

Universidad de las Ciencias Informáticas

Facultad 5



Módulo de vaciado de sólidos para la herramienta AsiXMec 1.0

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autor: Osnier Cuesta Aguiar

Tutor: Ing. Adrián Hernández Aguilera

Co-tutores: Ing. José Angel Lores Estrada

Ing. Juan Manuel Veiga León

La Habana, Julio de 2015

“Año 57 de la Revolución”

"La única lucha que se
pierde es la que se
abandona."

"Che Guevara"



Shane

Shane

"Che Guevara"

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y recomiendo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Firma del Autor

Osnier Cuesta Aguiar

Firma del Tutor

Ing. Adrián Hernández Aguilera

Firma del Co-tutor

Ing. José Angel Lores Estrada

Firma del Co-tutor

Ing. Juan Manuel Veiga León

DATOS DE CONTACTO

Tutores:

Tutor principal: Ing. Adrián Hernández Aguilera (ahaguilera@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Con cuatro años de experiencia en el proyecto Diseño y Simulación de Estructuras Mecánicas, teniendo a su cargo varios de los módulos de la herramienta AsiXMec 1.0. Ocupando la plaza de profesor con categoría de instructor.

Co-tutor: Ing. José Angel Lores Estrada (lores@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2012. Se ha desempeñado desde entonces como programador del proyecto Diseño y Simulación de Estructuras Mecánicas, teniendo a su cargo varios de sus módulos de la herramienta AsiXMec 1.0. Ha participado en eventos de corte científico e investigativo, cursado más de una veintena de cursos de postgrado, ha sido tutor, profesor de la asignatura Práctica Profesional y co-tutor de Trabajos de Diploma en años anteriores.

Co-tutor: Ing. Juan Manuel Veiga León (juanvl@uci.cu)

Graduado de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el año 2014. Desempeña su labor como especialista del Centro FORTES.

DEDICATORIA

Dedico este trabajo:

En primer lugar a mi madre Carmen, por ser la persona que se ha sabido sacrificar tanto por mí, por ser siempre la que ha sabido ayudarme en todos los momentos malos y buenos que he vivido. ¡A ti mamá va dedicado este trabajo!

A mi hermano Reinier, a mi prima Susefy, a mi tía María y a mi padre Israel.

AGRADECIMIENTOS

A mis padres por su apoyo y comprensión.

A mis antiguas amistades del IP, Livan, Manolo, Rodny, Pedro Pablo, Humberto, Iris, Yohana, y los que me faltan por mencionar.

A las nuevas amistades que conocí en esta larga carrera de la universidad, José Carlos, Anyelo, Yarianna, Ale, Pedro Alejandro, Luis Ángel, Frank.

A mi antiguo grupo de aula en la universidad, Pedro, Lore, Luis K., Joaquín, Chang, Yusvel, El Pelly, El Rafa, Yariel, Anislei, Judit, Elizabet y los que me faltan por mencionar.

A mi actual grupo de aula en la universidad, El Niche compañero de batallas, Los Hermanos Isnagas (Lionar y Reinier), El Gio (Eulogio), Nelson, Yuriaski, Leiser, Yosvany y los que me faltan por mencionar.

A mis tutores por confiar en mí y ayudarme en todo momento.

A todo el personal que de un modo u otro apporto ideas para lograr el desarrollo del presente trabajo investigativo.

A todos ustedes gracia.

RESUMEN

Las herramientas computacionales empleadas para mejorar la calidad del diseño de piezas mecánicas en el menor tiempo y al menor costo posible, han pasado a ser una de las mejores opciones dentro del ámbito de las industrias que producen en estas ramas. Actualmente el proyecto Diseño y Simulación de Estructuras Mecánicas, se encuentra inmerso en la culminación de la primera versión de una herramienta de diseño mecánico asistida por computadora, denominada AsiXMec 1.0. Los pasos de diseño en esta herramienta, para la realización de vaciado de sólidos sobre estructuras mecánicas, resultan ser excesivos, provocando que el proceso de construcción de estructuras mecánicas sea en determinadas ocasiones engorroso.

Para reducir los pasos de diseño en AsiXMec se decide desarrollar la operación de vaciado de sólidos. Esta operación permite retirar material de la parte interior del sólido, dejando una cavidad hueca en el interior de la pieza. La operación de vaciado fue desarrollada sobre la arquitectura definida para la primera versión de la herramienta AsiXMec, usando C++ como lenguaje de programación, Qt como entorno de desarrollo y el Proceso Unificado Ágil (AUP-UCI) como metodología de desarrollo.

La integración del módulo posibilitó realizar el vaciado de sólidos sobre estructuras mecánicas y determinar el espesor para los bordes de las caras resultantes. Además permite realizar los distintos tipos de vaciados (interior, exterior e híbrido), reduciendo los pasos de diseño para la obtención de estructuras mecánicas más complejas.

Palabras claves:

Herramientas CAD, Modeladores Geométricos, Operación de vaciado de sólidos.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	6
1.1. HERRAMIENTAS CAD.....	6
1.2. MODELADORES GEOMÉTRICOS.....	6
1.3. OPERACIONES 3D.....	7
1.4. FORMULACIONES MATEMÁTICAS.....	13
1.5. METODOLOGÍAS DE DESARROLLO DE <i>SOFTWARE</i>	16
1.6. TECNOLOGÍAS EMPLEADAS PARA LA CONFECCIÓN DEL MÓDULO VACIADO DE SÓLIDOS.....	18
CONCLUSIONES DEL CAPÍTULO.....	21
CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN	22
2.1. PROPUESTA DE SOLUCIÓN.....	22
2.2. MODELO DE DOMINIO DE LA OPERACIÓN DE VACIADO.....	23
2.3. TÉCNICAS EMPLEADAS PARA LA CAPTURA DE REQUISITOS.....	24
2.4. CASOS DE USO DEL SISTEMA.....	26
2.5. PATRÓN ARQUITECTÓNICO.....	30
2.6. MODELO DE DISEÑO.....	33
2.7. PATRONES DE DISEÑO.....	41
CONCLUSIONES DEL CAPÍTULO.....	42
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS	43
3.1. ESTÁNDAR DE CODIFICACIÓN.....	43
3.2. API UTILIZADAS DE LA BIBLIOTECA <i>OPEN CASCADE</i>	43
3.3. DIAGRAMA DE COMPONENTES.....	43
3.4. MÉTODOS DE PRUEBAS.....	45
CONCLUSIONES DEL CAPÍTULO.....	50
CONCLUSIONES	51
RECOMENDACIONES	52
REFERENCIAS BIBLIOGRÁFICAS	53

BIBLIOGRAFÍA	56
ANEXOS	58
ANEXO 1 DESCRIPCIÓN DE LOS CASOS DE USO.....	58
ANEXO 2 DESCRIPCIÓN DE LAS CLASES	63
ANEXO 3 CASOS DE PRUEBAS ELABORADOS	67
GLOSARIO DE TÉRMINOS	69

ÍNDICE DE FIGURAS

Figura 1. Bocetos creados	2
Figura 2. Extrusión de los bocetos	3
Figura 3. Ventana de dialogo de la operación de vaciado en FreeCAD	11
Figura 5. Modelo de dominio para la operación de vaciado	24
Figura 6. Diagrama de Casos de Uso del Sistema.....	27
Figura 7. Arquitectura del módulo desarrollado.....	32
Figura 8. Diagrama de paquetes correspondiente a la operación de vaciado de sólidos.....	34
Figura 9. Diagrama de Clases del diseño correspondiente a la operación de vaciado de sólidos	35
Figura 10. Diagrama de componentes genérico del módulo de vaciado de sólidos.....	44
Figura 11. Aplicación del vaciado interior en la herramienta AsiXMec.....	49
Figura 12. Aplicación del vaciado interior en la herramienta Inventor.....	49
Figura 13. Resultados de las pruebas.....	50

ÍNDICE DE ECUACIONES

Ecuación 1. Volumen del cubo	14
Ecuación 2. Volumen del prisma.....	14
Ecuación 3. Volumen del paralelepípedo.....	14
Ecuación 4. Volumen del ortoedro.....	15
Ecuación 5. Volumen del cilindro.....	15
Ecuación 6. Volumen del cilindro.....	15
Ecuación 7. Fórmula del método de capas.....	16
Ecuación 8. Proporción directa.....	16

ÍNDICE DE TABLAS

Tabla 1. Variación de AUP-UCI.....	17
Tabla 2. Requerimientos funcionales	25
Tabla 3. Determinación y descripción de los actores del sistema.....	26
Tabla 4. Descripción del CU-1_Aplicar el vaciado interior	28
Tabla 5. Descripción de la clase ShellController	36
Tabla 6. Descripción de la clase ShellBuilder.....	38
Tabla 7. Descripción de la clase ShellDriver	39
Tabla 8. Descripción de la clase ShellCommand	40
Tabla 9. Estrategia de prueba diseñada.....	47
Tabla 10. Caso de prueba basado en el CU-1_Aplicar el vaciado interior	47
Tabla 11. Descripción del CU-2_Aplicar el vaciado exterior	58
Tabla 12. Descripción del CU-3_Aplicar el vaciado híbrido	60
Tabla 13. Descripción de la clase ShellState	63
Tabla 14. Descripción de la clase ShellDialog.....	63
Tabla 15. Descripción de la clase ShellNode	65
Tabla 16. Descripción de la clase ShellItem.....	66
Tabla 17. Caso de prueba basado en el CU-2_Aplicar el vaciado exterior.....	67
Tabla 18. Caso de prueba basado en el CU-3_Aplicar el vaciado híbrido.....	68

INTRODUCCIÓN

La capacidad de diseño y desarrollo de nuevos productos, o la modificación de los existentes, es un factor fundamental para la competitividad de las empresas industriales. Por esta razón, es necesario ofrecer productos de mayor valor añadido, para lo que resulta prácticamente imprescindible adquirir, desarrollar y aplicar eficazmente tecnologías de apoyo a la función de diseño e ingeniería [1].

El concepto de herramientas de diseño e ingeniería hace referencia a términos como: Diseño asistido por computadoras (CAD¹), Fabricación asistida por computadoras (CAM²) e Ingeniería asistida por computadoras (CAE³). Se entiende como CAD a los sistemas informáticos que sirven de soporte a los procesos de diseño y fabricación de cualquier tipo de producto [2]. El uso de las herramientas CAD ha pasado a ser una de las mejores opciones en el ámbito industrial, pues resulta necesario mejorar la calidad, disminuir los costos y acortar los tiempos de diseño y producción.

En la Universidad de las Ciencias Informáticas (UCI), en el proyecto Diseño y Simulación de Estructuras Mecánicas (DISEM), perteneciente al centro Vertex (Entornos Interactivos 3D), se realiza la herramienta CAD, AsiXMec 1.0. La versión actual de AsiXMec no satisface totalmente las necesidades de los usuarios (ingenieros mecánicos). Una vez creados los objetos 3D a través de operaciones como Extruir o Revolucionar, el *software* permite aplicar sobre el sólido resultante operaciones 3D (Redondear, Biselar y Combinar) para modificar su estructura y generar los diseños requeridos por los ingenieros mecánicos, sin embargo la herramienta carece de una operación que permita retirar material en la parte interior de una estructura mecánica, mediante la selección de las superficies, operación que permitiría reducir los pasos de diseño para la obtención de estructuras mecánicas más complejas.

La carencia de esta operación en la herramienta, provoca que no se pueda añadir el grosor deseado a una malla proporcionada con parámetros simples, por lo que, a la hora de realizar el vaciado de una estructura, el proceso se vuelve complicado para obtener el resultado deseado. También provoca que no se pueda generar caras en ambas direcciones de manera simultánea y automática para definir el grosor de las paredes que se

¹ Del inglés *Computer Aided Design*

² Del inglés *Computer Aided Manufacturing*

³ Del inglés *Computer Aided Engineering*

generan. Estas limitaciones traen como consecuencias que el proceso de construcción de estructuras mecánicas dentro de la herramienta sea en determinadas ocasiones muy engorroso. Actualmente para lograr que sobre una estructura mecánica se realice un vaciado de sólidos, se tienen que realizar varios pasos de diseño para lograr el resultado deseado:

1. Crear dos bocetos de igual forma pero uno con mayor dimensión al otro, ubicando el de menor dimensión dentro del que tiene mayor dimensión (ver **¡Error! No se encuentra el origen de la referencia.**).

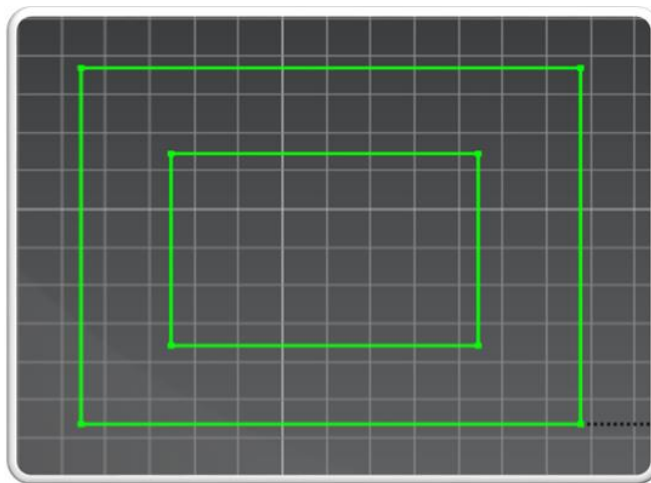


Figura 1. Bocetos creados

2. Aplicar la operación Extruir sobre los bocetos creados, dándole al boceto de mayor dimensión una distancia de extrusión mayor que la que se aplica al boceto de menor dimensión (ver Figura 2).

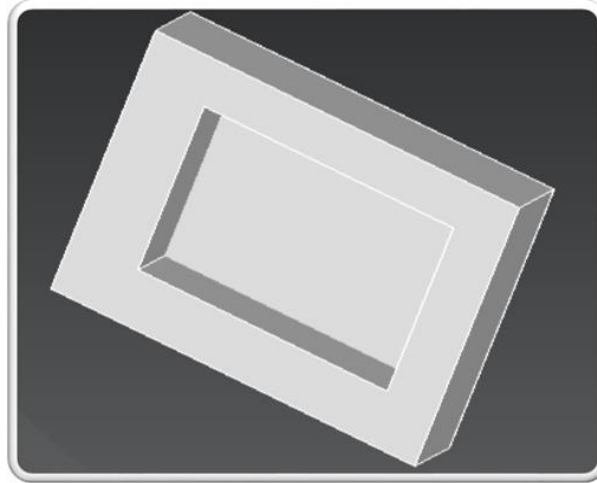


Figura 2. Extrusión de los bocetos

Como resultado se obtiene una estructura mecánica con un vaciado en su interior, pero no se logra con gran exactitud un espesor uniforme en los bordes generados. Además de no permitir la construcción de vaciados a partir de la eliminación de varias caras simultáneamente. Este proceso puede resultar fácil para los sólidos que no presenten mucha complejidad, mientras que en los sólidos que presentan más complejidad los pasos de diseño para obtener el resultado deseado pueden aumentar, sobrecargando el flujo de trabajo que se ha de seguir para lograr el resultado esperado.

Por la situación expuesta anteriormente, se plantea como **problema a resolver**:

¿Cómo realizar el vaciado de sólidos para la creación de estructuras mecánicas?

Este problema se enmarca en el **objeto de estudio** las operaciones 3D en la modelación de estructuras mecánicas. Quedando definido como **objetivo general de la investigación**:

Realizar el vaciado de sólidos en estructuras mecánicas.

El objeto lo delimita el **campo de acción**, la operación de vaciado de sólidos en la modelación de estructuras mecánicas.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas investigativas**:

- Realizar un análisis detallado de la operación de vaciado de sólidos en la modelación de estructuras mecánicas en distintas herramientas CAD.

- Investigar y determinar fórmulas matemáticas para realizar operaciones con el volumen de las estructuras creadas en la herramienta AsiXMec.
- Definir las herramientas, tecnologías y metodología empleadas para el desarrollo del módulo vaciado de sólidos.
- Determinar las API⁴ (interfaz de programación de aplicaciones) proporcionadas por el *framework Open Cascade* para obtener la previsualización y visualización de la operación de vaciado de sólidos en la herramienta AsiXMec.
- Generar los artefactos correspondientes a las fases de desarrollo: “Modelamiento del Negocio”, “Requisitos”, “Análisis y Diseño” e “Implementación”, propuestos por la metodología de desarrollo Proceso Unificado de Desarrollo de *Software* (AUP-UCI).
- Implementar el módulo vaciado de sólidos aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de *Software*.
- Incorporar el módulo vaciado de sólidos como una operación al árbol de *Open Cascade* para llevar a cabo la parametrización.
- Integrar la operación de vaciado de sólidos a la herramienta AsiXMec.
- Realizar pruebas a la solución desarrollada para validar los tipos de vaciado de sólidos generados por el módulo.

El diseño metodológico se realizó utilizando los métodos teóricos y empíricos que se describen a continuación.

Métodos teóricos:

- **Histórico-Lógico:** Se efectuaron búsquedas de información en diferentes trabajos de diploma de cursos anteriores, sitios web, revistas y libros. Con la utilización de este método se investigó sobre la historia y antecedentes relacionados con el tema de investigación propuesto. Este método constituye premisa indispensable para una comprensión, explicación e interpretación más profunda de la materia en cuestión.
- **Analítico-Sintético:** Se realizó el estudio de las principales técnicas, herramientas y estándares existentes para la creación de la operación de vaciado de sólidos.

⁴ Del inglés *Application Programming Interface*

- **Inducción-deducción:** Este método se utilizó para realizar el análisis y la definición de la estrategia a utilizar para el desarrollo del módulo vaciado de sólidos a partir de los problemas detectados.
- **Modelación:** Se empleó para realizar una representación simplificada de la realidad a través de diagramas de clases.

Métodos empíricos:

- **Observación:** Se empleó para constatar los resultados visuales alcanzados.
- **Consulta a especialistas:** Se consultaron distintos profesionales con conocimientos sobre el tema con el objetivo de recibir orientaciones.

El trabajo consta de una estructura en capítulos, además de la parte de los anexos y el glosario de términos, donde se pueden encontrar los detalles de la investigación, así como la propuesta y el diseño del sistema.

Capítulo 1: Fundamentación teórica. Constituye el respaldo teórico de la investigación, incluyendo un estudio del estado del arte de las principales herramientas computacionales que se emplean para el diseño de estructuras mecánicas, además el análisis y descripción de la metodología de desarrollo y las tecnologías empleadas en la solución.

Capítulo 2: Análisis y diseño de la propuesta de solución. En este capítulo se explica detalladamente el proceso de captura y refinamiento de los requisitos del *software* y se describe cómo debe ser realizado el módulo vaciado de sólidos. Además, mediante los diagramas generados por la metodología de desarrollo utilizada se explica la estructura de la aplicación.

Capítulo 3: Implementación y pruebas. En este capítulo se describen los aspectos relacionados con la construcción de la solución propuesta a través del conjunto de pruebas realizadas al módulo vaciado de sólidos para comprobar el funcionamiento de los requisitos funcionales y no funcionales asociados al mismo. Se abordarán los tipos de pruebas realizadas, resultados obtenidos en las mismas y la evaluación de esos resultados.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se encuentran los principales conceptos teóricos vinculados al desarrollo de la investigación para facilitar el entendimiento a partir de distintos enfoques, definiciones y puntos de vistas. Además se presentan las bases teóricas fundamentales relacionadas con las herramientas CAD y los beneficios que brindan. También se abordan elementos fundamentales de los modeladores geométricos, las operaciones 3D y formulaciones matemáticas para hallar volumen. Se define la metodología de desarrollo, las herramientas y tecnologías empleadas para el desarrollo del módulo.

1.1. Herramientas CAD

Actualmente las herramientas CAD son capaces de modelar las distintas piezas que componen el diseño, ya sea en dos o tres dimensiones (2D y 3D respectivamente), con funcionalidades y algoritmos que proporcionen mejor exactitud a los modelos, así como el ensamblado de las piezas previamente diseñadas en el modelador geométrico y la generación de sus planos [3].

El uso de las herramientas CAD ofrece una serie de beneficios para diseñar los productos desarrollados por los ingenieros mecánicos, ya que incluyen [4]:

- Menores costos de desarrollo de productos, aumento de la productividad, mejora en la calidad del producto y un menor tiempo de lanzamiento al Mercado.
- Mejor visualización del producto final, los sub-ensambles parciales y los componentes en un sistema CAD agilizan el proceso de diseño.
- Ofrece precisión en el diseño, lo que contribuye a reducir los errores.
- Permite una reutilización sencilla de diseños de datos y mejores prácticas.

Las herramientas CAD para lograr la modelación de los objetos requeridos por los ingenieros mecánicos se dividen fundamentalmente en módulos, tales como: modeladores geométricos, módulos de ensamble y módulos para la generación de planos; pero se centran fundamentalmente en el primer módulo para modelar objetos tridimensionales. A continuación se aborda este módulo.

1.2. Modeladores geométricos

Los modeladores geométricos son herramientas que permiten modelar todo tipo de objetos 2D y 3D y agrupan un conjunto de operaciones que facilitan el cumplimiento de sus propósitos, para esto incluyen dos módulos

fundamentales:

- *Sketcher*: permite diseñar modelos mediante el empleo de herramientas de diseño con operaciones ágiles e intuitivas.
- Módulo de partes: permite crear planos con las vistas de los modelos o ensamblajes de forma automática en muy poco tiempo.

De acuerdo a los estudios realizados se identifican dos tipos de modeladores geométricos [5]:

- Modelador Geométrico 2D: se basan en entidades geométricas vectoriales como puntos, líneas (polilíneas), círculos, rectángulos, arcos y polígonos, con las que se puede operar a través de una interfaz gráfica.
- Modelador Geométrico 3D: añaden superficies y sólidos 3D. Haciendo uso de un conjunto de funcionalidades como Biselado, Redondeo, División, Operaciones booleanas, entre otras.

En la actualidad han sido desarrollados diferentes modeladores geométricos para las herramientas CAD existentes. El modelador geométrico sobre el que trabaja la herramienta AsiXMec cuenta con todas las características descritas anteriormente permitiendo modelar todo tipo de estructuras mecánicas.

Los modeladores geométricos permiten modelar piezas tridimensionales aplicando un conjunto de operaciones 3D que favorecen el proceso de diseño de estructuras mecánicas. A continuación se presentan algunos términos fundamentales relacionados con dichas operaciones.

1.3. Operaciones 3D

El término operación, que en este entorno de modelado paramétrico 3D es utilizado como traducción del inglés *feature*, se refiere a cada uno de los procesos matemáticos empleados para construir el modelo; por lo que se hace referencia a este proceso como modelado basado en operaciones [6]. El modelo es el resultado de aplicar una secuencia de operaciones y modificaciones. La operación de extrusión es con frecuencia la primera de las operaciones empleadas en la construcción del modelo. Existen otros tipos de operaciones, algunas de las cuales generan nuevos sólidos a partir de bocetos y otras que modifican sólidos existentes.

Estas operaciones pueden clasificarse en operaciones 2D y operaciones 3D. Es importante destacar que existen algunas operaciones que se incluyen dentro de las dos clasificaciones expuestas con anterioridad. Ejemplo de estas son [7]:

Biselado: es un efecto que suaviza los bordes y las esquinas.

Redondeo: la operación redondear consiste en aplicar un redondeo de un borde interior o exterior de la pieza.

Patrón: permiten generar repeticiones tridimensionales mediante diversas formas (rectangular, circular o espejo).

Dentro del módulo *Sketcher*, existen operaciones y conjuntos de operaciones destinadas a la modificación de sólidos tridimensionales previamente creados, que se exponen a continuación [7]:

Vaciado: esta operación realiza el vaciado de sólidos, mediante la selección de la cara a eliminar y el grosor resultante de pared deseado.

Espesor: esta operación añade o elimina espesor de los sólidos mediante la selección de una o varias caras de los mismos.

División: esta operación permite cortar sólidos por superficies.

Superficies gruesas: esta operación proporciona espesor a una superficie previamente creada, dando lugar a un sólido.

Cerrar superficies: esta operación cierra superficies y las transforma en elementos sólidos.

Unir superficies: esta operación calcula la intersección entre un sólido y una superficie, uniéndolos y eliminando la parte deseada.

Operaciones booleanas: dentro de estas se encuentran operaciones tales como: suma, resta, entre otras.

Operaciones de transformación: dentro de estas se encuentran varias utilidades para la traslación, rotación, simetría y escalado de elementos sólidos.

Protrusión: añade material a la estructura mecánica. Existen cuatro modos de realizar esta operación, estos son: Extruir, Revolucionar, Mezclar, Barrer.

Agujero: permite la creación de agujeros en las estructuras mecánicas.

Según el estudio realizado sobre el conjunto de las operaciones que existen destinadas para la modificación de sólidos tridimensionales previamente creados, se enfocará el desarrollo del trabajo en la operación de vaciado, operación que soluciona el problema identificado en la herramienta AsiXMec.

1.3.1. Operación de vaciado de sólidos

Se define el vaciado de sólidos o *Shell* como un modificador que permite dar grosor a una superficie y añade un conjunto de caras en dirección opuesta a las existentes junto con aristas que, si es necesario, conectan las superficies interiores y exteriores en donde falten caras. Es una operación paramétrica utilizada para piezas moldeadas y de fundición. El material de la parte interior se elimina dejando una cavidad hueca. Si cambia las cotas de las piezas o del vaciado, ambos se redimensionarán automáticamente [8].

Para la definición de los tipos de vaciados se especifican una o varias caras de la estructura mecánica para eliminarlas del cuerpo y dejar el resto de las caras para las paredes del vaciado. De no especificar una cara para su eliminación, el vaciado creará una estructura mecánica hueca. Teniendo en cuenta la definición de dicha operación se decide desarrollar un módulo de vaciado para integrarlo a la herramienta de diseño AsiXMec, por lo tanto se realiza un estudio de cómo se maneja esta operación en varios de los sistemas CAD existentes.

1.3.2. Operación de vaciado de sólidos en varios sistemas CAD

La operación de vaciado de sólidos se lleva a cabo en varias de las herramientas CAD existentes en la actualidad, pues facilita el trabajo a la hora de eliminar material de una pieza, creando una cavidad con unas paredes cuyo espesor es especificado por el usuario. El espesor de las paredes no es sólo lateral, sino también se considera para el fondo de la pieza. A continuación se explica cómo se trabaja esta operación en algunos sistemas CAD:

En el sistema CAD *Autodesk Inventor*, en la ventana de diálogo de esta herramienta aparecen varios parámetros para definir el vaciado:

- el primero de ellos es “caras a eliminar”: en el cual se seleccionan las caras que se desean eliminar de la pieza, pueden ser dos caras contiguas. Para deseleccionar una cara ya señalada, se selecciona nuevamente la cara ya seleccionada con el botón izquierdo del ratón a la vez que se presiona la tecla Ctrl [9].
- otro de los parámetros sobre el que se puede actuar es la “dirección”: ésta puede ser interior, exterior o ambas. La primera realiza un vaciado de forma tal que las paredes externas de la pieza son las paredes externas de esta operación. En cambio, el vaciado exterior hace que la pared externa de la pieza original sea la interna del vaciado. El último caso es cuando se producen los dos vaciados

anteriormente descritos de manera simultánea [9].

- el último de los parámetros que ayudan a realizar esta operación es el “espesor”, del que ya se ha hablado en el primer párrafo del sub-epígrafe 1.4.2. En esta herramienta existe una ventana de despliegue en este parámetro, que permite definir el espesor del vaciado y que también muestra valores al usuario para aconsejar qué espesor considerar. Estos valores son el resultado del almacenamiento de los más comunes y se muestran como asesoramiento a la hora de elegir uno [9].
- La herramienta brinda la posibilidad de modificar el espesor para cada una de las paredes resultantes, aunque por defecto sea la opción uniforme la que esté activada. Para ello se ha de seleccionar el botón ">>" para desplegarlo e ir seleccionando un espesor para cada cara según se requiera [9].

En *SolidWorks* una vez seleccionado el icono de vaciado en la ventana de diálogo de la herramienta aparecen varios parámetros para definir el vaciado que se va a realizar [10]:

- El “espesor”: para establecer el espesor de las caras a conservar.
- El parámetro de “dirección”: para determinar el tipo de vaciado, este puede ser exterior e interior.
- Otro de los parámetros que ayudan a especificar esta operación es “caras a eliminar”: permite seleccionar una o más caras en el sólido.

En *AutoCAD* para realizar vaciados a un sólido se debe [11]:

- Teclear el comando *SOLIDEDIT* y luego seleccionar las opciones *Cuerpo* y *Vaciado*. También puede seleccionarse la opción *Solids Editing > Shell* del submenú *Modify* del menú estándar de *AutoCAD*.
- Seleccionar el sólido al cual se le realizará el vaciado (*Select a 3D solid*).
- Seleccionar, si es necesario, las caras que se desean excluir del vaciado.
- Establecer la distancia de vaciado (*Enter the shell offset distance*). Este valor corresponde al espesor de las caras que resultarán del vaciado; si el valor es positivo, el espesor se considerará hacia adentro del cuerpo; si es negativo, hacia fuera.

En *FreeCAD* para realizar el vaciado hacen uso de los siguientes parámetros (ver Figura 3) [12]:

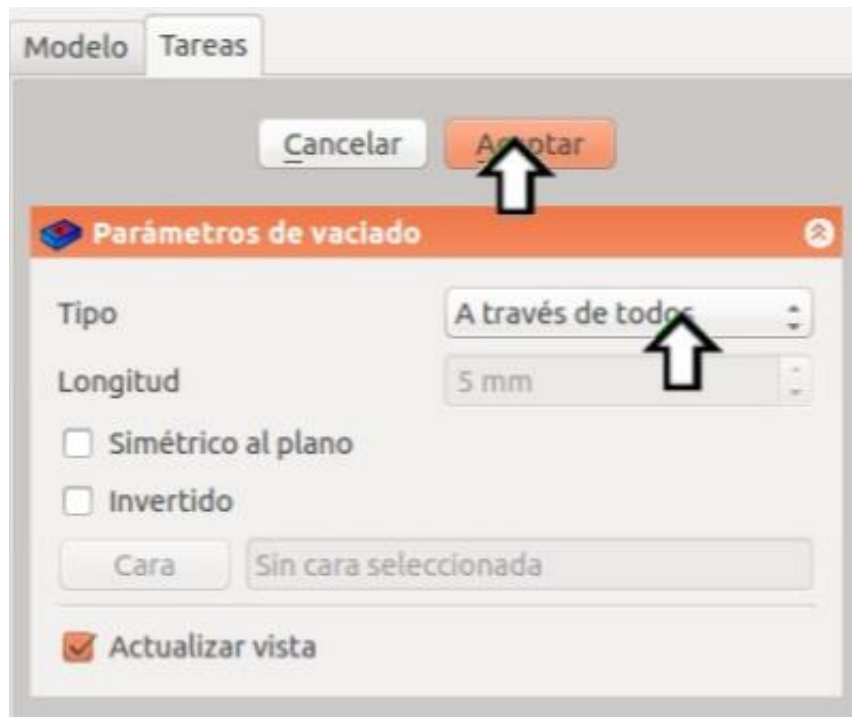


Figura 3. Ventana de dialogo de la operación de vaciado en FreeCAD

En el parámetro “Tipo” se encuentran varias funcionalidades:

- La funcionalidad “Cota”, para determinar la longitud de profundidad, habilitando el parámetro “Longitud”.
- También se encuentra la opción “A través de todos” para que la operación se haga a través de todo el sólido.
- La opción “Al primer lugar” para determinar la operación hasta un determinado lugar del sólido en su interior según el valor del parámetro “Longitud”.
- Por último la opción “Cerca de la cara” es para definir la operación sin la cara que fue seleccionada, habilitando el parámetro “Cara”.

Como parte de las tesis propuestas por el proyecto DISEM en años anteriores se realizó una tesis cuyo objetivo era realizar el módulo de vaciado de sólidos. Esta operación fue realizada y entre los parámetros que contaba la ventana de diálogo para definir el vaciado estaban:

- El parámetro de “determinar la dirección”: ésta podía ser exterior o interior. Interior hace un vaciado de

forma que las paredes externas de la pieza son las paredes externas de esta operación. En cambio, exterior hace que la pared externa de la pieza original sea la interna del vaciado.

- Otro de los parámetros era la “selección de caras”: aquí no se permitía seleccionar las caras en las que se quiere proceder al vaciado. Si no que se pasaban por parámetro todas las caras sobre las que se aplicaría el vaciado.
- El otro parámetro con el que cuenta dicha operación es el “espesor”: se trata de un espesor uniforme en todas las paredes.

Luego de analizar los resultados obtenidos en la tesis propuesta por el proyecto DISEM en años anteriores, se hace necesario el desarrollo de la investigación actual, debido a que la funcionalidad de vaciado de sólidos desarrollada anteriormente no se adapta a la arquitectura del modelador geométrico sobre la cual se está desarrollando la herramienta AsiXMec. La arquitectura actual del modelador 3D hace necesario la creación de un conjunto de clases que faciliten la selección por parte del usuario de las caras a eliminar, que gestionen los datos proporcionados por el usuario y lo registren en el árbol de OCAF⁵ de tal modo que siempre permita ir un paso hacia adelante o hacia atrás en la operación y que si existe algún cambio, este se propague de tal modo, que las operaciones realizadas con anterioridad se recalculen automáticamente. Por otra parte la operación de vaciado desarrollada con anterioridad solo implementaba el vaciado interior y el exterior, puesto que para ese momento eran los requeridos por el proyecto.

Según los estudios realizados, el trabajo con la operación de vaciado de sólidos es muy similar en varios de los sistemas CAD, el único parámetro que no cambia es el de la dirección, ya que, ésta puede ser interior, exterior o ambas. Con respecto a lo antes mencionado se identifican tres tipos de vaciado de sólidos:

- Vaciado interior: consiste en realizar un vaciado a partir de la eliminación de superficies. Posteriormente se proporciona un grosor a las paredes del vaciado, a través de la adición de caras, en dirección opuesta a la normal de las paredes del vaciado.
- Vaciado exterior: consiste en realizar un vaciado a partir de la eliminación de superficies. Posteriormente se proporciona un grosor a las paredes del vaciado, a través de la adición de caras, en la misma dirección de la normal de las paredes del vaciado.

⁵ Del inglés *Open CASCADE Application Framework*

- Vaciado híbrido: se aplican los dos tipos de vaciado de sólidos expuestos con anterioridad.

Teniendo en cuenta la investigación realizada sobre los sistemas similares que implementa la funcionalidad de vaciado de sólidos se decide que la ventana de diálogo del módulo a desarrollar cumpla con los siguientes parámetros:

- El primero de los parámetros sobre el que se puede actuar es la dirección. Ésta puede ser interior, exterior o ambas direcciones.
- El otro parámetro que ayuda a especificar dicha operación, es el tipo de espesor. Se trata de dos tipos de espesor uniforme en todas las paredes. Permitiendo al ingeniero mecánico que decida emplear uno de los dos tipos de espesor. El primero de estos maneja un valor en milímetros proporcionado por el usuario. El segundo tipo de espesor representa un porcentaje del valor total del volumen del sólido seleccionado por lo que se hace necesario el estudio del cálculo de volúmenes de sólidos en revolución.

1.4. Formulaciones matemáticas

Las formulaciones matemáticas son números y símbolos que muestran como obtener algún tipo de resultado. Son ecuaciones especiales que muestran la relación entre diferentes variables [13]. Para el desarrollo del módulo de vaciado de sólidos se decide realizar el estudio sobre fórmulas matemáticas para hallar el volumen de sólidos.

1.4.1. Fórmulas matemáticas para hallar el volumen de figuras geométricas

El volumen es la cantidad de espacio que ocupa un cuerpo. Es una característica cuantitativa del espacio ocupado por un sólido o sustancia. El volumen del sólido o la capacidad de un recipiente se definen por su forma y tamaño lineal. La unidad para medir volúmenes en el Sistema Internacional es el metro cúbico (m^3) que corresponde al espacio que hay en el interior de un cubo de 1 m de lado. Sin embargo, se utilizan más sus submúltiplos, el decímetro cúbico (dm^3) y el centímetro cúbico (cm^3) [14].

Fórmulas para hallar el volumen en algunas figuras geométricas:

La ecuación del volumen del cubo equivale a la longitud de su cara a tercera potencia [15].

$$V = a^3 \dots\dots\dots (Ecuación 1)$$

Donde:

V - cubo volumen

a - longitud de la cara del cubo

El volumen del prisma equivale a la multiplicación del área de la base por la altura [15].

Fórmula volumen de prisma $V = A_b * h$ (Ecuación 2)

Donde:

V - volumen del prisma

A_b - área de la base del prisma

h - longitud de la altura del prisma

Volumen del paralelepípedo equivale a la multiplicación del área de la base por la altura [15].

Fórmula volumen de paralelepípedo $V = A_b * h$ (Ecuación 3)

Donde:

V - volumen del paralelepípedo

A_b - área de la base del paralelepípedo

h - longitud de la altura del paralelepípedo

Volumen del ortoedro equivale a la multiplicación de su longitud, latitud y altura [15].

Fórmula volumen de ortoedro $V = a * b * h$ (Ecuación 4)

Donde:

V - volumen del ortoedro

a - longitud

b - latitud

h - altura

El volumen del cilindro equivale a la multiplicación del área de su base por la altura [15].

Fórmula volumen de cilindro

$V = A_b * h$ (Ecuación 5)

$V = \pi R^2 * h$ (Ecuación 6)

Donde:

V - volumen del cilindro

A_b - área de la base del cilindro

R^2 - radio del cilindro elevado a la dos.

h - longitud de la altura del cilindro

$\pi = 3.141592$

Según los estudios realizados sobre las formulaciones matemáticas, para hallar el volumen de varias figuras geométricas se percibe que, para calcular el volumen depende del tipo de figura que sea, se tiene en cuenta el área de la base (A_b), la longitud de la altura (h), el radio de la base (R) y el valor de la constante Pi (π), estos dos últimos valores solo son empleados para las figuras geométricas que su base sea en forma de esfera.

Teniendo en cuenta que, en la operación de vaciado de sólidos en los sistemas CAD, se eliminan las caras seleccionadas de la estructura mecánica y se muestra la estructura mecánica transformada según el espesor deseado para los bordes; se puede determinar el volumen de una estructura mecánica creada en la herramienta AsiXMec, ya que para realizar un objeto 3D en la herramienta, aplicando la operación Extruir, primero se crea una base y sobre dicha base se aplica la operación Extruir creando un objeto 3D, donde el valor del área de base (A_b) sería el área del boceto que se crea en el visor y la longitud de la altura (h) sería la distancia de la extrusión que se aplica sobre el boceto; los valores del radio de la base (R) y de Pi (π) no se tienen en cuenta para hallar el volumen en las estructuras mecánicas que no presentan una base esférica. Por tanto, la fórmula matemática a emplear para calcular el volumen de las estructuras mecánicas creadas en la herramienta sería (Ecuación 2, Ecuación 3 o Ecuación 5).

1.4.2. *Fórmula matemática para hallar el volumen de sólidos de revolución*

Los sólidos que se generan a partir de hacer girar una región plana alrededor de un eje son denominados sólidos en revolución. Para calcular el volumen de estos sólidos se puede aplicar uno de estos tres métodos: el de disco, el de la arandela y el de los castillos cilíndricos (también se le denomina método de capas) que es el más óptimo de los tres [16].

Por lo que para determinar el volumen de los sólidos de revoluciones creados en la herramienta AsiXMec se emplearía la fórmula del método de capas. Esta fórmula es la siguiente [16]:

$$V = \int_a^b 2\pi x f(x) dx \dots\dots\dots (Ecuación 7)$$

Donde el volumen del sólido de revolución que se genera al hacer girar alrededor del eje y la región que está comprendida entre la curva $y = f(x)$, con $f(x) > 0$, el eje x y las rectas verticales $x = a$ y $x = b$, donde $0 < a < b$ [16].

Definidas las fórmulas matemáticas que pueden ser utilizadas para determinar el volumen de los sólidos creados en la herramienta AsiXMec. Se decide aplicar proporcionalidad directa para determinar el valor del parámetro que maneja el espesor en por ciento. Dicha operación se emplea para conocer el valor del cuarto término de una proporción a partir de los valores de los otros términos.

Haciendo uso de la proporcionalidad directa se puede determinar el espesor en por ciento de las paredes que se generan al retirar las caras seleccionadas en la estructura mecánica. Dicha fórmula es [17]:

$$X = (t * (p/100))/100 \dots\dots\dots (Ecuación 8)$$

Donde:

x - valor que se obtiene de la operación

p - parte (valor entrado por el usuario)

t - todo (volumen de la estructura)

Después de haber determinado las fórmulas matemáticas para el cálculo de volumen de sólidos, los parámetros a considerar para la realización del vaciado de sólidos y los diferentes tipos de vaciados a realizar es necesario definir las metodologías y tecnologías para la realización del *software*.

1.5. Metodologías de desarrollo de *software*

Las metodologías de desarrollo de *software* son un conjunto de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto [18].

1.5.1. Proceso Unificado Ágil (AUP)

A pesar de la variedad de metodologías usadas en la UCI, el Proceso Unificado Ágil (AUP) es una versión simplificada del Proceso Unificado de *Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de *software* de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo [19]:

- Desarrollo Dirigido por Pruebas (TDD)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva [20]:

- Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

1.5.2. Fases de la variación AUP-UCI

En la UCI se realizó una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI; metodología que hace uso de buenas prácticas que se centran en el desarrollo de productos y servicios de calidad. De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición), en la variación de la metodología AUP-UCI 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 3 fases de AUP en una sola, a la que se le llama Ejecución y se agrega la fase de Cierre [21]. Para una mayor comprensión se muestra la siguiente Tabla 1 [21].

Tabla 1. Variación de AUP-UCI

Fases AUP	Fases Variación AUP-UCI	Objetivos de las fases (Variación AUP-UCI)
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planificación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
Elaboración Construcción Transición	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el <i>software</i> , incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se

		modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del <i>software</i> .
	Cierre	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Se decide utilizar la variación de AUP-UCI, por ser la que más se adapta al proyecto a desarrollar y a las condiciones de trabajo. Se eligió por ser muy apropiada para guiar proyectos de una complejidad y volumen no muy altos y que necesiten una rápida implementación, los cuales son los aspectos fundamentales a tener en cuenta para el desarrollo del módulo de vaciado de sólidos. Teniendo en cuenta las características de dicha metodología, que como se mencionó anteriormente, incorpora elementos ágiles para el desarrollo a los ya establecidos por RUP, este se convierte en la mejor opción para llevar a cabo el desarrollo del dicho módulo. Esto se debe a que en un equipo de trabajo pequeño formado en su mayoría por desarrolladores, el uso de una metodología ágil ha probado a nivel mundial ser la mejor vía para el desarrollo de un *software* que satisfaga los requisitos del usuario y reúna estándares aceptables de calidad.

1.6. Tecnologías empleadas para la confección del módulo vaciado de sólidos

La tecnología de desarrollo es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos (*software*) [22]. La herramienta AsiXMec tiene definida las tecnologías a emplear para la confección de la misma, por lo cual, el lenguaje de desarrollo y las herramientas a utilizar para desarrollar el módulo de vaciado de sólidos, son las que se fundamentan a continuación.

1.6.1. Biblioteca Open Cascade Community Edition (OCE)

Como principal plataforma de desarrollo se empleará la Tecnología *Open Cascade*, es un *framework* de desarrollo de *software* (SDK), para el desarrollo de aplicaciones que se ocupan de los datos CAD 3D, libremente disponibles en código abierto. Incluye un conjunto de librerías de clases C++ que prestan servicios

para la superficie 3D y modelado de sólidos, visualización, intercambio de datos y el desarrollo rápido de aplicaciones [23].

Se decide la utilización de dicha biblioteca para la realización de la operación de vaciado de sólidos, ya que la misma proporciona un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos), conocidas como API. Entre las API usadas para el desarrollo de la operación de vaciado de sólidos se encuentran:

BrepOffsetAPI_MakeThickSolid: describe las funciones para construir sólidos ahuecados. Construye un hueco sobre el sólido pasado por parámetro, mediante la eliminación del conjunto de caras (*ClosingFaces*) del sólido, donde el parámetro *Offset* define el espesor de las paredes. Su signo indica de qué lado de la superficie del sólido se realizará la forma ahuecada [24].

AIS_Shape: es el objeto interactivo que se utiliza en la mayor parte de las aplicaciones. Empleando funciones estándar disponibles que le permiten preparar las operaciones de selección de los elementos constitutivos en forma de vértices, aristas y caras en un contexto local abierto. Los modos de selección específicos para *Shape* se conocen como Modo de Activación Estándar [25].

BrepGProp: proporciona funciones globales para calcular las propiedades globales de una forma de líneas, superficies o volúmenes, y juntarlas con las propiedades globales ya calculadas para un sistema geométrico [26].

1.6.2. Lenguaje Unificado de Modelado

Se utilizará como lenguaje de modelado UML (acrónimo de *Unified Modeling Language*), es un lenguaje estándar en el análisis y diseño de sistemas de cómputo. Modela sistemas de información y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela. Se ha convertido en el lenguaje estándar de la industria para especificar, visualizar, construir y documentar sistemas de *software* [27].

UML permite representar el sistema gráficamente, de manera que se puedan entender y especificar las características del sistema antes de su construcción. Proporciona un conjunto de diagramas que permiten modelar los diferentes aspectos del sistema y su principal objetivo es la comunicación de información [28].

1.6.3. Herramientas CASE

Las herramientas de Ingeniería de *Software* Asistida por Ordenador (CASE por sus siglas en inglés), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el costo de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida del desarrollo del *software* en tareas como el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores [29].

Se decide utilizar *Visual Paradigm for UML* como herramienta CASE para seguir la política establecida en la universidad para el desarrollo de los productos. Además soporta el ciclo de vida completo de desarrollo de *software*: análisis y diseño orientado a objetivos, construcción, prueba y despliegue. Tiene soporte para los lenguajes de modelado UML 2.1, SysML⁶ y BPMN⁷.

1.6.4. Entorno de Desarrollo Integrado Qt-Creator

El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien se pueden utilizar varios de estos [30].

Qt-Creator es una herramienta para el desarrollo de programas. Es un IDE multiplataforma que se ajusta a las necesidades de los desarrolladores y creado para el desarrollo de aplicaciones con las bibliotecas Qt. Esta poderosa herramienta tiene la ventaja de permitir que un equipo de desarrolladores comparta un proyecto a través de diferentes plataformas de desarrollo (*Windows7* (en adelante) y *Linux*) con una herramienta común para desarrollo y depurado [31].

Teniendo en cuenta que en el proyecto DISEM se utiliza *Qt-Creator 5* como IDE de desarrollo se selecciona el mismo para la realización del módulo de vaciado de sólidos.

⁶ *Systems Modeling Language*

⁷ *Business Process Modeling Notation*

1.6.5. Lenguaje de programación C++

Un lenguaje de programación, es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis, que se pone a disposición del programador para que este pueda comunicarse con los dispositivos de *hardware* y *software* existentes [32].

C++ es un lenguaje de programación multiparadigma y de alto nivel, con el que se puede escribir cualquier programa. Una de las ventajas del C++ sobre otros lenguajes de programación es que soporta diferentes estilos de programación:

- Programación Estructurada: es un paradigma de programación orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora, utilizando únicamente subrutinas y tres estructuras: secuencia, selección (*if* y *switch*) e iteración (bucles *for* y *while*).
- Programación Orientada a Objetos (POO): es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Teniendo en cuenta que en el proyecto DISEM se utiliza C++ como lenguaje de programación se selecciona el mismo para la realización del módulo de vaciado de sólidos.

Conclusiones del capítulo

En este capítulo se realizó un estudio de las operaciones 3D que se emplean en varios de los sistemas CAD existentes en la actualidad, para la creación de estructuras mecánicas. Se determinó que la solución viable para resolver el problema planteado en la presente investigación, es implementar un módulo que permita realizar los diferentes vaciados de sólidos (interior, exterior e híbrido) de forma automática. Se deben controlar los parámetros de dirección, espesor y caras, para reducir los pasos de diseño en la creación de estructuras mecánicas en la herramienta CAD AsiXMec.

Además se fundamentó la selección de las tecnologías y herramientas que se utilizan para el modelado, diseño e implementación de la solución propuesta en la investigación.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En el presente capítulo se presenta la propuesta de solución del módulo para la realización de vaciado de sólidos en estructuras mecánicas. Se realiza la descripción del negocio; las características y funcionalidades que tendrá el mismo a partir de los requerimientos funcionales y no funcionales. Se presenta los diferentes diagramas empleados para representar la solución, las especificaciones de los casos de uso asociados al componente y la trazabilidad de los casos de uso con sus correspondientes requisitos funcionales.

2.1. Propuesta de solución

Se decide crear un módulo que permita realizar el vaciado sobre las estructuras mecánicas creadas en la herramienta AsiXMec. La creación de este módulo permitirá disminuir el flujo de trabajo en la herramienta, reduciendo los pasos de diseño en la creación de estructuras mecánicas.

El módulo tiene como nombre “Operación de vaciado”, el cual permitirá realizar tres tipos de vaciado (interior, exterior e híbrido) sobre las caras seleccionadas en las estructuras mecánicas. Para la realización del módulo propuesto se tiene en cuenta las normas que se establecen en la herramienta AsiXMec, para que exista semejanza en cuanto a la interfaz gráfica del módulo a desarrollar y los demás módulos que la conforman. La arquitectura de este módulo debe ser paramétrica, esto significa que cada una de las partes generadas de la pieza está regida por parámetros que controlan su forma y dimensión. Los parámetros pueden ser modificados y la pieza debe actualizarse a partir de los cambios realizados, se pueden deshacer o rehacer los pasos de diseño realizados y los cambios deben propagarse al resto de las operaciones relacionadas con dicha modificación.

Para el desarrollo del módulo, el estudio de las ecuaciones matemáticas para el cálculo de volumen constituyó parte fundamental de esta investigación, sin embargo no fue necesario hacer uso de las ecuaciones para hallar el volumen de los sólidos, debido a que la biblioteca *Open Cascade* proporciona la API *BrepGProp* que permite determinar el volumen del sólido. El espesor se calcula en la aplicación de dos formas:

- en la primera el usuario proporciona un valor, que representa el espesor que desea proporcionar a las caras resultantes luego de la eliminación de las caras seleccionadas.
- en la segunda el valor proporcionado por el usuario representa un porcentaje. Para determinar el espesor en porcentaje, se empleó la ecuación de proporción directa para tratar el valor que define el

usuario en por ciento (ver Ecuación 8). Donde para determinar este valor, se hace necesario saber el volumen del sólido para determinar un grosor uniforme en todas las paredes que se generan según el volumen del sólido y el valor definido por el usuario.

Para la realización de los vaciados interior y exterior se utilizó la API *BrepOffsetAPI_MakeThickSolid* que permite realizar el vaciado del sólido con los parámetros definidos por el usuario. Para la realización del vaciado híbrido se aplica al sólido el vaciado interior, posteriormente al resultado obtenido se le realiza el vaciado exterior, obteniendo así el vaciado híbrido.

2.2. Modelo de dominio de la operación de vaciado

Los Modelos de Dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de *software* o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir [33].

Para facilitar la comprensión del funcionamiento de la operación de vaciado se explica a continuación el flujo de trabajo del mismo:

- El proceso para realizar la operación de vaciado en la herramienta AsiXMec, comienza cuando se crea una estructura mecánica tridimensional. El ingeniero mecánico utiliza el módulo para realizar la forma ahuecada en el interior de la estructura. En dicho módulo primero se determinan las caras sobre las que se aplicará la operación de vaciado.
- Una vez seleccionadas las caras sobre las que se aplicará la operación, se selecciona el tipo de vaciado a realizar para el desarrollo de la operación.
- Por último se define el espesor deseado por el usuario (ingeniero mecánico) para generar los bordes de las nuevas paredes generadas de la forma ahuecada.

Cabe destacar que el ingeniero mecánico puede alterar el orden de los pasos sin afectar el correcto funcionamiento de la aplicación.

El siguiente diagrama es el Modelo de Dominio, referido al módulo para el vaciado de sólidos en la herramienta AsiXMec (ver Figura 5).



Figura 4. Modelo de dominio para la operación de vaciado

Descripción de los conceptos:

- Ingeniero Mecánico: es el encargado de diseñar las estructuras mecánicas y acceder al módulo para hacer uso de sus funcionalidades.
- Herramienta AsiXMec: es una herramienta utilizada para el modelado de sólidos a partir de las operaciones de diseño que trae consigo.
- Operación de vaciado: son operaciones que permiten modificar una estructura mecánica.
- Estructuras Mecánicas: son todas las estructuras mecánicas que pueden ser creadas o modificadas por el ingeniero mecánico a partir de la operación de vaciado.

2.3. Técnicas empleadas para la captura de requisitos

Para la recopilación y obtención de la información necesaria para la captura de requisitos, se utilizaron las técnicas: observación y tormenta de ideas.

La observación se llevó a cabo para percibir como se desarrolla la operación en varios de los sistemas CAD existentes, lográndose captar directamente las particularidades de los procesos en la operación.

Utilizando la técnica de tormenta de ideas, en conjunto con el tutor, el jefe de proyecto y el personal que trabaja sobre el desarrollo de la herramienta AsiXMec, se acumularon ideas para tener una perspectiva general para desarrollar un módulo que solucione las necesidades del sistema.

2.3.1. Requerimientos funcionales

Los requisitos funcionales son capacidad o condición que el sistema debe cumplir [34].

Como resultado de haber aplicado las técnicas de captura de requerimientos abordadas en el epígrafe anterior

se obtuvo como resultado un total de 3 requisitos funcionales por los que se rige el desarrollo del módulo, los cuales se exponen en la Tabla 2.

Tabla 2. Requerimientos funcionales

Número	Requerimiento	Descripción
RF1	Aplicar el vaciado interior.	Permite realizar el vaciado interior sobre las superficies seleccionadas de la estructura mecánica, creando una estructura ahuecada desde los bordes de la superficie seleccionada hacia el interior, transformando la estructura mecánica.
RF2	Aplicar el vaciado exterior.	Permite realizar el vaciado exterior sobre las superficies seleccionadas de la estructura mecánica, creando una estructura ahuecada desde los bordes de la superficie seleccionada hacia el exterior, transformando la estructura mecánica.
RF3	Aplicar el vaciado híbrido.	Permite realizar el vaciado híbrido sobre la superficie seleccionadas de la estructura mecánica, creando una estructura ahuecada desde los bordes de la superficie seleccionada hacia el interior y el exterior de la superficie selecciona, transformando la estructura mecánica.

2.3.2. Requerimientos no funcionales

Se refiere a aspectos técnicos que debe incluir el sistema y se relacionan con características de restricciones y de calidad que debe cumplir el *software* [34]. A continuación se describen los requisitos no funcionales tomados en cuenta en el presente trabajo:

- Funcionamiento:

RNF 1. Memoria RAM superior a un 1 GB.

RNF 2. CPU *Pentium* IV 3.0GHz o superior.

RNF 3. Sistema Operativo GNU/Linux.

- Restricciones del diseño

RNF 1. Uso de la metodología Proceso Unificado Ágil (AUP-UCI).

El análisis y diseño del componente será basado en la metodología AUP-UCI.

RNF 2. Uso del lenguaje de modelado UML 2.1.

El análisis y diseño del componente usará el lenguaje de modelado UML en la versión 2.1.

RNF 3. Uso de *Visual Paradigm for UML 8.0* como herramienta CASE.

Se utiliza como herramienta CASE *Visual Paradigm for UML 8.0*.

RNF 4. Se usa como lenguaje de programación C++.

Se emplea el lenguaje de programación C++ bajo el paradigma de Programación Orientada a Objetos.

RNF 5. Como IDE de desarrollo *Qt-Creator 5*.

Se utiliza como IDE de desarrollo *Qt-Creator 5*, para la realización de la operación de vaciado de sólidos.

2.4. Casos de Uso del Sistema

Luego de determinar los requerimientos funcionales, éstos se agruparon en casos de uso del sistema. Un caso de uso especifica una secuencia de acciones, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. Los Diagramas de Casos de Uso especifican las funcionalidades y el comportamiento de un sistema mediante la interacción con los usuarios y/u otros sistemas, muestra la relación entre los actores y los casos de uso [35].

2.4.1. Definición de los actores del sistema

Un actor es cualquier usuario que interactúa con el sistema, es decir, aquella persona o sistema que interactúa con él [36]. En este caso en particular quien usará el sistema será el ingeniero mecánico.

A continuación se muestra una tabla con la determinación del actor del sistema y su descripción.

Tabla 3. Determinación y descripción de los actores del sistema

Actores	Descripción
Ingeniero mecánico	Interactúa con el módulo. Diseña estructuras mecánicas ordenando posteriormente la ejecución del módulo para realizar vaciados a las mismas.

2.4.2. Diagrama de Casos de Uso del Sistema

El Diagrama de Casos de Uso del Sistema es la representación de las acciones que el sistema ejecuta y produce un resultado observable para un actor [37]. Es decir, son funcionalidades que el sistema brinda a los actores que interactúan con el mismo.

A continuación se muestra el Diagrama de Casos de Uso del Sistema, quedando representadas las funcionalidades del sistema y su interacción con el actor (ver Figura 6).

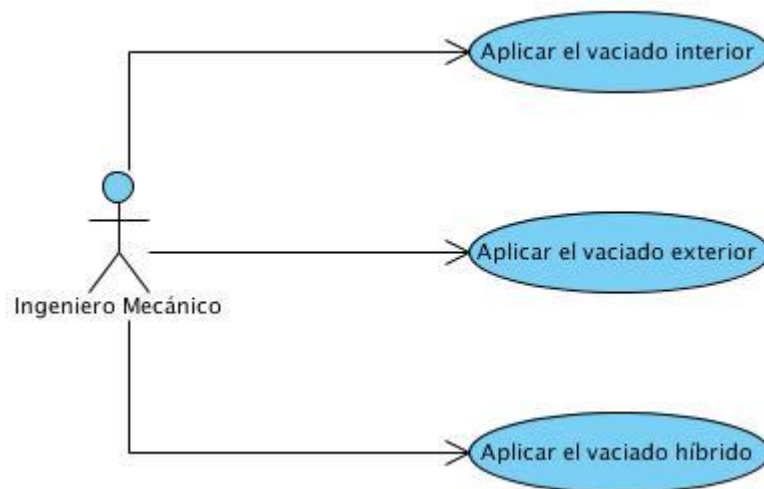


Figura 5. Diagrama de Casos de Uso del Sistema

2.4.3. Descripción de los Casos de Uso

La descripción de los Casos de Uso del Sistema especifica las interacciones del actor con el sistema, describe el flujo de actividades que realiza el actor al hacer uso del sistema y las correspondientes respuestas del mismo.

A continuación se muestra la descripción del caso de uso Aplicar el vaciado interior presente en el Diagrama de Casos de Uso del Sistema. Las descripciones de los demás casos de uso presentes en el Diagrama de Casos de Uso del Sistema se encuentran en el [Anexo 1](#).

Tabla 4. Descripción del CU-1_Aplicar el vaciado interior

Caso de uso:	Aplicar el vaciado interior	
Objetivo:	Permitir realizar sobre la cara seleccionada de la estructura mecánica el vaciado interior, determinando también el espesor de los bordes.	
Actores:	Ingeniero mecánico.	
Resumen:	El caso de uso se inicia cuando el ingeniero mecánico realiza un vaciado de sólido con un grosor determinado a una estructura mecánica.	
Complejidad:	Alta	
Prioridad:	Crítica	
Referencias:	RF1	
Precondiciones:	Debe existir al menos una estructura mecánica en el visor.	
Postcondiciones:	Se aplicó el vaciado interior a una estructura mecánica.	
Flujo de Eventos		
Flujo básico Aplicar el vaciado		
Acción del Actor:	Respuesta del Sistema:	
1.	El especialista selecciona la operación de vaciado.	1.1 Muestra el diálogo para la funcionalidad de vaciado.
2.	Selecciona la operación de vaciado interior.	
3.	Selecciona una cara del sólido.	3.1 Previsualiza el resultado de la operación.
4.	Selecciona el tipo de espesor: <ul style="list-style-type: none"> • Si es el primer espesor define el valor deseado en unidades de 	4.1 El sistema valida el espesor dado por el usuario y previsualiza el resultado de la operación según el nuevo valor del tipo de espesor seleccionado en caso de que sea

	<p>milímetro.</p> <ul style="list-style-type: none"> • Si es el segundo espesor define el valor deseado en porciento. 	correcto.
5.	<p>Selecciona la opción Hacer.</p> <ul style="list-style-type: none"> • Si selecciona la opción Cancelar ver el Flujo Alterno 2 Selecciona la opción Cancelar. 	5.1 Visualiza los resultados de la operación.
		Termina el caso de uso.
Flujo Alterno de Eventos		
Nº Evento 2 Selecciona la opción Cancelar.		
Acción del Actor:		Respuesta del Sistema:
1.	Selecciona la opción Cancelar del diálogo de Vaciado.	1.1 Se cierra el diálogo. No se ejecuta la operación.
		Termina el caso de uso.
Flujo Alterno de Eventos		
Nº Evento 2.1 Espesor invalido.		
Acción del Actor:		Respuesta del Sistema:
1.		<p>1.1 Valida la expresión teniendo en cuenta las siguientes condiciones:</p> <ol style="list-style-type: none"> 1. Espesor menor que 0. Muestra un mensaje de error. 2. Espesor igual a cero o mayor que 100 000: <p>Muestra el mensaje de error.</p> <ol style="list-style-type: none"> 1. Parámetros inválidos (no están definidos en el Editor de Parámetros): <p>Muestra mensaje de error.</p>
2.	Presiona el botón Aceptar.	2.1 Regresa al paso 1 del Flujo Básico Vaciado interior.

Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos no funcionales		No aplicable
Asuntos pendientes		No aplicable

2.5. Patrón arquitectónico

Los patrones arquitectónicos son los que definen la estructura de un sistema, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema [38].

2.5.1. Arquitectura de software para el desarrollo del módulo

La arquitectura de *software* representa la estructura o estructuras del sistema que consiste en componentes de *software*, las propiedades externas visibles de esos componentes y las relaciones entre ellos. Para el desarrollo del módulo se decide utilizar la arquitectura de *software* que está definida para la herramienta: arquitectura basada en capas, apoyada en la utilización de *plugins*⁸ para extender las funcionalidades del *software*. Donde cada módulo que contiene la herramienta, está conformado por tres capas diferentes, por un lado definen una capa de presentación para la interacción del usuario con el sistema, una segunda capa controladora que hace función de puente para controlar el flujo de trabajo y distribuir los datos recibidos por la capa de presentación, y por otro lado una capa de datos que recibe los datos enviados por la capa controladora para procesar la lógica de los datos.

El patrón arquitectónico por capas propone la construcción de tres capas, donde cada capa descansa sobre la inferior. En este esquema la capa más alta utiliza varios servicios definidos por la inferior, pero la última es inconsciente de la superior. Además, normalmente cada capa oculta las capas inferiores de las siguientes superiores a esta.

Funcionamiento del patrón arquitectónico por capas para la estructura del módulo de vaciado de sólidos:

⁸ Aplicaciones funcionales que puede anexarse a un programa para aumentar sus funcionalidades.

- La capa presentación: es con la que el ingeniero mecánico interactúa a través de la interfaz gráfica del módulo. La cual contiene las clases encargadas de capturar los nuevos datos que define el usuario para que sean reenviados al controlador.
- La capa controladora: es la que contiene la clase controladora, que recibe las solicitudes requeridas por los ingenieros mecánicos a través de la interfaz gráfica del módulo. Es la capa donde se maneja el flujo de negocio, enviando los datos recibidos a las clases que se encuentran en la capa de datos.
- La capa de datos: en esta capa se encuentran las clases que reciben los datos que provienen de la controladora para realizar la operación, devolviendo el resultado según los datos definidos por el usuario.

A continuación se muestra la arquitectura de *software* utilizada para el módulo desarrollado (ver Figura 7):

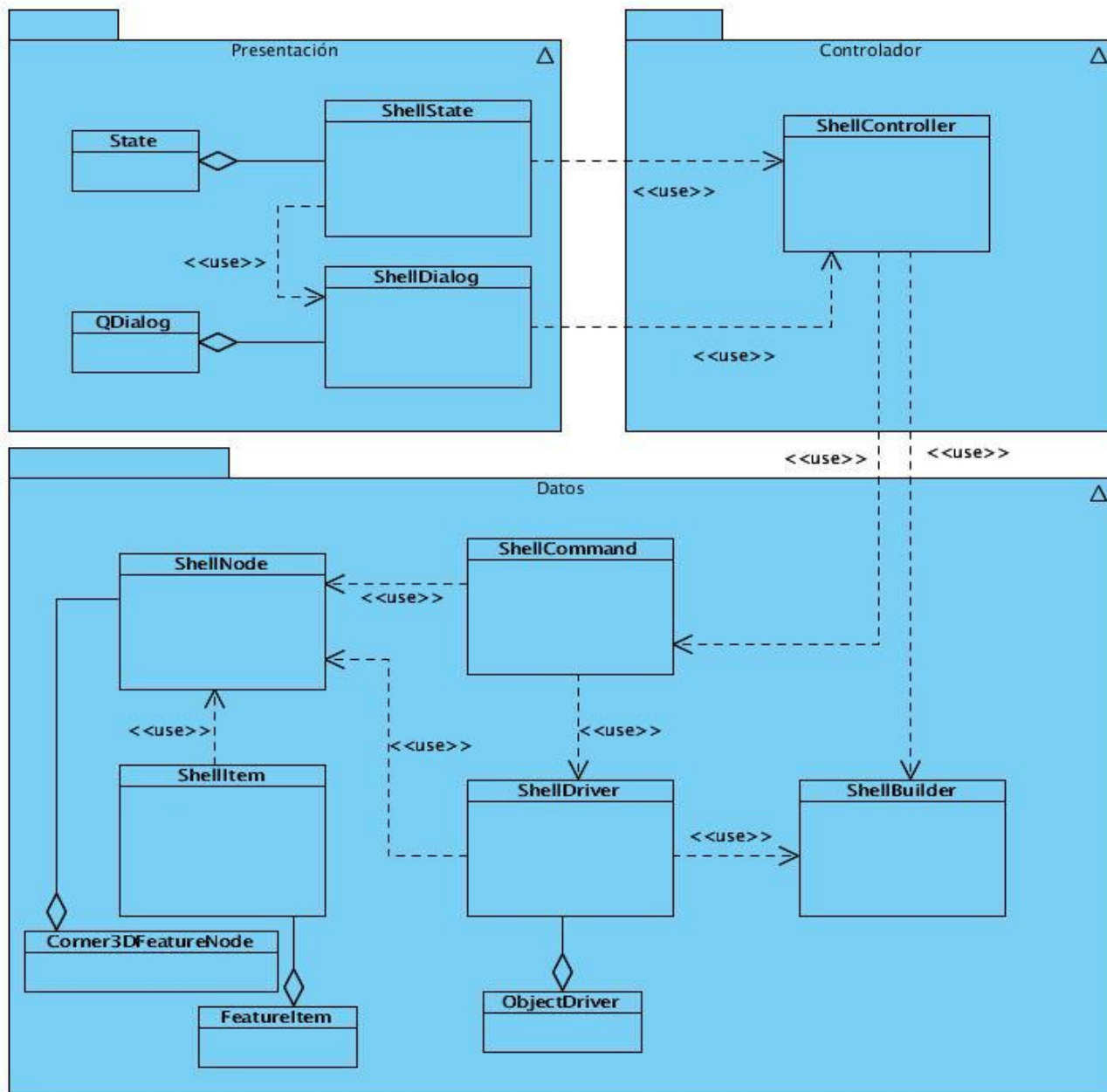


Figura 6. Arquitectura del módulo desarrollado

2.6. Modelo de diseño

El modelo de diseño es un modelo de objeto que describe la relación física de los casos de uso centrándose en los requisitos funcionales y no funcionales, relacionados con el entorno de implementación. Sirve de abstracción a la implementación del sistema y es utilizado como una entrada fundamental de las actividades de la implementación y debe ser mantenido sobre todo el ciclo de vida del *software* [39].

2.6.1. Diagrama de paquetes

El objetivo de estos diagramas es obtener una visión más clara del sistema de información orientado a objetos, organizándolo en subsistemas, agrupando los elementos del análisis, diseño o construcción y detallando las relaciones de dependencia entre ellos. El mecanismo de agrupación se denomina Paquete [40].

Estrictamente hablando, los paquetes y sus dependencias son elementos de los diagramas de casos de uso, de clases y de componentes, por lo que se podría decir que el diagrama de paquetes es una extensión de éstos. El diagrama de paquetes es tratado como una técnica aparte, que se aplica en el análisis para la agrupación de casos de uso o de clases de análisis, en el diseño de la arquitectura para la agrupación de clases de diseño y en el diseño detallado para agrupar componentes [40].

Descripción de los paquetes que conforman el módulo desarrollado (ver Figura 8):

Paquete de Presentación: en este paquete se encuentran las clases que manejan la parte visual del módulo, donde el especialista puede interactuar con el módulo para definir los datos y realizar la operación de vaciado.

Paquete de Negocio o Controlador: en el paquete de negocio se encuentra la clase controladora principal, la que maneja la lógica del negocio. Recibiendo los datos que fueron definidos por el usuario para redireccionarlos a las clases encargadas de realizar la operación según sus funcionalidades en la solución.

Clases que se encuentran en el Paquete de datos.

Paquete de Datos: en este paquete se encuentran las clases encargadas de realizar la operación según los datos recibidos por la clase controladora.

A continuación se muestra el diagrama de paquetes del módulo desarrollado según la arquitectura de desarrollo utilizada:

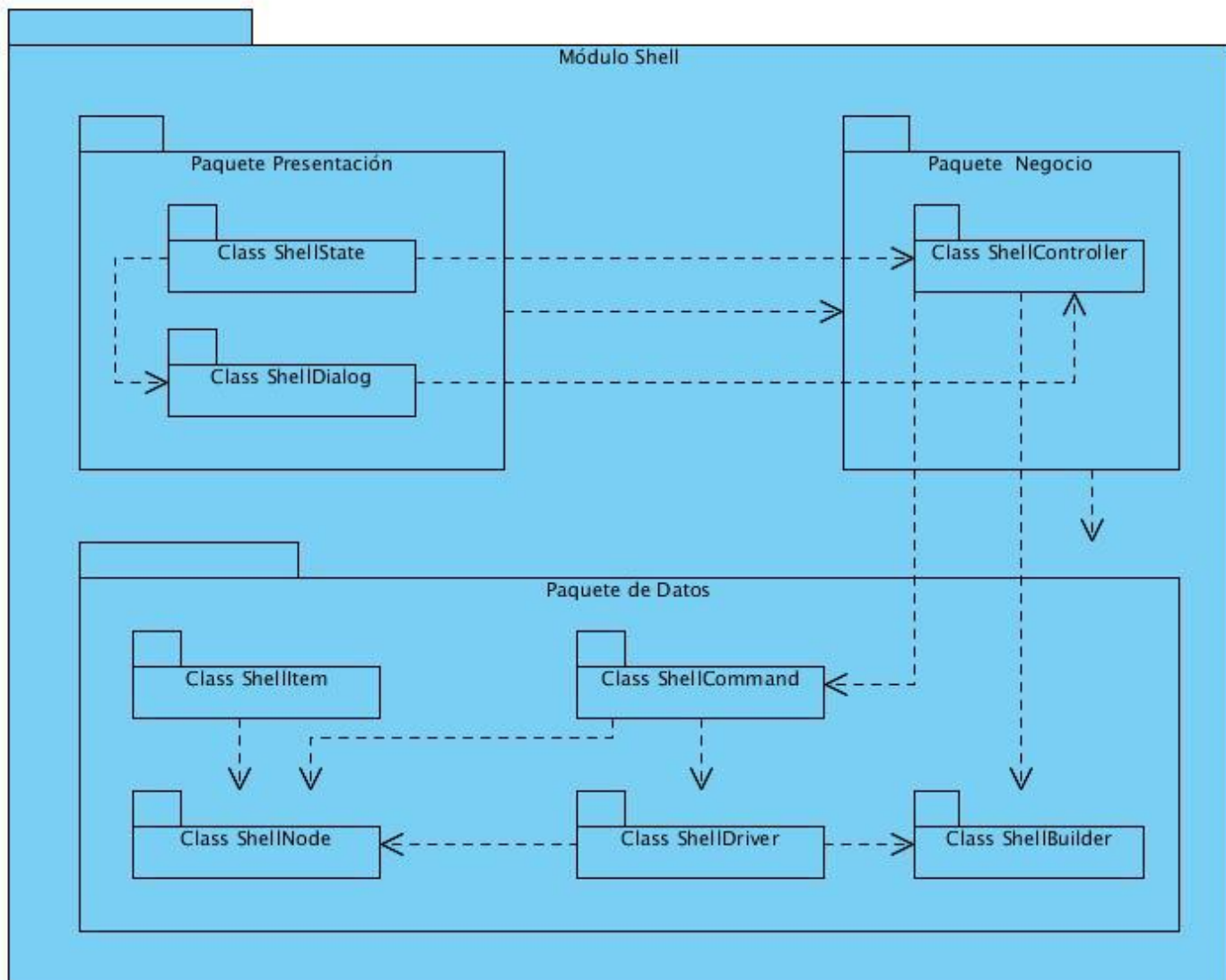


Figura 7. Diagrama de paquetes correspondiente a la operación de vaciado de sólidos

2.6.2. Diagrama de clases del diseño

Un Diagrama de Clases del Diseño representa las clases del sistema con sus relaciones. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal [41]. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones (ver Figura 9).

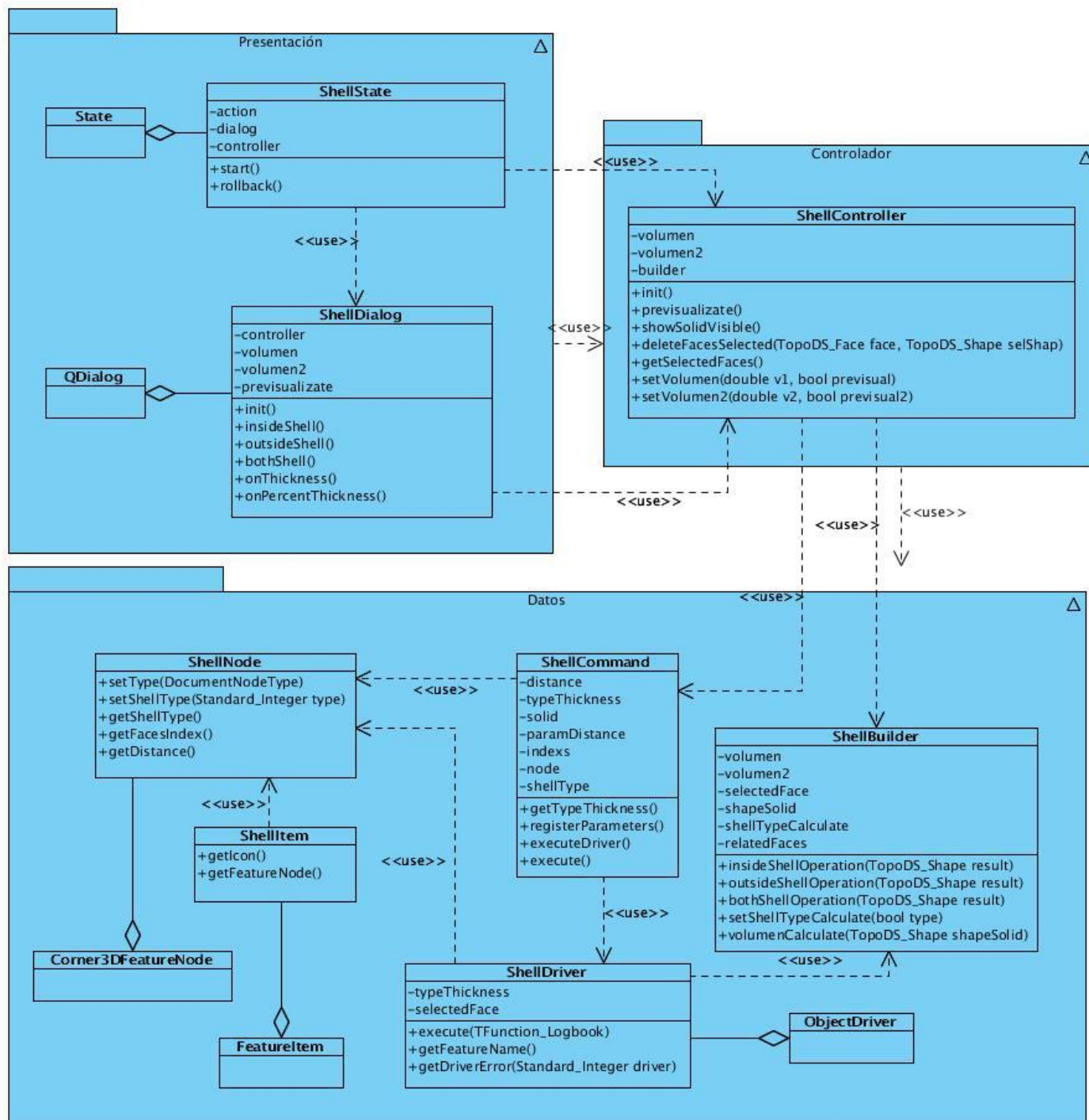


Figura 8. Diagrama de Clases del diseño correspondiente a la operación de vaciado de sólidos

A continuación se describe un conjunto de clases usadas en la solución del módulo desarrollado, las restantes descripciones pueden ser vistas en el [Anexo 2](#).

Tabla 5. Descripción de la clase ShellController

Descripción de la clase ShellController	
Nombre: <i>ShellController</i>	
Tipo de clase: Controladora	
Resumen: Es la clase controladora principal, la cual es la encargada de controlar el flujo del negocio. La misma redirecciona los datos proveniente de las clases <i>ShellState</i> y <i>ShellDialog</i> , correspondiente a cada caso de uso. Además de manejar la previsualización a partir de los parámetros definidos por el usuario.	
Atributo:	Tipo:
<i>volumen</i>	<i>Standard_Real</i>
<i>volumen2</i>	<i>Standard_Real</i>
<i>builder</i>	<i>ShellBuilder</i>
Responsabilidades	
Nombre:	<i>init()</i>
Descripción:	Funcionalidad encargada de inicializar las acciones de acuerdo a los datos recibidos por la clase <i>ShellDialog</i> .
Nombre:	<i>previsualizate()</i>
Descripción:	Funcionalidad encargada de previsualizar el resultado esperado según los datos recibidos por la clase <i>ShellDialog</i> .
Nombre:	<i>setVolumen(double v1, bool previsual)</i>
Descripción:	Funcionalidad encargada de modificar el valor del volumen y previsualizar el resultado según el valor del volumen.
Nombre:	<i>setVolumen2(double v2, bool previsual2)</i>

Descripción:	Funcionalidad encargada de modificar el valor del volumen2 y previsualizar el resultado según el valor del volumen2.
Nombre:	<i>getSelectedFaces()</i>
Descripción:	Funcionalidad que devuelve una lista que contiene todas las caras seleccionadas.
Nombre:	<i>setShellType(SHELL_TYPE type)</i>
Descripción:	Funcionalidad encargada de modificar el tipo de vaciado que se quiere aplicar.
Nombre:	<i>getShellType()</i>
Descripción:	Funcionalidad encargada de devolver el tipo de vaciado que se va aplicar.
Nombre:	<i>selectFaces()</i>
Descripción:	Funcionalidad encargada de previsualizar el resultado según las caras seleccionadas del sólido sobre el que se esté aplicando la operación.
Nombre:	<i>deleteFacesSelected(TopoDS_Face face, TopoDS_Shape selShap)</i>
Descripción:	Funcionalidad encargada de deseleccionar las caras que fueron seleccionadas anteriormente en el sólido.
Nombre:	<i>setExpForDistance(TCollection_ExtendedString volumen)</i>
Descripción:	Funcionalidad encargada de modificar el valor del volumen en el editor de parámetros de la herramienta AsiXMec.
Nombre:	<i>setExpForDistance2(TCollection_ExtendedString volumen2)</i>
Descripción:	Funcionalidad encargada de modificar el valor del volumen2 en el editor de parámetros de la herramienta AsiXMec.
Nombre:	<i>getExpForDistance()</i>
Descripción:	Funcionalidad encargada de devolver el valor del tipo de volumen que se maneja en el editor de parámetros de la herramienta AsiXMec.
Nombre:	<i>getFacesIndex(TopoDS_Shape solid, vector<TopoDS_Face> faceArray)</i>

Descripción:	Funcionalidad encargada de devolver un arreglo que contiene los índices (valores enteros) de las caras de un sólido.
Nombre:	<i>showSolidVisible()</i>
Descripción:	Funcionalidad encargada de mostrar el sólido sobre el que se aplicara la operación de vaciado cuando existan varios sólidos en el visor.
Nombre:	<i>setTypeThickness(bool type)</i>
Descripción:	Funcionalidad encargada de modificar el tipo de espesor con el que se quiere trabajar.

Tabla 6. Descripción de la clase ShellBuilder

Descripción de la clase ShellBuilder	
Nombre: <i>ShellBuilder</i>	
Tipo de clase: Modelo	
Resumen: Es la clase que recibe la lógica del negocio, la cual es la encargada de realizar la operación de vaciado para darle respuestas a los datos proveniente de la clase controladora principal.	
Atributo:	Tipo:
<i>volumen</i>	<i>Standard_Real</i>
<i>volumen2</i>	<i>Standard_Real</i>
<i>relatedFaces</i>	<i>vector<TopoDS_Face></i>
<i>shapeSolid</i>	<i>TopoDS_Shape</i>
<i>selectedFace</i>	<i>TopTools_ListOfShape</i>
<i>shellTypeCalculate</i>	<i>bool</i>
Responsabilidades	
Nombre:	<i>volumenCalculate(TopoDS_Shape shapeSolid)</i>
Descripción:	Funcionalidad encargada de calcular el volumen de un sólido pasado por parámetro.

Nombre:	<i>insideShellOperation(TopoDS_Shape& result)</i>
Descripción:	Funcionalidad encargada de realizar la operación de vaciado desde el interior de los bordes de la cara seleccionada.
Nombre:	<i>outsideShellOperation(TopoDS_Shape &result)</i>
Descripción:	Funcionalidad encargada de realizar la operación de vaciado desde el exterior de los bordes de la cara seleccionada.
Nombre:	<i>bothShellOperation(TopoDS_Shape &result)</i>
Descripción:	Funcionalidad encargada de realizar la operación de vaciado en ambos sentidos de los bordes de la cara seleccionada.
Nombre:	<i>setShellTypeCalculate(bool type)</i>
Descripción:	Funcionalidad encargada de modificar el tipo de espesor a emplear para realizar la operación de vaciado.

Tabla 7. Descripción de la clase ShellDriver

Descripción de la clase ShellDriver	
Nombre: <i>ShellDriver</i>	
Tipo de clase: Modelo	
Resumen: Es la clase donde se maneja la operación según los parámetros definidos por el usuario, provenientes de la clase controladora principal.	
Atributo:	Tipo:
<i>typeThickness</i>	<i>bool</i>
<i>selectedFace</i>	<i>TopTools_ListOfShape</i>
Responsabilidades	
Nombre:	<i>execute(TFunction_Logbook & /*log*/)</i>

Descripción:	Funcionalidad encargada de adquirir los datos necesarios para la realización del vaciado y realizar el vaciado.
Nombre:	<i>getDriverError(Standard_Integer driver)</i>
Descripción:	Funcionalidad encargada de manejar el tratamiento de los mensajes de error.
Nombre:	<i>getFeatureName()</i>
Descripción:	Funcionalidad encargada de agregar la operación de vaciado dentro el historial de operaciones del árbol de OCAF.

Tabla 8. Descripción de la clase ShellCommand

Descripción de la clase ShellCommand	
Nombre: <i>ShellCommand</i>	
Tipo de clase: Modelo	
Resumen: Es la clase que se encarga de contener todos los datos necesarios para realizar la operación de vaciado.	
Atributo:	Tipo:
<i>distance</i>	<i>Standard_Real</i>
<i>typeThickness</i>	<i>bool</i>
<i>indexs</i>	<i>vector<Standard_Integer></i>
<i>shellType</i>	<i>Standard_Integer</i>
<i>selectedFace</i>	<i>TopTools_ListOfShape</i>
<i>solid</i>	<i>SolidNode</i>
<i>node</i>	<i>ShellNode</i>
<i>paramDistance</i>	<i>ParameterNode</i>
Responsabilidades	

Nombre:	<i>getTypeThickness()</i>
Descripción:	Funcionalidad encargada de devolver el tipo de espesor con el que se desea aplicar la operación de vaciado.
Nombre:	<i>registerParameters()</i>
Descripción:	Funcionalidad encargada de registrar los parámetros.
Nombre:	<i>executeDriver()</i>
Descripción:	Funcionalidad encargada de llamar driver para ejecutar la operación según los datos recibidos.
Nombre:	<i>execute()</i>
Descripción:	Funcionalidad encargada de crear las características del nodo y registrar los datos para mandar a ejecutar al <i>driver</i> .

Para la modelación de la solución propuesta se propone el uso de los patrones de diseño, con el propósito de optimizar recursos y organizar el código.

2.7. Patrones de diseño

Los patrones ayudan a capturar conocimiento y a crear un vocabulario técnico, hacen el diseño orientado a objetos más flexible, elegante y en algunos casos reusable. En la implementación desarrollada para la operación de vaciado se usan los patrones generales de *software* para asignación de responsabilidades (GRASP) los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos.

Patrones GRASP

- **Experto:** Este patrón se pone en evidencia en la clase *ShellController* que contiene toda la información necesaria recibida por la clase *ShellDialog*. Enviando los datos necesarios a las clases encargadas de realizar la operación de vaciado.
- **Creador:** Con el uso de este patrón las clases tienen la información necesaria para realizar la operación de vaciado. Dicho patrón usa directamente las instancias creadas del objeto y almacena o maneja varias instancias de la clase. Este patrón se evidencia en la clase *ShellDriver* y *ShellBuilder*.

- **Alta Cohesión:** Con el uso de este patrón se permite asignarle a cada clase de la solución, una labor única dentro del sistema, no desempeñada por el resto de los elementos, logrando más facilidad a la hora de la reutilización.

Conclusiones del capítulo

En el capítulo, se describió el negocio. Se especificaron los requisitos funcionales y no funcionales. Se identificaron y describieron los casos de uso del módulo así como el modelado del diagrama de casos de uso que muestra las principales funcionalidades del componente. Se mostraron los diagramas de clases del diseño. También se realizó la propuesta de solución al problema planteado a partir de la descripción de las clases implementadas, las API utilizadas y las consideraciones necesarias para el correcto funcionamiento de las mismas.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS

El presente capítulo resumirá el conjunto de pruebas realizadas en el módulo desarrollado para comprobar la satisfacción de los requisitos funcionales y no funcionales asociados al mismo. Abordando los métodos y los niveles de pruebas existentes, logrando el análisis y la evaluación de los resultados obtenidos de dicho módulo.

3.1. Estándar de codificación

Los estándares de codificación son reglas específicas de cada lenguaje de programación, cuyo cumplimiento reduce de forma significativa, la posibilidad de cometer errores no detectados por los compiladores. Al implementar y verificar el cumplimiento de estos estándares de codificación, se evitan los errores en la introducción del código, reduciendo el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismos [42]. Para el desarrollo del módulo de vaciado se decide seguir el estándar de codificación definidos en el desarrollo de la herramienta AsiXMec.

3.2. API utilizadas de la Biblioteca *Open Cascade*

Para la realización de los algoritmos y las clases propuestas como solución al problema planteado en el presente trabajo se usaron dentro de las API con que cuenta *Open Cascade* dos funcionalidades fundamentales:

- *BrepGProp*, determina el volumen de una superficie según el sólido pasado por parámetro.
- *BrepOffsetAPI_MakeThickSolid*, se empleó para realizar la operación de vaciado. Está permite pasarle por parámetro el sólido al que se le aplicará la operación, las caras que fueron seleccionadas y el tipo de espesor deseado para el grosor de las nuevas paredes que se generan. Donde el signo de dicho espesor determina el tipo de vaciado que se desee realizar, y la combinación de ambos genera el vaciado híbrido. En el caso del vaciado exterior se acoto el valor máximo (igual a 100000) para garantizar el correcto funcionamiento de la API.

3.3. Diagrama de componentes

Los diagramas de componentes describen la descomposición física de los elementos de un sistema (módulo, base de datos, programa ejecutable, entre otros) y sus relaciones. Muestran las opciones de realización

incluyendo código fuente, binario y ejecutable, pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente, entre otros [43].

En el siguiente diagrama de componentes, se muestra la relación que existe entre las principales clases que contiene el módulo desarrollado (ver Figura 10).

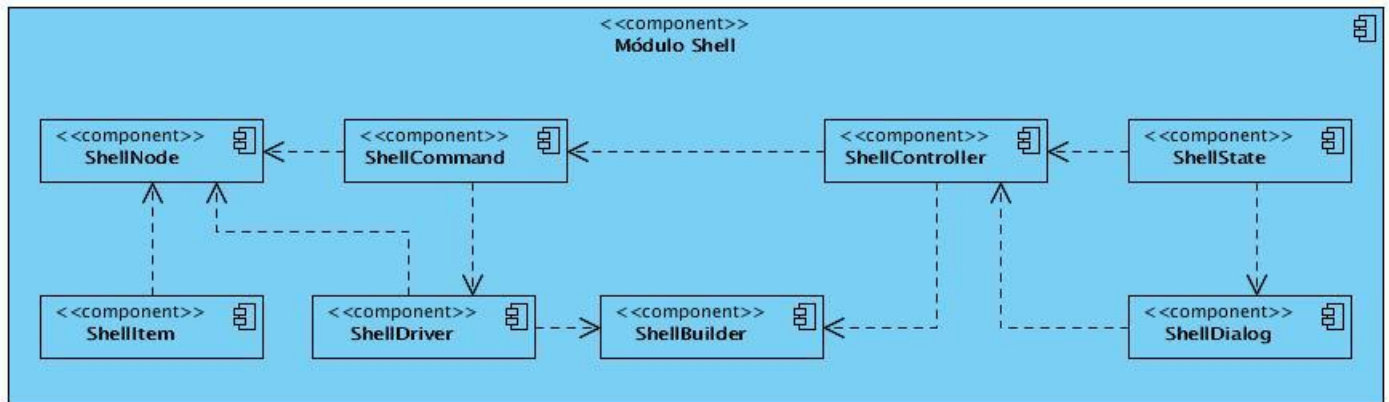


Figura 9. Diagrama de componentes genérico del módulo de vaciado de sólidos

A continuación se describen los componentes más importantes que se relacionan con la operación de *vaciado de sólidos*:

- *ShellController*: es el componente que contiene la clase controladora principal, la cual es encargada de redireccionar los datos proveniente de las clases *ShellState* y *ShellDialog*, correspondiente a cada caso de uso. Maneja la previsualización a partir de los parámetros definidos por el usuario.
- *ShellBuilder*: es el componente que contiene la clase de la lógica del negocio, la cual es la encargada de realizar las operaciones de vaciado para darle respuestas a los datos provenientes de la clase controladora principal.
- *ShellState*: es el componente que contiene la clase que maneja el estado de la operación, la cual es la encargada de redireccionar el estado de los datos de la operación de vaciado a la clase controladora principal.
- *ShellDialog*: es el componente que contiene la clase que maneja los datos de la operación, la cual es la encargada de redireccionar los datos de la operación a la clase controladora principal.
- *ShellCommand*: componente que contiene la clase que se encarga de contener todos los datos

necesarios para realizar la operación.

- *ShellDriver*: es el componente que contiene la clase donde se maneja la operación según los parámetros definidos por el usuario.
- *ShellNode*: es el componente que registra los datos en el árbol de OCAF accedido por la clase *ShellDriver* para la realización de la operación.
- *ShellItem*: es el componente que define qué operación debe mostrar el *object browser* como realizada.

3.4. Métodos de pruebas

Los métodos de pruebas son elementos que garantizan la calidad del *software* sometiéndose a una serie de acciones que verifiquen si cumple con los requerimientos definidos. Estas se realizan con el fin de emitir una evaluación de un producto desde un punto de vista crítico. Los métodos de pruebas se clasifican en:

Pruebas de Caja Negra: pruebas que se llevan a cabo sobre la interfaz del *software*. El objetivo es demostrar que las funciones del *software* son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene, sin tener en cuenta su funcionamiento interno. Se encargan de saber qué es lo que hace el *software*, pero sin dar importancia a cómo lo hace [44].

Pruebas de Caja Blanca: se denomina pruebas de caja blanca a un tipo de prueba de *software* que se realiza sobre las funciones internas de un módulo. Así como las pruebas de caja negra ejercitan los requisitos funcionales desde el exterior del módulo, las de caja blanca están dirigidas a las funciones internas. Las pruebas de caja blanca se llevan a cabo en primer lugar, sobre un módulo concreto, para luego realizar las de caja negra sobre varios subsistemas [44].

3.4.1. Niveles de prueba

Las pruebas pueden agruparse en varios niveles de pruebas tales como:

Pruebas Unitarias: comienzan con la prueba de cada módulo. Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los estos funcione correctamente por separado. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el fragmento de código debe

satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas: fomentan el cambio, simplifican la integración, documentan el código, separan la interfaz del código y hacen que los errores estén más acotados y sean fáciles de localizar [45].

Pruebas de Integración: a partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de *software* una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes de *software* funcionan juntos [45].

Prueba del Sistema: el *software* ensamblado totalmente con cualquier componente *hardware* que requiera, se prueba para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de *software* completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento [45].

Pruebas de Aceptación: estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado [45].

3.4.2. Análisis de las pruebas realizadas

Para verificar que todos los requisitos fuesen implementados correctamente se decide realizar sobre las funciones del módulo desarrollado las pruebas de caja negra como una de los métodos de pruebas. Entre los niveles de pruebas se decide seleccionar pruebas de aceptación. Los casos de pruebas resultantes estarán dirigidos a evidenciar la integración de cada componente y el funcionamiento del sistema como un todo. A la solución propuesta se le realizaron pruebas de funcionalidad ya que permitieron validar la implementación de funciones, métodos y casos de uso, así como verificar su integración y disponibilidad en la herramienta AsiXMec.

A continuación se muestra una tabla con la estrategia de prueba diseñada:

Tabla 9. Estrategia de prueba diseñada

Niveles de prueba	Tipos de prueba	Métodos de prueba
Aceptación	Funcionalidad	Caja negra

3.4.3. Pruebas realizadas al módulo de vaciado de sólidos

Tras aplicar la estrategia de prueba diseñada anteriormente y concluida la primera etapa de implementación del módulo desarrollado, se procede a realizar las pruebas. Para el inicio de la validación de la solución propuesta, se estructuraron los casos de prueba a partir de los cuales se chequeó el cumplimiento de los requisitos funcionales del *software*.

Se entiende por Caso de Prueba, según la Ingeniería del *software*, al conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio [46].

Con la realización de los casos de pruebas realizados se obtuvieron resultados satisfactorios, que validan en cierta medida la calidad del módulo desarrollado. A continuación se muestra el caso de prueba realizado al caso de uso “Aplicar el vaciado interior”. Los demás casos de pruebas elaborados se encuentran en el [Anexo 3](#).

Caso de prueba utilizando técnicas de funcionalidad.

Tabla 10. Caso de prueba basado en el CU-1_Aplicar el vaciado interior

Escenarios	Descripción	Espesor	Espesor por por ciento	Respuesta del sistema	Flujo central
EC 1: Aplicar el vaciado interior.	1.1. El especialista selecciona la Operación de Vaciado. - Vaciado interior 1.3. Selecciona la cara sobre la que desea realizar el vaciado.	20	20	1.2. Muestra el diálogo para la funcionalidad Vaciado. 1.4. Previsualiza el resultado de la operación.	Cinta de opciones

	<p>1.5. Especifica la distancia del espesor deseada mediante un número.</p> <p>1.8. Selecciona la opción Hacer.</p> <p>- Si selecciona la opción Cancelar ver EC 3.1 Selecciona la opción Cancelar.</p>			<p>1.6. Previsualiza el resultado de la operación.</p> <p>1.7. Valida el valor de la distancia. Ver EC 2.1 Distancia inválida.</p> <p>1.9. Crea y visualiza el vaciado teniendo en cuenta los valores especificados por el usuario.</p>	
EC 2.1: Distancia inválida	2.2. Presiona el botón aceptar.	150000	150000	2.2. Valida el espesor teniendo en cuenta las siguientes condiciones:	
		0	0	<p>1. Distancia menor que cero: Muestra el mensaje de error.</p> <p>2. Distancia igual a cero o mayor que 100 000: Muestra el mensaje de error.</p> <p>3. Parámetros inválidos (no están definidos en el Editor de Parámetros): Muestra mensaje de error.</p>	

El resultado de las pruebas realizadas constató que el módulo desarrollado cumple con los requisitos funcionales definidos en el Capítulo 2. Dicho resultado se observa en la siguiente figura.

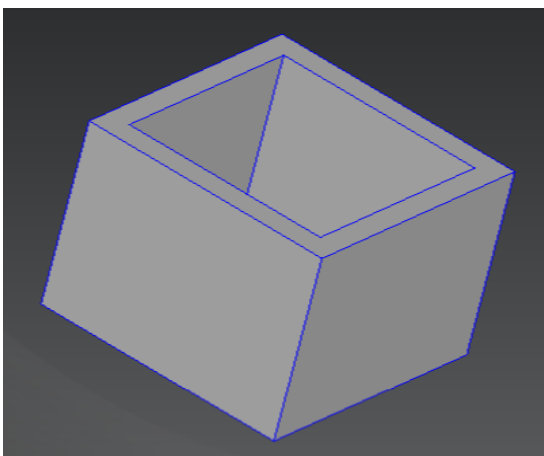


Figura 10. Aplicación del vaciado interior en la herramienta AsiXMec

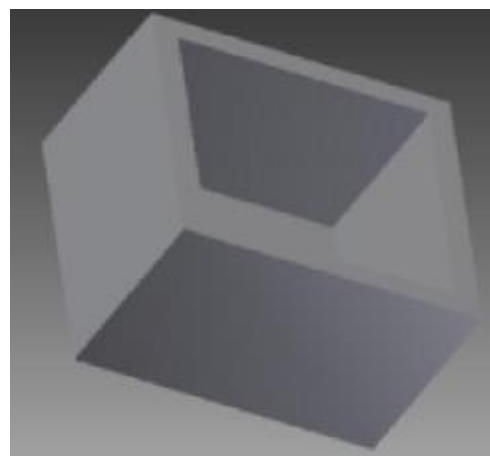


Figura 11. Aplicación del vaciado interior en la herramienta Inventor

Se decide realizar en la herramienta CAD *Autodesk Inventor* la aplicación de la operación de vaciado a una estructura mecánica, obteniendo resultados similares. Según los resultados obtenidos se puede concluir que las pruebas fueron satisfactorias. Durante las pruebas se documentaron las no conformidades surgidas dándole la categoría de Alta, Media o Baja según su impacto en el negocio y el sistema.

Las no conformidades encontradas durante las tres iteraciones estuvieron estrechamente relacionadas con las funciones incorrectas del módulo. El resto de las no conformidades estuvieron vinculadas a los elementos del diseño de la interfaz y la arquitectura de la información en el *software*. Para eliminar las causas de las no conformidades detectadas se toman acciones correctivas, en el caso de que la no conformidad sea potencial se llevarán a cabo acciones preventivas.

A continuación se muestra el comportamiento del módulo desarrollado a lo largo de cada ciclo de pruebas según las no conformidades detectadas por las pruebas realizadas por varios integrantes del proyecto DISEM.

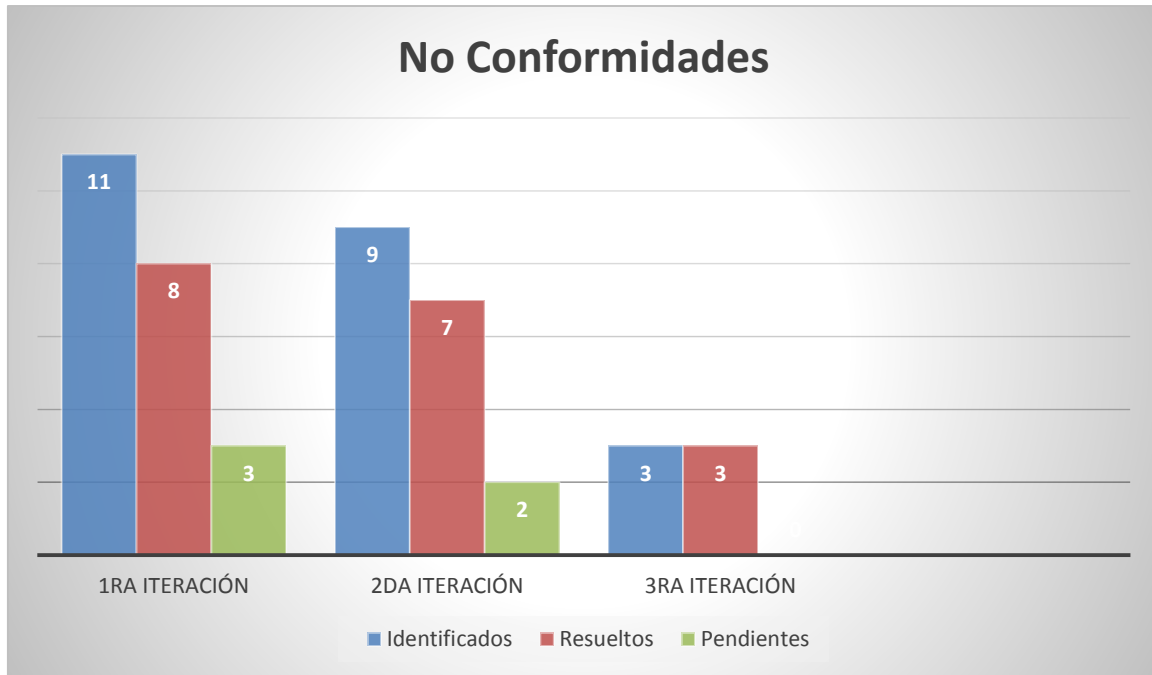


Figura 12. Resultados de las pruebas

Conclusiones del capítulo

En el presente capítulo se presentó el diagrama de componentes, el cual refleja el resultado del desarrollo del módulo, los elementos que componen el sistema, la interacción entre ellos y se describieron cada uno de los componentes con los que cuenta dicho módulo. También se expusieron los resultados obtenidos durante las pruebas realizadas al módulo desarrollado.

CONCLUSIONES

Como resultado de la investigación desarrollada se logró integrar el módulo de vaciado de sólidos a la herramienta AsiXMec garantizando así que se reduzcan los pasos de diseño para la obtención de estructuras mecánicas complejas.

RECOMENDACIONES

Aún con el cumplimiento del objetivo de la aplicación y teniendo como base el estudio efectuado durante la realización de esta investigación, se recomienda para próximas versiones:

- realizar un estudio sobre cómo realizar la operación de vaciado a través de la topología del sólido para mejorar el resultado del módulo desarrollado cuando se haga uso del parámetro que maneja el espesor en porcentaje.
- permitir al usuario definir un espesor para cada cara del sólido resultante.

REFERENCIAS BIBLIOGRÁFICAS

- [1] “El libro de Catia V5: Módulos Part Desing, Wireframe & Surface Desing, Assembly Desing y Drafting, María Gloria del Rio Cidoncha, María Eugenia Martínez Lomas, Juan Martínez Palacios y Silvia Pérez Díaz. Editorial Tébar, SL. Madrid, 2007”
- [2] “CAD.” [Online]. Available: <http://www.mastermagazine.info/termino/4142.php>. [Accessed: 13-May-2015].
- [3] “Herramientas CAD/CAM/CAE y prototipado virtual y rápido para la obra pública,” Interempresas. [Online]. Available: <http://www.interempresas.net/ObrasPublicas/Articulos/34772-Herramientas-CAD-CAM-CAE-y-prototipado-virtual-y-rapido-para-la-obra-publica.html>. [Accessed: 22-Jun-2015].
- [4] “3 - Indicar las ventajas que tiene usar software CAD/CAM - stigestionydesarrollo.” [Online]. Available: <https://sites.google.com/site/stigestionydesarrollo/recuperacion/desarrollo-1/recuperacion-tema-4---desarrollo/safdsadsa>. [Accessed: 27-May-2015].
- [5] Doc_U, “Sistema CAD.” [Online]. Available: <http://es.scribd.com/doc/17754860/Sistema-CAD#scribd>.
- [6] “3.3. Operaciones: Crear el sólido 3D. - De Formas y Funciones.” [Online]. Available: <http://www.togores.net/inventor-01/autodeskinventor/creacion-automatica-de-una-pieza/crea-solido>. [Accessed: 03-May-2015].
- [7] Juan Manuel Hernández Guijarro, “Modelado y mecanizado mediante CNC del molde de una pala de pádel.” .
- [8] C. G. L, “Tutorial 06: Operaciones con sólidos,” - Blog de Carlos González Larenas, docente de Dibujo y construcción, destinados a informar a sus alumnos y compartir experiencias. .
- [9] César Otero González and Cristina Manchado del Val, “Open Couser Ware.” .
- [10] “Ayuda de SOLIDWORKS.” [Online]. Available: http://help.solidworks.com/2013/spanish/SolidWorks/sldworks/HIDD_DVE_FEAT_SHELL.htm?id=0bf5fb446d514871bb864e8a3edf03a9#Pg0.
- [11] Ramón Quiza Sardiñas, “El dibujo en tres dimensiones con AutoCAD 2000.” .
- [12] FÉLIX FERNÁNDEZ ESCUREDO, “Manual de FreeCAD.” 0.15, 02-Nov-2014.
- [13] “Ecuaciones y Fórmulas.” [Online]. Available: <http://www.disfrutalasmatematicas.com/algebra/ecuaciones-formulas.html>.
- [14] “El volumen y su medida.” [Online]. Available: http://concurso.cnice.mec.es/cnice2005/93_iniciacion_interactiva_materia/curso/materiales/propiedades/volumen.htm.

- [15]“Fórmulas para calcular áreas y volúmenes | Cuerpos Sólidos Geométricos.” .
- [16]“How Should Integration be Defined.”
- [17]“Relacion-de-proporcionalidad-directa-e-inversa.” [Online]. Available:
<http://www.portaleducativo.net/octavo-basico/806/Relacion-de-proporcionalidad-directa-e-inversa>.
[Accessed: 22-Jun-2015].
- [18]Kareny Brito Acuña, “Metodologías tradicionales y metodologías ágiles,” in Metodologías de desarrollo para aplicaciones web,
- [19]“Proceso unificado ágil (aup),” 12:28:52 UTC.
- [20]“AUP Ingeniería de Software.” [Online]. Available: http://ingenieriadesoftware.mex.tl/63758_AUP.html.
[Accessed: 05-May-2015].
- [21]Tamara Rodríguez Sánchez, “Metodología de desarrollo para la Actividad productiva de la UCI.” 11-Dec-2014.
- [22]“Desarrollo de Software,” Universitaria Agustiniiana - Uniagustiniana. [Online]. Available:
<http://www.uniagustiniana.edu.co/index.php/conocenos-desarrollo-de-software.html>. [Accessed: 22-Jun-2015].
- [23]“Open CASCADE Technology, 3D modeling & numerical simulation.” [Online]. Available:
<http://www.opencascade.org/>.
- [24]“Open CASCADE Technology,” Open CASCADE Technology BRepOffsetAPI_MakeThickSolid Class Reference. .
- [25]“Open CASCADE Technology,” Open CASCADE Technology AIS_Shape Class Reference. .
- [26]“Open CASCADE Technology,” Open CASCADE Technology BRepGProp Class Reference. .
- [27]“Curso de UML - Desarrollo Orientado a Objetos con UML.” [Online]. Available:
<http://www.lawebdelprogramador.com/cursos/UML/2336-Desarrollo-Orientado-a-Objetos-con-UML.html>.
[Accessed: 27-May-2015].
- [28]S. Pérez Lovelle, C. Galindo, J. C, A. Hernández Perenzuela, A. Arredondo López, R. Nápoles, L. R, and F. Carnero González, “Una propuesta para incorporar más semántica de los modelos al código generado,” Polibits, no. 49, pp. 47–50, Jun. 2014.
- [29]“Herramientas Case - Monografias.com.” [Online]. Available:
<http://www.monografias.com/trabajos14/herramicase/herramicase.shtml#tec>. [Accessed: 05-May-2015].
- [30]fergarcia, “Entorno de Desarrollo Integrado (IDE).,” fergarcia. .

- [31]salomeher, "Qt Creator," TEIDE2. .
- [32]"Curso C++," OrganiZATOR. [Online]. Available: http://www.zator.com/Cpp/E1_2.htm.
- [33]"Modelo de Dominio," Tecnología y Synergix. .
- [34]"REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES (Clase # 7)," SISTEMAS DE INFORMACIÓN GERENCIAL, SEGURIDAD Y PROTECCIÓN. .
- [35]"Casos de uso." [Online]. Available: <http://www.mastermagazine.info/termino/4184.php>. [Accessed: 11-Jun-2015].
- [36]"Análisis de Sistemas." [Online]. Available: <https://sites.google.com/site/jojooa/analisis-de-sistemas/definicion-de-actor-que-es-un-actor>
- [37]"diagrama de casos de uso del negocio y del sistema," 18:58:53 UTC.
- [38]ingeniods, "Patrones Arquitectónicos," Ingenio DS. .
- [39]Craig Larman, "El Mdelo de Diseño."
- [40]"Diagrama de Paquetes." [Online]. Available: <http://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-paquetes>. [Accessed: 05-Jun-2015].
- [41]José Enrique González Cornejo "El Lenguaje de Modelado Unificado (UML)." [Online]. Available: <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>
- [42]Manuel Arias Calleja, "Estándares de codificación."
- [43]F. Coronel, "Unidad 5 Modelo de implementacion.: 5.1 Diagrama de componentes," Unidad 5 Modelo de implementacion., 08-Jun-2013. .
- [44]"Pruebas de caja blanca y negra," 20:36:02 UTC.
- [45]F. Carrillo, "Pruebas del Software: Niveles de Prueba del Software," Pruebas del Software, 03-Jan-2011. .
- [46]"Casos de pruebas," 21:21:18 UTC.

BIBLIOGRAFÍA

- [1] E. 6 Marzo and 2015 En 17:00, "Blog Mp3.es » 5 herramientas de diseño CAD para que imagines tu casaBlog Mp3.es," Blog.mp3. . : <http://blog.mp3.es/5-herramientas-de-diseno-cad-editores-cad/>
- [2] Carlos Gonzalez Larenas, "MVNlog.cl." [Online]. Available: <http://www.mvblog.cl/2013/03/17/autocad-3d-tutorial-06-operaciones-con-solidos/>. [Accessed: 05-Jun-2015].
- [3] "Cómo calcular el volumen," wikiHow. [Online]. Available: <http://es.wikihow.com/calcular-el-volumen>. [Accessed: 05-Jun-2015].
- [4] "Metodologías de desarrollo," 03:04:28 UTC. : <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>
- [5] "Metodología RUP," 13:27:16 UTC. : <http://es.slideshare.net/cortesalvarez/metodologa-rup>
- [6] P. Letelier and M. C. Penadés, "Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)," www.cyta.com.ar/ta0502/v5n2a1.htm, 15-Apr-2006. [Online]. Available: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>. [Accessed: 05-Jun-2015].
- [7] J. Srividhya, Y. Li, and J. R. Pomerening, "Open cascades as simple solutions to providing ultrasensitivity and adaptation in cellular signaling," *Phys. Biol.*, vol. 8, no. 4, p. 046005, Aug. 2011. : <http://iopscience.iop.org/1478-3975/8/4/046005>
- [8] ingeniods, "Patrones Arquitectónicos," Ingenio DS. . : <https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/>
- [9] "BPM con Reglas de Negocio (Business Process Software) | AuraPortal." [Online]. Available: <http://www.auraportal.com/es/reglas-negocio-bpm-workflow-software>. [Accessed: 05-Jun-2015].
- [10] "Arquitectura de Software," Software Guru. [Online]. Available: <http://sg.com.mx/revista/27/arquitectura-software>. [Accessed: 05-Jun-2015].
- [11] rubenfa, "Patrones de diseño: qué son y por qué debes usarlos," Genbeta Dev, 14-Jul-2014. [Online]. Available: <http://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos>. [Accessed: 05-Jun-2015].
- [12] "PRUEBAS DEL SOFTWARE," PRUEBAS DEL SOFTWARE. [Online]. Available: <https://pruebasdelsoftware.wordpress.com/>. [Accessed: 05-Jun-2015].
- [13] I. «. M. K. B. dice, "MODELO DE ANÁLISIS," Mundo Kramer's Blog. .

- [14] C. A. Guerrero, J. M. Suárez, and L. E. Gutiérrez, "Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web," *Inf. Tecnológica*, vol. 24, no. 3, pp. 103–114, Jan. 2013.
- [15] "Proceso unificado ágil (aup)," 12:28:52 UTC. : <http://es.slideshare.net/joseluisdifu/proceso-unificado-gil-aup-17171038>
- [16] "Técnicas para explotar objetos en 3ds Max |." : <http://blog.acaddemia.com/tecnicas-para-explotar-objetos-en-3ds-max/>
- [17] "Manual modelado 3d auto cad," 00:11:31 UTC. : <http://es.slideshare.net/fernandovegacerna/manual-modelado-3-d-auto-cad>
- [18] "Pruebas de Aceptación del Cliente (página 2) - Monografias.com." [Online]. Available: <http://www.monografias.com/trabajos36/pruebas-de-aceptacion/pruebas-de-aceptacion2.shtml>. [Accessed: 11-Jun-2015].
- [19] "Codificación | Recursos web." : <http://recursosweb.unam.mx/recursos-web/creacion-de-paginas-web/estandares-de-codificacion/>
- [20] "Qt Creator," SourceForge. [Online]. Available: <http://sourceforge.net/projects/qtcreator.mirror/>. [Accessed: 11-Jun-2015].
- [21] E. Guerrero, "análisis y diseño de software: Arquitectura en tres capas," *análisis y diseño de software*, 18-May-2012. . : <http://www.frainguerrero.blogspot.com/2012/06/arquitectura-en-tres-capas.html>
- [22] "PROPORCIONALIDAD DIRECTA - Enciclopedia de Tareas." .
- [23] "Herramientas CAD/CAM/CAE y prototipado virtual y rápido para la obra pública," *Interempresas*. [Online]. Available: <http://www.interempresas.net/ObrasPublicas/Articulos/34772-Herramientas-CAD-CAM-CAE-y-prototipado-virtual-y-rapido-para-la-obra-publica.html>. [Accessed: 22-Jun-2015].

ANEXOS

Anexo 1 Descripción de los Casos de uso.

Tabla 11. Descripción del CU-2_Aplicar el vaciado exterior

Caso de uso:	Aplicar el vaciado exterior	
Objetivo:	Permitir realizar sobre la cara seleccionada de la estructura mecánica el vaciado exterior, determinando también el espesor de los bordes.	
Actores:	Ingeniero mecánico.	
Resumen:	El caso de uso se inicia cuando el ingeniero mecánico realiza un vaciado de sólido con un grosor determinado a una estructura mecánica.	
Complejidad:	Alta	
Prioridad:	Crítica	
Referencias:	RF1	
Precondiciones:	Debe existir al menos una estructura mecánica en el visor.	
Postcondiciones:	Se aplicó el vaciado exterior a una estructura mecánica.	
Flujo de Eventos		
Flujo básico Aplicar el vaciado		
Acción del Actor:	Respuesta del Sistema:	
1. El especialista selecciona la operación de vaciado.	1.1 Muestra el diálogo para la funcionalidad de vaciado.	
2. Selecciona la operación de vaciado exterior.		
3. Selecciona una cara del sólido.	3.1 Previsualiza el resultado de la operación.	
4. Selecciona el tipo de espesor:	4.1 El sistema valida el espesor dado por el usuario y	

	<ul style="list-style-type: none"> • Si es el primer espesor define el valor deseado en unidades de milímetro. • Si es el segundo espesor define el valor deseado en porciento. 	previsualiza el resultado de la operación según el nuevo valor del tipo de espesor seleccionado en caso de que sea correcto.
5.	<p>Selecciona la opción Hacer.</p> <ul style="list-style-type: none"> • Si selecciona la opción Cancelar ver el Flujo Alterno 2 Selecciona la opción Cancelar. 	5.1 Visualiza los resultados de la operación.
		Termina el caso de uso.
Flujo Alterno de Eventos		
Nº Evento 2 Selecciona la opción Cancelar.		
Acción del Actor:		Respuesta del Sistema:
1.	Selecciona la opción Cancelar del diálogo de Vaciado.	<p>1.1 Se cierra el diálogo. No se ejecuta la operación.</p> <p>Termina el caso de uso.</p>
Flujo Alterno de Eventos		
Nº Evento 2.1 Espesor invalido.		
Acción del Actor:		Respuesta del Sistema:
1.		<p>1.1 Valida la expresión teniendo en cuenta las siguientes condiciones:</p> <ol style="list-style-type: none"> 1. Espesor menor que 0. Muestra un mensaje de error. 2. Espesor igual a cero o mayor que 100 000: Muestra el mensaje de error. <ol style="list-style-type: none"> 1. Parámetros inválidos (no están definidos en el

		Editor de Parámetros): Muestra mensaje de error.
2.	Presiona el botón Aceptar.	2.1 Regresa al paso 1 del Flujo Básico Vaciado exterior.
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos no funcionales		No aplicable
Asuntos pendientes		No aplicable

Tabla 12. Descripción del CU-3_Aplicar el vaciado híbrido

Caso de uso:	Aplicar el vaciado híbrido	
Objetivo:	Permitir realizar sobre la cara seleccionada de la estructura mecánica el vaciado híbrido, determinando también el espesor de los bordes.	
Actores:	Ingeniero mecánico.	
Resumen:	El caso de uso se inicia cuando el ingeniero mecánico realiza un vaciado de sólido con un grosor determinado a una estructura mecánica.	
Complejidad:	Alta	
Prioridad:	Crítica	
Referencias:	RF1	
Precondiciones:	Debe existir al menos una estructura mecánica en el visor.	
Postcondiciones:	Se aplicó el vaciado híbrido a una estructura mecánica.	
Flujo de Eventos		
Flujo básico Aplicar el vaciado		
Acción del Actor:	Respuesta del Sistema:	

1.	El especialista selecciona la operación de vaciado.	1.1 Muestra el diálogo para la funcionalidad de vaciado.
2.	Selecciona la operación de vaciado híbrido.	
3.	Selecciona una cara del sólido.	3.1 Previsualiza el resultado de la operación.
4.	<p>Selecciona el tipo de espesor:</p> <ul style="list-style-type: none"> • Si es el primer espesor define el valor deseado en unidades de milímetro. • Si es el segundo espesor define el valor deseado en porciento. 	4.1 El sistema valida el espesor dado por el usuario y previsualiza el resultado de la operación según el nuevo valor del tipo de espesor seleccionado en caso de que sea correcto.
5.	<p>Selecciona la opción Hacer.</p> <ul style="list-style-type: none"> • Si selecciona la opción Cancelar ver el Flujo Alterno 2 Selecciona la opción Cancelar. 	5.1 Visualiza los resultados de la operación.
		Termina el caso de uso.

Flujo Alterno de Eventos

Nº Evento 2 Selecciona la opción Cancelar.

Acción del Actor:

Respuesta del Sistema:

1.	Selecciona la opción Cancelar del diálogo de Vaciado.	1.1 Se cierra el diálogo. No se ejecuta la operación. Termina el caso de uso.
----	---	--

Flujo Alterno de Eventos

Nº Evento 2.1 Espesor invalido.

Acción del Actor:

Respuesta del Sistema:

1.		1.1 Valida la expresión teniendo en cuenta las siguientes condiciones: <ol style="list-style-type: none"> 1. Espesor menor que 0. Muestra un mensaje de error. 2. Espesor igual a cero o mayor que 100 000: Muestra el mensaje de error. <ol style="list-style-type: none"> 1. Parámetros inválidos (no están definidos en el Editor de Parámetros): Muestra mensaje de error.
2.	Presiona el botón Aceptar.	2.1 Regresa al paso 1 del Flujo Básico Vaciado híbrido.
Relaciones	CU Incluidos	No aplicable
	CU Extendidos	No aplicable
Requisitos no funcionales		No aplicable
	Asuntos pendientes	No aplicable

Anexo 2 Descripción de las clases

Tabla 13. Descripción de la clase ShellState

Descripción de la clase ShellState	
Nombre: <i>ShellState</i>	
Tipo de clase: Vista	
Resumen: Es la clase que maneja el estado de la operación, la cual es la encargada de redireccionar el estado de los datos de la operación de vaciado a la clase controladora principal e iniciar la ventana de dialogo del módulo.	
Atributo:	Tipo:
<i>action</i>	<i>QAction</i>
<i>dialog</i>	<i>ShellDialog</i>
<i>controller</i>	<i>ShellController</i>
Responsabilidades	
Nombre:	<i>start()</i>
Descripción:	Funcionalidad encargada de mostrar el dialogo con los datos inicializados.
Nombre:	<i>rollback()</i>
Descripción:	Funcionalidad encargada de mandar a cerrar el contexto y reiniciar los datos, mostrando todos los sólidos que existan en el contexto.

Tabla 14. Descripción de la clase ShellDialog

Descripción de la clase ShellDialog	
Nombre: <i>ShellDialog</i>	
Tipo de clase: Vista	
Resumen: Es la clase que maneja los datos de la operación, la cual es la encargada de redireccionar los	

datos de la operación a la clase controladora principal. Es la que realiza la interacción con el usuario para que sean determinados los datos.

Atributo:	Tipo:
<i>controller</i>	<i>ShellController</i>
<i>volumen</i>	<i>ParameterEditor</i>
<i>volumen2</i>	<i>ParameterEditor</i>
<i>previsualizate</i>	<i>bool</i>

Responsabilidades

Nombre:	<i>init()</i>
Descripción:	Funcionalidad encargada de inicializar los datos para que sean enviados al <i>controller</i> .
Nombre:	<i>insideShell()</i>
Descripción:	Funcionalidad encargada de enviar los datos para realizar el vaciado interior.
Nombre:	<i>outsideShell()</i>
Descripción:	Funcionalidad encargada de enviar los datos para realizar el vaciado exterior.
Nombre:	<i>bothShell()</i>
Descripción:	Funcionalidad encargada de enviar los datos para realizar el vaciado híbrido.
Nombre:	<i>onThickness()</i>
Descripción:	Funcionalidad encargada de activar el espesor del volumen en el primer <i>ParameterEditor</i> ..
Nombre:	<i>onPercentThickness()</i>
Descripción:	Funcionalidad encargada de activar el espesor del volumen2 en el segundo <i>ParameterEditor</i> .

Tabla 15. Descripción de la clase ShellNode

Descripción de la clase ShellNode	
Nombre: ShellNode	
Tipo de clase: Modelo	
Resumen: Es la clase que registra los datos en el árbol de OCAF para que sean accedidos por la clase ShellDriver.	
Atributo:	Tipo:
Esta clase no contiene atributos.	
Responsabilidades	
Nombre:	<i>setType(DocumentNodeType /*type*/)</i>
Descripción:	Funcionalidad encargada de manejar el tipo de vaciado según un <i>LabelNode</i> .
Nombre:	<i>setShellType(Standard_Integer type)</i>
Descripción:	Funcionalidad encargada dado un tipo de estándar entero, contener el tipo de vaciado que se aplicara.
Nombre:	<i>getDistance()</i>
Descripción:	Funcionalidad encargada de obtener el valor de la distancia del <i>ParameterNode</i> .
Nombre:	<i>getSecondParameter()</i>
Descripción:	Funcionalidad encargada de devolver un <i>ParameterNode</i> según la distancia del <i>Label</i> .
Nombre:	<i>setFacesIndex(vector<Standard_Integer> array)</i>
Descripción:	Funcionalidad encargada de contener un conjunto de caras según sus índices, dado por el arreglo (vector <Estándar_Entero>) pasado por referencia.
Nombre:	<i>getFacesIndex()</i>
Descripción:	Funcionalidad encargada de obtener el arreglo que contiene los índices de las caras.

Tabla 16. Descripción de la clase ShellItem

Descripción de la clase ShellItem	
Nombre: <i>ShellItem</i>	
Tipo de clase: Modelo	
Resumen: Es la clase que define qué operación debe mostrar el <i>object browser</i> como realizada.	
Atributo:	Tipo:
Esta clase no contiene atributos.	
Responsabilidades	
Nombre:	<i>getIcon()</i>
Descripción:	Funcionalidad encargada de obtener el icono de la operación de vaciado.
Nombre:	<i>getFeatureNode()</i>
Descripción:	Funcionalidad encargada de obtener las características del nodo.

Anexo 3 Casos de pruebas elaborados

Tabla 17. Caso de prueba basado en el CU-2_Aplicar el vaciado exterior

Escenarios	Descripción	Espesor	Espesor en porcentaje	Respuesta del sistema	Flujo central
EC 1: Aplicar el vaciado exterior.	<p>1.1. El especialista selecciona la Operación de Vaciado.</p> <p>- Vaciado exterior</p> <p>1.3. Selecciona la cara sobre la que desea realizar el vaciado.</p> <p>1.5. Especifica la distancia del espesor deseada mediante un número.</p> <p>1.8. Selecciona la opción Hacer.</p> <p>- Si selecciona la opción Cancelar ver EC 3.1 Selecciona la opción Cancelar.</p>	20	20	<p>1.2. Muestra el diálogo para la funcionalidad Vaciado.</p> <p>1.4. Previsualiza el resultado de la operación.</p> <p>1.6. Previsualiza el resultado de la operación.</p> <p>1.7. Valida el valor de la distancia. Ver EC 2.1 Distancia inválida.</p> <p>1.9. Crea y visualiza el vaciado teniendo en cuenta los valores especificados por el usuario.</p>	Cinta de opciones
EC 2.1: Distancia inválida.	2.2. Presiona el botón aceptar.	150000	150000	<p>2.2. Valida el espesor teniendo en cuenta las siguientes condiciones:</p> <ol style="list-style-type: none"> 1. Distancia menor que cero: Muestra el mensaje de error. 2. Distancia igual a cero o mayor que 100 000: Muestra el mensaje de error. 3. Parámetros inválidos (no están definidos en el Editor de Parámetros): Muestra mensaje de error. 	
		0	0		

Tabla 18. Caso de prueba basado en el CU-3_Aplicar el vaciado híbrido

Escenarios	Descripción	Espesor	Espesor en porcentaje	Respuesta del sistema	Flujo central
EC 1: Aplicar el vaciado híbrido.	<p>1.1. El especialista selecciona la Operación de Vaciado.</p> <p>- Vaciado híbrido</p> <p>1.3. Selecciona la cara sobre la que desea realizar el vaciado.</p> <p>1.5. Especifica la distancia del espesor deseada mediante un número.</p> <p>1.8. Selecciona la opción Hacer.</p> <p>- Si selecciona la opción Cancelar ver EC 3.1 Selecciona la opción Cancelar.</p>	20	20	<p>1.2. Muestra el diálogo para la funcionalidad Vaciado.</p> <p>1.4. Previsualiza el resultado de la operación.</p> <p>1.6. Previsualiza el resultado de la operación.</p> <p>1.7. Valida el valor de la distancia. Ver EC 2.1 Distancia inválida.</p> <p>1.9. Crea y visualiza el vaciado teniendo en cuenta los valores especificados por el usuario.</p>	Cinta de opciones
EC 2.1: Distancia inválida.	2.2. Presiona el botón aceptar.	150000	150000	<p>2.2. Valida el espesor teniendo en cuenta las siguientes condiciones:</p> <ol style="list-style-type: none"> 1. Distancia menor que cero: Muestra el mensaje de error. 2. Distancia igual a cero o mayor que 100 000: Muestra el mensaje de error. 3. Parámetros inválidos (no están definidos en el Editor de Parámetros): Muestra mensaje de error. 	
		0	0		

GLOSARIO DE TÉRMINOS

CAD: *Computer Aided Design* (Diseño Asistido por Computadoras).

CAE: *Computer Aided Engineering* (Ingeniería asistida por computadoras)

CAM: *Computer Aided Manufacturing* (Fabricación Asistida por Computadoras)

CASE: *Computer Aided Software Engineering*. Ingeniería de *software* asistida por computadora.

Herramientas CASE: Herramientas utilizadas para el desarrollo de proyectos de Ingeniería de *Software*.

AUP: *Agile Unified Process* (Proceso Unificado Ágil). Metodología para el desarrollo de *Software*.

Software: Programas de sistema, utilerías o aplicaciones expresadas en un lenguaje de máquina.

Framework: Es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

API: *Application Programming Interface* (Interfaz de Programación de Aplicaciones).

RAM: *Random Access Memory* (Memoria de Acceso Aleatorio).

GRASP: *General Responsibility Assignment Software Patterns* (Patrones Generales de *Software* para Asignación de Responsabilidades).

UML: *Unified Modeling Language* (Lenguaje Unificado de Modelado).

Requisitos: Capacidades, condiciones o cualidades que el sistema debe cumplir y tener.

Caso de uso: Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.