



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 4

MÓDULO PARA EL DISEÑO DE CASCOS DE EMBARCACIONES Y LOS CÁLCULOS HIDROSTÁTICOS

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Kevin Pérez Mandina

Tutores: Dr.C Augusto Cesar Rodríguez Medina

Dr.C Leonel Galeano Vasconcelos

Ing. Fernando Martín Padrón

Ing. Manuel Fornés Martínez

La Habana, 2018

The greatest pleasure in life is to do what people say you can not do.
Walter Bagehot

Dedicatoria

A mis padres y hermano, en especial a mi mamá.

Agradecimientos

Agradezco a mis padres por su sacrificio día a día, apoyo incondicional y por haberme hecho el hombre que soy y del cual estoy orgulloso, por haberse quedado hasta tarde ayudándome a terminar este trabajo. En especial a mi querida mamá, a la cual esta tesis esta dedicada. A mi hermano, a mis primos, tíos y abuelos, entre ellos a mi tía Tania, que en más de una ocasión me ayudó en esta ardua tarea. A mis tutores por apoyarme en la realización de este trabajo. Agradezco a los amigos que he hecho a lo largo de estos cinco años, algunos están todavía, otros se han marchado por cuestiones de la vida. A Anabel por brindarme su apoyo en momentos difíciles para mi. A Jessica la primera amistad que hice en la UCI y que hasta el día de hoy ha estado ahí para mi. A Mario Sánchez por aguantarme las charlas de mis crisis existenciales y por darme consejos exactos en la hora precisa para no perder el rumbo. A Dayani mi compañera excepcional de equipo desde el 5103. A Maykel, Jonathan, José Armando, Gustavo, Cabo, Feito y Machin que me apoyaron cuando perdí mi laptop. A mis compañeros de apartamento Erick, Pabel, Michel. En especial a Gabriel, Mario y Frank.compañeros míos en toda esta etapa de mi vida. A todos muchas gracias.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Kevin Pérez Mandina
Autor

Dr.C Augusto Cesar Rodríguez Medina
Tutor

Dr.C Leonel Galeano Vasconcelos
Tutor

Ing. Fernando Martín Padrón
Tutor

Ing. Manuel Fornés Martínez
Tutor

En el presente trabajo se exponen los resultados de un proceso de investigación y desarrollo destinado a la implementación de funcionalidades para el modelado por computadora de cascos de embarcaciones, encapsuladas en un módulo informático. Entre los objetivos planteados se encuentran el modelado de cascos de buques a partir de planos de forma, obtener las líneas del plano de forma a partir de un modelo virtual y la realización de los cálculos hidrostáticos preliminares. El módulo se desarrolló empleando el lenguaje C++, el framework Qt y la tecnología Open Cascade para el modelado. La metodología de desarrollo de software empleada fue AUP-UCI en su escenario 4. El resultado obtenido consiste en un módulo informático para el diseño naval. Se implementaron funcionalidades para el empleo de curvas y superficies interpoladas tipo NURBS. La importancia de este resultado reside en que ha sido desarrollado con tecnologías de código abierto, lo que permite elevar la soberanía tecnológica en la industria naval. Ofrece además, una alternativa al empleo de sistemas comerciales, que poseen elevados precios y son de difícil acceso para el país debido a las restricciones impuestas por el bloqueo económico y financiero de los Estados Unidos de América contra Cuba. Logra implementar funcionalidades para informatizar las tareas en el diseño de cascos de buques, esto reduce el tiempo de ejecución de esa tarea en caso de que se hiciera por procedimientos manuales.

Palabras clave: cálculos hidrostáticos, CAD, modelado naval, NURBS, plano de forma.

Introducción	1
1 Fundamentación Teórica	3
1.1 Aspectos preliminares sobre la situación problemática y el proceso de investigación	3
1.2 Modelado del casco utilizando curvas NURBS	7
1.3 Módulos o sistemas existentes para el diseño asistido por computadora de cascos de buques y cálculos hidrostáticos	12
1.3.1 Rhinoceros 3D	12
1.3.2 MaxSurf	12
1.3.3 Delftship	13
1.3.4 FreeShip	13
1.3.5 Módulo Ship del FreeCAD	14
1.3.6 Defcar	14
1.3.7 FORAN	15
1.3.8 Consideraciones del análisis de los sistemas existentes	15
1.4 Cálculos Hidrostáticos	16
1.4.1 Método Empírico	17
1.5 Metodología para el desarrollo	20
1.5.1 AUP-UCI	20
1.6 Herramientas y Tecnologías para el desarrollo	22
1.6.1 Visual Paradigm	22
1.6.2 Open Cascade Technology	24
1.6.3 Lenguaje de programación	24
1.6.4 Framework de desarrollo Qt	25
1.6.5 Sistema de control de versiones	25
1.7 Conclusiones del capítulo	26
2 Propuesta de Solución	27
2.1 Mapa Conceptual	27
2.2 Descripción de las funcionalidades	28

2.3	Requisitos	29
2.3.1	Requisitos Funcionales	29
2.3.2	Requisitos No Funcionales	30
2.4	Historias de usuario	30
2.5	Diseño	33
2.5.1	Estilo y patrón arquitectónico del software	33
2.5.2	Patrones de Diseño	35
2.5.3	Diagrama de Clases del Diseño	36
2.6	Conclusiones del Capítulo	36
3	Implementación y Pruebas	37
3.1	Implementación	37
3.1.1	Detalles técnicos de la implementación	37
3.1.2	Estándar de codificación	38
3.1.3	Diagrama de Componentes	38
3.1.4	Resultados de la implementación	38
3.2	Pruebas	40
3.2.1	Niveles de prueba	43
3.2.2	Métodos de prueba	43
3.2.3	Diseño de Casos de Prueba	44
3.2.4	Pruebas Unitarias	45
3.2.5	Pruebas de Aceptación	47
3.2.6	Resultados de las pruebas	49
3.3	Conclusiones del capítulo	50
	Conclusiones	51
	Recomendaciones	52
	Glosario	53
	Acrónimos	54
	Referencias bibliográficas	56
	Apéndices	58
A	Anexos	59
A.1	Historias de usuarios	59
A.2	Casos de Prueba	63

A.3 Aval 65

Índice de figuras

1.1	Planos transversales del casco de un barco	8
1.2	Planos longitudinales del casco de un barco	9
1.3	Planos horizontales del casco de un barco	9
1.4	Esloras en un barco	10
1.5	Calado y puntal en un barco	10
1.6	Curvas B-Spline modelando cuadernas	11
1.7	Red de puntos de control modelando casco de barco	11
2.1	Mapa Conceptual	27
2.2	Diagrama de Arquitectura de programa principal/subprograma	34
2.3	Diagrama de Clases del Diseño	36
3.1	Estándar de codificación	39
3.2	Estándar de codificación	39
3.3	Estándar de codificación	40
3.4	Diagrama de Componentes	40
3.5	Líneas transversales	41
3.6	Líneas horizontales	41
3.7	Líneas longitudinales	41
3.8	Plano de forma y casco modelado	42
3.9	Modelo completo y mitad del modelo	42
3.10	Datos y cálculos hidrostáticos	42
3.11	Pruebas Unitarias	48
3.12	Iteraciones de las pruebas	50

Índice de tablas

1.1	Valores de f	18
1.2	Valores de C_{KG}	20
2.1	Historia de Usuario Importar bitmap con el plano de forma del casco	30
2.2	Historia de Usuario Introducir los parámetros del casco	31
2.3	Historia de Usuario Realizar cálculos hidrostáticos	32
3.1	Datos Técnicos	38
3.2	Diseño de Casos de Prueba RF 4	45
3.3	No Conformidades	49
A.1	Historia de Usuario Importar modelo de casco de buque	59
A.2	Historia de Usuario Completar el modelo del casco de buque importado	60
A.3	Historia de Usuario Modelar un casco a partir de una imagen bitmap.	61
A.4	Historia de Usuario Modelar las líneas de los contornos del casco de buque	62
A.5	Diseño de Casos de Prueba RF 1	63
A.6	Diseño de Casos de Prueba RF 2	63
A.7	Diseño de Casos de Prueba RF 3	63
A.8	Diseño de Casos de Prueba RF 5	64
A.9	Diseño de Casos de Prueba RF 6	64
A.10	Diseño de Casos de Prueba RF 7	64

En el diseño de embarcaciones se emplean sistemas de Diseño Asistido por Computadora (CAD, por sus siglas en inglés) tanto de uso específico como de uso general. Como todas las tecnologías CAD son productos costosos e inaccesibles para la industria cubana es factible el desarrollo de tales herramientas empleando sistemas de código abierto existentes. Estos pueden ser estudiados o reutilizados en nuevas implementaciones en plazos relativamente cortos.

El empleo de los sistemas CAD se realiza por ingenieros, dibujantes y diseñadores, especializados en el diseño y construcción de buques. En los apartados que siguen se muestran las funcionalidades principales, que se han implementado en el desarrollo del módulo, que permitirá modelar los cascos de embarcaciones por dos vías: la primera es a partir de la captura de líneas de construcción características, existentes en los planos de forma y la segunda importando los datos numéricos asociados a un prototipo de embarcación.

Durante el proceso de investigación y desarrollo se realizó un estudio preliminar para obtener información sobre el estado del arte en esta rama de la tecnología, así como de los elementos matemáticos que sustentan la construcción de configuraciones complejas. Por la consulta bibliográfica realizada se verificó la existencia de herramientas informáticas comerciales de código abierto, que son empleadas en el diseño de cascos de buques.

En el presente trabajo se describe la implementación de las funcionalidades, que permiten al diseñador naval modelar el casco de un buque, a partir de sus planos de forma por medios computacionales. También se muestra como emplear los datos de las configuraciones de buques obtenidas, para realizar los cálculos hidrostáticos preliminares.

En el caso de las tecnologías de código abierto los sistemas de uso específico para el diseño naval no son abundantes y su desarrollo en relación con las herramientas comerciales resulta limitado. No obstante, existen emprendimientos, como FreeShip, el módulo Ship de la aplicación FreeCAD, y una implementación en C++ de FreeShip denominada ShipCAD, que al momento de la presente investigación presentaba poco avance en su desarrollo.

Una ventaja importante en el desarrollo de sistemas para el diseño e ingeniería asistidos por computadoras con herramientas de código abierto, es la existencia de la tecnología Open Cascade, núcleo gráfico existente en software libre. Las funcionalidades que la misma provee, facilita aumentar la productividad del proceso de desarrollo.

El estudio preliminar realizado permitió establecer las ideas rectoras para el presente trabajo. En los capítulos siguientes se exponen los materiales y métodos utilizados, la propuesta de solución, el resultado

obtenido, las pruebas realizadas y las reflexiones sobre el mismo.

En este capítulo se explican aspectos de la situación problemática y el diseño de la investigación a realizar. Contiene información sobre sistemas CAD, tanto de código abierto como de herramientas comerciales, y su uso en la industria naval cubana. Además, aborda temas como la metodología y las herramientas de desarrollo que se emplean.

1.1. Aspectos preliminares sobre la situación problemática y el proceso de investigación

Los sistemas CAD se consideran tecnologías de avanzada con un peso importante en la automatización de los procesos productivos. En la bibliografía consultada se puede inferir que existen diversas herramientas informáticas que son empleadas para diseñar los cascos de los buques, algunas de uso general y otras especializadas. Entre las de uso general se pueden mencionar Rhinoceros 3D, Catia, SolidWork. Entre las especializadas se encuentran FORAN, Defcar, MaxSurf, Delftship; pero en ambos casos los sistemas son costosos y considerados como de alta tecnología.

También, se puede afirmar que no todos los sistemas satisfacen algunas tareas de diseño realizadas por los constructores navales como:

- Visualizar y manipular modelos de casco de buque.
- Construir el casco a partir del plano de forma.
- Obtener los planos de forma a partir del casco del buque.

El diseño manual de cascos para embarcaciones tiene una serie de desventajas como (Colmena, 2015):

- Requiere de grandes superficies con equipos costosos y mucha mano de obra.
- El alisado de líneas depende de los criterios del trazador y no de normas absolutas e independientes del operario.
- El proceso requiere de varias transmisiones de datos, que inevitablemente, conducirán a errores en el montaje con gastos elevados en las rectificaciones.
- Necesita mucho tiempo y no se puede esperar a que esté terminado para pedir los materiales, por lo que producirá una necesidad de mayor cantidad de materiales.
- Es difícil convertir los datos del trazador en formas adecuadas para el trabajo con las máquinas dirigidas electrónicamente.

Principalmente los ingenieros navales utilizan tres formas para llevar a cabo el diseño de una nueva embarcación; series hidrodinámicas, canales de experiencia hidrodinámica y a partir de un prototipo, para saber más ver (Wu, Cui y Zhou, 2001). En este trabajo solo se aborda el diseño a partir de prototipos. En esta forma los ingenieros navales siempre se apoyan en un prototipo funcional, del cual obtienen los planos de forma. A partir de estos modelan el casco del modelo a construir variando los parámetros dimensionales de una embarcación ya sean estos: eslora, manga y puntal. (Castro, Azpíroz y Fernández, 1997)

El plano de forma contiene los cortes en planos de las tres dimensiones del casco del buque. En los sistemas CAD este proceso se encuentra automatizado. Existen software que mediante un reconocimiento de patrones modelan el casco del buque insertándole el plano de forma. Otros software utilizan el dibujo del usuario sobre el plano de forma donde se obtiene el casco del buque. Esta última funcionalidad se encuentra en el Rhinoceros 3D. Además del modelado, estos software facilitan los cálculos hidrostáticos del casco.

En Cuba existen varias instituciones que se dedican a la construcción, reparación y transformación de embarcaciones. Algunas de ellas son: el Astillero de Casa Blanca (CDC), el Centro de Proyectos Navales (CEPRONA), el Centro de Investigación y Desarrollo Naval (CIDNAV) y DAMEX.

En el CEPRONA se construyen y diseñan embarcaciones pesqueras como: camarones, catamaranes, bicicletas acuáticas y canoas. También se construyeron una serie de patanas remolcadas en el 2005. Este centro utiliza el software WOLFSON, para a partir del diseño de formas de las embarcaciones realizar el cálculo de resistencia al avance.

El CIDNAV utiliza el software MaxSurf para el diseño naval y el AutoCAD e Inventor para el diseño mecánico. El CDC está avalado por el Lloyd's Register. En este se construyeron remolcadores de altura para PDVSA, avalados por el Lloyd's Register. Actualmente existen tres modelos de este tipo de servicio en PDVSA.

CDC y DAMEX son los principales astilleros de construcción y reparación de barcos en Cuba. La sucursal

DAMEX que se encuentra en Cuba está situada en Santiago de Cuba, donde se construyeron catamaranes para la Isla de la Juventud empleando aluminio. Actualmente se están construyendo buques multipropósito para Venezuela en DAMEX, CDC y Asticar. Este último es un astillero que se encuentra en la bahía de La Habana, está avalado por el buró francés VERITAS.

Con la actualización del modelo económico existe una reactivación de la industria naval. Se trazó como objetivo desarrollar la bahía de La Habana para el turismo y mejorar las condiciones de la lancha de Regla como transporte urbano. Las empresas del MINTRANS como: la Naviera Cubana de Cabotaje, Navegación Caribe demandan la necesidad de producción en la parte de diseño, construcción, transformación y reparación de embarcaciones. En la actualidad hay un desarrollo de lanchas rápidas, vehículos submarinos, conversión de embarcaciones de la economía a embarcaciones de guerra. También existe un proyecto de producción de limpia bahías y de embarcaciones patrulleras.

En Cuba como los laboratorios hidrodinámicos son limitados, se utiliza el canal de remolque del Empresa Nacional de Análisis Técnico del Agua (ENAST). Su función es el estudio y cálculo de las corrientes en los ríos, CIDNAV lo utiliza para hacer los cálculos de resistencia al avance. Al existir una gran limitación para realizar ensayo de laboratorio, se necesita utilizar software de ingeniería naval para realizar la generación de formas del casco y la realización de cálculos hidrostáticos.

En el CDC se utiliza el Defcar del cual se pagó su licencia de uso. Esta estaba limitada a una versión de Windows, al establecerse otra versión de Windows en la empresa no existía compatibilidad con el nuevo sistema. Esto obligaba a comprar nuevamente el software para la nueva versión de Windows o mantener el sistema operativo viejo.

Cuba tiene prohibido el acceso a los softwares considerados de alta tecnología debido a las cláusulas del bloqueo económico y financiero impuesto por el gobierno de los Estados Unidos de América a nuestro país. Por otra parte, estos sistemas comerciales son costosos; y su adquisición en caso de no existir el bloqueo, solamente estaría justificada si la comercialización de los productos resultantes asegura la recuperación de la inversión. Los mismos se distribuyen, principalmente, bajo licencias de uso, lo que significa que el sistema informático no se adquiere como propiedad, sino que se autoriza su uso por un tiempo determinado. Además, a los resultados de diseño en ingeniería, como obras del intelecto humano, le son aplicables las regulaciones sobre la propiedad intelectual, por lo que si fueran obtenidas con herramientas sin licencia no es posible su comercialización.

Otro problema al usar los softwares extranjeros es que tienen problemas técnicos por la parte de ingeniería naval e informáticos por las versiones. Cuando se quiere generar una forma de casco nueva que se ajuste a las condiciones de Cuba, hay que utilizar formas semejantes que trae el software y a estas le faltan datos que son muy específicos de Cuba. Por tanto, se puede afirmar, que los software disponibles para la ingeniería naval no satisfacen las exigencias del diseño naval en Cuba.

A partir de lo antes mencionado se llega al siguiente **problema de investigación**:

- ¿Cómo automatizar las tareas de diseño de cascos de buques y realizar los cálculos hidrostáticos pre-

liminares?

El **objeto de estudio** de la investigación esta asociado a la línea del gráfico por computadora, teniendo como campo de acción, los algoritmos para el modelado geométrico y topológico aplicado al diseño naval.

Para la solución del problema antes mencionado se propone como **objetivo general**:

- Desarrollar un módulo para el diseño de cascos para embarcaciones y los cálculos hidrostáticos preliminares con tecnologías de código abierto.

A partir del objetivo general definido, se derivan los siguientes **objetivos específicos**:

- Diseñar las funcionalidades para el modelado de cascos de buques y los cálculos hidrostáticos.
- Implementar funcionalidades para el modelado de cascos de buques y los cálculos hidrostáticos.
- Realizar pruebas a las funcionalidades desarrolladas.

Para dar cumplimiento a estos objetivos específicos se proponen las siguientes **tareas de la investigación**:

- Búsqueda y revisión bibliográfica sobre el objeto de investigación.
- Fundamentación de las necesidades y requerimientos de proyecto que avalan la conformación del módulo.
- Análisis de los códigos fuentes disponibles, como FreeShip, ShipCAD, el módulo Ship de FreeCAD; esto implica además la comparación con las funcionalidades de los módulos homólogos existentes en aplicaciones comerciales para el diseño y la ingeniería asistidos por computadoras en el área de la tecnología naval.
- Proceso de síntesis (obtención de conclusiones, resultados, definición de las formas de hacer y conformación de la estructura del módulo).
- Estudios de los aspectos de la metodología de desarrollo de software Agile Unified Process versión UCI (AUP-UCI, por sus siglas en inglés) que se aplicarán en la investigación.

- Obtención de requisitos a partir de los sistemas disponibles, ya sean estos propietarios o de código abierto, o mediante videos de los mismos y consideraciones de los potenciales usuarios.
- Definición de la arquitectura del módulo, lo que implica además definir los patrones de diseño con los que cumplirá.
- Determinación de la estructura de clases del módulo.
- Diseño de la interfaz gráfica del módulo.
- Estudio de las metodologías de diseño de cascos para embarcaciones y los cálculos hidrostáticos.
- Implementación del módulo destinado al diseño de cascos para embarcaciones y los cálculos hidrostáticos.(incluye adicionar funcionalidades requeridas por el usuario o existentes en aplicaciones comerciales y que no están presentes en FreeCAD).
- Pruebas al módulo.

Para resolver y dar cumplimiento a los objetivos y las tareas propuestas se empleó como métodos de investigación:

Métodos teóricos:

Análisis y síntesis: se empleó para la construcción y desarrollo de la teoría, para la profundización en el tema y la sistematización del conocimiento.

Métodos empíricos:

Observación: se empleó para caracterizar las soluciones, teniendo en cuenta distintos datos y formas de modelado, de otras aplicaciones CAD (FreeCAD, FreeShip, SolidWork, Rhinoceros 3D, Delftship), y así establecer los requisitos con las principales funcionalidades de estos.

Experimentación: para realizar pruebas de funcionamiento al módulo.

Medición: durante los experimentos o pruebas de funcionamiento se toman los tiempos de modelado y cálculo para poder estimar de manera objetiva la eficiencia del módulo.

1.2. Modelado del casco utilizando curvas NURBS

Para facilitar la comprensión de los aspectos relacionados con el casco de un barco, resulta conveniente exponer algunas definiciones y conceptos relacionados con el diseño naval que serán utilizados en las descripciones.

Los planos de forma son la representación de las líneas características de la configuración del casco del buque, obtenidas a partir de cortes realizados al modelo con series de planos horizontales, longitudinales y transversales, cuyas intercepciones se proyectan sobre los tres planos de coordenadas principales (Tupper, 1996).

Las líneas de agua son las curvas del plano de formas resultantes de la intersección del modelo con planos de corte horizontales, uno de los cuales es el que coincide con la línea de flotación del buque (ibíd.).

Las líneas longitudinales se forman a partir de la intersección del modelo con planos de corte longitudinales, uno de ellos coincide con la línea de crujía o eje simétrico del casco (ibíd.).

Las líneas transversales o de cuadernas se forman cuando se intersecta al modelo del casco con planos de corte transversales, uno de ellos coincide con la cuaderna maestra (ibíd.).

En el caso de los planos de forma se distinguen tres tipos: el Plano diametral o transversal es perpendicular al horizontal (también denominado plano base) y corta el modelo del buque para obtener secciones transversales ver figura 1.1; el plano vertical o longitudinal trazado en dirección Proa-Popa que también es perpendicular al horizontal y contiene las proyecciones de las curvas de forma longitudinales ver figura 1.2; el plano horizontal contiene las proyecciones de las curvas obtenidas por los cortes del modelo con planos horizontales, denominadas líneas de agua ver figura 1.3 (ibíd.).

La eslora es la longitud del buque, se mide frecuentemente en pies (SailandTrip, 2017) ver figura 1.4.

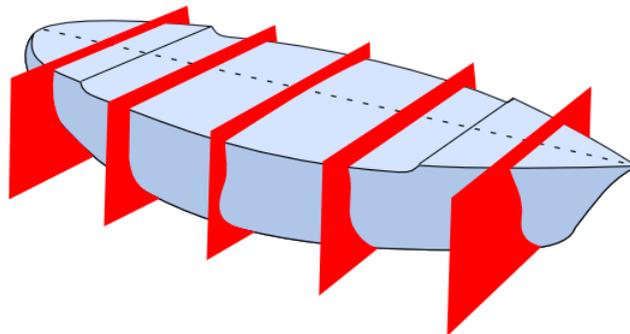


Figura 1.1. Planos transversales del casco de un barco

La eslora entre perpendiculares es la medida entre las perpendiculares de proa y popa. Entendemos perpendicular de popa la medida generalmente tomada en línea al eje del timón y como perpendicular de proa a la intersección del casco con la línea de flotación a plena carga y con asiento nulo, es decir, que el calado de proa y el calado de popa son iguales (ibíd.) ver figura 1.4.

La manga es el ancho del barco. Como la manga no es constante a lo largo de todo el barco, llamaremos manga máxima a la parte más ancha del barco que normalmente suele coincidir con la cuaderna maestra (ibíd.).

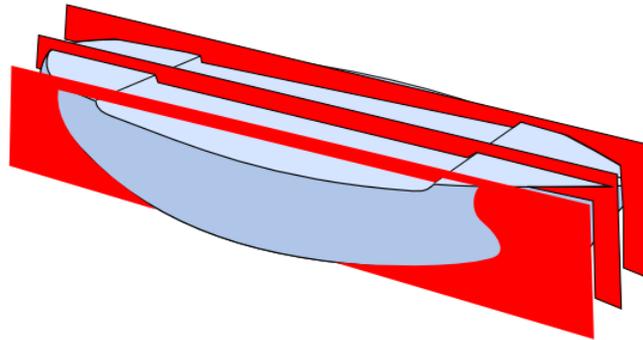


Figura 1.2. Planos longitudinales del casco de un barco

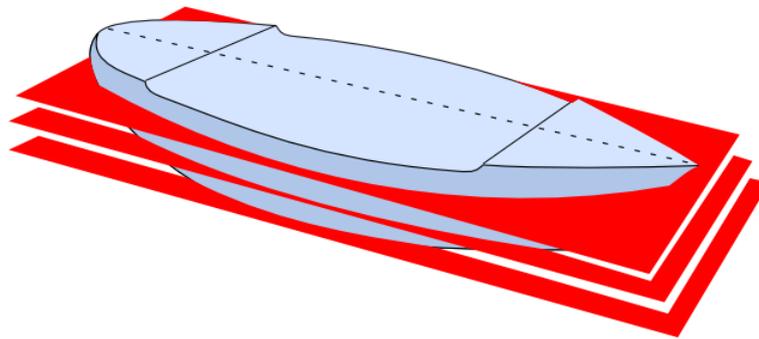


Figura 1.3. Planos horizontales del casco de un barco

El puntal es la altura del buque o distancia vertical en metros medido desde la cara inferior del casco en su intersección con la quilla y la línea de cubierta principal o la cara superior del trancanil (SailandTrip, 2017) ver figura 5.

El calado es la altura de la parte sumergida del casco, también lo podemos definir como la medida vertical tomada desde la quilla hasta la línea de flotación (ibíd.) ver figura 1.5.

El casco de un buque es una superficie compleja, por lo que la generación y definición de las formas es casi un trabajo artesanal. Con el uso del ordenador se posibilita el trabajo a pesar de que el diseño a mano de las formas mediante plantillas todavía sigue extendido (Arribas, 2004).

Si el casco del buque se corta mediante planos perpendiculares al eje longitudinal del mismo, se obtienen unas curvas denominadas cuernas. También, al cortar el casco del buque por planos paralelos al plano de



Figura 1.4. Esloras en un barco



Figura 1.5. Calado y puntal en un barco

la base, se obtienen unas curvas denominadas líneas de agua. Secciones perpendiculares a la vez a los planos que contienen las cuadernas y al plano base, forman unas curvas denominadas longitudinales.

Las superficies que mejor modelan el casco de un buque son las denominadas superficies Non Uniform Rational B-Spline (NURBS, por sus siglas en inglés). La base de estas superficies son unas curvas planas denominadas B-Spline y es imprescindible entender estas curvas para comprender el funcionamiento de una superficie NURBS (Arribas, 2004).

NURBS es el modelo matemático más común utilizado para el modelado de curvas y superficies complejas en CAD / Fabricación Asistida por Computadora (CAM, por sus siglas en inglés) y gráficos por computado-

ra (Farin, 1999).

Una curva B-Spline es una secuencia de segmentos de curva que están conectados entre sí a partir de un polígono de vértices denominado polígono de control. Estos forman una sola curva continua que se dota de un esquema de interpolación que relaciona vértices del polígono conectados de extremo a extremo. Pueden existir muchas curvas que estén asociadas el mismo polígono de control (Sederberg, 2014).

En la figura 1.6 aparecen algunas cuadernas modeladas con puntos reales mediante B-Spline.

Para modelar el casco en su totalidad es necesario obtener una red de puntos de control a partir de las

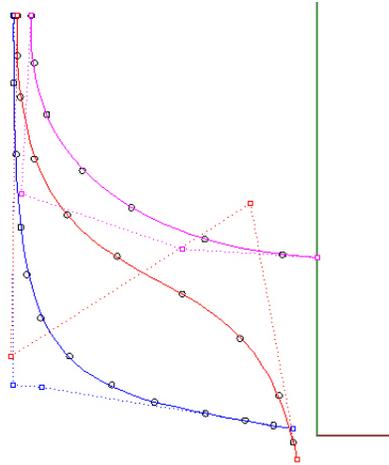


Figura 1.6. Curvas B-Spline modelando cuadernas

B-Spline que definen a las curvas características del casco. En la figura 1.7 se muestra como es una red de puntos de control. Un aspecto importante es la manipulación de los puntos de control de cada curva. Estos deberán coincidir con algunos puntos de intersección o existir la posibilidad de ajustar tal coincidencia. De esta manera se consigue el esqueleto que define el forro o casco del buque. Luego se interpolan estos puntos, obteniendo finalmente la superficie del casco del buque.

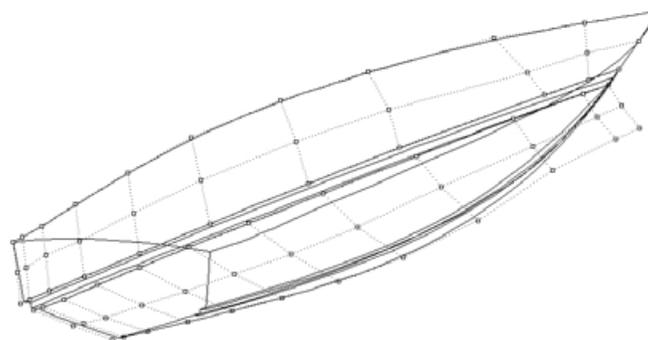


Figura 1.7. Red de puntos de control modelando casco de barco

1.3. Módulos o sistemas existentes para el diseño asistido por computadora de cascos de buques y cálculos hidrostáticos

Se describen los módulos o sistemas existentes tanto comerciales como libres y los conceptos sobre los cálculos hidrostáticos.

1.3.1. Rhinoceros 3D

Rhinoceros 3D es una herramienta para modelado en 3D basado en NURBS . Se ha popularizado en las diferentes industrias, por su diversidad, funciones multidisciplinarias y el relativo bajo costo. Las extensas opciones para importación y exportación en el programa es una razón del crecimiento de su uso. La gran variedad de formatos con los que puede operar, le permite actuar como una herramienta de conversión, permitiendo romper las barreras de compatibilidad entre programas durante el desarrollo del diseño. Existe también la versión RhinoMarine, que se adapta a las características especiales requeridas por los diseñadores navales incluidos el carenado, las superficies desarrollables y el análisis de curvaturas, etc. Además, Rhinomarine tiene el complemento Orca3D que es un plug-in externo que ofrece un paquete completo de herramientas especializadas para el diseño naval y el análisis en Rhinoceros (Rhinoceros, 2018).

Una de las ventajas que aporta el Rhinoceros 3D como modelador de cascos de buques, es su facilidad para aprender a manejarlo. Permite ajustar fácilmente la geometría del casco a lo que se quiere modelar. También tiene herramientas para realizar el alisado del casco (fairing) lo cual es indispensable para obtener un buen modelo 3D y que pueda ser utilizado posteriormente en la fase de análisis numérico (ibíd.).

1.3.2. MaxSurf

Es un software australiano. Ha sido ampliamente utilizado durante más de 25 años en miles de embarcaciones comerciales, barcos militares, embarcaciones y yates de motor y de vela en todo el mundo. Se ha utilizado para desarrollar los catamaranes de aluminio de alta velocidad más grandes y rápidos del mundo (más de 40 nudos) (Surf, 2017).

Este sistema proporciona a los arquitectos navales herramientas potentes integradas para el modelado del casco, análisis de estabilidad, resistencia y potencia, movimientos, definición de estructura del casco primario y análisis de elementos finitos de la estructura del casco (ibíd.).

Para la virtualización de cascos se utiliza la tecnología de modelado de superficies NURBS en la creación de formas complejas de cascos. Incluye muchas capacidades potentes como el recorte dinámico, la transformación paramétrica, el cambio de tamaño proporcional del modelo y la compatibilidad con otras plataformas de software de Arquitectura Naval y CAD como AutoCAD, Microstation, Rhinoceros, GHS y SHCP. En particular, MaxSurf importa y exporta archivos nativos de Rhinoceros y el árbol de ensamblaje MaxSurf se importa a Rhinoceros, lo que hace que la edición de modelos sea muy sólida. Por ejemplo, los usuarios

pueden realizar todos los modelos de casco en MaxSurf e importar la geometría en Rhinoceros para agregar elementos como superestructura, equipamiento y otros detalles que no son necesarios para el análisis de estabilidad y la predicción de rendimiento (Surf, 2017).

En los cálculos hidrostáticos y de estabilidad se utiliza el modelo de casco 3D para analizar la estabilidad (daño intacto, dañado y probabilístico), ángulo de escora, resistencia longitudinal, longitud inundable, puntos de inundación y verificación con bibliotecas completas integradas de criterios de estabilidad (ibíd.).

1.3.3. Delftship

Es un completo y bastante sencillo programa de diseño en 3D de barcos, válido tanto para barcos a vela como a motor. Desde un simple boceto hecho a mano se puede obtener perfectamente el casco, ya que permite trabajar con imágenes de fondo. Calcula y exporta un informe con los cálculos hidrostáticos del modelo, así como de un montón de parámetros como superficie lateral, superficie mojada y volumen (Delftship, 2016).

Ofrece de forma simultánea 4 vistas del modelo: alzado, planta, perfil y perspectiva, con lo que vemos inmediatamente las modificaciones que hacemos. Muestra una imagen con volumen en 3D (render). Permite trabajar con capas. Muestra a la vez las coordenadas del punto seleccionado y las del cursor. Permite exportar/importar en diferentes formatos. Tiene un completo plano de líneas del modelo (ibíd.).

El software tiene dos versiones una gratis y otra de pago. La edición gratuita del software tiene características que incluyen cálculos hidrostáticos básicos y varios formatos de exportación de archivos. Se puede usar para visualizar rápidamente sus conceptos de diseño y es adecuado para uso personal o profesional (ibíd.).

La versión profesional incluye todas las características del modelador de casco Delftship gratuito. Además, tiene agregado características como carenado automático, desarrollo de placas, informes nativos personalizables, y herramientas que le permiten al diseñador naval una mejor completitud en el diseño (ibíd.).

1.3.4. FreeShip

FreeShip diseña en tres dimensiones el casco de un barco. Además, realiza cálculos hidrostáticos para modelar el casco más adecuado al uso de la embarcación (FreeShip, 2006).

Normalmente las aplicaciones para el diseño en 3D suelen tener demasiadas opciones y menús que hacen complicado si se quisiera comenzar a dibujar. No ocurre en este caso. La función de FreeShip es muy concreta. Tiene todo lo necesario para cumplirla sin distracciones (ibíd.).

El paquete de instalación de FreeShip Plus incluye catorce diseños de naves de diferentes categorías y tipos: veleros, pesqueros, cargueros, etc. Son de gran ayuda como ejemplo o como punto a partir del cual modificar el modelo (ibíd.).

1.3.5. Módulo Ship del FreeCAD

FreeCAD-Ship trabaja sobre Ship entities, que deben crearse previamente sobre la geometría proporcionada. La geometría puede ser un conjunto de sólidos, pero debe seguir los criterios definidos a continuación (FreeCAD, 2017):

- Se debe proveer todo el casco (incluyendo el caso de buques no simétricos).
- La geometría de estribor debe incluirse en el dominio de las Y negativas.
- El origen (0,0,0) es la intersección de la Cuaderna maestra y la Línea base.

Para crear una Instancia de barco selecciona una geometría proporcionada por el módulo y se ejecuta la herramienta de creación de barco. Se deben introducir las dimensiones principales del buque. FreeCAD-Ship emplea un diseño de introducción progresiva de datos. Las operaciones más simples se pueden realizar con una cantidad de información relativamente pequeña, y conforme se desee realizar operaciones más complejas, se deberá ir introduciendo más información (ibíd.).

1.3.6. Defcar

El Sistema Defcar un paquete completo de software CAD/CAM para el proyecto y la producción de buques. Este Sistema compuesto por módulos independientes entre sí, permite al usuario diseñar formas, realizar cálculos de arquitectura naval, desarrollar el forro, modelar la estructura interna, hacer el despiece y preparar los ficheros de anidado y corte por control numérico. Hasta la fecha en torno a 60 empresas de 13 países distintos proyectan y construyen con Defcar todo tipo de buques. Como ejemplo, el Grupo ARMON entrega del orden de 45 remolcadores y buques pesqueros cada año con el Sistema Defcar. Todos los proyectos de la prestigiosa Oficina Técnica CINTRANAVAL también son realizados con este software. Por otra parte, grandes astilleros como IZAR Gijón, IZAR Carenas Cartagena o la oficina IZAR Carenas Ingeniería (que da soporte técnico a los 5 astilleros de IZAR Carenas) también utilizan el Sistema Defcar (Defcar, 2010).

Aunque los módulos del Defcar son independientes, todos ellos constituyen un sistema integrado que le permite no perder información alguna en la transferencia de ficheros. No se necesita otros programas para cubrir el proyecto y la producción de buques. El Sistema permite la importación/exportación de líneas (2 o 3D) mediante ficheros DXF, así como de superficies a través de IGES (ibíd.).

1.3.7. FORAN

El Sistema FORAN es un sistema de CAD/CAM/CAE naval creado por SENER para el diseño y producción de cualquier tipo de buque y artefacto offshore. Es un sistema multidisciplinario e integrado, que puede ser utilizado en todas las fases de diseño y producción del buque y disciplinas. El Sistema recoge toda la información en una base de datos única. FORAN está enfocado principalmente al diseño de (SENNER, 2018):

- Mercantes, pasaje, graneleros, quimiqueros, portacontenedores, cementeros o petroleros.
- Buques militares de superficie y submarinos, donde permite realizar un control de configuración, analizar alternativas de diseño (prototipos), gestionar formas, materiales y estándares especiales, así como introducir criterios personalizables.
- Buques específicos, remolcadores y buques de trabajo, buques hotel, pesqueros, buques de transporte de pescado, oceanográficos, etc.
- Offshore tanto para plataformas flotantes -ancladas y fijas- como para buques de transporte de personal a plataformas, ancleros, buques de suministro, rescate, contra-incendios o antipolución.

FORAN es el único sistema naval desarrollado por una compañía que al mismo tiempo es oficina técnica naval, y que fue concebido para solucionar los problemas de la propia oficina. Ello implica que se habla el mismo idioma del usuario y se entiende la problemática del cliente. FORAN permite el trabajo simultáneo sobre una misma base de datos desde centros ubicados en diferentes localizaciones geográficas. Gracias a ello, cualquier cambio es visible inmediatamente para todos, lo que garantiza que haya una total integración en el diseño y que no se produzcan inconsistencias (ibíd.).

1.3.8. Consideraciones del análisis de los sistemas existentes

A pesar de que Rhinoceros 3D es una buena herramienta a tener en cuenta a la hora del diseño naval no se puede utilizar debido a que no se ajusta a las condiciones del diseño naval cubano. Además Cuba no tiene permitido su acceso por ser un software considerado de alta tecnología. Sin embargo, su estudio permitió conocer de funcionalidades como: modelar el casco de la embarcación a partir del plano de forma. Este tipo de funcionalidades posibilita modelar cualquier casco de buque, ya que todos los barcos poseen su plano de forma por lo que se decide implementar esta funcionalidad.

MaxSurf no cuenta con los estándares para el diseño naval cubano. Además, los ingenieros navales no pueden emplear el sistema para la comercialización de diseños propios, por las restricciones del bloqueo.

El Delftship no se utiliza en Cuba a pesar de que tiene una versión libre, a causa de las restricciones del bloqueo. La funcionalidad más importante de este sistema es modelar a partir de imágenes el casco del buque, la cual fue tenida en cuenta a la hora de desarrollo del módulo.

FreeShip es un sistema de software libre. La compañía que lo desarrolló transformó este software en el actual Delftship, quitándole su soporte, por lo que su utilización en la industria naval es discreta. Está escrito en lenguaje Pascal lo que dificulta su utilización a la hora de reutilizar su código. La funcionalidad de tener varios modelos de ejemplo y como punto de partida para el diseño, fue implementada en el módulo de esta tesis.

El módulo Ship de la plataforma FreeCAD solo permite construir una nueva embarcación a partir de modelos ya definidos. Luego de asignarle los parámetros dimensionales se realizan los cálculos hidrostáticos para mostrarlos en forma tabular o gráfica los que reciben la denominación de tablas hidrostáticas o curvas hidrostáticas respectivamente y de la estabilidad de la embarcación. También permite observar como es el plano de forma del casco del modelo. Estos cálculos consumen recursos de la máquina ya que tienen implementados métodos numéricos de integración que se llevan a cabo en numerosas iteraciones.

El Defcar es utilizado en Cuba por el CDC, pero posee el mismo problema que los softwares anteriores. Este no contiene los estándares de diseño naval cubano, además posee problemas técnicos que afectan a la producción. Este software aportó a la investigación, la información sobre el uso real en la industria cubana de softwares para el diseño naval.

FORAN es muy utilizado en la actualidad por sus funcionalidades. Cuba no tiene acceso a este software al igual que en los anteriores por las políticas del bloqueo. Aunque se tuviera acceso la licencia de uso anual de este software es muy alta, lo que afectaría a la economía cubana. Además, no contiene los estándares de diseño naval cubano.

Por tanto, la utilización de un software propio en la industria cubana, posibilitaría la eliminación del gasto en licencias para el uso de los sistemas comerciales, tendría los estándares de diseño naval cubano, que estas herramientas no poseen y permitiría a los ingenieros navales cubanos a comercializar sus diseños a nivel mundial.

1.4. Cálculos Hidrostáticos

Los cálculos hidrostáticos en ingeniería naval tienen como salida las curvas hidrostáticas, que se emplean en la explotación del buque para conocer los parámetros hidrostáticos en función de la posición del buque en una condición determinada. Estas curvas se conforman por el volumen sumergido o desplazamiento volumétrico, la posición del centro de carena y el centro de gravedad, el área de la superficie de flotación, el coeficiente de bloque, el área de la cuaderna maestra y otros parámetros que por la exigencia de los cálculos pueden representarse o no en función del calado del buque (Lewis, 1988).

Existen cinco métodos de cálculos hidrostáticos:

- El Método Empírico
- El Método Estadístico
- El Método Numérico
- El Método del Canal de Experiencia
- El Método de Prueba a Escala Real

En este trabajo se realizan los cálculos hidrostáticos utilizando el Método Empírico.

1.4.1. Método Empírico

Este método de cálculo se basa en el empleo de fórmulas obtenidas a través de la experiencia en el mundo naval. A continuación, se entra en más detalle sobre las fórmulas y relaciones empíricas que se usan para los cálculos hidrostáticos.

Los valores a calcular son los siguientes: Coeficiente de Bloque (CB), Volumen sumergido (V), Área de Cuaderna Maestra (A_{wp}), Coeficiente Prismático (CP), Coeficiente de Flotación (CWP), Centro de Carena (KB), Centro de Gravedad (KG).

Coeficiente de Bloque

El coeficiente de bloque se calcula mediante el método de Katsoulis (Castro, Azpíroz y Fernández, 1997):

$$CB = 0.817 \cdot f \cdot Lpp^{0.42} \cdot B^{-0.3072} \cdot T^{0.1721} \cdot v^{-0.6135}$$

Donde f es un factor de corrección que está en dependencia del tipo de embarcación a calcular ver tabla 1.1, se emplean también los valores de la Eslora entre perpendiculares (Lpp), Manga (B), Velocidad máxima (v), Calado (T).

Tabla 1.1. Valores de f

Tipos de Buques	f
Petrolero	0.99
Carga General	0.99
Bulkcarrier	1.03
LNG	1.04
LPG	1.04
Quimiquero	1.04

Volumen sumergido

El volumen sumergido V está dado por la relación entre la Eslora (L), la B , el T y el CB (Watson, 1998) y se define:

$$V = CB \cdot L \cdot B \cdot T$$

Área de Cuaderna Maestra

Este parámetro se calcula mediante el despeje de la siguiente relación entre B , Coeficiente del Área de la Cuaderna Maestra (CM) y L (Castro, Azpíroz y Fernández, 1997):

$$A_{wp} = CM \cdot B \cdot L$$

Donde el CM se calcula utilizando la fórmula del HSVA (ibíd.):

$$CM = \frac{1}{1 + (1 - CB)^{3.5}}$$

Coeficiente Prismático

Se calcula utilizando la ecuación de L. Troost (ibíd.) para estimar CP :

$$CP = 1.20 - 2.12 \cdot FN$$

Y el número de Froude (FN) (Baldomir, 2014) se calcula:

$$FN = \frac{v}{q \cdot L}$$

Se emplean los valores de coeficiente empírico q tiene un valor de 9,81, v y L.

Coefficiente de Flotación

Este coeficiente se calcula empleando la fórmula de J. Torroja (Castro, Azpíroz y Fernández, 1997):

$$CWP = A + B \cdot CB$$

Donde A y B están en función del grado U/V de las secciones transversales, grado que se representa por un parámetro G que vale 0 para formas acusadas en U, y 1 para formas acusadas en V (ibíd.).

$$A = 0,248 + 0,049G$$

$$B = 0,778 + 0,035G$$

Centro de Carena

El KB se calcula utilizando la fórmula de Morrish empleando los valores de T, V y A_{wp} (Lewis, 1988):

$$KB = \frac{1}{3} * \left(\frac{5T}{2} - \frac{V}{A_{wp}} \right)$$

Centro de Gravedad

KG se calcula mediante la siguiente ecuación (Schneekluth y Bertram, 1998):

$$KG = C_{KG} \cdot D$$

Tabla 1.2. Valores de C_{KG}

Tipos de Buques	C_{KG}
Barco de Pasajeros	[0.67 – 0.72]
Barco Carguero Grande	[0.58 – 0.64]
Barco Carguero Pequeño	[0.60 – 0.80]
Bulkcarrier	[0.55 – 0.58]
Tanquero	[0.52 – 0.54]

Donde D es el Puntal del barco. C_{KG} es un coeficiente empírico cuyo valor se puede apreciar en la tabla 1.2 y está en dependencia del tipo de embarcación.

1.5. Metodología para el desarrollo

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado.

1.5.1. AUP-UCI

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decidió hacer una variación de la metodología Agile Unified Process (AUP, por sus siglas en inglés), de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la Universidad de las Ciencias Informáticas (UCI) (Sánchez, 2015).

Descripción de las Fases

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre.

Descripción de las disciplinas

AUP propone 7 disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos, Análisis y Diseño en AUP están unidos en la disciplina Modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de Prueba se desagrega en tres disciplinas: Pruebas Internas, de Liberación y Aceptación. Las restantes tres disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI- DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto) (Sánchez, 2015).

Descripción de los roles

AUP propone 9 roles (Administrador de proyecto, Ingeniero de procesos, Desarrollador, Administrador de BD, Modelador ágil, Administrador de la configuración, Stakeholder, Administrador de pruebas, Probador), se decide para el ciclo de vida de los proyectos de la UCI tener 11 roles, manteniendo algunos de los propuestos por AUP y unificando o agregando otros.

Esta versión de AUP define cuatro escenarios en los que se puede ubicar el desarrollo de una aplicación de acuerdo a sus características, los cuales son (ibíd.):

- Escenario 1: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que puedan modelar una serie de interacciones entre los trabajadores del negocio/actores del sistema (usuario), similar a una llamada y respuesta.
- Escenario 2: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio.
- Escenario 3: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio con procesos muy complejos, independientes de las personas que los manejan y ejecutan, proporcionando objetividad, solidez, y su continuidad.
- Escenario 4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos.

Consideraciones

Siguiendo la política de desarrollo de software de la institución, se define como metodología a emplear la AUP-UCI en el escenario número 4, debido a la necesidad de una metodología que responda con facilidad a los cambios continuos, por estar el cliente siempre acompañando el desarrollo y por estar bien definido el negocio.

1.6. Herramientas y Tecnologías para el desarrollo

El desarrollo continuo de las Tecnologías de la Información y las Comunicaciones (TIC) propicia el perfeccionamiento de las herramientas informáticas, por lo que se hace necesario realizar una investigación acerca de las posibles herramientas a utilizar para el desarrollo del módulo. Seguidamente se describen las tecnologías y herramientas empleadas en la presente investigación, realizando un análisis de las principales características de cada una de ellas.

1.6.1. Visual Paradigm

Es una herramienta Ingeniería de Software Asistida por Computación (CASE, por sus siglas en inglés). La misma propicia un conjunto de funcionalidades que apoyan al desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Pressman, 2003).

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition (Community Edition, ya que existe la Enterprise, Professional, etc.). Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos.

Se caracteriza por (ibíd.):

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.

- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, con diferentes especificaciones.
- Generación de código para Java y exportación como HTML.
- Soporte de UML versión 2.1.
- Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento.
- Ingeniería inversa - Código a modelo, código a diagrama.
- Ingeniería inversa Java, C++, Esquemas XML, XML, NET exe/dll, CORBA IDL.
- Generación de código - Modelo a código, diagrama a código.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Generación de código y despliegue de EJB - Generación de beans para el desarrollo y despliegue de aplicaciones.
- Diagramas de flujo de datos.
- Soporte ORM - Generación de objetos Java desde bases de datos.
- Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.

- Generador de informes.
- Distribución automática de diagramas - Reorganización de las figuras y conectores de los diagramas UML.

1.6.2. Open Cascade Technology

La Tecnología Open Cascade Technology (OCCT, por sus siglas en inglés), es una plataforma de desarrollo de software que proporciona servicios para superficies 3D y modelado de sólidos, el intercambio de datos CAD, y la visualización. La mayor parte de la funcionalidad OCCT se encuentra disponible en forma de bibliotecas de C ++. OCCT puede ser aplicado en el desarrollo de software cuyo objetivo sea el modelado 3D CAD, fabricación / medición CAM o simulación numérica Ingeniería Asistida por Computadora (CAE, por sus siglas en inglés). Es una tecnología de software libre; se puede distribuir y/o modificar bajo los términos de la licencia Pública General de GNU (LGPL) versión 2.1, con excepción adicional (Cascade, 2018).

Alternativamente, Open Cascade puede ser utilizada bajo los términos de licencia comercial o acuerdo contractual. Está diseñada para ser altamente portátil y es conocida por trabajar en una amplia gama de plataformas (UNIX, Linux, Windows, Mac OS X, Android). La versión (7.0.0. beta) está certificada oficialmente en las plataformas Windows (IA-32 y x86-64), Linux (x86-64), MAC OS X (x86-64) y Android (4.0.4 ARMv7). Opencascade Community Edition (OCE, por sus siglas en inglés) es una versión de OCCT en la que la comunidad del software libre aporta sus experiencias y recomendaciones de optimización a las versiones liberadas mediante foros o la página oficial de desarrollo de esta tecnología (ibíd.).

Se decide el empleo en la presente investigación de OCE por todas las características antes mencionadas de la tecnología de Open Cascade, por ser de software libre y poseer un acelerado desarrollo por parte de la comunidad.

1.6.3. Lenguaje de programación

C++ es un lenguaje imperativo orientado a objetos derivado del C. En realidad, un superconjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que, como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción (Stroustrup, 2013). Estrictamente hablando, C no es un subconjunto de C++; de hecho, es posible escribir código C que es ilegal en C++. Pero a efectos prácticos, dado el esfuerzo de compatibilidad desplegado en su diseño, puede considerarse que C++ es una extensión del C clásico. La definición "oficial" del lenguaje nos dice que C++ es un lenguaje de propósito general basado en el C, al que se han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios

de nombres, funciones inline, sobrecarga de operadores, referencias, operadores para manejo de memoria persistente, y algunas utilidades adicionales de librería (Stroustrup, 2013).

Tanto C como C++ son lenguajes de programación de propósito general. Todo puede programarse con ellos, desde sistemas operativos y compiladores hasta aplicaciones de bases de datos y procesadores de texto, pasando por juegos, aplicaciones a medida, etc (ibíd.).

Se elige este lenguaje de programación para el desarrollo de la propuesta de solución a causa de todas las características antes mencionadas y por su uso en la tecnología de Open Cascade. También es el que se usa actualmente en el proyecto al que está integrado la investigación.

1.6.4. Framework de desarrollo Qt

Es un marco de desarrollo de aplicaciones multiplataforma basado en C++, dentro de las que se encuentran: Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS y otras. Está disponible bajo varias licencias: licencia comercial y para software libre con varias versiones de la GPL y la LGPL. Este es mucho más que un conjunto de herramientas de Interfaces Gráficas de Usuario (GUI, por sus siglas en inglés). Proporciona módulos para el desarrollo multiplataforma en las áreas de redes, bases de datos, OpenGL 11, tecnologías web, sensores, protocolos de comunicación (Bluetooth, puertos serie), XML 12 y procesamiento JSON 13, impresión, la generación de PDF, y mucho más (Creator, 2018). El Entorno Integrado de Desarrollo recomendado para el empleo de este framework es el Qt Creator.

Se decide el uso de Qt en la presente investigación por ser un framework para el desarrollo de aplicaciones basado en C++ y por poseer una gran cantidad de módulos para disímiles tareas.

1.6.5. Sistema de control de versiones

Git es un sistema de control de versiones distribuido, libre y de código abierto, diseñado para manejar proyectos pequeños o muy grandes con rapidez y eficiencia (Torvalds y Hamano, 2018).

La característica Git que realmente lo separa de casi todos los otros SCM es su modelo de ramificación. Permite tener múltiples ramas locales que pueden ser totalmente independientes entre sí. La creación, la fusión y la supresión de esas líneas de desarrollo toma segundos (ibíd.).

Esto significa que se pueden ejecutar acciones como las indicadas a continuación (ibíd.):

- Conmutación de contexto sin fricción. Crear una rama para probar una idea. Comenzar varias veces o cambiar a la rama de donde se ramificó y aplicar un parche.
- Reglas Basadas en el Rol. Tener una rama que siempre contiene solo lo que va a la producción, otra que se fusiona en el trabajo para la prueba, y varias más pequeñas para el trabajo diario.

- Flujo de trabajo basado en funciones. Crear nuevas ramas para cada nueva función en la que esté trabajando, de modo que pueda cambiar sin problemas entre ellas y, a continuación, suprimir cada una de las ramas cuando se fusione en su línea principal.
- Experimentación desechable. Crear una rama para experimentar, darse cuenta de que no va a funcionar, y solo eliminarlo o abandonar el trabajo.

GitLab es un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Es la primera aplicación individual construida desde cero para todas las etapas del ciclo de vida de Desarrollo y Operaciones (DevOps) para equipos de Producto, Desarrollo, Aseguramiento de la Calidad (QA, por sus siglas en inglés), Seguridad y Operaciones para que trabajen simultáneamente en el mismo proyecto. También, permite a los equipos colaborar y trabajar desde una única aplicación, en lugar de administrar múltiples herramientas. GitLab proporciona a los equipos un único almacén de datos, una interfaz de usuario y un modelo de permiso en todo el ciclo de vida de DevOps, lo que permite a los equipos colaborar, reduciendo significativamente el tiempo de ciclo y centrándose exclusivamente en crear software de gran calidad rápidamente (GitLab, 2018).

Durante el desarrollo de la propuesta de solución debido a las características antes mencionadas y a las indicaciones del proyecto al que está integrado la presente investigación, se decidió escoger como sistema de control de versiones a GitLab.

1.7. Conclusiones del capítulo

- Los software disponibles de ingeniería naval no se ajustan a las características del diseño naval en Cuba.
- Como las curvas características del casco recogidas en el plano de forma se pueden modelar mediante NURBS, se puede desarrollar un algoritmo utilizando la biblioteca Open Cascade que interpole estas curvas generando la superficie del casco.

En este capítulo se plantea la propuesta de solución que incluye el mapa conceptual, la descripción de las funcionalidades, los requisitos funcionales y no funcionales, las historias de usuario y aspectos de diseño (el estilo y patrón arquitectónico, los patrones de diseño, el modelo de clases).

2.1. Mapa Conceptual

Los mapas conceptuales son uno de los tipos de Recursos Esquemáticos que mayor aplicación tiene en los procesos de creación, representación y adquisición de la Información y el Conocimiento; útil para el desarrollo del aprendizaje de los estudiantes. Desarrollado por Novak en la Universidad de Cornell en Estados Unidos (Ecured, 2018).

A continuación, en la figura 2.1 se presenta el mapa conceptual definido en la presente investigación:

El casco de un buque tiene un plano de forma, el cual contiene los planos: longitudinal, transversales y

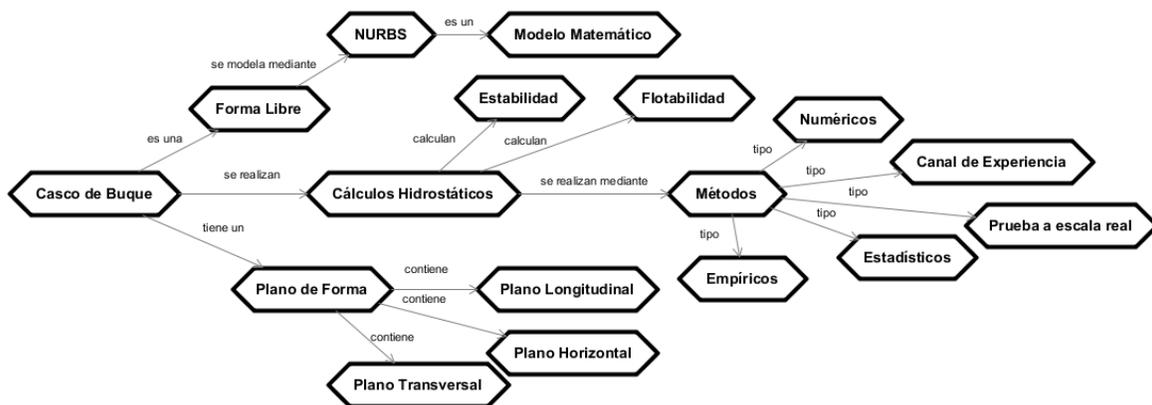


Figura 2.1. Mapa Conceptual

horizontal que contienen las líneas características de la configuración del casco del buque. El casco de un

buque es una forma libre, o sea no se pueden describir utilizando funciones matemáticas, se modela mediante superficies NURBS es un modelo matemático de curvas interpoladas. Al casco de un buque se le realizan cálculos hidrostáticos mediante métodos que pueden ser: numéricos, empíricos, estadísticos, de prueba a escala real o en canal de experiencia. Se calcula estabilidad y flotabilidad.

2.2. Descripción de las funcionalidades

Las funcionalidades serán diseñadas para el modelado y cálculo hidrostático de cascos de buques. Estas estarán integradas a la aplicación CAD que está desarrollando el grupo de investigación Soluciones Informáticas para la Ingeniería y la Industria (SIPII) la cual poseerá una sesión *Ship* donde se encontrarán estas. Al activarse este modo se mostrarán una serie de botones que tendrán asociados los algoritmos necesarios para ejecutarlas. Se enuncian a continuación los botones asociados a las funcionalidades:

- El botón *Import* permite importar el modelo de un casco y prepararlo para trabajar sobre él en este módulo. Los formatos soportados en esta funcionalidad son *brep*, *stl*, *step* y *txt* en este último se recogen los conjuntos de puntos que describen las características del casco del buque.
- *Mirror* permite completar el modelo importado si este solo contiene la mitad. También completa simétricamente un conjunto de puntos importados a partir de un archivo *txt*. Luego de ejecutar la funcionalidad no permite mostrar la mitad del modelo nuevamente.
- *Import Plane* permite importar una foto bitmap con los planos de forma. Se utiliza el módulo *Sketcher* para trazar curvas interpoladas del tipo NURBS o B-Spline sobre cada una de las líneas del plano de forma introducido. Esta funcionalidad solo modela el casco a partir de las líneas características de uno de los tres planos de forma. Para que el sistema modele el casco en su totalidad es necesario que recoja los puntos de intercepción entre las líneas características, guardarlos ordenados en una matriz y pasarla a *GeomAPI_PointsToBSplineSurface*. Esta funcionalidad también permite introducir el valor de escala entre las líneas características.
- *OutlineDraw* permite introducir la cantidad de secciones a realizar sobre el modelo, formando las líneas características del casco. Las líneas formadas están en dependencia del plano en el cual se trabaje.
- *Create* permite crear una instancia de los parámetros del casco dígame; eslora, eslora entre perpendiculares, manga, puntal, calado y velocidad. Estos son los parámetros para realizar los cálculos hidrostáticos. No se corresponden con los parámetros a escala del modelo.

- *Hidrostatic Calculation* permite realizar cálculos hidrostáticos mediante el método empírico. En esta funcionalidad se calculan los valores de V , CB , A_{wp} , CP , CWP , KB , KG . Los parámetros calculados son necesarios para calcular posteriormente la estabilidad y flotabilidad del buque.

Mediante estas funcionalidades se logra informatizar algunas tareas de diseño realizadas por los constructores navales como:

- Visualizar y manipular modelos de casco de buque.
- Construir el casco a partir del plano de forma.
- Obtener los planos de forma a partir del casco del buque.

2.3. Requisitos

Se pueden clasificar en funcionales y no funcionales. Existen diferentes métodos para la descripción de los requisitos, uno de ellos es el uso de las Historias de Usuario.

2.3.1. Requisitos Funcionales

A continuación, se enumeran los requisitos funcionales definidos para el módulo:

- RF 1. Importar modelo de casco de buque.
- RF 2. Completar el modelo del casco de buque importado.
- RF 3. Importar bitmap con el plano de forma del casco.
- RF 4. Modelar un casco a partir de una imagen bitmap.
- RF 5. Modelar las líneas de los contornos del casco de buque.
- RF 6. Introducir los parámetros del casco.
- RF 7. Realizar cálculos hidrostáticos.

2.3.2. Requisitos No Funcionales

A continuación, se exponen los requisitos no funcionales del módulo:

- RNF 1. Software: SO.: GNU-Linux, GCC: 4.2.4
- RNF 2: Restricciones de diseño e implementación: Arquitectura en capas, lenguaje de programación C++, biblioteca OpenCascade.

2.4. Historias de usuario

Las historias de usuario son tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento se pueden modificar, reemplazar por otras más específicas o generales o añadirse nuevas. Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarla (Jeffries, Anderson y Hendrickson., 2001).

Tabla 2.1. Historia de Usuario Importar bitmap con el plano de forma del casco

Número: 3	Nombre del requisito: Importar bitmap con el plano de forma del casco		
Programador: Kevin Pérez Mandina	Iteración Asignada: 1ra		
Prioridad: Alta	Tiempo Estimado: 2 semanas		
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas		
Descripción:			
1- Objetivo:			
Permitir importar un fichero bitmap con el plano de forma del casco.			
2- Acciones para lograr el objetivo (precondiciones y datos):			
Para importar un fichero bitmap con el plano de forma del casco se debe tener en cuenta los siguientes aspectos:			
- Un fichero bitmap con el plano de forma del casco.			
- La clase SketchObject.			
- La clase ViewProviderSketch.			
3- Flujo de la acción a realizar:			
- Se selecciona el botón Import Plane y se busca el fichero bitmap y se presiona en aceptar.			
- Se muestra en el área de trabajo la imagen con el plano de forma.			
- Si se selecciona el botón Cancel se cierra la ventana.			
Observaciones:			
Prototipo de interfaz:			

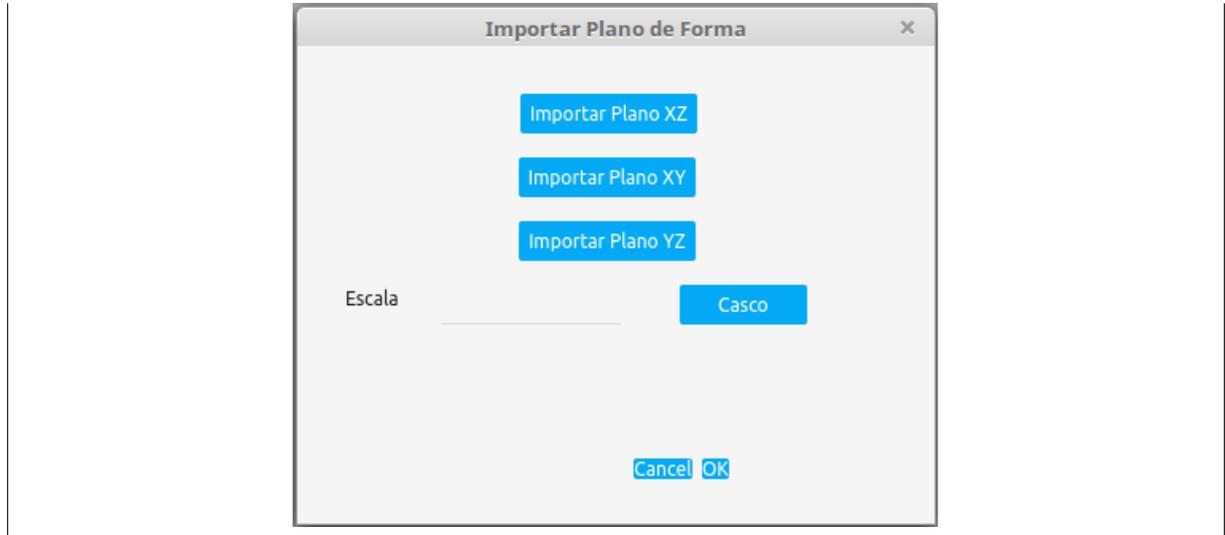


Tabla 2.2. Historia de Usuario Introducir los parámetros del casco

Número: 6	Nombre del requisito: Introducir los parámetros del casco
Programador: Kevin Pérez Mandina	Iteración Asignada: 2da
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas
<p>Descripción:</p> <p>1- Objetivo: Introducir los parámetros del casco.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para introducir los parámetros del casco del buque se debe tener en cuenta los siguientes datos: - El valor de la eslora, la eslora entre perpendiculares, el calado, la manga, el puntal, la velocidad.</p> <p>3- Flujo de la acción a realizar: - El usuario introduce los datos necesarios y selecciona la opción Create. - Si se selecciona el botón Cancel se cierra la ventana.</p>	
Observaciones:	
Prototipo de interfaz:	

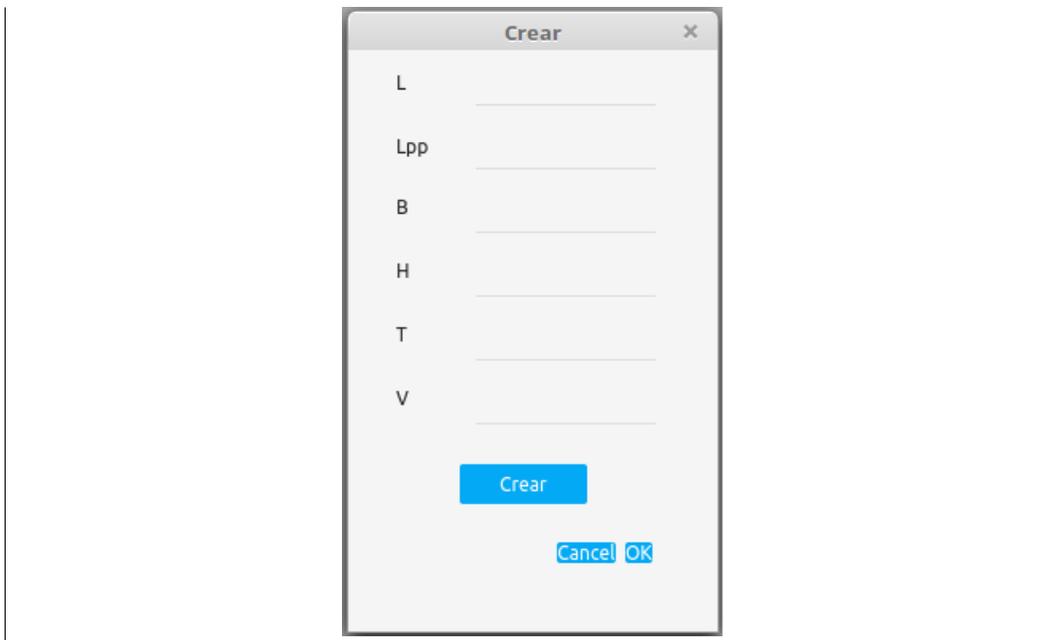
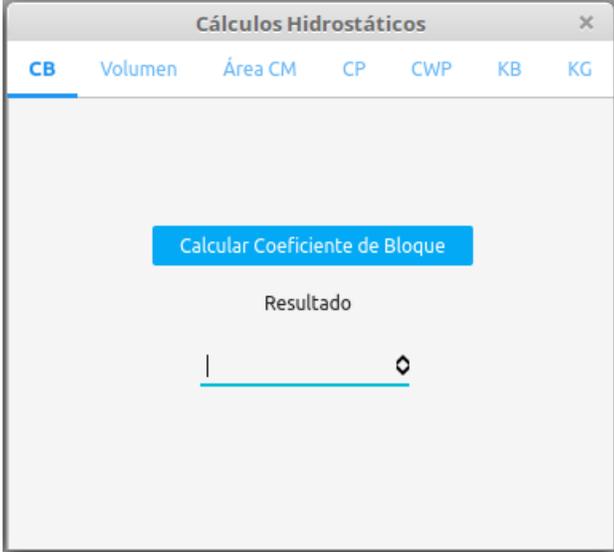


Tabla 2.3. Historia de Usuario Realizar cálculos hidrostáticos

Número: 7	Nombre del requisito: Realizar cálculos hidrostáticos
Programador: Kevin Pérez Mandina	Iteración Asignada: 2da
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas
<p>Descripción:</p> <p>1- Objetivo: Permitir realizar los cálculos hidrostáticos para casco de buque.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para realizar cálculos hidrostáticos se debe tener en cuenta: - El usuario realizó la funcionalidad Create. - Programar en el siguiente orden los cálculos hidrostáticos: 1- Coeficiente de Bloque 2- Volumen sumergido 3- Área de Cuaderna Maestra 4- Coeficiente Prismático 5- Coeficiente de Flotación 6- Centro de Carena 7- Centro de Gravedad</p> <p>3- Flujo de la acción a realizar: - El usuario selecciona las pestañas para realizar los cálculos hidrostáticos en el siguiente orden: - Si se selecciona el botón Cancel se cierra la ventana. 1- Coeficiente de Bloque</p>	

2- Volumen sumergido 3- Área de Cuaderna Maestra 4- Coeficiente Prismático 5- Coeficiente de Flotación 6- Centro de Carena 7- Centro de Gravedad
Observaciones:
<p>Prototipo de interfaz:</p> 

2.5. Diseño

La esencia del diseño del software es la toma de decisiones sobre la organización lógica del software. Algunas veces, se representa esta organización lógica como un modelo en un lenguaje definido de modelado tal como UML y otras veces simplemente se utiliza notaciones informales y esbozos para representar el diseño (Sommerville, 2006). El proceso de diseño tiene asociado la decisión del tipo arquitectura y los patrones de diseño que empleará el sistema, así como la confección de distintos diagramas que favorezcan el trabajo en la fase de implementación.

2.5.1. Estilo y patrón arquitectónico del software

Un estilo arquitectónico es una transformación impuesta al diseño de todo un sistema. El objetivo es establecer una estructura para todos los componentes del sistema. En caso de que una arquitectura existente se vaya a someter a reingeniería, la imposición de un estilo arquitectónico desembocará en cambios fundamentales en la estructura del software, incluida una reasignación de la funcionalidad de los componentes.

En cambio los patrones arquitectónicos aplicados al software, definen un enfoque específico para el manejo de alguna característica de comportamiento del sistema (Pressman, 2005).

Estilo arquitectónico del software

La arquitectura de llamada y retorno es un estilo arquitectónico que permite a un diseñador de software obtener una estructura de programa que resulte relativamente fácil modificar y cambiar de tamaño. En esta categoría hay dos subestilos (ibíd.):

- Arquitectura de programa principal/subprograma. Esta estructura de programa clásica separa la función en una jerarquía de control donde un programa principal invoca a varios componentes de programa, que a su vez pueden invocar a otros componentes. En la figura 2.2 se ilustra una arquitectura de este tipo.
- Arquitectura de llamada de procedimiento remoto. Los componentes de una arquitectura de programa principal/subprograma se distribuyen entre varias computadoras de una red. El módulo pertenece a un sistema que contiene un programa principal donde se invocan componentes para realizar las funcionalidades que necesita el usuario. Por tanto el estilo arquitectónico apropiado para el módulo es llamada y retorno, en la categoría de arquitectura de programa principal/subprograma.

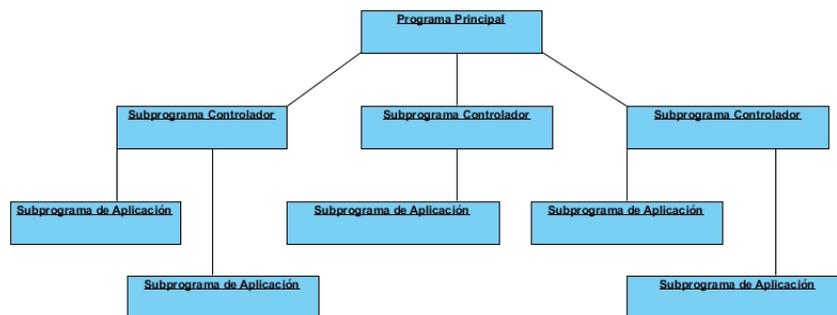


Figura 2.2. Diagrama de Arquitectura de programa principal/subprograma

Patrón arquitectónico del software

El patrón capas ayuda a estructurar aplicaciones que pueden descomponerse en grupos de subtareas en la que cada grupo de subtareas está en un nivel particular de abstracción o sea en una capa diferente. Donde cada capa provee servicios a la capa superior y es servido por la capa inferior. Si las interfaces de las capas

sufren cambios o se añaden nuevas funcionalidades a una capa, solamente se ven afectadas las capas adyacentes. Entre las desventajas de este modelo es que la estructuración de los sistemas puede resultar difícil y el rendimiento se puede ver afectado, pues se puede incurrir en que una funcionalidad debe pasar por muchas capas para alcanzar la capa superior que solicitó su servicio (Sommerville, 2007).

El patrón arquitectónico más apropiado para el desarrollo del módulo es por capas. Este está estructurado en dos capas, una de presentación o vista y otra de lógica del negocio. Permitiendo la comunicación con el núcleo de la aplicación y con otros módulos de la misma. Además, este patrón permite que soporte bien los cambios y el desarrollo incremental.

2.5.2. Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interfaces.

Los patrones General Responsibility Assignment Software Patterns (GRASP, por sus siglas en inglés) definidos en las funcionalidades:

- Experto: mediante su uso, se asignan responsabilidades a la clase que cuenta con la información necesaria (Larman, 2003). Este patrón se evidencia en la clase Ship que se encarga del modelado y la clase ImportPlaneShapeShip de recoger la información del plano de forma.
- Creador: permite crear objetos de una clase determinada (ibíd.). Este patrón se evidencia en las clases de interfaz DlgCreateShip, DlgHidrostaticCalculation, DlgPlaneImport, DlgShapeImport, DlgShipOutlineDraw y DlgMirror ya que tienen la información necesaria para la creación de objetos tipo Ship e ImportPlaneShapeShip.
- Polimorfismo: se emplea para asignar comportamientos distintos según el tipo de clase (ibíd.). Se evidencia en las clases Ship esta hereda de la clase Feature y reimplementan los métodos execute, onChanged, mustExecute y getViewProviderName, la clase ImportPlaneShapeShip hereda de la clase Sketcher y reimplementa el método getBackRoot

Patrón GOF definido en las funcionalidades:

- Observador: es un patrón que define una dependencia del tipo uno a muchos entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. (Microsoft, 2018). Este patrón se evidencia en la función mustExecute.

2.5.3. Diagrama de Clases del Diseño

Un diagrama o modelo de clases en Lenguaje Unificado de Modelado (UML, por sus siglas en inglés) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, métodos, atributos, y las relaciones entre los objetos (herencia, agregación, asociación, entre otras). En la figura 2.3 se muestra el Diagrama de Clases del Diseño empleado en la propuesta de solución.

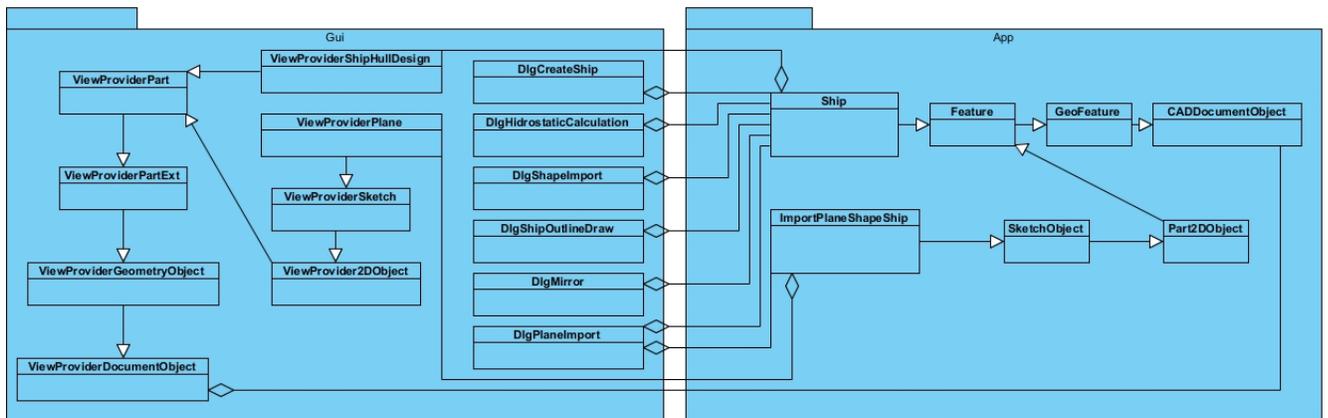


Figura 2.3. Diagrama de Clases del Diseño

2.6. Conclusiones del Capítulo

- Como el módulo pertenece a un sistema que contiene un programa principal donde se invocan componentes para realizar las funcionalidades que necesita el usuario, se identificó como estilo arquitectónico más apropiado llamada y retorno; y como patrón a emplear en capas, específicamente en dos capas.

En el presente capítulo se abordan cada uno los componentes que integran la etapa de implementación y prueba del módulo a desarrollar. Se expone el estándar de codificación, los diseños de los casos de pruebas, así como los resultados obtenidos del proceso de verificación de la calidad.

3.1. Implementación

La fase de implementación del software posee como entradas los artefactos de diseño, como: diagramas de clases, especificación de arquitectura, patrones a emplear en el sistema, entre otros. En la implementación se define el estándar de codificación a emplear y el diagrama de componentes del sistema. También se realizan las implementaciones a las historias de usuarios, entre otras actividades.

3.1.1. Detalles técnicos de la implementación

En la implementación del módulo se utilizaron diferentes Application Programming Interface (API, por sus siglas en inglés) de la biblioteca de modelado Open Cascade como:

- BRepBuilderAPI_Transform, esta API se utilizó para rotar y trasladar modelos en los métodos de construcción de figuras. También se empleó en la funcionalidad *Mirror* para completar las mitades simétricas de los modelos.
- BRepBuilderAPI_MakeFace, con esta API se construyeron superficies simples, como las caras de las figuras geométricas en el espacio.

- `GeomAPI_PointsToBSplineSurface`, esta se empleó para construir el prototipo del casco a partir de la interpolación de los puntos contenidos en las curvas B-Spline del plano de forma.
- `BRepAlgoAPI_Section`, utilizando esta API se interceptan planos con el modelo del casco, obteniendo secciones de este, con lo que se obtiene las curvas características del casco.

En la tabla 3.1 se observan datos técnicos sobre el código del módulo como son: el lenguaje de los tipos de archivos que tiene el módulo, la cantidad de archivos, la cantidad de líneas de código, comentarios y líneas en blanco.

Tabla 3.1. Datos Técnicos

Lenguaje	Archivos	Líneas en blanco	Comentarios	Líneas de código
C++	18	888	265	2092
C/C++ Header	19	211	35	436
Total:	37	1099	300	2528

3.1.2. Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Aunque la forma de escribir código es propia de cada programador. La forma usada va a depender de la facilidad para entender el código y retomar ciertas partes realizadas por otros integrantes, así como la depuración de las mismas (ASPL, 2018).

3.1.3. Diagrama de Componentes

Un diagrama de componentes muestra dependencias entre los componentes, que no son más que una unidad física de implementación con interfaces bien definidas pensada para ser utilizada como parte reemplazable de un sistema. Puede mostrar un sistema configurado, con la selección de componentes usados para construirlo o un conjunto de componentes disponibles (una biblioteca de componentes) con sus dependencias (Jacobson, Rumbaugh y Booch, 2000). En la figura 3.4 se observa el Diagrama de Componentes del sistema.

3.1.4. Resultados de la implementación

En este apartado se muestran imágenes sobre el resultado de la implementación de las funcionalidades. En las figuras 3.5, 3.6, 3.7 se muestra el resultado de la funcionalidad *OutlineDraw*.

Descripción	Ejemplo
Definición de Objetos, Clases, funciones y atributos	
Todos los nombres de las clases implementadas comenzarán con letra mayúscula. En caso de poseer un nombre compuesto se escribirán de acuerdo a la normativa CamelCase-UpperCamelCase.	<pre>class Foo{ cuerpo de la clase } class FooFirst{ cuerpo de la clase }</pre>
Siempre se declara para todas las clases implementadas su respectivo destructor de clase.	<pre>virtual ~Foo()</pre>
La declaración de funciones o métodos siempre comenzarán en letra inicial minúscula. En caso de ser un nombre compuesto se regirá por la normativa CamelCase-lowerCamelCase.	<pre><Tipo dato retorno> funcion() <Tipo dato retorno> funcionCompuesta() <Tipo dato retorno>funcionDobleCompuesta()</pre>

Figura 3.1. Estándar de codificación

Los atributos siempre estarán escritos con letra minúscula. En caso de ser un nombre compuesto se regirá por la normativa CamelCase-lowerCamelCase.	<pre><Tipo dato> atributo; <Tipo dato> atributoNombreCompuesto;</pre>
Definición de parámetros dentro de las funciones y constructores de clases	
Los nombres de los identificadores de los parámetros en las funciones deben estar escritos con minúscula separados a un espacio cada uno y en caso de ser compuesto utilizar normativa CamelCase-lowerCamelCase.	<pre><Tipo dato retorno> funcion(<tipo><id1>, <tipo><id2>, <tipo><idN>)</pre>
Los identificadores de los parámetros dentro de los constructores de las clases deben estar escritos con minúscula separados a un espacio cada uno y en caso de ser compuesto utilizar normativa CamelCase-lowerCamelCase.	<pre>Clase(<tipo><id1>, <tipo><id2>, <tipo><idN>)</pre>
Definición de expresiones	
Para una mejor comprensión en la lectura y legibilidad del código los operadores binarios exceptuando los punteros, función de llamado a miembros, escritura de un arreglo y paréntesis de una función se escribirán con un espacio entre ellos	<pre>x + y; x == y; idFuncion.miembro(); idFuncion->miembro(); array[];</pre>

Figura 3.2. Estándar de codificación

En la figura 3.8 se muestra el resultado de la funcionalidad *Import Plane* .

En la figura 3.9 se muestra el resultado de la funcionalidad *Mirror* .

En la figura 3.10 se muestra el resultado de la funcionalidades *Create* y *Hidrostatic Calculation* .

Definición de estructuras de control y bucles	
Las estructuras de control y los bucles estarán definidos de igual manera en ambos casos siguiendo el estándar determinado por el <i>framework</i> de Qt.	Para las estructuras if, else, if else : <pre><estructura control>(condición){ tarea a ejecutar }</pre> Para los bucles while, for, do while y otros: <pre><bucle>(condiciones){ tarea a ejecutar }</pre>
Comentarios en el código según el estándar de C++.	
Comentarios pequeños.	<code>/* comentario sencillo */</code>
Otros comentarios.	<code>/* *Comentario */</code>
Comentario de versión, descripción de clase y otras características de la clase o paquete.	<code>/* ***** *Comentario amplio * ***** */</code>

Figura 3.3. Estándar de codificación

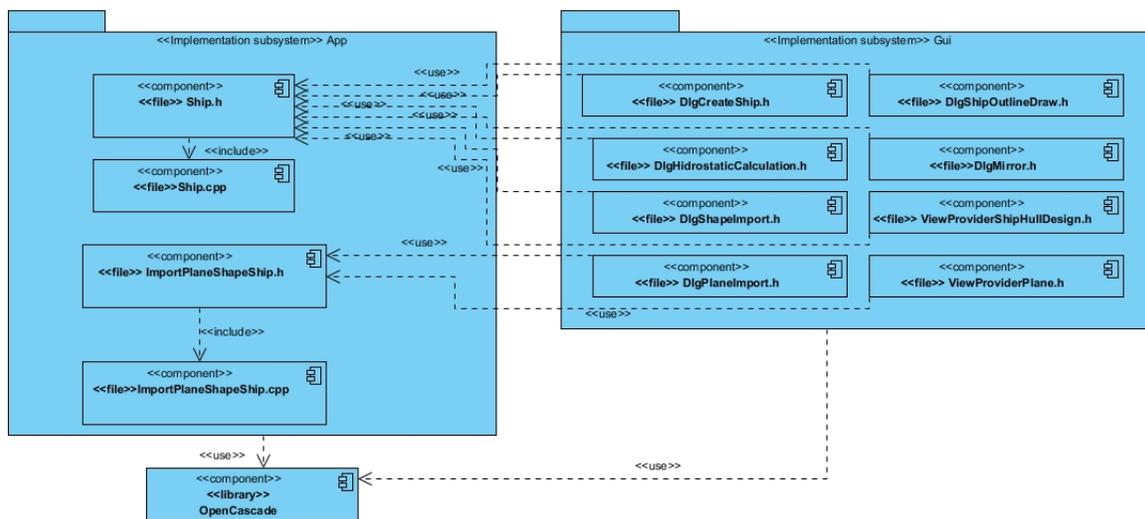


Figura 3.4. Diagrama de Componentes

3.2. Pruebas

El proceso de pruebas de software tiene los siguientes objetivos (Sommerville, 2006):

- Demostrar al desarrollador y al cliente que el software satisface sus requisitos. Esto significa que debería haber al menos una prueba para cada requerimiento o característica que se incorporará a la entrega del producto.

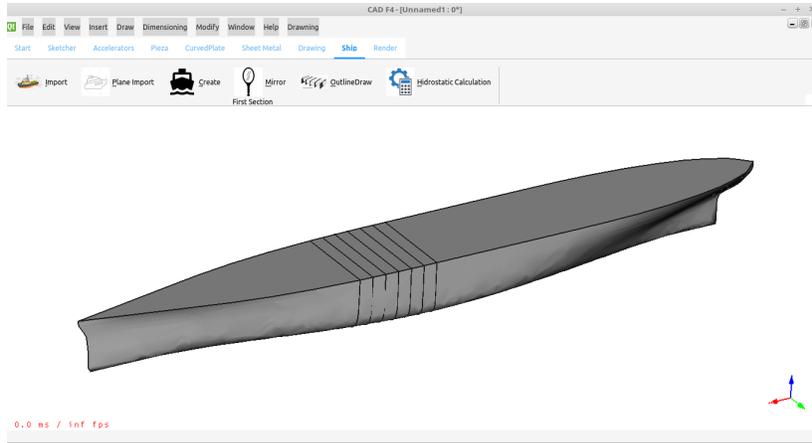


Figura 3.5. Líneas transversales

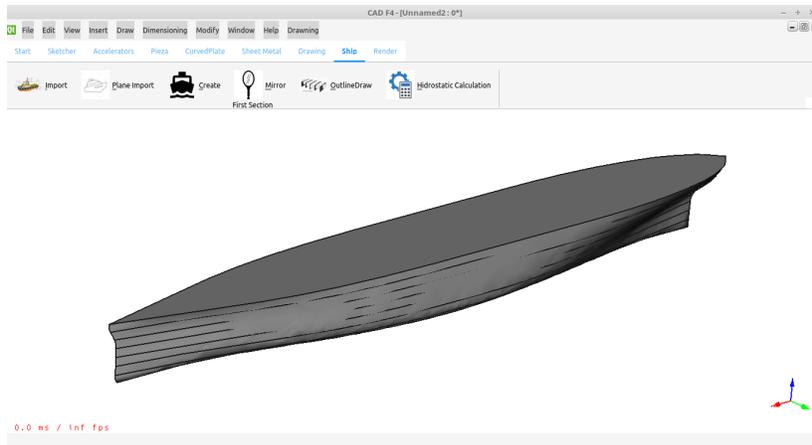


Figura 3.6. Líneas horizontales

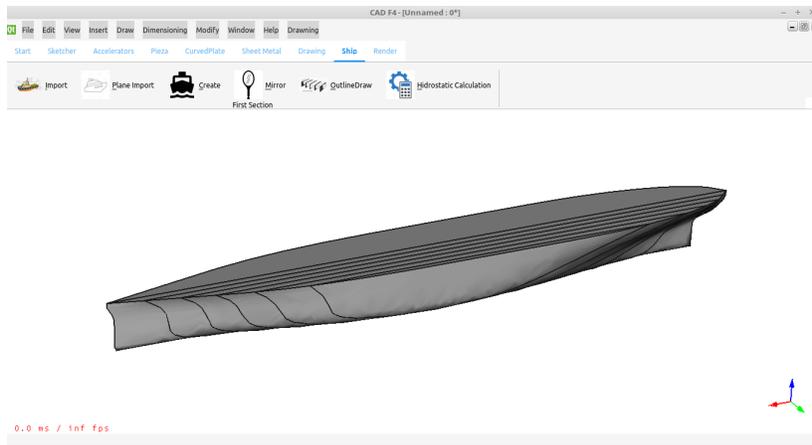


Figura 3.7. Líneas longitudinales

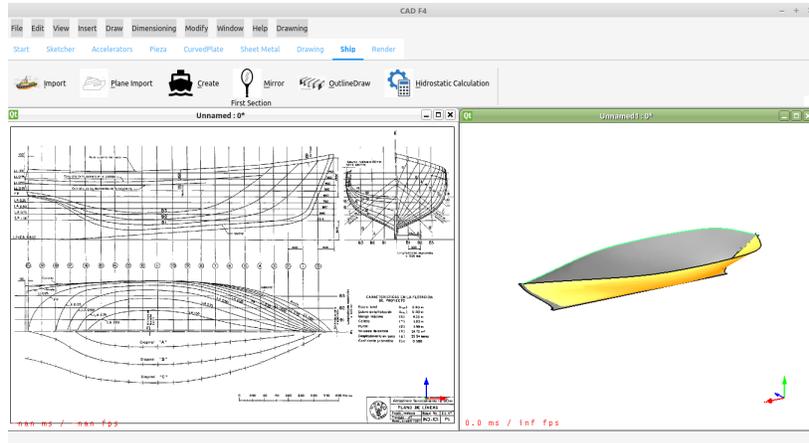


Figura 3.8. Plano de forma y casco modelado

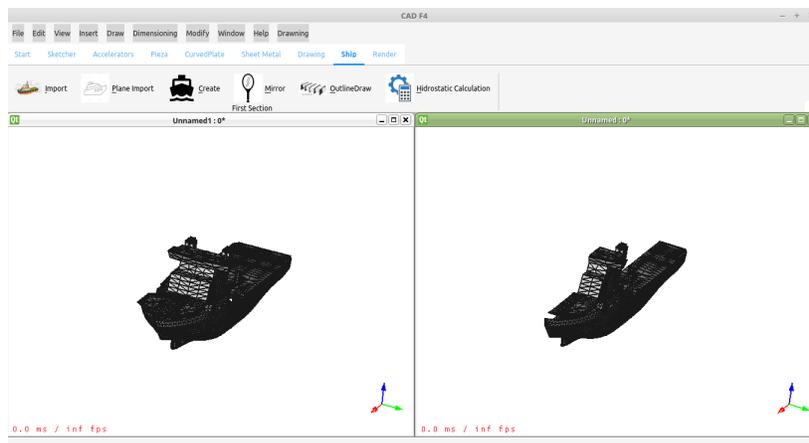


Figura 3.9. Modelo completo y mitad del modelo

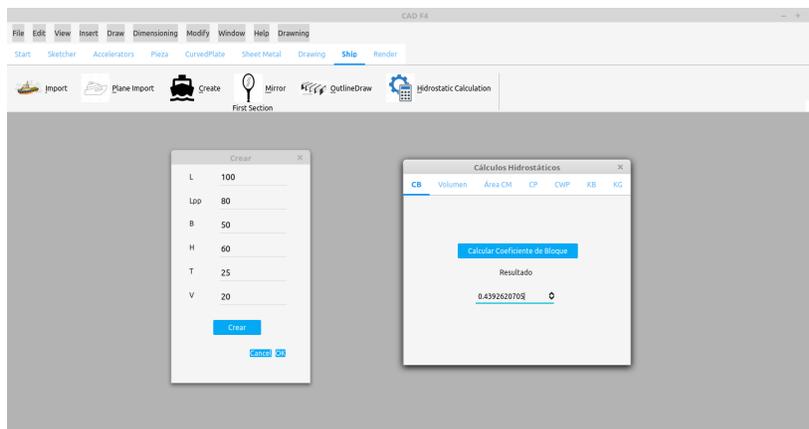


Figura 3.10. Datos y cálculos hidrostáticos

- Descubrir defectos en el software en el que el comportamiento de este es incorrecto, no deseable o no cumple su especificación. La prueba de defectos está relacionada con la eliminación de todos los tipos de comportamientos del sistema no deseables, tales como caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrupción de datos.

3.2.1. Niveles de prueba

Para aplicarle pruebas a un sistema se deben tener en cuenta una serie de objetivos en diferentes escenarios y niveles de trabajo, debido a que las pruebas son agrupadas por niveles que se encuentran en distintas etapas del proceso de desarrollo. Los niveles de prueba son (Tenorio, 2010):

- Prueba Unitaria o de Unidad: se centra en el proceso de verificación de la menor unidad del diseño del software: el componente de software o módulo. Se emplea para detectar errores debidos a cálculos incorrectos, comparaciones incorrectas o flujos de control inapropiados. Las pruebas del camino básico y de bucles son técnicas muy efectivas para descubrir una gran cantidad de errores en los caminos.
- Prueba de Integración: es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados mediante la prueba unitaria y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.
- Pruebas de Sistema: está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. Algunas de estas son: pruebas de recuperación, seguridad, resistencia, entre otras.
- Pruebas de Aceptación: es la realización de una serie de pruebas de caja negra que demuestran la conformidad con los requisitos. Un plan de prueba traza la clase de pruebas que se han de llevar a cabo, y un procedimiento de prueba define los casos de prueba específicos en un intento por descubrir errores de acuerdo con los requisitos.

3.2.2. Métodos de prueba

El principal objetivo del diseño de casos de prueba es obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Para llevar a cabo este objetivo, se emplearán los

dos métodos de prueba (Tenorio, 2010):

- Prueba de Caja Blanca: se centran en la estructura de control del programa. Se obtienen casos de prueba que aseguren que durante la prueba se han ejecutado, por lo menos una vez, todas las sentencias del programa y que se ejercitan todas las condiciones lógicas.
- Prueba de Caja Negra: son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno de un programa, solo se fijan en las funciones que realiza el software. Las técnicas de prueba de caja negra se centran en el ámbito de información de un programa, de forma que se proporcione una cobertura completa de prueba.

3.2.3. Diseño de Casos de Prueba

Los casos de prueba incluyen todas las funciones que el programa es capaz de realizar. Deben tener en cuenta el uso de todo tipo de datos de entrada/salida, cada comportamiento esperado, todos los elementos de diseño, y cada clase de defecto. Todos los requisitos deberán ser cubiertos por los casos de prueba, de manera tal que cubra el software a fondo. Se puede decir que son la descripción de las actividades que se van a ejecutar con el fin de validar la aplicación (Duarte, 2015).

Algunas características son (ibíd.):

- Se usan las mismas técnicas, pero con otro objetivo.
- No hay programas de prueba, sino sólo el código final de la aplicación.
- Se prueba el programa completo.
- Uno o varios casos de prueba por cada requisito.
- Se prueba también rendimiento, capacidad, etc. (y no sólo resultados correctos).
- Pruebas alfa(desarrolladores) y beta (usuarios).

Los Casos de Pruebas han sido realizados sobre la base de las Historias de Usuarios y tienen como objetivo fundamental encontrar la mayor cantidad posible de deficiencias existentes en las funcionalidades implementadas ver tabla 3.2.

Tabla 3.2. Diseño de Casos de Prueba RF 4

Descripción general			
Permitir modelar un casco a partir de una imagen bitmap			
SC 1 Modelar casco			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Pestaña Sketcher.	Selecciona la pestaña Sketcher la cual mostrará una sección para dibujar sobre la imagen del plano de forma.	Brinda la posibilidad de dibujar curvas y líneas sobre la imagen.	ShipHullDesgin /ImportPlaneShapeShip /Sketcher
EC 1.2 Opción Casco	Entra el valor de la escala y selecciona la opción Casco.	Crea la superficie con los datos de entrada.	ShipHullDesgin/Ship /Casco
EC 1.3 Opción de Ok	Selecciona la opción de Ok	Crea la Superficie con los datos de entrada y cierra la ventana.	ShipHullDesgin/Ship /Apply and Close
EC 1.4 Opción de Close	Selecciona la opción de Close	Cierra la ventana.	ShipHullDesgin/Ship /Close

3.2.4. Pruebas Unitarias

Para la realización de las pruebas unitarias se empleó Qt Test, el cual es un framework para pruebas de este tipo a aplicaciones y librerías. Qt Test provee todas las funcionalidades comunes de los framework de pruebas unitarias, así como extensiones para probar interfaces gráficas de usuario (Test, 2018). Ver figura 3.11 donde se muestra el resultado de las pruebas unitarias realizadas durante el proceso de implementación, en la que se evidencia la aceptación de todas las verificaciones.

Los valores utilizados en la realización de las pruebas unitarias fueron obtenidos mediante el cálculo manual realizado por un especialista naval. Este utilizó las ecuaciones del método empírico para los cálculos hidrostáticos descritos anteriormente. También, se estableció un margen de error, porque en la comparación de los resultados de las pruebas unitarias los valores flotantes obtenidos pueden ser diferentes para cada máquina que realice el cálculo. El valor máximo de error es de 0.0001m, este es mucho menor que el error introducido por los instrumentos de medición utilizados en la etapa de construcción naval.

Código fuente 3.1. Código para ejecutar *Qt Test*

```
#include <QtTest>
#include <typeinfo>
#include <bits/stdc++.h>
#include "HidrostaticCalculation/ship.h";
using namespace std;

class TestUnitTest : public QObject
```

```

{
    Q_OBJECT

private Q_SLOTS:

    void testCreateShip ();
    void testCalculationCB ();
    void testCalculationVolumen ();
    void testCalculationAwp ();
    void testCalculationCP ();
    void testCalculationCF ();
    void testCalculationKB ();
    void testCalculationKG ();

private:
    Ship * ship = new Ship ();
    double epsilon = 0.0001;
};
void TestUnitTest::testCreateShip () {
// eslora - esloraperpendicular - manga - puntal - calado - velocidad
ship->CreateShip (100,80,50,60,25,20);
}
void TestUnitTest::testCalculationCB () {
// 0.4392620705 = 0.817 * f * Lpp042 * B-0.3072 * T0.1721 * V-0.6135
// qDebug ()<<ship->CalculationCB ()<<"CalculationCB () ";
QVERIFY2 (fabs (ship->CalculationCB () - 0.43926) < epsilon, "No coincide
CalculationCB ()");
}
void TestUnitTest::testCalculationVolumen () {
// 54 907.758806766 = CB * L * B * T
// qDebug ()<<ship->CalculationVolumen ()<<"CalculationVolumen () ";
QVERIFY2 (fabs (ship->CalculationVolumen () - 54907.75781) < epsilon, "No
coincide CalculationVolumen ()");
}
void TestUnitTest::testCalculationAwp () {
// 4416.8593248921 = CM * B * L 0.883371865 = 1/1+(1 - CB)3.5
// qDebug ()<<ship->CalculationAwp ()<<"CalculationAwp () ";
QVERIFY2 (fabs (ship->CalculationAwp () - 4416.85920) < epsilon, "No
coincide CalculationAwp ()");
}
void TestUnitTest::testCalculationCP () {
// -4.7962655045 = 1.20 - 2.12 * Fn
// 2.8284271247 = V/raiz q*L
// qDebug ()<<ship->CalculationCP ()<<"CalculationCP () ";
QVERIFY2 (fabs (ship->CalculationCP () - -4.79627) < epsilon, "No coincide

```

```

    CalculationCP ( " );
}
void TestUnitTest::testCalculationCF () {
// 0.623372 = A + B * CB
// Donde G puede ser 0 si es forma U o 1 si es forma V
// A 0.297 = 0.248 + 0,049 * G
// B 0.743 = 0,778 - 0,035 * G
// qDebug() << "ship -> CalculationCF ( true ) << "CalculationCF ( ) ";
QVERIFY2( fabs ( ship -> CalculationCF ( true ) - 0.62337 ) < epsilon , "No coincide CalculationCF ( ) );
}
void TestUnitTest::testCalculationKB () {
// 16.6895322922 = 1/3 * (5T/2 - V/Awp)
// qDebug() << "ship -> CalculationKB ( ) << "CalculationKB ( ) ";

// printf ( " %.4f \n " , 8.32454354 );
QVERIFY2( fabs ( ship -> CalculationKB ( ) - 16.68953 ) < epsilon , "No coincide CalculationKB ( ) );
}
void TestUnitTest::testCalculationKG () {
// 37.2 = Ckg * D
// Ckg valor promedio 0.62
// qDebug() << "ship -> CalculationKG ( ) << "CalculationKG ( ) ";
QVERIFY2( fabs ( ship -> CalculationKG ( ) - 37.2 ) < epsilon , "No coincide CalculationKG ( ) );
}

QTEST_APPLESS_MAIN ( TestUnitTest )

#include "qtest.moc"

```

1

3.2.5. Pruebas de Aceptación

Una Prueba de Aceptación tiene como propósito demostrar al cliente el cumplimiento de un requisitos del software. Estas se caracterizan por (Letelier, 2007):

- Describir un escenario (secuencias de pasos) de ejecución o uso del sistema desde la perspectiva del cliente.
- Puede estar asociada a requisitos funcionales o no funcionales.

```
***** Start testing of TestUnitTest *****
Config: Using QTest library 5.5.1, Qt 5.5.1 (x86_64-l
PASS  : TestUnitTest::initTestCase()
PASS  : TestUnitTest::testCreateShip()
PASS  : TestUnitTest::testCalculationCB()
PASS  : TestUnitTest::testCalculationVolumen()
PASS  : TestUnitTest::testCalculationAwp()
PASS  : TestUnitTest::testCalculationCP()
PASS  : TestUnitTest::testCalculationCF()
PASS  : TestUnitTest::testCalculationKB()
PASS  : TestUnitTest::testCalculationKG()
PASS  : TestUnitTest::cleanupTestCase()
Totals: 10 passed, 0 failed, 0 skipped, 0 blacklisted
***** Finished testing of TestUnitTest *****
```

Figura 3.11. Pruebas Unitarias

- Un requisito tiene una o más Pruebas de Aceptación asociadas.
- Las Pruebas de Aceptación cubren desde escenarios típicos/frecuentes hasta los más excepcionales.

Las Pruebas de Aceptación se pueden utilizar para (Letelier, 2007):

- Obligar a definir requisitos que sean verificables.
- Valorar adecuadamente el esfuerzo asociado a la incorporación de un requisito.
- Negociar con el cliente el alcance del sistema.
- Planificar el desarrollo iterativo e incremental del sistema.
- Guiar a los desarrolladores.
- Identificar oportunidades de reutilización.

Al software se le verificaron las funcionalidades siguientes:

- Importar modelo de casco de buque.
- Completar el modelo del casco de buque importado.
- Importar bitmap con el plano de forma del casco.
- Modelar un casco a partir de una imagen bitmap.

- Modelar las líneas de los contornos del casco de buque.
- Introducir los parámetros del casco.
- Realizar cálculos hidrostáticos.

A los estudiantes, oficiales de la Academia Naval de las FAR y jefes de empresas navales, se les realizó una encuesta de opinión el día 6 de junio a las 10:00 de la mañana. Los criterios acerca del módulo fueron en su totalidad positivos y se mostró un alto grado de aceptación por parte de los encuestados. El registro de la encuesta se encuentra archivado en el Vicedecanato de Investigación de la Facultad 4, como parte de la documentación del grupo de investigación SIPII. También se emitió un aval por la Academia Naval *Granma* como muestra de la conformidad a las pruebas realizadas sobre el módulo, ver Anexo A.3.

3.2.6. Resultados de las pruebas

El resultado de las pruebas de software tiene gran importancia, porque ayuda a mejorar la calidad del mismo. Se utiliza para obtener las fallas que presenta el sistema y poder analizar las futuras. Al corregir estas fallas se disminuye el mayor número de no conformidades posible, mejorando así el grado de aceptación por parte del cliente.

La realización de pruebas funcionales permitió identificar una serie de no conformidades que se pueden observar en la tabla 3.3:

Tabla 3.3. No Conformidades

No. NC	Requisito Funcional	Descripción	Complejidad	Estado
1	RF 1	No importaba un modelo en formato txt.	Baja	Resuelta
2	RF 2	No completaba un modelo importado en stl	Alta	Resuelta
3	RF 2	No completaba un modelo importado en txt	Baja	Resuelta
4	RF 2	No completaba un modelo importado en brep	Alta	Resuelta
5	RF 3	Se cerraba la aplicación cuando se importaba la imagen	Alta	Resuelta
6	RF 3	No mostraba la imagen importada en bitmap	Baja	Resuelta
7	RF 4	No se diferenciaba el plano del sketcher en que se trabajaba	Baja	Resuelta
8	RF 4	No interpolaba las curvas B-Spline dibujadas sobre la imagen	Alta	Resuelta
9	RF 5	Las líneas no se mostraban juntos con el casco	Alta	Resuelta
10	RF 7	El resultado de los cálculos fue incorrecto	Baja	Resuelta

Se puede afirmar que la efectividad del módulo para el diseño de cascos de buques y cálculos hidrostáticos, está avalada por los resultados de las pruebas realizadas por el equipo de desarrollo y especialistas en

construcción naval.

En la figura 3.12 se muestran los datos correspondientes a cada iteración de prueba por las que transitó el módulo.

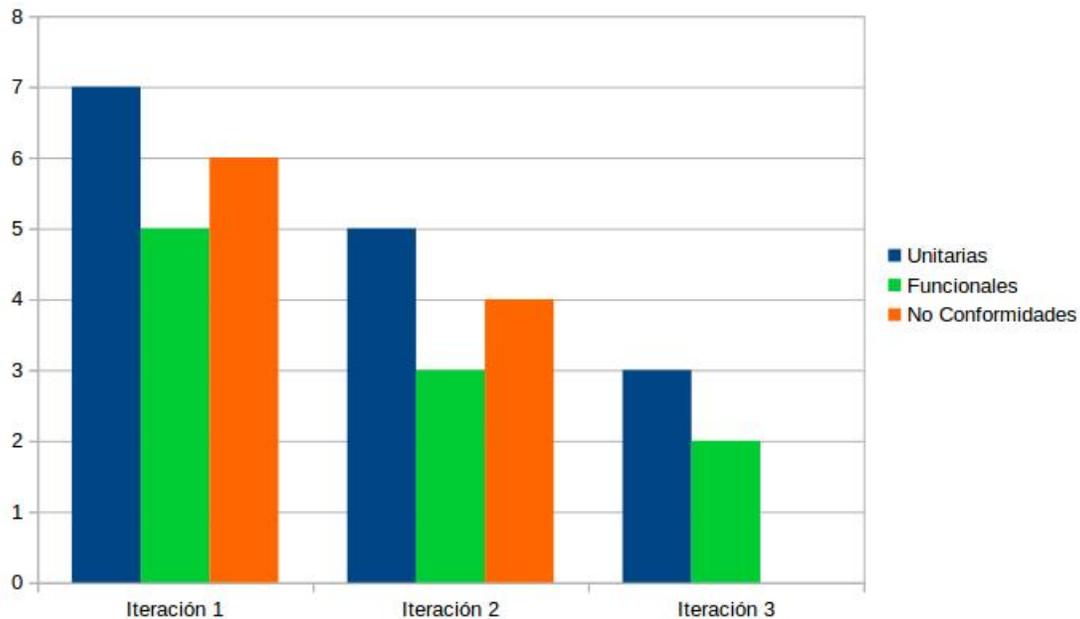


Figura 3.12. Iteraciones de las pruebas

3.3. Conclusiones del capítulo

- Se caracterizaron los estándares de codificación establecidos por el grupo de investigación para la implementación de las clases y los métodos definidos en la fase de diseño.
- Las comparaciones entre los valores flotantes de los cálculos hidrostáticos de la implementación y los proporcionados por especialistas no son exactas, por tanto fue necesario determinar un margen de error de 0.0001 m.
- Con las pruebas funcionales se detectaron un total de 10 no conformidades que fueron corregidas, garantizando una mejor calidad de la solución.

Como resultado del proceso de Investigación-Desarrollo realizado se arribó a las siguientes conclusiones generales:

- Con las funcionalidades del módulo obtenido se informatizó el modelado de cascos de buques a partir de su plano de forma, la obtención de los planos de forma a partir de un modelo de casco, así como la visualización de modelos importados en formatos de intercambio de información gráfica normalizados.
- El resultado obtenido contribuye a la soberanía tecnológica en el área de la construcción naval y reduce el tiempo de diseño durante el proceso de modelado, con respecto a procedimientos manuales.
- Para poder modelar el casco a partir de las líneas características de los tres planos de forma es necesario obtener los puntos de intercepción entre las líneas características, guardarlos ordenados en una matriz y pasar esta a la API `GeomAPI_PointsToBSplineSurface`.

Recomendaciones

A partir del estudio realizado y luego de haber analizado los resultados obtenidos se recomiendan los siguientes elementos a tener en cuenta para futuros trabajos:

- Completar la funcionalidad de modelar un casco de barco a partir de una imagen con el plano de forma.
- Incorporar la funcionalidad que permita realizar los cálculos hidrostáticos utilizando el método numérico.
- Incorporar la funcionalidad para dibujar un plano de forma.

Bulkcarrier Es un barco que se dedica al transporte de cargas secas a granel. 18, 20

Cabotaje Navegación costera. 5

carenado Se llama así la parte del barco sumergida en el agua considerada tanto exteriormente como referida al material que lo compone. Recibe también las denominaciones de vivo, obra viva y vientre de la nave. En sistemas CAD se define como el proceso de diseño de la carena del barco. 12, 13

DAMEX Es el nombre de las sucursales en el extranjero de la compañía de construcción naval holandesa DAMEN. 4, 5

entities Son modelos parametrizados creados a partir de los prototipos de series existentes en el módulo Ship del FreeCAD. 14

escora Inclinación lateral que toma un buque por la fuerza del viento, el ladeamiento de la carga u otra causa. 13

Lloyd's Register Es una organización global que ofrece servicios de auditoría independiente a empresas que operan con activos de alto riesgo, y de capital intensivo en los sectores de energía y transporte, para mejorar la seguridad de la vida, los bienes, y el medio ambiente. 4

LNG Es un buque dedicado al transporte de gas natural licuado. 18

LPG Barco tanque para el transporte de gas propano licuado. 18

SENER Es una empresa internacional de servicios profesionales de ingeniería. 15

trancanil En construcción naval, se denomina trancanil a la zona de unión entre la traca cinta, última traca superior del forro exterior del casco y la cubierta principal. 9

VERITAS Es una empresa que realiza ensayos, inspección y emite certificación en la industria, la marina, la construcción, el comercio internacional y procesos que puedan afectar el medio ambiente. 5

- A_{wp}* Área de Cuaderna Maestra. 17, 19, 29
- API** Application Programming Interface. 37, 38, 51
- AUP** Agile Unified Process. 20, 21
- AUP-UCI** Agile Unified Process versión UCI. 6, 22
- B** Manga. 17, 18
- CAD** Diseño Asistido por Computadora. 1, 3, 4, 7, 10, 24, 28
- CAE** Ingeniería Asistida por Computadora. 24
- CAM** Fabricación Asistida por Computadora. 10, 24
- CASE** Ingeniería de Software Asistida por Computación. 22
- CB** Coeficiente de Bloque. 17, 18, 29
- CDC** Astillero de Casa Blanca. 4, 5, 16
- CEPRONA** Centro de Proyectos Navales. 4
- CIDNAV** Centro de Investigación y Desarrollo Naval. 4, 5
- CM** Coeficiente del Área de la Cuaderna Maestra. 18
- CP** Coeficiente Prismático. 17, 18, 29
- CWP** Coeficiente de Flotación. 17, 29
- DevOps** Desarrollo y Operaciones. 26
- ENAST** Empresa Nacional de Análisis Técnico del Agua. 5
- GRASP** General Responsibility Assignment Software Patterns. 35
- GUI** Interfaces Gráficas de Usuario. 25
- KB** Centro de Carena. 17, 19, 29
- KG** Centro de Gravedad. 17, 19, 29
- L** Eslora. 18, 19
- L_{pp}** Eslora entre perpendiculares. 17
- NURBS** Non Uniform Rational B-Spline. 10, 12, 28
- OCCT** Open Cascade Technology. 24
- OCE** Opencascade Community Edition. 24
- QA** Aseguramiento de la Calidad. 26
- SIPII** Soluciones Informáticas para la Ingeniería y la Industria. 28, 49

T Calado. 17–19

TIC Tecnologías de la Información y las Comunicaciones. 22

UCI Universidad de las Ciencias Informáticas. 20, 21

UML Lenguaje Unificado de Modelado. 36

V Volumen sumergido. 17–19, 29

v Velocidad máxima. 17, 19

Referencias bibliográficas

- Arribas, Francisco Pérez (2004). «Modelado Geométrico del casco de un buque». En: *Universidad Politécnica de Madrid* (vid. págs. 9, 10).
- ASPL (2018). *Estándares de codificación*. URL: <http://www.aspl.es/fact/files/aspl-fact/estandares-node2.html> (vid. pág. 38).
- Baldomir, Carlos Fernández (2014). «Diseño de Formas». En: *Universidad de Coruña* 14-03 (vid. pág. 19).
- Cascade, Open (2018). *About Open Cascade*. URL: <https://www.opencascade.com/content/documentation> (vid. pág. 24).
- Castro, Ricardo Alvariño, Juan José Azpíroz Azpíroz y Manuel Meizoso Fernández (1997). *El Proyecto Básico del Buque Mercante*. Ed. por Fondo Editorial de Ingeniería Naval. Madrid: Colegio Oficial de Ingenieros Navales. ISBN: 84-921750-2-8 (vid. págs. 4, 17-19).
- Colmena, Cristina Cerón (2015). «Estudio sobre esquemas de subdivisión no lineales en mallados no uniformemente espaciados». En: *Universidad Politécnica de Cartagena* (vid. pág. 3).
- Creator, Qt (2018). *Qt Creator*. URL: https://wiki.qt.io/About_Qt (vid. pág. 25).
- Defcar (2010). *DEF CAR CAD/CAM systems for Shipdesign anda Shipbuilding*. URL: http://www.defcar.com/index_esp.htm (vid. pág. 14).
- Delfship (2016). *DelfShip*. URL: <http://www.delftship.net/> (vid. pág. 13).
- Duarte, Maria Elena Valle (2015). *Casos de prueba y tipos de pruebas*. URL: <https://prezi.com/jcygaztotrn/casos-de-prueba-y-tipos-de-pruebas/> (vid. pág. 44).
- Ecured (2018). *Mapa Conceptual*. URL: https://www.ecured.cu/Mapa_conceptual (vid. pág. 27).
- Farin, Gerald E. (1999). *NURBS From Projective Geometry to Practical Use*. 2da. ISBN: 1568810849 (vid. pág. 11).
- FreeCAD (2017). *FreeCAD-Ship s60 tutorial*. URL: https://www.freecadweb.org/wiki/FreeCAD-Ship_s60_tutorial/es# (vid. pág. 14).
- FreeShip (2006). «FREE!ship manual». En: URL: http://www.marin.ntnu.no/havromsteknologi/Prosjekter/Den_Unge_Skips_Designeren/Freeship/Freeship_Manual_2.6.pdf (vid. pág. 13).
- GitLab (2018). *About GitLab*. URL: <https://about.gitlab.com/> (vid. pág. 26).
- Jacobson, I., J. Rumbaugh y G. Booch (2000). *El Lenguaje Unificado de Modelado. Manual de Referencia* (vid. pág. 38).
- Jeffries, R., A. Anderson y C. Hendrickson. (2001). *Extreme Programming Installed* (vid. pág. 30).
- Larman, C. (2003). *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. ISBN: 978-84-205-3438-1 (vid. pág. 35).
- Letelier, Patricio (2007). «Pruebas de Aceptación como conductor del Proceso Software». En: *Universidad Politécnica de Valencia* (vid. págs. 47, 48).
- Lewis, Edward V., ed. (1988). *Principles of Naval Architecture*. Vol. Volume I Stability and Strength. 601 Pavonia Avenue Jersey City, NJ: The Society of Naval Architects y Marine Engineers. ISBN: 0-939773-00-7 (vid. págs. 16, 19).

- Microsoft (2018). *Usando el Patrón de Diseño Observer Manual de Laboratorio*. URL: <https://msdn.microsoft.com/es-es/library/bb972192.aspx> (vid. pág. 35).
- Pressman, Roger S. (2003). *Ingeniería de Software. Un enfoque práctico*. 5ta Edición (vid. pág. 22).
- (2005). *Ingeniería del Software. Un enfoque práctico*. Sexta Edición (vid. pág. 34).
- Rhinoceros (2018). *Características de Rhino 6*. URL: <https://www.rhino3d.com/6/features> (vid. pág. 12).
- SailandTrip (2017). *Terminología Naval, Dimensiones del barco*. URL: <https://sailandtrip.com/partes-del-barco-dimensiones/> (vid. págs. 8, 9).
- Sánchez, T. Rodríguez (2015). *Metodología de desarrollo para la actividad productiva de la UCI* (vid. págs. 20, 21).
- Schneekluth, H. y V. Bertram (1998). *Ship Design for Efficiency and Economy*. Linacre House, Jordan Hill, Oxford OX2 8DP: Reed Educational y Professional Publishing Ltd. ISBN: 0 7506 4133 9 (vid. pág. 19).
- Sederberg, Thomas W. (2014). *Computer Aided Geometric Design*. Brigham Young University: Department of Computer Science (vid. pág. 11).
- SENNER (2018). *FORAN*. URL: <http://www.marine.sener/es/foran> (vid. pág. 15).
- Sommerville, Ian (2006). *Ingeniería de Software*. 8va Edición (vid. págs. 33, 40).
- (2007). *Software Engineering*. Eighth Edition. China Machine Press. ISBN: 0-321-313779-8 (vid. pág. 35).
- Stroustrup, B. (2013). *The C++ Programming Language* (vid. págs. 24, 25).
- Surf, Max (2017). *About MaxSurf*. URL: <http://www.maxsurf.net/aboutmaxsurf.html> (vid. págs. 12, 13).
- Tenorio, R. Ruiz (2010). *Las Pruebas de Software y su Importancia en las Organizaciones* (vid. págs. 43, 44).
- Test, Qt (2018). *Qt Test*. URL: <http://doc.qt.io/qt-5/qttest-index.html> (vid. pág. 45).
- Torvalds, L. y J. Hamano (2018). *Git*. URL: <https://git-scm.com/> (vid. pág. 25).
- Tupper, E. C (1996). *Introduction to Naval Architecture*. Linacre House, Jordan Hill, Oxford OX2, 8DP. Elsevier Science Ltd. ISBN: 0 7506 2529 5 (vid. pág. 8).
- Watson, David G.M. (1998). *Practical Ship Design*. Ed. por R. Bhattacharyya & M.E. McCormick. OCEAN ENGINEERING SERIES EDITORS. Vol. Volume I. ELSEVIER OCEAN ENGINEERING BOOK SERIES. The Boulevard, Langford Lane Kidlington, Oxford OX5 IGB, UK: British Library Cataloguing. ISBN: 0-08-044054- I (vid. pág. 18).
- Wu, You-Sheng, Wie-Cheng Cui y Guo-Jun Zhou, eds. (2001). *Practical Design of Ships and Other Floating Structures*. Vol. I. The Boulevard, Langford Lane Kidlington, Oxford OX5 IGB, UK: ELSEVIER. ISBN: 0080439500 (vid. pág. 4).

Apéndices

A.1. Historias de usuarios

Tabla A.1. Historia de Usuario Importar modelo de casco de buque

Número: 1	Nombre del requisito: Importar modelo de casco de buque	
Programador: Kevin Pérez Mandina	Iteración Asignada: 1era	
Prioridad: Alta	Tiempo Estimado: 2 semanas	
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas	
Descripción:		
1- Objetivo: Permitir importar un modelo a partir de formatos stl, step, brep y txt.		
2- Acciones para lograr el objetivo (precondiciones y datos): Para importar un modelo se debe tener en cuenta los siguientes datos: - El modelo en los formatos stl, step, brep o txt.		
3- Flujo de la acción a realizar: - Se selecciona el botón Import y se busca el fichero con el modelo. - Se presiona aceptar y se importa el modelo.		
Observaciones:		
Prototipo de interfaz:		

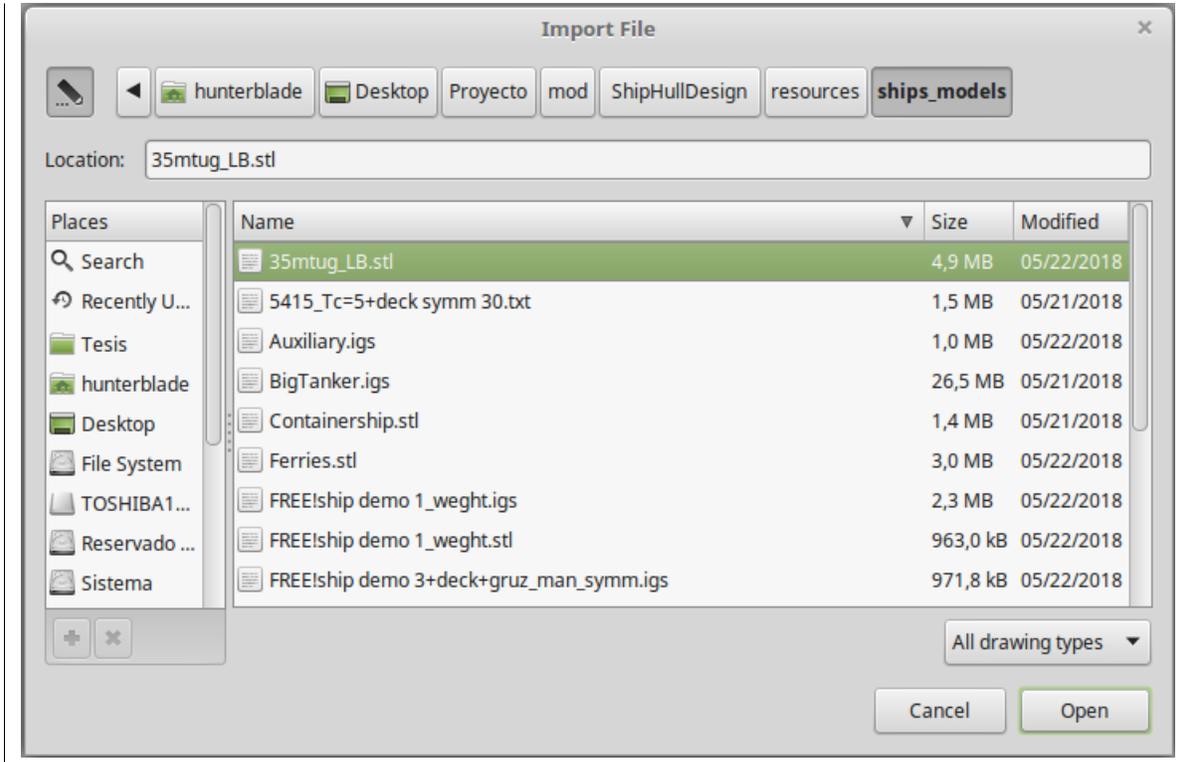


Tabla A.2. Historia de Usuario Completar el modelo del casco de buque importado

Número: 2	Nombre del requisito: Completar el modelo del casco de buque importado
Programador: Kevin Pérez Mandina	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas
Descripción:	
1- Objetivo: Permitir completar la forma de un modelo si se importa su mitad.	
2- Acciones para lograr el objetivo (precondiciones y datos): Para completar el modelo del casco de buque importado se debe tener en cuenta los siguientes aspectos: - La mitad de un modelo - Usar la API de OpenCascade BRepBuilderAPI-Transform.	
3- Flujo de la acción a realizar: - Se selecciona el botón Mirror y se muestra el modelo completo - Si se selecciona el botón Cancel se cierra la ventana.	
Observaciones:	
Prototipo de interfaz:	

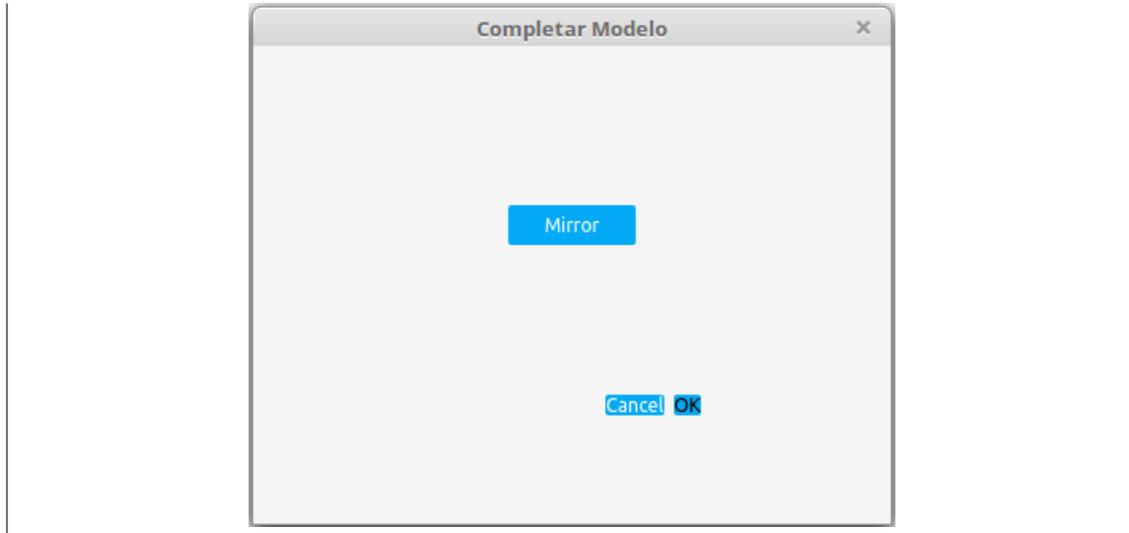


Tabla A.3. Historia de Usuario Modelar un casco a partir de una imagen bitmap.

Número: 4	Nombre del requisito: Modelar un casco a partir de una imagen bitmap.
Programador: Kevin Pérez Mandina	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas
<p>Descripción:</p> <p>1- Objetivo: Permitir modelar un casco a partir de una imagen bitmap.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para Modelar un casco a partir de una imagen bitmap se debe tener en cuenta los siguientes aspectos:</p> <ul style="list-style-type: none"> - Una imagen bitmap con el plano de forma del casco en él área de trabajo. - La clase SketchObject. - La clase ViewProviderSketch. - Usar las API de OpenCascade BRepBuilderAPI-MakeFace y GeomAPI-PointsToBSplineSurface. - El valor de escala entre las líneas del plano de forma. <p>3- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - Se introduce el valor de escala entre las líneas del plano de forma y se selecciona el botón Casco. - Se muestra en el área de trabajo el modelo del casco. - Si se selecciona el botón Cancel se cierra la ventana. 	
Observaciones:	
Prototipo de interfaz:	

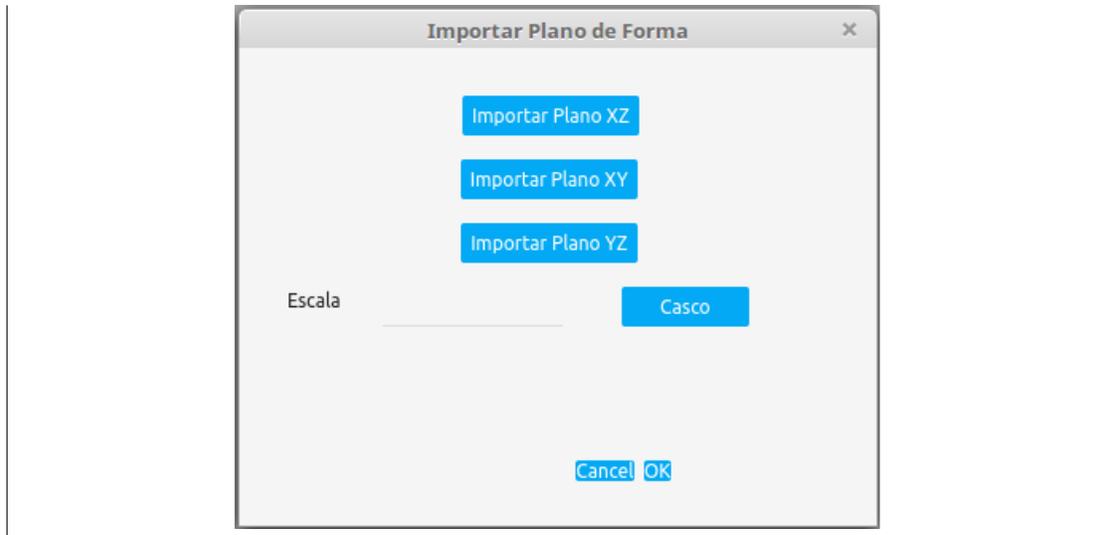
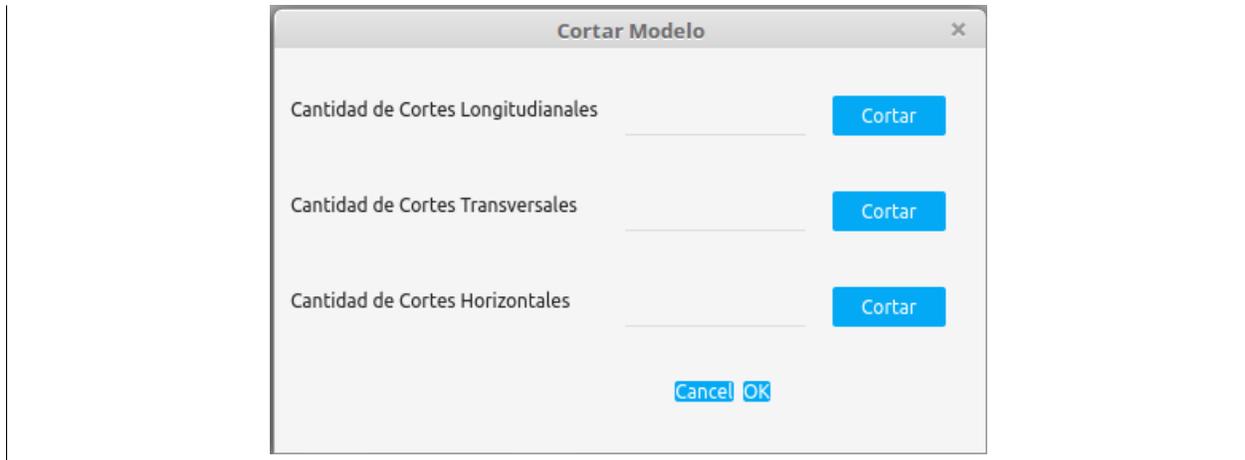


Tabla A.4. Historia de Usuario Modelar las líneas de los contornos del casco de buque

Número: 5	Nombre del requisito: Modelar las líneas de los contornos del casco de buque
Programador: Kevin Pérez Mandina	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 2 semanas
Riesgo en Desarrollo: N/A	Tiempo Real: 1, 5 semanas
Descripción:	
1- Objetivo: Permitir el modelado de las líneas de los contornos del casco de buque.	
2- Acciones para lograr el objetivo (precondiciones y datos): Para el modelado de las líneas de los contornos del casco de buque se debe tener en cuenta los siguientes aspectos: - El modelo de un casco de buque. - El plano sobre el cual se obtendrán las líneas. - La cantidad de líneas a modelar. - Usar la API de OpenCascade BRepAlgoAPI-Section.	
3- Flujo de la acción a realizar: - Cuando el usuario introduzca correctamente los datos necesarios y seleccione la opción Cortar en dependencia del plano, se crea el conjunto de líneas sobre el casco. - Si se selecciona el botón Cancel se cierra la ventana.	
Observaciones:	
Prototipo de interfaz:	



A.2. Casos de Prueba

Tabla A.5. Diseño de Casos de Prueba RF 1

Descripción general			
Importar modelo de casco de buque			
SC 1 Importar Modelo			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Miror.	Selecciona la opción de Import	Brinda la posibilidad de importar un modelo con las extensiones .brep, step y txt.	ShipHullDesgin/Ship/Import

Tabla A.6. Diseño de Casos de Prueba RF 2

Descripción general			
Permitir el completado de una figura dada la mitad de un modelo			
SC 1 Completar figura			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción de Miror.	Selecciona la opción Miror el cual mostrará una ventana para seleccionar el botón Miror para completar el modelo	Brinda la posibilidad de completar un modelo importado.	ShipHullDesgin/Ship/Mirror
EC 1.2 Opción Miror	Selecciona la opción Miror	Completa el modelo importado.	ShipHullDesgin/Ship/Mirror
EC 1.3 Opción de Close	Selecciona la opción de Close	Cierra la ventana	ShipHullDesgin/Ship/Close

Tabla A.7. Diseño de Casos de Prueba RF 3

Descripción general			
Importar bitmap con el plano de forma del casco			
SC 1 Importar bitmap			
Escenario	Descripción	Respuesta del sistema	Flujo central

EC 1.1 Opción Import Plane.	Selecciona la opción de Import Plane	Brinda la posibilidad de importar un bitmap con el plano de forma.	ShipHullDesgin/Ship/ImportPlane
EC 1.2 Opción de Ok	Selecciona la opción Ok	Muestra la imagen en el área de trabajo.	ShipHullDesgin/Ship/Ok
EC 1.3 Opción de Close	Selecciona la opción de Close	Cierra la ventana	ShipHullDesgin/Ship/Close

Tabla A.8. Diseño de Casos de Prueba RF 5

Descripción general			
Modelar las líneas de los contornos del casco de buque			
SC 1 Modelar líneas características			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción OutlineDraw.	Selecciona la opción de OutlineDraw	Brinda la posibilidad de modelar líneas características del plano de forma se introduce la cantidad de líneas en el plano que se quiera modelar	ShipHullDesgin/Ship/OutlineDraw
EC 1.2 Opción de Apply	Selecciona la opción Cortar	Muestra las líneas en el modelo del casco.	ShipHullDesgin/Ship/Cortar
EC 1.3 Opción de Close	Selecciona la opción de Close	Cierra la ventana	ShipHullDesgin/Ship/Close

Tabla A.9. Diseño de Casos de Prueba RF 6

Descripción general			
Introducir los parámetros del casco			
SC 1 Introducir parámetros			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Create.	Selecciona la opción de Create	Brinda la posibilidad de introducir los parámetros del casco.	ShipHullDesgin/Ship/Create
EC 1.2 Opción de Create	Selecciona la opción Create	Introduce los parámetros al módulo.	ShipHullDesgin/Ship/Create
EC 1.3 Opción de Close	Selecciona la opción de Close	Cierra la ventana	ShipHullDesgin/Ship/Close

Tabla A.10. Diseño de Casos de Prueba RF 7

Descripción general			
Realizar cálculos hidrostáticos			
SC 1 Realizar cálculos hidrostáticos			
Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Opción Hydrostatic Calculation.	Selecciona la opción de Hydrostatic Calculation	Brinda la posibilidad de realizar los cálculos hidrostáticos.	ShipHullDesgin/Ship/Hydrostatic Calculation
EC 1.2 Opción de CB	Selecciona la opción CB	Calcula CB	ShipHullDesgin/Ship/CB

EC 1.3 Opción de Volumen	Selecciona la opción Volumen	Calcula Volumen	ShipHullDesgin/Ship/Volumen
EC 1.4 Opción de Área CM	Selecciona la opción Área CM	Calcula Área de la cuaderna maestra	ShipHullDesgin/Ship/Área CM
EC 1.5 Opción de CP	Selecciona la opción CP	Calcula CP	ShipHullDesgin/Ship/CP
EC 1.5 Opción de CWP	Selecciona la opción CWP	Calcula CWP	ShipHullDesgin/Ship/CWP
EC 1.6 Opción de KB	Selecciona la opción KB	Calcula KB	ShipHullDesgin/Ship/KB
EC 1.7 Opción de KG	Selecciona la opción KG	Calcula KG	ShipHullDesgin/Ship/KG
EC 1.8 Opción de Close	Selecciona la opción de Close	Cierra la ventana	ShipHullDesgin/Ship/Close

A.3. Aval



ACADEMIA NAVAL "GRANMA"



ORDEN "ANTONIO MACEO"



NC-ISO 9001: 2015
REGISTRO NO. 021-2017

La Habana, 6 de junio de 2018
"Año 60 de la Revolución"

AVAL

El software de Diseño Naval desarrollado por el compañero Kevin Pérez Medina de la Universidad de Ciencias Informáticas constituye la solución a necesidades de la especialidad de Ingeniería Naval para la formación de especialistas, el empleo en centros de proyectos y astilleros de construcción y reparación naval. Este programa está basado en la generación de formas mediante solucionador NURBS, técnica empleada por los programas de diseño profesionales que permiten la descripción de las formas complejas del casco y otros elementos del buque. Aunque el programa se encuentra en desarrollo se aprecian herramientas avanzadas para la obtención y descripción de la geometría del casco de los buques que permiten la generación directa de la geometría 3D del casco del buque a partir de una imagen del plano de formas, opción que no brinda los programas profesionales como el DEFCAR, MAXSURF, empleados en la institución, el Centro de Investigación y Desarrollo Naval y el Astillero CDC de Casablanca.

El desarrollo de este programa implica la independencia tecnológica del sector naval en las especialidades de diseño, construcción y reparación naval, pues en la actualidad la compra de programas profesionales afectan económicamente por la necesidad de actualización, compatibilidad con los sistemas operativos y los elevados precios, además de presentar limitaciones para el desarrollo de embarcaciones que respondan a las necesidades reales de Cuba como por ejemplo:

- Diseño de embarcaciones para la Marina de Guerra fuera de los parámetros de diseño estandarizados que presentan los programas profesionales.
- Conversión y transformación de embarcaciones para el desarrollo del transporte marítimo, la marina de Guerra y las Tropas Guardafronteras.
- Formación de las especialidades de Ingeniería en Construcción Naval cuyo software de diseño permita el desarrollo de embarcaciones acorde a las necesidades reales de Cuba.

Todas estas acciones requieren del Plano de Formas que describe la geometría del casco del buque, para la obtención de las cualidades marinerías del buque y como documentación técnica oficial de las embarcaciones. El Plano de Formas es la "huella digital del casco del buque", toda la documentación técnica del buque, así como los cálculos, dependen de las formas descritas en 2D para la explotación y en 3D para los cálculos.

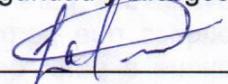
Existen clientes potenciales para la explotación de un software como este para el diseño, desarrollo, construcción y reparación de embarcaciones como son:

- Academia Naval "Granma", MGR, MINFAR, Universidad marítimo – naval de Cuba.

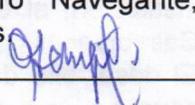
- Centro de Investigación y Desarrollo Naval (CINAVAL), UIM, MINFAR, Diseño y desarrollo de embarcaciones.
- Centro de Proyectos Navales (CEPRONA), MINAL, diseño y desarrollo de embarcaciones.
- Astillero ASTICAR, MINAL, construcción y reparación de embarcaciones.
- Astilleros "CDC Havana", MITRANS, construcción y reparación de embarcaciones.
- EMI Astilleros ASTIMAR, UIM, MINFAR, construcción y reparación de embarcaciones.

Entre otros centros en Cuba que utilizan como herramienta estos programas. Por todo lo antes expuesto consideramos la necesidad de desarrollo de este programa, la utilidad de los resultados alcanzados en esta etapa de desarrollo y la posibilidad de colaboración futura con nuestros especialistas y alumnos. Se avala la profundidad y validez de las formulaciones empleadas en el software, así como la secuencia de desarrollo planificada en función de la secuencia lógica de estudio del Diseño Naval: Generación de Formas, Cálculo Hidrostático, Estabilidad, Insumergibilidad, Resistencia al Avance, Propulsión, Gobierno y Balance de los Buques.

CF Dr. C Liván Lazo Porta, Profesor Titular, Jefe de Cátedra de Marinería, Ingeniero navegante, Especialista en Técnicas Avanzadas de Seguridad y Riesgos en la especialidad de Construcción Naval.

_____ 

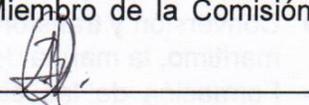
CF Mr. Cs Frank Acosta Chipi, Profesor Auxiliar, Ingeniero Navegante, Especialista de Modelación y Simulación del Centro de Simuladores

_____ 

1er Tte Ing. Odannis Gutiérrez Rodríguez, Ingeniero en Construcción Naval, Jefe de Base de Reparaciones Navales de Tropas Guardafronteras, Especialista en Construcción y Reparación Naval. Miembro de la Comisión Ministerial de Exámenes.

_____ 

Ing. Alejandro Rodríguez Conde, Ingeniero en Construcción Naval, Jefe de Brigada del taller de Acero del Astillero CDC Havana. Miembro de la Comisión Ministerial de Exámenes.

_____ 

2^{do} Jefe de Cátedra de Ingeniería Naval
Teniente de Navío

Rubelis Sarmiento Rodicio

