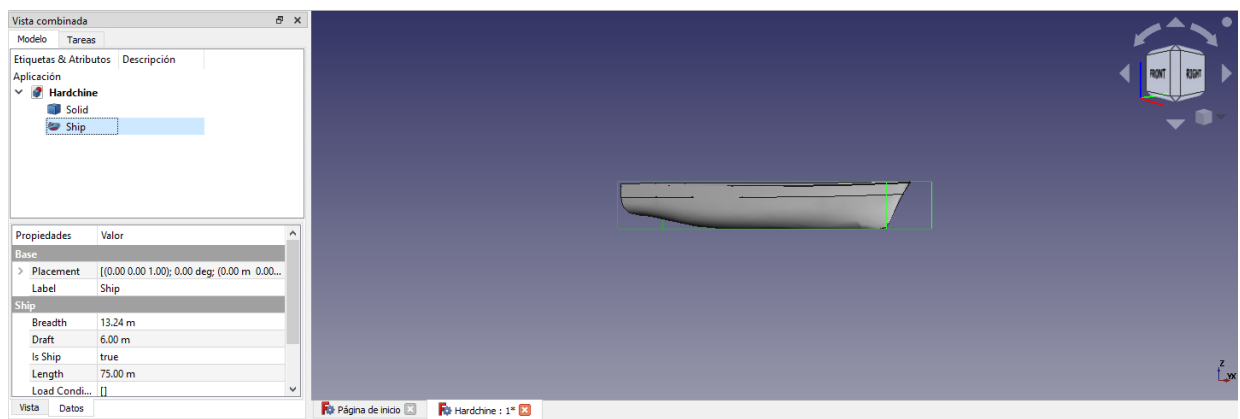


Figura H.14: Vista de ejemplo tras crear un nuevo buque en *FreeCAD Ship*.

Finalmente, para obtener la predicción de resistencia al avance con el modelo se siguen los mismos pasos que en el caso de utilizar un modelo de ejemplo. Con el modelo seleccionado, se accede al método deseado, a continuación, se rellenan los valores que no son obtenibles a través de un modelo en tres dimensiones y se acepta. Posteriormente, se presentan los valores obtenidos.

Predicción de resistencia al avance sin modelo en tres dimensiones.

Este supuesto es más sencillo de realizar que los anteriores. Tras acceder al módulo “Ship” se abre un nuevo documento. A continuación, se seleccionará el método deseado para realizar la predicción de resistencia al avance y se rellenan, en este caso, todos los valores de la interfaz gráfica que se despliega, teniendo en cuenta algunos detalles:

- Método de Amadeo. La eslora en la flotación y la superficie mojada se pueden calcular a



través de fórmulas empíricas, en caso de ser desconocidos. Tal y como se muestra en la ayuda que aparece si se mantiene el ratón encima de las diversas etiquetas, en caso de no conocer dichos valores, en *FreeCAD Ship* habrá que introducir el valor igual a cero.

- Método de Holtrop-Mennen. Al igual que en el método de Amadeo, pero en este caso únicamente con la superficie mojada, si el valor se fija igual a cero, el módulo obtendrá el valor de la superficie mojada del buque a través de fórmulas empíricas.

Finalmente, tras validar los valores introducidos, se pulsa aceptar y se calculará la predicción de resistencia al avance.

Valores obtenidos.

Finalmente, tras realizar la predicción de resistencia al avance, en cualquiera de los supuestos mencionados anteriormente, se presentarán los valores obtenidos de la siguiente forma.

- Gráficas. Se presentarán tres gráficas, comparando: la variación de la resistencia total en función de la velocidad; la variación de la potencia, tanto efectiva como al freno, en función de la velocidad; y la variación de los coeficientes resistivos en función de la velocidad. Bajo las gráficas, se sitúan varias pestañas que definen cada una de las gráficas para poder acceder a ellas.
- Hoja de cálculo. Todos los valores obtenidos se guardan en una hoja de cálculo que queda a disposición del usuario. Se puede acceder a dicha hoja de cálculo a través del árbol que se encuentra en el lateral izquierdo. En él aparecerá el nombre de la hoja de cálculo y haciendo doble click sobre ella, se abrirá.

A continuación, en la imagen H.15, se muestra la presentación de los resultados y cómo acceder a ellos.

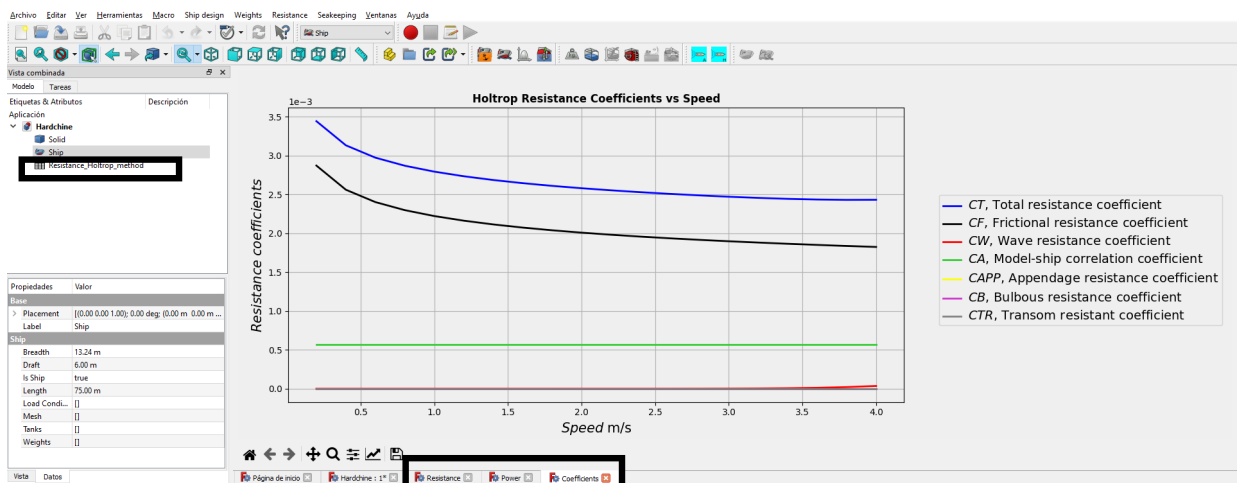


Figura H.15: Vista ejemplo de la presentación de resultados obtenidos en *FreeCAD Ship*.