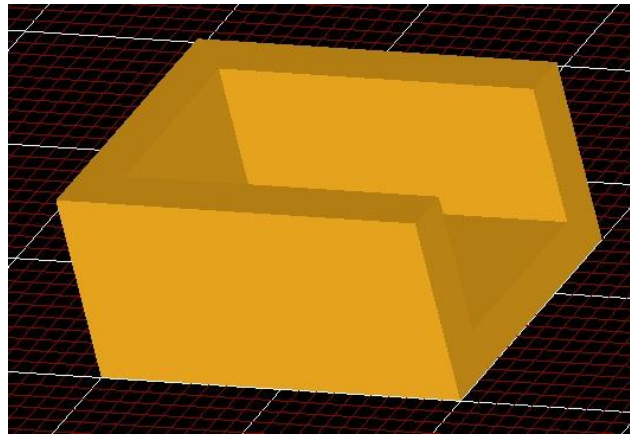


**Universidad de las Ciencias Informáticas**

**Facultad 5**



**Módulo Shell para entidades 3D**



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

**Autor: Adrián Hernández Aguilera**

**Tutor: MSc. Marvyn Amado Márquez Rodríguez**

**Co-Tutores:**

**Ing. Fabio Casado Ruiz**

**Ing. Yaima Fiallo Valle**

**Asesor: MSc. Pedro Carlos Pérez Martinto**

**Junio del 2012**

## **DECLARACIÓN DE AUTORÍA**

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

Para que así conste firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Adrián Hernández Aguilera

Autor

---

Marvyn Amado Márquez

Tutor

## **DATOS DE CONTACTO**

**Tutor:** MSc. Marvyn Amado Márquez Rodríguez

Edad: 27

Ciudadanía: Cubana.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

Título: MSc. Informática Aplicada

Categoría Docente: Ninguna.

E-mail: mamarquez@uci.cu

Graduado de la UCI, master en ciencias.

**Co-tutor:** Ing. Fabio Casado Ruiz.

Edad: 24

Ciudadanía: Cubana.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniero en Ciencias Informáticas.

Categoría Docente: Ninguna

**E-mail:** fcasado@uci.cu

Graduado de la UCI, con 3 años de experiencia trabajando con gráficos por computadoras.

**Co-tutor:** Ing. Yaima Fiallo Valle

**Edad:**30

**Ciudadanía:** Cubana.

**Institución:** Universidad de las Ciencias Informáticas (UCI).

**Título:** Ingeniero en informática

**Categoría Docente:** Asistente

**E-mail:** yfiallo@uci.cu

Graduado de ingeniería informática en la escuela José Antonio Echeverría (CUJAE) posee 5 años de experiencia en el tema de la ingeniería de software.

## **DEDICATORIA**

A mi madre que tanto ha sacrificado por verme llegar hasta aquí, que sufrió mis derrotas y festejó mis triunfos como nadie, que confió en mí y en mi proceder.

A mis abuelos que tantos años lucharon por mí, que me aconsejaron en todo momento, que fueron incondicionales conmigo y hoy no están para compartir este momento.

## **AGRADECIMIENTOS**

De alguna manera quisiera agradecer a todas las personas que han influido en el cumplimiento de este sueño.

A mi madre que me apoyó durante estos 5 años incondicionalmente, a mi padre José Alberto que tantas veces me dio la mano en los peores momentos y en el que tuve gran apoyo incondicional.

A mis tías que me han visto crecer ante las dificultades y han estado ahí para mí.

A mi prima Lisandra que tanto ánimo me dio en los momentos de mayor desgano.

No puedo dejar de mencionar a mis amigos: Dayron, Yassel, Pepe, Yaneisi, las jimaguas y Yissel que son muy especiales para mí por poder siempre contar con ellos, al igual que al equipo kafunga compuesto por Karel, Lefebre y Luanner, así como los holguineros , Iván, Neurys, Ofelia, Esmérida.

A Rosalia, Isel, Miguel Ángel, Solange, Aylin María y Aylin caballero, Rene, Katisleydis, a mi tutor, a mi cotutor que tanta fe tuvieron en mí.

Quiero agradecer además a una persona que marcó mi vida para bien y supo darme su apoyo en todo momento.

A Fidel Castro, autor de la idea de crear la Universidad de Ciencias Informáticas, a la Revolución Cubana que me permitió estudiar en ella.

## **RESUMEN**

El proyecto CDSEM (Centro de Desarrollo y Simulación de Estructuras Mecánicas) se encuentra en la actualidad asociado con la empresa venezolana PDVSA (Petróleos de Venezuela SA). Dicha asociación tiene como objetivo la entrega de la herramienta GALBA-CAD por parte del centro de desarrollo.

Esta herramienta carece en su diseño de un módulo **Shell** que trae como consecuencias que no se pueda añadir grosor a una superficie, extruir caras en ambas direcciones y convertir una curva cerrada en un objeto sólido 3D. Todas estas restricciones provocan que el proceso de construcción de estructuras mecánicas, dentro de la herramienta, sea en determinadas ocasiones, engorroso, lo que trae consigo que la productividad no sea la mejor.

El presente trabajo tiene como propósito la elaboración del módulo mencionado con anterioridad. La aplicación del módulo **Shell** permite mejorar la productividad en el proceso de construcción de las estructuras mecánicas.

Al concluir este trabajo se logró aplicar el **Shell** a las estructuras mecánicas teniendo en cuenta inicialmente si éstas son mallas o sólidos. Además se identificaron distintos tipos de **Shell** y la forma en que son creados.

Palabras Clave: **Shell**, modelador geométrico, herramientas CAD.

# Índice

ÍNDICE	8	
INTRODUCCIÓN	10	
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	14	
1.1 DISEÑO DE TALADROS DE PERFORACIÓN PETROLERA		14
1.2 HERRAMIENTAS CAD/CAE/CAM		14
1.3 MODELADORES GEOMÉTRICOS		16
1.4 BIBLIOTECA VTK Y OPEN CASCADE		19
1.5 OPERACIÓN SHELL		19
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA	26	
2.1 REGLAS DEL NEGOCIO		26
2.2 HERRAMIENTAS DE MODELADO Y METODOLOGÍAS DE DESARROLLO		27
2.3 MODELO DE DOMINIO		27
2.4 CAPTURA DE REQUISITOS		28
2.4.1 <i>Requisitos funcionales</i>		29
2.4.2 <i>Requisitos no funcionales</i>		29
2.5 MODELO DE CASOS DE USO		30
2.5.1 <i>Diagrama de casos de uso del sistema</i>		30
2.5.2 <i>Descripción de los casos de uso</i>		31
2.6 DISEÑO DEL SISTEMA		37
2.7 DIAGRAMAS DE SECUENCIAS		39
CAPÍTULO 3. SOLUCIÓN PROPUESTA	44	
3.1 HERRAMIENTAS DE DESARROLLO		44
3.2 ESTRUCTURAS UTILIZADAS DE OPEN CASCADE		45
3.3 APIs UTILIZADAS DE OPEN CASCADE		47
3.3.1 <i>BRepBuilderAPI</i>		47
3.3.2 <i>BRepPrimAPI</i>		48
3.3.3 <i>BRepOffsetAPI</i>		48
3.4 CLASE SHELL		49
3.4.1 <i>Método ApplyTypeShells</i>		49
3.4.2 <i>Método Apply</i>		51
3.5 DIAGRAMA DE COMPONENTES DEL MÓDULO DESARROLLADO		53
3.6 VALIDACIÓN DE LOS RESULTADOS		55
3.6.1 <i>CASOS DE PRUEBA</i>		55
CONCLUSIONES	63	
RECOMENDACIONES	64	
BIBLIOGRAFÍA	65	
ANEXOS	68	
ANEXO A: IMÁGENES DE LA APLICACIÓN		68





## Introducción

---

El surgimiento de las herramientas CAD/CAE/CAM ((**CAD**) diseños asistido por computadoras, (**CAE**) ingeniería asistida por computadoras y fabricación asistida por computadoras(**CAM**)) se asocia a la evolución de los ordenadores. Los primeros pasos tienen lugar en la década de 1950 con el surgimiento de las primeras pantallas gráficas, el lápiz óptico y la aparición de las máquinas herramientas. Posteriormente la aparición del término CAD y las investigaciones realizadas sobre el mismo, así como la generalización del término CAD/CAE/CAM propiciaron que las industrias entraran en una constante automatización de sus procesos en los que se va generalizando la integración de las diversas técnicas de diseño, análisis, simulación y fabricación. En la actualidad el uso de las herramientas CAD/CAE/CAM ha pasado a ser una de las mejores opciones dentro del ámbito de la industria.

Las herramientas CAD/CAE/CAM, son herramientas que posibilitan el diseño, ingeniería y fabricación asistido por computadoras de productos industriales, que revisten una gran importancia a nivel mundial por el potencial que brindan. Estas herramientas basan su trabajo en un módulo conocido comúnmente como modelador geométrico.

Se definen a los modeladores geométricos como herramientas poderosas encargadas del modelado de sólidos y que estimulan la creatividad y la capacidad intelectual, además de reforzar el conocimiento de las tres dimensiones.

El proyecto CDSEM perteneciente al Centro de Informática Industrial (CEDIN) que reside en la universidad de las ciencias informáticas (UCI) se encuentra junto con la empresa venezolana PDVSA en el desarrollo de una herramienta para el apoyo a la creación de taladros de perforación de pozos de petróleo. La herramienta en desarrollo, GALBA-CAD, carece en su diseño de un módulo **Shell** lo que trae consigo que no se pueda añadir grosor a una malla proporcionada con parámetros simples, extruir caras en ambas direcciones de manera simultánea y automática y convertir una curva cerrada en un objeto sólido 3D de forma automática.

Todas estas limitaciones traen consigo que el proceso de construcción de estructuras mecánicas, dentro de la herramienta, sea en determinadas ocasiones engorroso, que trae como consecuencias que la productividad no sea la mejor.

Teniendo en cuenta la situación problemática planteada anteriormente podemos definir como **problema de investigación**:

¿Cómo proporcionar grosor a una superficie para mejorar la productividad en el proceso de construcción de estructuras mecánicas?

A partir del problema de investigación planteado podemos definir como **objeto de investigación** el proceso de modelación de estructuras mecánicas en herramientas CAD.

Se define como **objetivo general** desarrollar un módulo **Shell** que permita proporcionar grosor a una superficie.

Definiendo como el **campo de acción** la operación **Shell** en la modelación de estructuras mecánicas.

Para dar cumplimiento al objetivo general planteado es necesario realizar un grupo de tareas investigativas tales como:

- Buscar la Bibliografía referente al tema para lograr un mejor entendimiento del problema y una conducción satisfactoria hacia la solución.
- Caracterizar los modeladores geométricos que se han desarrollado en las disímiles herramientas CAD/CAE/CAM para conocer su completo funcionamiento.
- Definir la operación **Shell** utilizada en las distintas variantes de las herramientas CAD identificadas para decidir cómo llevarla a cabo en el módulo y procurar solución al problema.
- Implementar el algoritmo seleccionado para realizar la operación **Shell**.
- Comparar los resultados obtenidos en el módulo desarrollado con los obtenidos de aplicar el **Shell** en la herramienta **Inventor** para validar dichos resultados.

Para la realización de la investigación y elaboración del presente trabajo se emplearon varios **métodos científicos de investigación**, entre los cuales se pueden mencionar los siguientes:

**Métodos teóricos:**

**Análisis Histórico – Lógico:** Mediante este método se analizó la evolución y desarrollo del objeto de investigación y sus elementos más importantes.

**Analítico – Sintético:** Se usó para analizar la información de las técnicas existentes que definen un **Shell**.

**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.

Se muestra a continuación la estructura del trabajo de diploma, incluyendo una síntesis de los capítulos y secciones fundamentales:

**Capítulo 1.** Fundamentación Teórica.

En este capítulo se definen los principales conceptos que serán empleados durante todo el trabajo, se presentan las bases teóricas fundamentales relacionadas con las herramientas CAD/CAE/CAM y se profundiza en los conceptos para definir la operación **Shell**.

**Capítulo 2.** Características del sistema

Durante este capítulo se describe el sistema desde la perspectiva de Ingeniería de Software, usando el Proceso Unificado de Desarrollo como metodología. Se presentan las reglas específicas del negocio y el modelo de dominio del problema. Se realiza la captura de requisitos y el modelo de casos de uso del Sistema. Posteriormente se muestran los Diagramas de Secuencia correspondientes a los Casos de Uso.

**Capítulo 3.** Descripción de la solución.

Se propone una solución técnica al problema planteado, haciendo uso de algoritmos y funciones implementadas por las herramientas CAD/CAE/CAM o alguna biblioteca de clases que facilite dicho trabajo, así como las adaptaciones realizadas por el autor para resolver el problema existente. Además se dan a conocer las herramientas seleccionadas para resolver el problema, así como los principales resultados obtenidos en el trabajo.

**Anexos:** En esta sección se incluyen imágenes obtenidas con la realización de la operación Shell.

**Glosario de Términos:** Se elaboró un Glosario de Términos con el objetivo de facilitar la comprensión del lenguaje utilizado.

## Capítulo 1. Fundamentación teórica

---

En este capítulo se definen los principales conceptos que serán empleados durante todo el trabajo y se presentan las bases teóricas fundamentales relacionadas con el diseño de taladros de perforación petrolera y las bases concernientes a las herramientas CAD/CAE/CAM. Además se abordan elementos fundamentales de los modeladores geométricos y la operación **Shell**.

### **1.1 Diseño de taladros de perforación petrolera**

---

En el mundo actual la manera para determinar la existencia de hidrocarburos en un depósito propuesto por la investigación geológica es a través de la perforación de pozos de petróleo. Para ello es de vital importancia el uso de los taladros de perforación petrolera (1). Para llevar a cabo este proceso de perforación se solicita una serie de taladros que ofrecen diferentes diámetros y niveles de profundidad en el momento requerido. Dada la importancia que reviste para el mundo la extracción de petróleo, el diseño y construcción de los taladros de perforación ha ido evolucionando, haciendo necesario que los fabricantes presenten productos cada vez más adecuados a las necesidades de los clientes.

A medida que pasa el tiempo se hace de vital importancia el mejoramiento de las herramientas que llevan a cabo el diseño de estos taladros por el aumento de la demanda. Estas herramientas mejoradas han de ser de fácil manejo, en donde los usuarios con conocimientos básicos puedan hacer uso de las mismas y de fácil obtención para que sean comercialmente accesibles para todo aquel que las necesite (2).

Las herramientas CAD son la base fundamental para los grandes proyectos de ingeniería (3).

### **1.2 Herramientas CAD/CAE/CAM**

---

Una fábrica con una integración completa mediante computadoras, consistirá en subsistemas modulares, controlados por computadoras, que estarán interconectadas, constituyendo un sistema de cálculo distribuido. El mismo facilitará enormemente el diseño y desarrollo de nuevos productos e implicará que la distancia entre diseño y fabricación disminuya (4).

Otros de los beneficios que brindan estas herramientas son la interactividad, la posibilidad de simular el comportamiento del modelo antes de la construcción del prototipo modificando sus parámetros, la

generación de planos con todo tipo de vistas, detalles y secciones y la posibilidad de conexión con un sistema de fabricación (5).

En el mundo actualmente existe una amplia gama de herramientas CAD/CAE/CAM que se usan de acuerdo al proceso industrial que se quiere favorecer y cuyo objetivo específico es el modelado de sólidos. Ejemplo de estas herramientas son:

- **CATIA**: Es un potente programa de diseño, fabricación e ingeniería realizado por **Dassault Systèmes** (6).
- **AutoCAD**: Es un programa CAD para entornos de usuarios finales y corporativos (3).
- **ArchiCAD**: Es una de las principales alternativas al AutoCAD, difiere de este en utilizar un modelo integrado 2D/3D y además tiene una orientación abierta a la arquitectura (7).
- **QCad**: Es una aplicación informática multiplataforma de diseño asistido por computadora para diseño 2D (8).
- **GstarCAD**: Es un programa CAD de bajo costo y compatible con AutoCAD (9).
- **AutodeskInventor**: Es un paquete de modelado paramétrico de sólidos en 3D (10).
- **SolidWorks**: Es un programa de diseño asistido por computadora para el modelado mecánico. El mismo modela estructuras mecánicas y conjuntos. Además de extraer de ellos tanto planos como otro tipo de información necesaria para la producción (11).
- **SalomeMeca**: Es un software libre que integra un preprocesador (CAD y maderador), un código de elementos finitos no-lineales y herramientas de post-análisis. La plataforma emplea el código de elementos finitos no-lineales **Code\_Aster** y la interfaz gráfica Salome. Por otro lado, todos los servicios (CAD, maderador, post-análisis) de Salome son accesibles vía comandos empleando el lenguaje de **Code\_Aster** basado en **Python** (12).

Para favorecer el proceso de diseño de taladros de perforación petrolera se decide, no emplear ninguna de estas herramientas, debido a que las mismas son elaboradas bajo licencias propietarias, en algunos casos, dificultando su comercialización.

En el caso particular de las herramientas **Salome Meca** y **QCAD** cabe destacar que las mismas son herramientas que presentan licencia LGPL en el caso de la primera y GPL en el segundo caso, pero la

interacción con el usuario en ambas se vuelve difícil y engorroso provocando que el proceso de diseño de estructuras mecánicas sea complejo por lo que se decide no utilizarlas.

De acuerdo a todas las limitaciones presentadas por las herramientas estudiadas se decide crear una herramienta CAD/CAE/CAM nueva para favorecer el proceso de diseño de taladros de perforación de petróleo (GALBA-CAD).

Las herramientas especializadas en el CAD/CAE/CAM se centran en el dibujo a partir de un módulo comúnmente conocido como modelador geométrico. Su uso depende del objeto a modelar y la finalidad para la que se construya el modelo.

### 1.3 Modeladores geométricos

---

Los modeladores geométricos son herramientas poderosas que permiten modelar todo tipo de objetos. Estos incluyen tres módulos fundamentales:

- **Sketcher**: Constituye un módulo que permite diseñar modelos mediante el empleo de herramientas de diseño con operaciones ágiles e intuitivas.
- **Ensamblaje**: Es un módulo preparado para crear conjuntos o ensamblajes mediante la inserción de los modelos 3D que fueron obtenidos en el módulo **sketcher**. Los ensamblajes se definen por el establecimiento de relaciones geométricas entre las estructuras mecánicas integrantes. La creación de estos permite analizar las posibles interferencias o choques entre los componentes insertados, así como simular el conjunto mediante motores lineales, rotativos, resortes y gravedad y evaluar la correcta cinemática del conjunto.
- **Dibujo técnico o plano**: Módulo que permite crear planos con las vistas de los modelos o ensamblajes de forma automática en muy poco tiempo.



Dichos módulos agrupan un conjunto de operaciones que facilitan el cumplimiento de sus propósitos. 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:

- **Chamfer**. Permite realizar **chaflán** a las estructuras mecánicas.
- **Round**: Se utiliza para redondear aristas u otras partes de las estructuras mecánicas.
- **Pattern**: Permite copiar una función un número determinado de veces.

Dentro de las operaciones 3D se incluyen:

- **Protusión**: Añade material a la estructuras mecánicas. Existen cuatro modos de realizar esta operación, estos son: **Extrude, Revolve, Blend, Sweep**.
- **Hole**: Permite la creación de agujeros en las estructuras mecánicas.
- **Rib**: Crea nervios. Se define el plano donde se va a dibujar la sección del nervio, su sección o forma y el espesor.
- **Shell**: Vacía el interior del volumen sólido quedando como resultado una cáscara con un espesor de pared determinado.

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

**Modelador Geométrico 2D**: Son usados para generar geometrías y perfiles que serán usados posteriormente para la construcción de superficies y sólidos. Se basan en entidades geométricas vectoriales como puntos, líneas, arcos y polígonos, con las que se puede operar a través de una interfaz gráfica.

**Modelador Geométrico 3D**: Son usados para modelar superficies y sólidos.

A raíz de estos estudios han sido desarrollados diferentes modeladores geométricos para herramientas CAD entre los que se puede mencionar:

- **ACIS**: El modelador **ACIS** es un núcleo (o motor) geométrico de modelamiento tridimensional que provee las funcionalidades necesarias para el modelado en 3D (13).
- **Parasolid**: Proporciona funcionalidades básicas que permiten a los usuarios modelar rápidamente y de forma robusta los productos más complejos de la industria. Está basado en una tecnología de representación de límites de alta precisión (14).
- **Stereolithography**: Permite que la creación de objetos tridimensionales sea más rápida, exacta, confiable y económica.
- **Pro/EngineerWildfire**: Es un producto de diseño, fabricación e ingeniería asistida por computadora. Como uno de sus principales beneficios resalta que dentro de las herramientas de su rango es una de las más fácil de adquirir económicamente (15).
- **Unigraphics**: Es un completo sistema CAD/CAM/CAE en tres dimensiones, con una estrecha relación entre todos sus módulos. Combinando todos ellos se puede diseñar, analizar, crear planos y programas de mecanizado del producto (16).
- **Solid Edge**: Es un programa de diseño asistido por computadoras de piezas tridimensionales que permite el modelado de distintos materiales, doblado de chapas, ensamblaje de conjuntos, soldadura y funciones de dibujo en plano para ingenieros (17).

Se decide no utilizar ninguno de estos modeladores geométricos ya que varios de estos fueron elaborados bajo licencias privativas, como es el caso de **ACIS** y **Solid Edge**. Otro de los factores por los cuales se decide no emplear dichos modeladores es que los mismos no se ajustan con la arquitectura propuesta para herramienta GALBA-CAD.

Para lograr el modelado de sólidos en la herramienta GALBA-CAD y teniendo en cuenta las limitantes mencionadas con anterioridad de los modeladores existentes, se decide crear un modelador geométrico que se ajuste a la arquitectura de dicha herramienta. Para ello se consultaron la biblioteca **VTK** y la biblioteca **Open Cascade (OCC)**.

## 1.4 Biblioteca VTK y OPEN CASCADE

---

La biblioteca **VTK** es una biblioteca de clases en C++ compuesta por un conjunto de herramientas para el trabajo con gráficos 3D, procesamiento de imágenes y visualización de código abierto. Esta biblioteca es multiplataforma y soporta una amplia variedad de algoritmos de visualización (escalar, vectorial, tensorial, textura y métodos volumétricos), y avanzadas técnicas de modelado como: modelos implícitos, reducción de polígonos, suavizados de malla, corte y contorno. Cuenta además con un extenso marco de visualización, y un conjunto de **widgets** de interacción 3D. Se decide no emplear dicha biblioteca ya que la misma para la visualización de gráficos realiza un gran consumo de memoria, además de no poseer funcionalidades que faciliten el desarrollo de modeladores geométricos (18).

La biblioteca **Open Cascade** es una biblioteca disponible en código abierto. La misma incluye componentes elaborados en C++ para el modelado de sólidos y superficies, visualización, intercambio de datos y desarrollo rápido de aplicaciones (19). Se decide el uso de esta biblioteca ya que la misma está especializada en la creación de herramientas CAD/CAE/CAM proporcionando un conjunto de funciones que facilitan el desarrollo de este tipo de herramientas así como de modeladores geométricos utilizados por las mismas. Además de lo antes expuesto, dicha biblioteca ha sido utilizada por varios clientes comerciales en diversos sectores desde su inclusión en la industria pesada.

Teniendo en cuenta lo expuesto en el epígrafe 1.3 el modelador geométrico desarrollado, con la ayuda de la biblioteca **Open Cascade** y para la herramienta GALBA-CAD, permite modelar piezas 2D y 3D contando con cada uno de los módulos referidos en dicho epígrafe. De las operaciones contenidas por cada uno de estos módulos el modelador geométrico desarrollado no cuenta con la operación **Shell** la cual es tema en el presente trabajo y sobre la que se profundizará en el epígrafe siguiente.

## 1.5 Operación Shell

---

Se define el **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 (20).

La operación Shell 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 (21).

Para la definición de estos vaciados se especifica 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.

A continuación se muestra una imagen que ejemplifica el proceso descrito anteriormente:

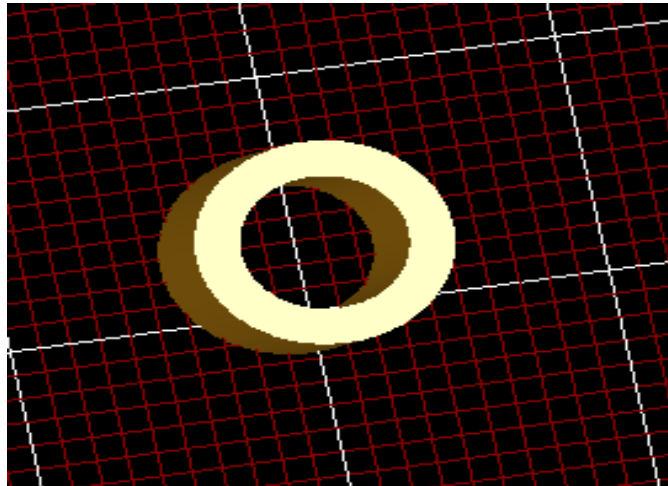


Figura 1 Aplicación del *Shell* a un Sólido

Se identifican tres tipos de **Shell** en los estudios realizados sobre el tema:

- **Shell** 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.
- **Shell** 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.
- **Shell** híbrido: Se aplican los dos tipos de **Shell** expuestos con anterioridad.

En algunas herramientas dedicadas al diseño en general, como es el caso de 3dmax, después de definido el vaciado este puede ser modificado por una serie de parámetros (20) que son descritos a continuación:

- **Cantidad de Caras Interiores/ Exteriores:** La cantidad de caras interiores o exteriores, se define como la distancia existente al mover la superficie interna hacia el interior y la superficie externa hacia el exterior, de sus posiciones originales.
- **Segmentos:** Se crea un plano de corte que puede manipularse para cortar un objeto por donde interese o para refinar la malla resultante, según las opciones que se activen. Resulta mucho más práctico que una operación binaria en bastantes casos propios de modelos arquitectónicos. El número de subdivisiones para cada borde por defecto es 1. Este parámetro solo es cambiado si es necesario una mayor resolución en el borde para el uso de modelos posteriores u otros modificadores.
- **Bordes Biselados:** Se utiliza para definir el perfil de los bordes y la resolución.
- **Spline Biselado:** Permite introducir inclinaciones en la dirección de extrusión, suavizar los bordes y añadir tres niveles diferentes en dicha dirección. Se aplica con frecuencia a textos pero puede aplicarse a cualquier otra forma.
- **Cambio de Material Exterior:** Especifica los materiales que serán aplicados a todos los polígonos exteriores a la superficie. De estar deshabilitada esta operación se usan los materiales originales de las caras.
- **Cambio de Material Interior:** Especifica los materiales que serán aplicados a todos los polígonos interiores a la superficie. De estar deshabilitada esta operación se usan los materiales originales de las caras.
- **Material Exterior:** Especifica el material para las caras exteriores del polígono. Para su uso es necesario que la operación cambio de material exterior esté habilitada.
- **Material Interior:** Especifica el material para las caras interiores del polígono. Solamente está disponible cuando la función cambio de materiales interiores está habilitada.
- **Cambio de Material de Bordes:** Especifica él o los materiales que serán aplicados a los bordes. De no especificar ninguno se toma por defecto los materiales originales de las caras de los cuales los bordes se derivan.

- **Material de Bordes:** Especifica el material de las caras del borde. Para su uso es necesario habilitar la función cambio de material de los bordes.
- **Auto Suavizado de Bordes:** Se aplica automáticamente. Los ángulos basados en la suavización son aplicados a los bordes de las caras usando los parámetros de los ángulos.
- **Ángulo:** Especifica el ángulo máximo que puede existir entre las caras de los bordes que se suavizan a través de la operación auto suavizado de bordes. Su uso depende de que la operación con anterioridad mencionada este activa. Es necesario aclarar que las caras con un ángulo mayor que el especificado no se suavizan.
- **Cambio de Suavizado de Grupos:** Permite especificar un grupo de suavizado de bordes a los nuevos polígonos usando la configuración de la función suavizado de grupos. Solamente está habilitada cuando la operación de auto suavizado de bordes esta deshabilitada.
- **Suavizado de Grupos:** Establece el grupo de suavizado de bordes de los polígonos. Solamente está activa cuando la función cambio de suavizado de grupo está habilitada. Cuando la operación de suavizado de grupos establece como valor por defecto el 0 indica que no hay suavizado que aplicar a los bordes de polígonos. Los valores que se pueden aplicar a dicha función oscilan entre 1 y 32.
- **Mapeo de Bordes:** Especifica el tipo de mapeado de texturas que se aplica a los nuevos bordes. Los tipos de mapeado disponibles son:
  - **Copia:** Copia cada borde de las caras usando las mismas coordenadas UVM que las caras originales del que se deriva.
  - **Ninguno:** Cada borde de cara le asigna a U un valor de 0 y a V un valor de 1 Por tanto cuando se realice el mapeado los bordes tomarán el color del borde superior izquierdo.
  - **Tira:** Los bordes son mapeados a través de tiras continuas.
  - **Interpolación:** El mapeado de bordes se interpola a partir de los polígonos adyacentes a las superficies exteriores e interiores.

- **Desplazamiento**: Determina la separación de los vértices de la textura a través de la habilitación de los bordes. Al incrementar este valor, aumenta la repetición de los mapas de texturas a través de los bordes de los polígonos.
- **Selección de Bordes**: Selecciona las caras del borde. Esta selección pasa a la pila otros modificadores.
- **Selección de Caras Interiores**: Selecciona las caras internas. Esta selección pasa a la pila otros modificadores.
- **Selección de caras exteriores**: Selecciona las caras externas. Esta selección pasa a la pila otros modificadores.
- **Ajuste de Esquinas**: Ajusta los vértices de las esquinas para mantener la línea recta de los bordes. Como se muestra en la siguiente figura:

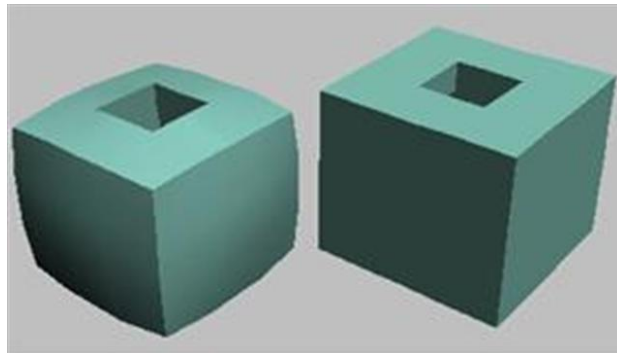


Figura 2 Ajuste de los vértices de las esquinas usando la propiedad *de ajuste de esquinas*

De los parámetros descritos con anterioridad, en la solución, solo se llevará a cabo la realización del modificador de cantidad de caras interiores y exteriores ya que la utilización de estos permite la obtención de los distintos tipos de **Shell** y en herramientas profesionales dedicadas al diseño mecánico, como es **Autodesk Inventor**, solo se tienen en cuenta estos, debido a que los mismos proporcionan realismo a los diseños. En el caso de los restantes parámetros a pesar de que agregan mayor funcionalidad al módulo no son de interés para los ingenieros mecánicos.



Después de estudiadas las bibliografías referentes al tema del presente trabajo y adquirido los conocimientos necesarios se procede al análisis desde el punto de vista de ingeniería de software de forma tal que se logre proporcionar una solución viable a dicho problema y aplicar eficazmente todos los conceptos definidos.

## Capítulo 2 Características del sistema

---

Durante este capítulo se describe el sistema desde la perspectiva de Ingeniería de Software, usando el Proceso Unificado de Desarrollo como metodología. Se presentan las reglas específicas del negocio y el modelo de dominio del problema. Posteriormente se muestran los requisitos funcionales y no funcionales detectados durante la captura de requisitos y el modelo de casos de uso del sistema; dentro de este último, los actores del sistema, los casos de uso y sus respectivas descripciones. Del flujo de trabajo diseño del sistema se muestran los diagramas de secuencia correspondientes a los casos de uso.

### 2.1 Reglas del negocio

---

La herramienta GALBA-CAD carece en su diseño de un módulo **Shell**. Esto acarrea consigo que el proceso de construcción de estructuras mecánicas sea en determinados momentos engorroso, trayendo como consecuencias que la productividad en dicho proceso no sea la mejor. Para ello se crea un módulo el cual se encargará de aplicar los distintos tipos de **Shell** que estarán dados por la cantidad de caras exteriores e interiores con que cuente dicha estructura.

Para la realización de los vaciados y teniendo en cuenta lo descrito con anterioridad se establece como regla del negocio:

- Cuando se modifique la cantidad de caras exteriores no se puede modificar la cantidad de caras interiores. Lo mismo ocurre en sentido contrario, si se modifica la cantidad de caras interiores no se puede modificar la cantidad de caras exteriores.

## 2.2 Herramientas de modelado y metodologías de desarrollo

---

Para guiar el proceso de desarrollo de la herramienta GALBA-CAD el arquitecto del proyecto CDSEM decidió utilizar como metodología de desarrollo **Rational Unified Process (RUP)**. Dicha metodología es centrada en la arquitectura, lo que permite mostrar la visión común del sistema completo y describe los elementos del modelo que son más importantes, dirigida por casos de uso, además de iterativa e incremental haciendo que los proyectos bajo esta metodología se desarrollen en fases y estas a su vez en iteraciones donde cada iteración resultará un incremento del proceso de desarrollo (22).

Se pretende integrar el módulo desarrollado a dicha herramienta por lo que se decide usar como metodología la antes mencionada para evitar incoherencias e incompatibilidades en la integración.

Se decidió utilizar como lenguaje de modelado, UML, y como herramienta de modelado, **Visual Paradigm**, Este último utiliza el lenguaje UML para describir los distintos artefactos que se generan durante la realización de la ingeniería de software del proyecto (23).

## 2.3 Modelo de dominio

---

El modelo del dominio es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real en lugar de componentes de software (22).

El entorno se puede describir cuando el ingeniero mecánico diseña una estructura mecánica a través de las operaciones de diseño contenidas por los modeladores geométricos. A continuación se muestra la descripción del negocio con el objetivo de facilitar la comprensión del sistema.

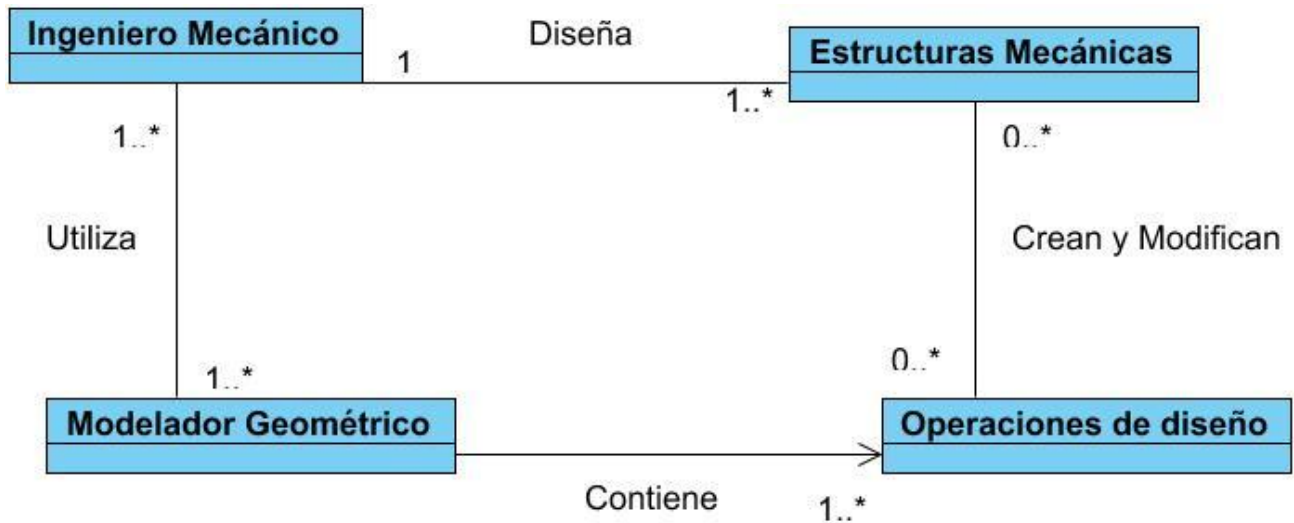


Figura 3 Diagrama de clases del dominio

- **Ingeniero Mecánico:** Es el encargado de diseñar las estructuras mecánicas y acceder al módulo para hacer uso de sus funcionalidades.
- **Modelador Geométrico:** Es una herramienta utilizada para el modelado de sólidos a partir de las operaciones de diseño que trae consigo.
- **Operaciones de diseño:** Son operaciones que permiten crear o modificar una estructura mecánica.
- **Estructuras Mecánicas:** Son todos los **Shape** o estructuras mecánicas que pueden ser creados o modificados por el ingeniero mecánico a partir de las operaciones de diseño.

## 2.4 Captura de requisitos

Podemos definir como requisito la condición o capacidad que debe exhibir o poseer un sistema o componente para satisfacer un contrato, estándar, especificación, u otra documentación formalmente expuesta facilitando el entendimiento entre clientes y desarrolladores (24).

Estos pueden clasificarse en dos tipos:

- **Requisitos funcionales:** Proporcionan una descripción de lo que un sistema debe hacer, es decir, específica algo que el sistema entregado debe ser capaz de realizar.
- **Requisitos No funcionales:** Especifican algo sobre el propio sistema, y cómo debe realizar sus funciones. Existen varios tipos de requisitos no funcionales los cuales serán abordados posteriormente.

A continuación se exponen los requisitos funcionales y no funcionales, por los cuales se rige el desarrollo del módulo:

---

#### 2.4.1 Requisitos funcionales

---

- **RF1** Aplicar **Shell** interior.
- **RF2** Aplicar **Shell** exterior.
- **RF3** Aplicar **Shell** híbrido

---

#### 2.4.2 Requisitos no funcionales

---

Los requisitos no funcionales tomados en cuenta en el presente trabajo son los que se describen a continuación:

- **Software:** La herramienta podrá ser utilizada en cualquiera de los sistemas operativos mencionados a continuación:
  - Windows XP.
  - Windows 7.
  - Ubuntu 10.04 o superior.
- **Hardware:** El software trabajará sobre máquinas que sean del modelo Pentium 4 en adelante, que posean memoria RAM superior a los 512 MB y tarjeta de video de gama baja.
- **Seguridad:**
  - **Fiabilidad:** Las estructuras mecánicas diseñadas deben ser representadas de manera similar a como son representadas en otras herramientas profesionales dedicadas al diseño mecánico.
  - **Integridad:** Los datos originales no deben sufrir pérdidas durante su representación.

- **Apariencia o Interfaz Gráfica de Usuario:** La interfaz gráfica de usuario debe proporcionar, de forma coherente y sencilla, interactividad para todas las funcionalidades de la aplicación.
- **Soporte:** Se brindará soporte para los sistemas operativos Windows XP, Windows 7 y Ubuntu 10.04 o superiores.
- **Restricciones en el Diseño e Implementación:** Se empleará el lenguaje de programación C++ bajo el paradigma de Programación Orientada a Objetos.

## 2.5 Modelo de casos de uso

---

La forma en que los actores usan el sistema es representada a través de los casos de usos. Estos son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del actor.

Los actores de un sistema son agentes externos, roles que las personas (usuarios) o dispositivos juegan cuando interactúan con el software. En este caso en particular quien usará el sistema será el ingeniero mecánico.

**Tabla 1 Actor Del Sistema**

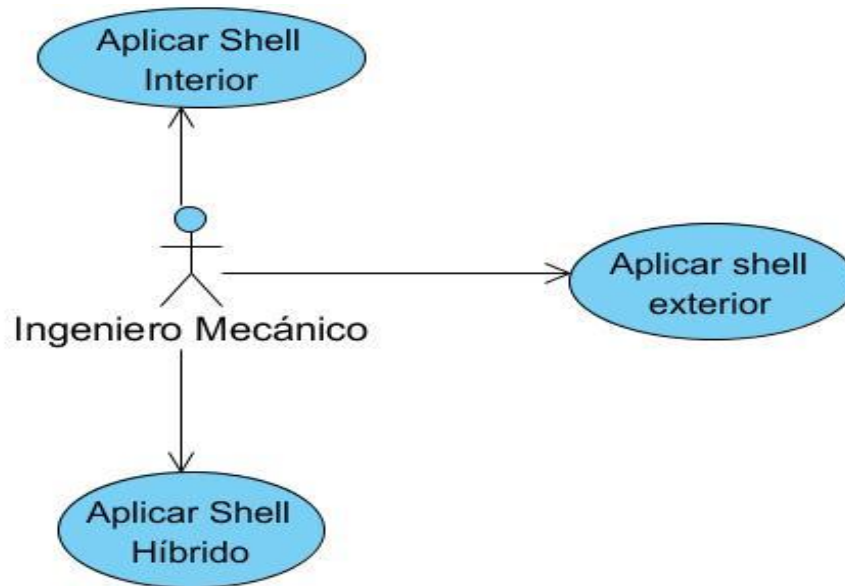
Actores	Justificación
Ingeniero mecánico	Interactúa con el módulo. Diseña estructuras mecánicas ordenando posteriormente la ejecución del módulo para realizar vaciados a las mismas.

### 2.5.1 Diagrama de casos de uso del sistema

---

El Diagrama de casos de uso del sistema (DCUS) representa gráficamente los casos de uso y su interacción con los actores.

A continuación se muestra el diagrama de casos de uso del sistema elaborado a raíz del presente trabajo:



Figuran 4 Diagrama de Casos de Uso del Sistema.

---

### 2.5.2 Descripción de los casos de uso

---

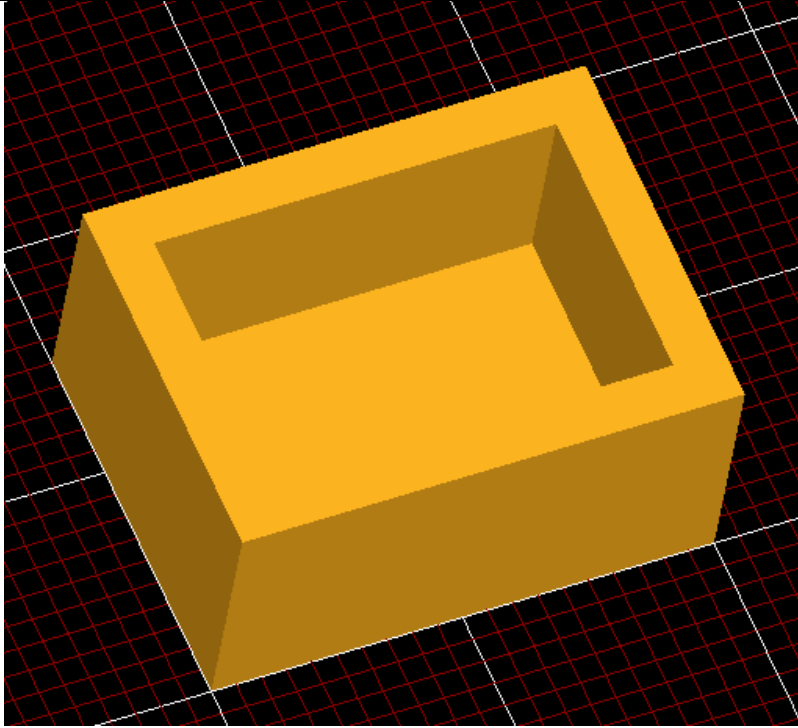
Cada caso de uso posee una descripción de las acciones que realizará el sistema como respuesta a las peticiones del usuario. A continuación se relacionan las tablas correspondientes a las descripciones de los casos de uso detectados y se argumentan los flujos operacionales de cada uno.

**Tabla 2. Descripción del caso de uso aplicar *Shell* interior.**

Caso de Uso:	Aplicar <b>Shell</b> interior.	
Actores:	Ingeniero mecánico.	
Propósito:	Aplicar <b>Shell</b> interior a la estructura mecánica representada en el visor.	
Pre-condiciones:	Debe existir al menos una estructura mecánica en el visor.	
Resumen:	El caso de uso se inicia cuando el ingeniero mecánico realiza un vaciado con un grosor determinado a una estructura mecánica, terminando así el caso de uso.	
.Referencia:	RF1.	
Flujo Normal de Eventos:		
Acción del Actor:	Respuesta del Sistema:	
1. Selecciona las superficies que desea eliminar de la estructura mecánica, establece el grosor, y aplica el <b>Shell</b> interior.	2. Verifica que se haya seleccionado al menos una superficie. 3. Captura las superficies seleccionadas de la estructura mecánica representada en el visor. 4. Elimina las superficies previamente definidas de las estructuras mecánicas representadas en el visor. 5. Invierte las normales de las caras. 6. Proporciona grosor a las paredes restantes del vaciado. 7. Muestra el vaciado creado. 8. Termina el caso de uso	
Flujo Alterno de Eventos:		
	2.1 El sistema muestra el mensaje de error “Debe seleccionar al menos una superficie para su eliminación.”	
Post-condiciones:	Se aplicó el <b>Shell</b> interior a una estructura mecánica.	
Prioridad:	Crítica	



Prototipo del caso de uso aplicar *Shell* interior



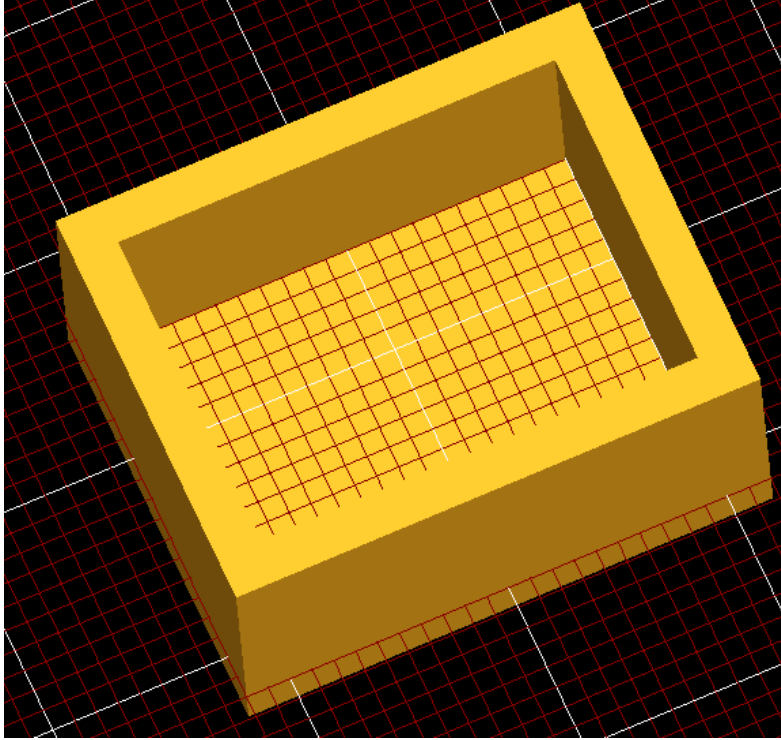
**Tabla 3. Descripción del caso de uso aplicar *Shell* exterior.**

Caso de Uso:	Aplicar <b>Shell</b> exterior.	
Actores:	Ingeniero mecánico.	
Propósito:	Aplicar el <b>Shell</b> exterior a la estructura mecánica representada en el visor.	
Pre-condiciones:	Debe existir al menos una estructura mecánica en el visor.	
Resumen:	El caso de uso se inicia cuando el ingeniero mecánico realiza un vaciado con un grosor determinado a una estructura mecánica, terminando así el caso de uso.	
.Referencia:	RF2.	
Flujo Normal de Eventos:		
Acción del Actor:	Respuesta del Sistema:	
1. Selecciona las superficies que desea eliminar de la estructura mecánica, establece el grosor y aplica el <b>Shell</b> exterior.	2. Verifica que se haya seleccionado al menos una superficie. 3. Captura las superficies seleccionadas de la estructura mecánica representada en el visor. 4. Elimina las superficies previamente definidas de la estructura mecánica representada en el visor. 5. Proporciona grosor a las paredes restantes del vaciado. 6. Muestra el vaciado creado en el visor. 7. Termina el caso de uso.	
Flujo Alternativo de Eventos:		
	2.1 El sistema muestra el mensaje de error "Debe seleccionar al menos una superficie para su eliminación."	
Post-condiciones:	Se aplicó el <b>Shell</b> exterior a una estructura mecánica.	

Prioridad:

Crítica

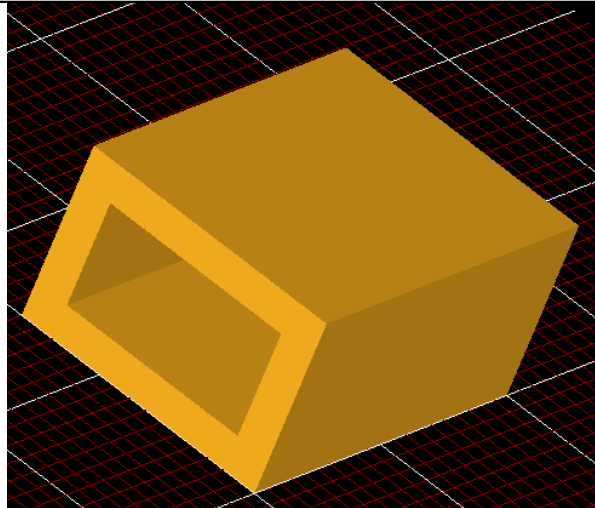
Prototipo del caso de uso aplicar *Shell* exterior



**Tabla 4 Descripción del caso de uso aplicar *Shell* híbrido.**

Caso de Uso:	Aplicar <b>Shell</b> híbrido
Actores:	Ingeniero mecánico.
Propósito:	Aplicar <b>Shell</b> a una estructura mecánica.
Pre-Condiciones:	Debe existir al menos una estructura mecánica en el visor.
Resumen:	El caso de uso se inicia cuando el ingeniero mecánico selecciona la estructura mecánica y aplica el <b>Shell</b> híbrido a la misma terminando así el caso de uso.
.Referencia:	RF3.
Flujo Normal de Eventos:	
Acción del Actor:	Respuesta del Sistema:
<ol style="list-style-type: none"> <li>1. Selecciona las superficies que desea eliminar de la estructura mecánica, establece el grosor y aplica el <b>Shell</b> híbrido.</li> </ol>	<ol style="list-style-type: none"> <li>2. Captura las superficies seleccionadas de la estructura mecánica representada en el visor.</li> <li>3. Elimina las superficies previamente definidas de la estructuras mecánica representada en el visor.</li> <li>4. Aplica el <b>Shell</b> interior en dependencia del parámetro cantidad de caras el cual establece el grosor de las paredes internas del vaciado.</li> <li>5. Aplica el <b>Shell</b> exterior en dependencia del parámetro cantidad de caras el cual establece el grosor de las paredes externas del vaciado.</li> <li>6. Muestra el vaciado creado en el visor.</li> <li>7. Termina el caso de uso</li> </ol>
Post-condiciones:	Se aplicó el <b>Shell</b> híbrido a una estructura mecánica.
Prioridad:	Crítica

Prototipo del caso de uso aplicar *Shell* híbrido



## 2.6 Diseño del sistema

---

En el flujo de trabajo de diseño la elaboración de los diagramas de clases de diseño juega un papel fundamental, estos muestran las clases finales para la realización de los casos de uso modelados anteriormente.

Los diagramas de clases de diseño muestran las clases con sus atributos y métodos y la forma en que se relacionan. En este flujo también se realizan los diagramas de interacción que muestran la comunicación y las relaciones entre los objetos, con el objetivo de dar cumplimiento a los requisitos.

La interacción de las clases en ocasiones tiende a hacerse compleja. Con el propósito de optimizar recursos y organizar el código se decide el uso de los patrones de diseño.

Los patrones de diseño son ampliamente usados en la actualidad ya que los mismos ayudan a la organización del código, encapsulan funciones, optimizan recursos y asignan responsabilidades. En el presente trabajo se decide el uso de los patrones GRASP (**General Responsibility Assignment Software Patterns**). Los mismos son patrones de software para asignación de responsabilidades que entre sus principales ventajas se encuentran:

- Mantiene el encapsulamiento.
- Los objetos utilizan su propia información para llevar a cabo sus tareas.
- Se distribuye el comportamiento entre las clases que contienen la información requerida.
- Son más fáciles de entender y mantener.

Dentro de los patrones GRASP en el módulo se propone el uso del patrón de alta cohesión. El mismo se refiere a la capacidad de que cada elemento del diseño propuesto realice una única labor

En la figura 5 se muestra el diagrama de clases del diseño correspondiente a una aplicación a la cual se le ha incluido el módulo donde se evidencia el uso de los patrones:

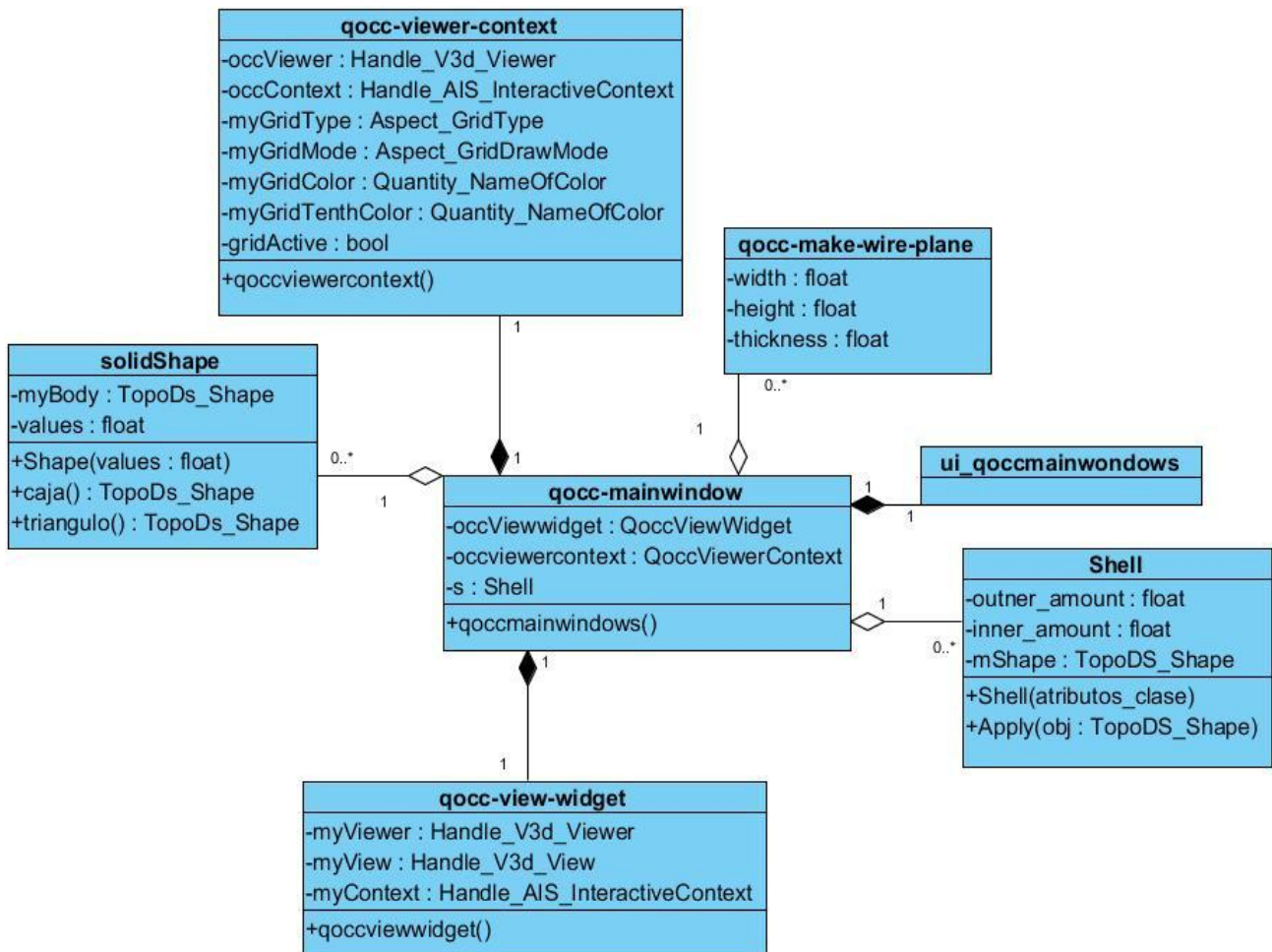


Figura 5 Diagrama de clases del diseño correspondiente al demo desarrollado

En el diagrama mostrado en la figura 5 se identifica otro patrón GRASP, el patrón creador. Dicho patrón ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Para esto la nueva instancia deberá ser creada por la clase que cumpla alguna de las reglas expuestas a continuación:

- La clase tiene la información necesaria para realizar la creación del objeto
- La clase use directamente las instancias creadas del objeto o almacene
- La clase almacena o maneja varias instancias de la clase
- Contiene o agrega las clases.

Esto último se evidencia en la figura 5, donde la clase `qocmainwindow` sigue el patrón creador porque contiene y agrega al resto de las clases y con esto crea instancias de las mismas.

## **2.7 Diagramas de secuencias**

---

Los diagramas de secuencia son diagramas que se usan para modelar la interacción entre objetos en un sistema a través del tiempo y se modela para cada caso de uso (25).

A continuación se muestran los diagramas de secuencias elaborados a partir de los casos de uso del sistema identificados:

Figura 6 Diagrama de secuencia del caso de uso aplicar *Shell* interior

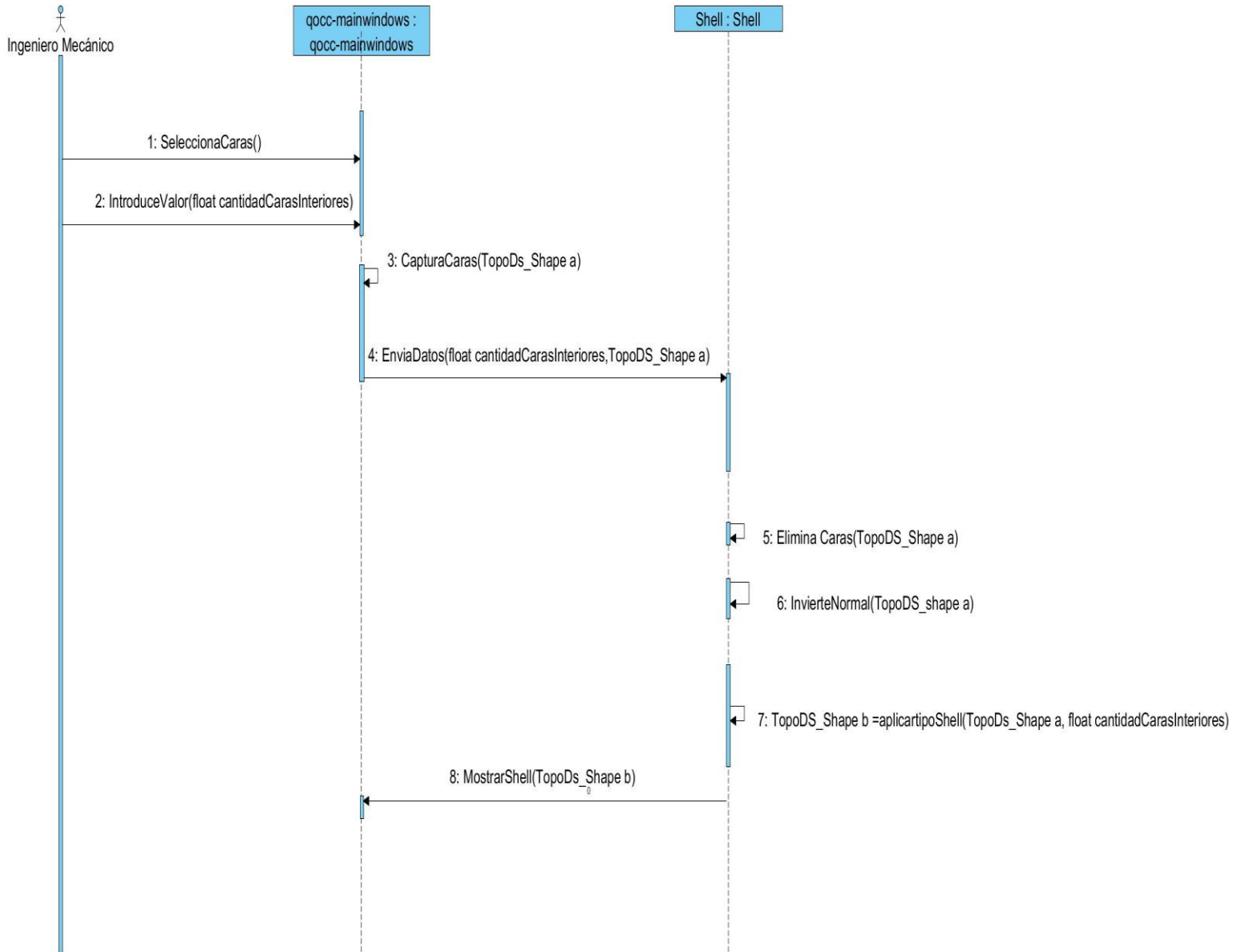




Figura 7 Diagrama de secuencia del caso de uso aplicar *Shell* exterior

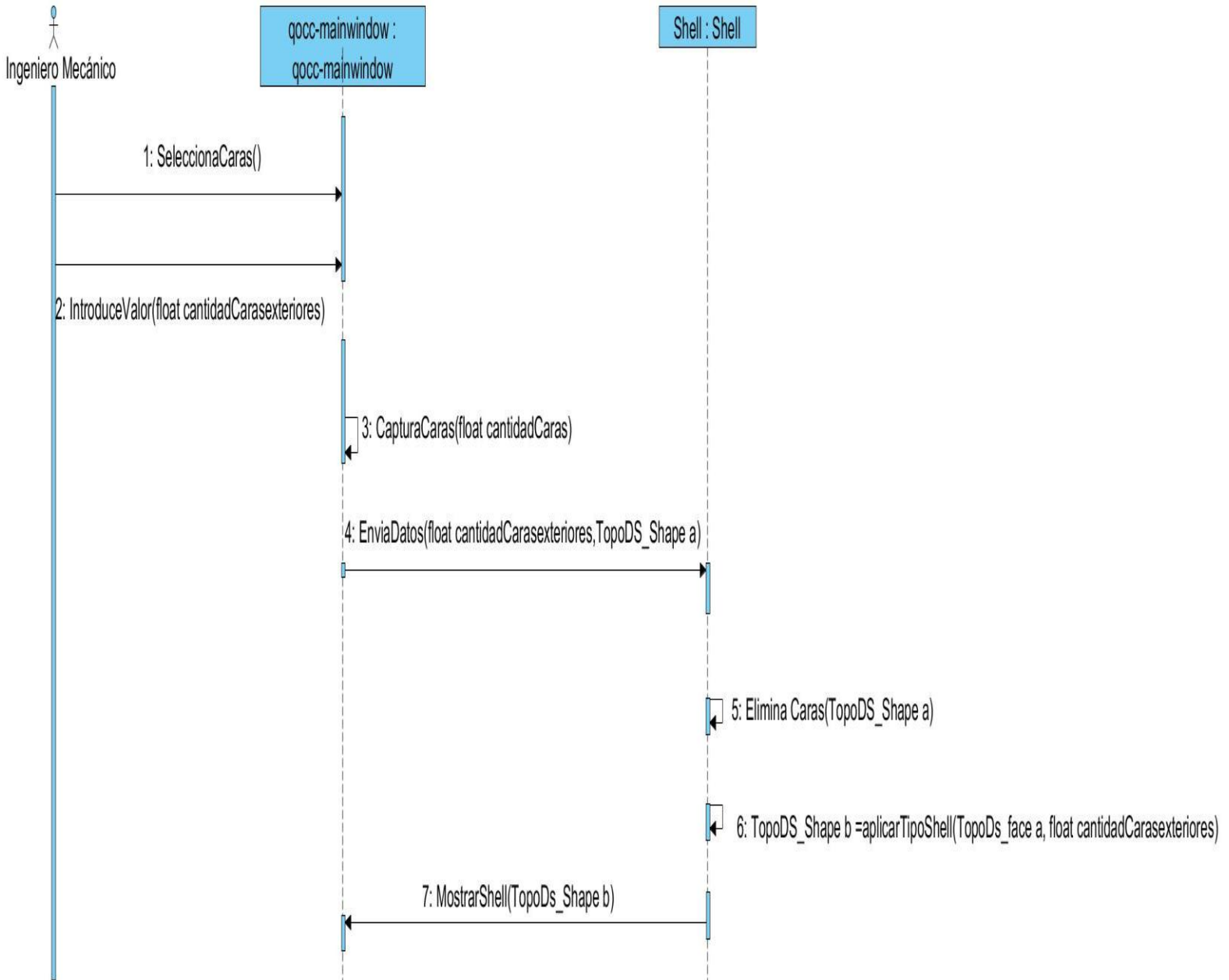
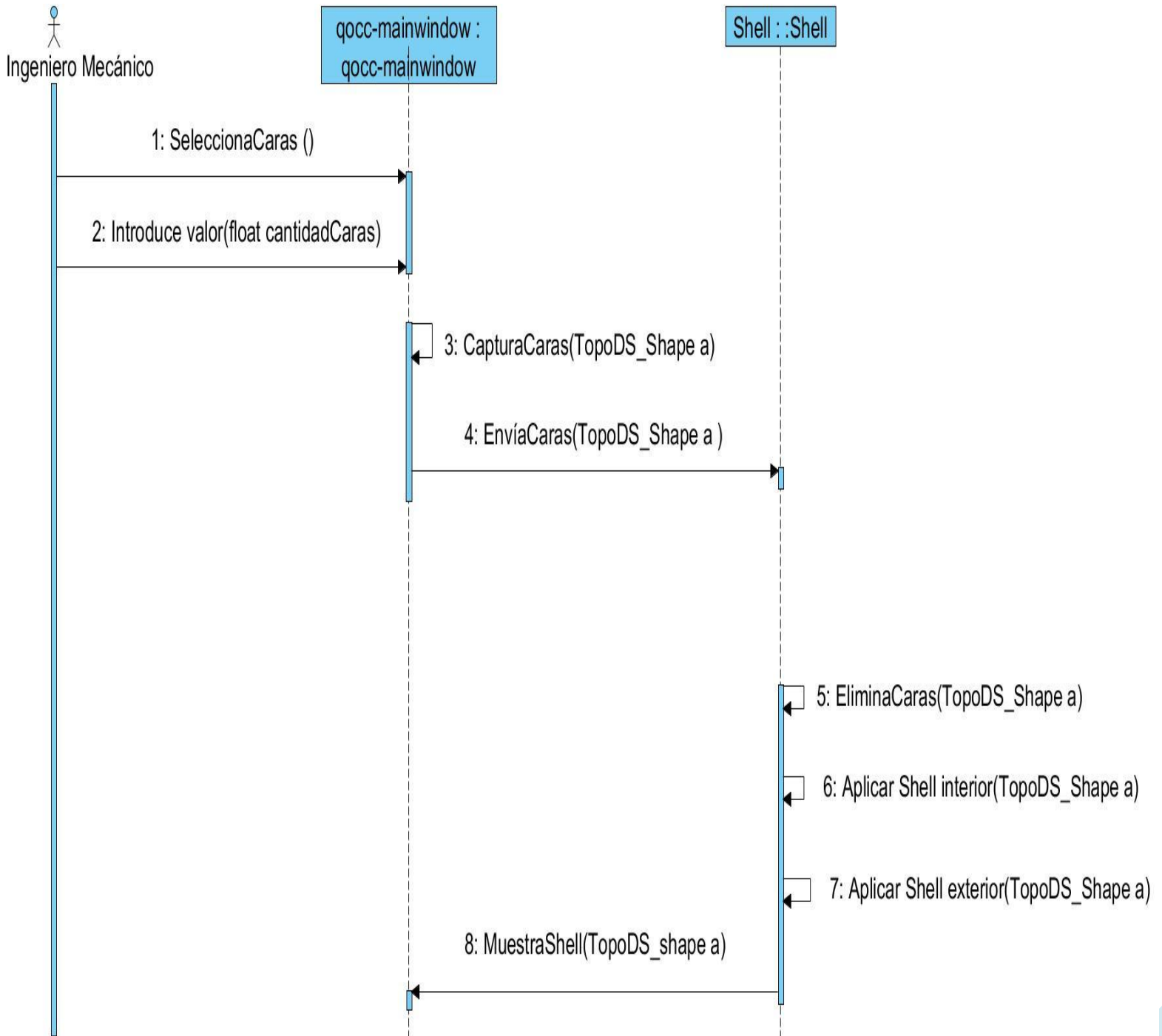


Figura 8 Diagrama de secuencia del caso de uso aplicar *Shell* híbrido



Después de analizado el sistema desde el punto de vista de ingeniería de software, logrando un mejor entendimiento del mismo, se procede a la descripción de la solución propuesta para dar respuesta a la problemática definida.

## Capítulo 3. Solución propuesta

---

En el presente capítulo se propone una solución al problema planteado con anterioridad, se describen las bibliotecas, clases y métodos usados, además de los pasos lógicos necesarios para llevar a cabo la implementación de algoritmos capaces de proporcionar respuesta a la problemática. Se suministra conjuntamente con lo antes expuesto las herramientas de desarrollo seleccionadas para la realización del presente trabajo.

### 3.1 Herramientas de desarrollo

---

Para dar solución a la problemática planteada se definió el uso de herramientas que facilitarán el proceso de desarrollo.

Para la creación de la interfaz gráfica de usuario (GUI) se utilizó el **framework QT**. Este permite la portabilidad de la aplicación hacia diferentes sistemas operativos y facilita en gran medida el desarrollo de nuevos componentes gráficos (26). Entre sus principales características se encuentran:

- Con el mismo código base, permite desplegar el sistema en múltiples plataformas.
- Producir aplicaciones de alto rendimiento con apariencia nativa.
- La concentración de los desarrolladores en la producción de código y no en las particularidades del sistema operativo.
- Acceso total al código fuente para su revisión y modificación.

No se utiliza **Qt-Creator** como IDE de desarrollo pues la integración con el subversión no es buena, además de que presenta problemas en el completamiento de código en el momento de usar los estándar de C++ y **Open Cascade**.

Debido a lo expuesto con anterioridad se elige como entorno de desarrollo integrado (IDE) el uso de **Visual Studio**. Este es uno de los entornos de programación más completos que existen en la actualidad y el cual cuenta con la corrección automática de errores y el auto-completado inteligente (27).

El módulo desarrollado utiliza para el cumplimiento de su propósito además de lo antes expuesto, la biblioteca **Open Cascade**, que brinda funcionalidades que posibilitan la realización de la operación **Shell**.

### 3.2 Estructuras utilizadas de Open Cascade

---

**Open Cascade** proporciona un conjunto de estructuras que facilitan el desarrollo de modeladores geométricos, así como las operaciones que integran cada uno de sus módulos. Entre las estructuras empleadas se encuentran:

**TopoDS:** Proporciona métodos para almacenar objetos de la clase **TopoDS\_Shape** (28).

**TopoDS\_Shape:** Se encarga de describir una forma en términos de geometría. Dicha forma posee orientación y ubicación. Los **TopoDS\_Shape** se pueden clasificar de diferentes formas, cada una de ellas se encarga de manejar el **shape** de forma diferente. En la figura siguiente se muestran las diferentes clasificaciones de los **TopoDS\_Shape** (28):

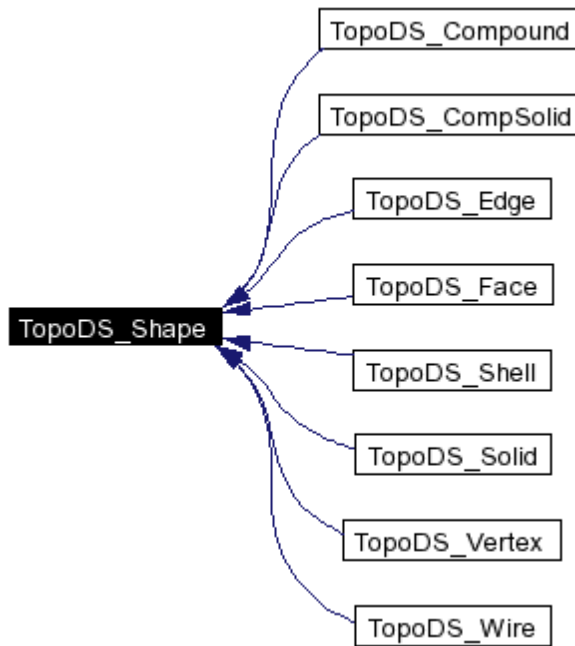


Figura 9 Clasificación de los TopoDS\_Shape

De las clasificaciones de los **TopoDS\_Shape** expuestas en la figura 9, las empleadas fueron:

**TopoDS Wire:** Describe un alambre que proporciona una orientación y ubicación en términos geométricos (28).

**TopoDS Face:** Describe cara que brindan ubicación y orientación en términos geométricos (28).

**TopoDS Edge:** Describe un borde que hace referencia a un límite que proporciona una orientación y ubicación (28).

**TopoDS Compound:** Describe un compuesto de varios **TopoDS\_Shape**. Dicho compuesto proporciona una ubicación y orientación en términos geométricos (28).

Otras de las estructuras usadas fue el **TopExp\_Explorer** la cuales utilizada para visitar los datos topológicos y formar el paquete **TopoDS** (28). Para la construcción de un explorador es necesario tener en cuenta los aspectos expuestos a continuación:

- El **Shape** a explorar.
- El tipo de **Shape** a buscar
- El ejemplo de **Shape** a evitar. Por defecto el tipo es **Shape**.

El explorador visita a toda la estructura para encontrar formas del tipo solicitado que no son contenidas en el prototipo a evitar.

### 3.3 APIS UTILIZADAS DE OPEN CASCADE

---

Para la realización de los algoritmos y clases propuestas como solución al problema planteado en el presente trabajo se usaron dentro de las **APIs** con que cuenta **Open Cascade** tres paquetes fundamentales: el **BrepBuilderAPI**, el **BrepPrimAPI** y el **BrepOffsetAPI** en las que se profundizará a continuación.

#### 3.3.1 BRepBuilderAPI

El paquete **BrepBuilderAPI** proporciona **APIs** para la estructura topológica de datos **Brep** (28). El paquete **BrepBuilderAPI** posee un conjunto de clases destinadas a proporcionar alto nivel y llamadas simples a cualquier operación común, así como mantener el acceso al bajo nivel a través de llamadas de alto nivel. Ejemplo de esto último es la cobertura completa de los modelos que incluye:

- Creación de vértices, aristas, caras y sólidos.
- Operaciones de barrido.
- Operaciones binarias.
- Cálculo de las propiedades globales.

Dentro de las **APIs** que integran este paquete podemos mencionar: **BRepBuilderAPI\_MakeWire** y **BrepBuilderAPI\_Copy**, las cuales serán abordadas a continuación.

#### **BRepBuilderAPI\_MakeWire**

Se define **BRepBuilderAPI\_MakeWire** como una API que es utilizada para la construcción de alambres a partir de bordes. Un alambre puede ser construido a partir de cualquier número de aristas. Para la construcción de estos se inicializa y a continuación se añade un número ilimitado de bordes. Dicha inicialización puede ser sin bordes, con bordes de un alambre existente o con la unión de cuatro bordes conectables (28).

Un objeto **MakeWire** proporciona un marco para inicializar la construcción de un alambre, la adición de bordes para la malla en construcción y consultar el resultado.

La **API BrepBuilderAPI\_MakeWire** es usada en el método **Apply**, que será descrito más adelante, en el proceso de construcción del **Shell** a partir de una malla.

### **BrepbuilderAPI\_Copy**

La **API BrepBuilderAPI\_Copy** permite realizar el duplicado de una forma o **Shape**. Proporcionando un marco para la aplicación de la construcción del algoritmo y la consulta del resultado.

Esta **API** es utilizada por el método **Apply** para realizar copias a las estructuras mecánicas creadas permitiendo así manipular las mismas sin afectar el modelo original.

### **3.3.2 BRepPrimAPI**

El paquete **BRepPrimAPI** contiene **APIs** que tienen como propósito la construcción de objetos tales como: cajas, esferas, cilindros, entre otros. Dentro de este paquete se encuentra la API **BRepPrimAPI\_MakePrism** la cual es imprescindible para la aplicación del **Shell** a una malla, realizada en el método **Apply**.

Dicha **API** describe las funciones de barrido lineal para la construcción de topologías a través del llamado de prismas (28).

Para la definición de un prisma es necesario tener una forma base que es barrida y la dirección del barrido.

Un objeto **MakePrism** proporciona un marco para la definición y construcción de un prisma, la implementación del algoritmo de construcción de prismas y la consulta de resultados.

### **3.3.3 BRepOffsetAPI**

Este paquete es parte fundamental en la solución debido a que contiene las **APIs** utilizadas para la realización de los distintos tipos de **Shell**.

Dentro de este paquete existen dos **APIs** fundamentales: la **BRepOffsetAPI\_MakeOffsetShape** y **BRepOffsetAPI\_MakeThickSolid**.

La primera se encarga de describir las funciones de construcción de un **Shell** cuyo resultado es un **Shape** acíclico paralelo al original (28).

En el caso de la **API BRepOffsetAPI\_MakeThickSolid** permite la creación de sólidos huecos con un grosor que puede ser determinado por el usuario (28).



### 3.4 Clase Shell

---

La clase **Shell** diseñada como parte de la solución tiene como objetivo aplicar los distintos tipos de **Shell** a las estructuras mecánicas, de manera dinámica a partir de atributos y métodos que requiere dicha funcionalidad. Las variantes de la operación **Shell** pueden ser aplicadas tanto a planos, como a sólidos.

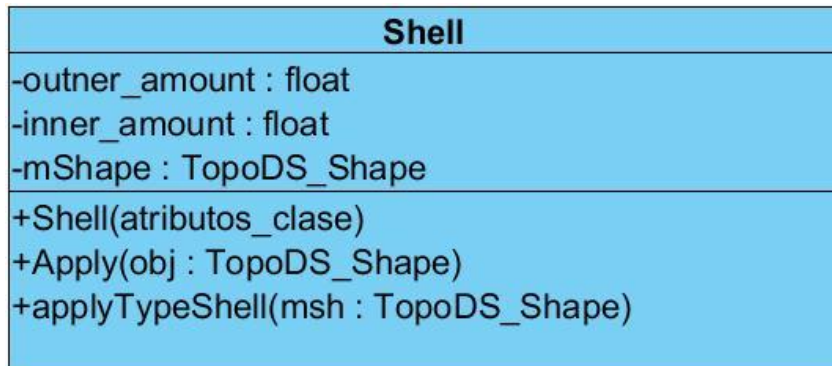


Figura 10 UML de la Clase Shell con los atributos de la misma.

Dentro de la clase **Shell** fueron implementados los métodos fundamentales para brindar funcionalidad al módulo los que gestionan todo el proceso de construcción de los distintos tipos de **Shell**. Dichos métodos son: el método **applyTypeShell** el cual aplica el **Shell** interior y exterior a las estructuras mecánicas y el método **Apply** encargado de aplicar el **Shell** híbrido. Para un mejor entendimiento de estos algoritmos se explica en los epígrafes 3.4.1 y 3.4.2 respectivamente cómo funcionan los mismos. También se mostrarán los diagramas de componentes para brindar una mejor comprensión de la solución en el epígrafe 3.5.

#### 3.4.1 Método ApplyTypeShells

El método **ApplyTypeShells** tiene como función principal aplicar el **Shell** interior o el **Shell** exterior a la estructura mecánica representada en el visor. Para ello es necesario inicialmente definir las caras del sólido que se desean suprimir, posteriormente se establece el grosor de las paredes del vaciado.

En el caso de aplicar el **Shell** interior antes de establecer el grosor es importante invertir la dirección de la normal de las caras del vaciado.

A continuación se muestra el pseudocódigo del mismo:

<b>Pseudocódigo del algoritmo ApplyTypeShells</b>
<p><b>Entradas:</b></p> <p>msh: Estructura mecánica.</p> <p>facesToRemove: Lista de caras a eliminar.</p> <p>aPnt: Punto de ubicación de la estructura mecánica.</p> <p>mybody: Resultado de aplicar el <b>Shell</b>.</p> <p>aVariable: Dirección de la cara</p> <p>cantidadCaras: Grosor del vaciado.</p>
<p><b>Para todo</b> Vi que pertenece a V <b>hacer</b> donde V es la lista de caras de la estructura msh.</p> <p>Convertir Vi en una superficie</p> <p><b>si</b> Vi es un plano <b>entonces</b></p> <p>almacena la ubicación del plano en aPnt</p> <p>almacena la dirección de la cara que se encuentra en el aPnt en aVariable</p> <p>adiciona a la lista facesToRemove la cara que se encuentra en esa dirección</p> <p><b>fin condición</b></p> <p><b>fin ciclo</b></p> <p>elimina las caras que se encuentran en la lista facesToRemove</p> <p><b>si</b> cantidadCaras es menor que 0 <b>entonces</b></p> <p>invierte la dirección de la caras restantes</p> <p>añade grosor al resto de las paredes y almacena el resultado en mybody</p> <p><b>Sino entonces</b></p> <p>añade grosor al resto de las paredes y almacena el resultado en mybody</p> <p><b>fin condicion</b></p> <p>devuelve mybody</p> <p><b>fin algoritmo</b></p>

Figura 11 Pseudocódigo del algoritmo ApplyTypeShells.

### 3.4.2 MétodoApply

**Apply** es la funcionalidad encargada de crear y aplicar el Shell híbrido a la estructura mecánica existente en el visor. Para ello elimina las superficies seleccionadas por el usuario y posteriormente aplica el **Shell** interior y el **Shell** exterior dándole grosor a las superficies restantes.

A continuación se muestra el pseudocódigo del mismo:

Pseudocódigo del algoritmo Apply
<p><b>Entradas:</b></p> <p>ms: Estructura mecánica.</p> <p>facesToRemove: Lista de caras a eliminar.</p> <p>aPnt: Punto de ubicación de la estructura mecánica.</p> <p>mybody: Resultado de aplicar el <b>Shell</b>.</p> <p>aVariable: Dirección de la cara</p> <p>cantidadCaras: Grosor del vaciado.</p> <p>aSurface: Superficie de una estructura mecánica</p>

**si** ms es una malla **entonces**

convertir la malla en cara

convertir la cara en superficie

convertir la superficie en plano y almacenarla en asurface

**fin de la condición**

**si** aSurface es un plano **entonces**

convierte la cara en prisma

almacena el prisma en mybody

**fin de la condición**

elimina las caras que se encuentran en la lista facesToRemove

cantidadCaras toma valor negativo

invierte la dirección de la caras restantes

añade grosor al resto de las paredes y almacena el resultado en mybody

cantidad de caras toma valor positivo

añade grosor al resto a las paredes de la estructura mybody

devuelve mybody

**fin del algoritmo**

Figura 12 pseudocódigo del algoritmo Apply

### 3.5 Diagrama de componentes del módulo desarrollado

---

Un componente representa una parte física del sistema, por ejemplo, una biblioteca, un ejecutable o una tabla, que abarca la implementación de un grupo de clases del diseño. Cada componente define una interfaz que describe su funcionalidad y forma de empleo. El diagrama de componentes, permite conocer a los desarrolladores y clientes la estructura física que tiene el sistema y cómo se relacionan sus partes.

Para un mejor entendimiento del módulo se decidió agruparlos en dos partes:

- El primer grupo, que se mostrará en la figura 13, estará conformado por el ejecutable, las bibliotecas y las relaciones existentes entre ellos.
- El segundo grupo mostrado en la figura 14 está integrado por los componentes generados a partir del código fuente.

A continuación se muestran los diagramas de componentes correspondientes al módulo desarrollado:

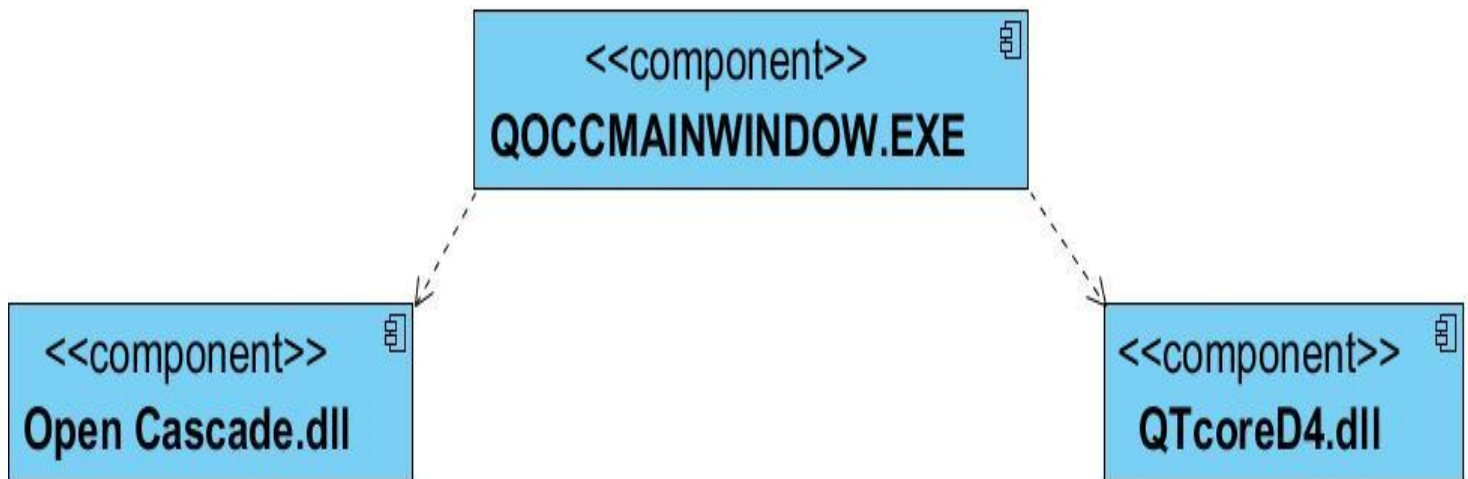


Figura 13 Diagrama de componentes del módulo desarrollado.

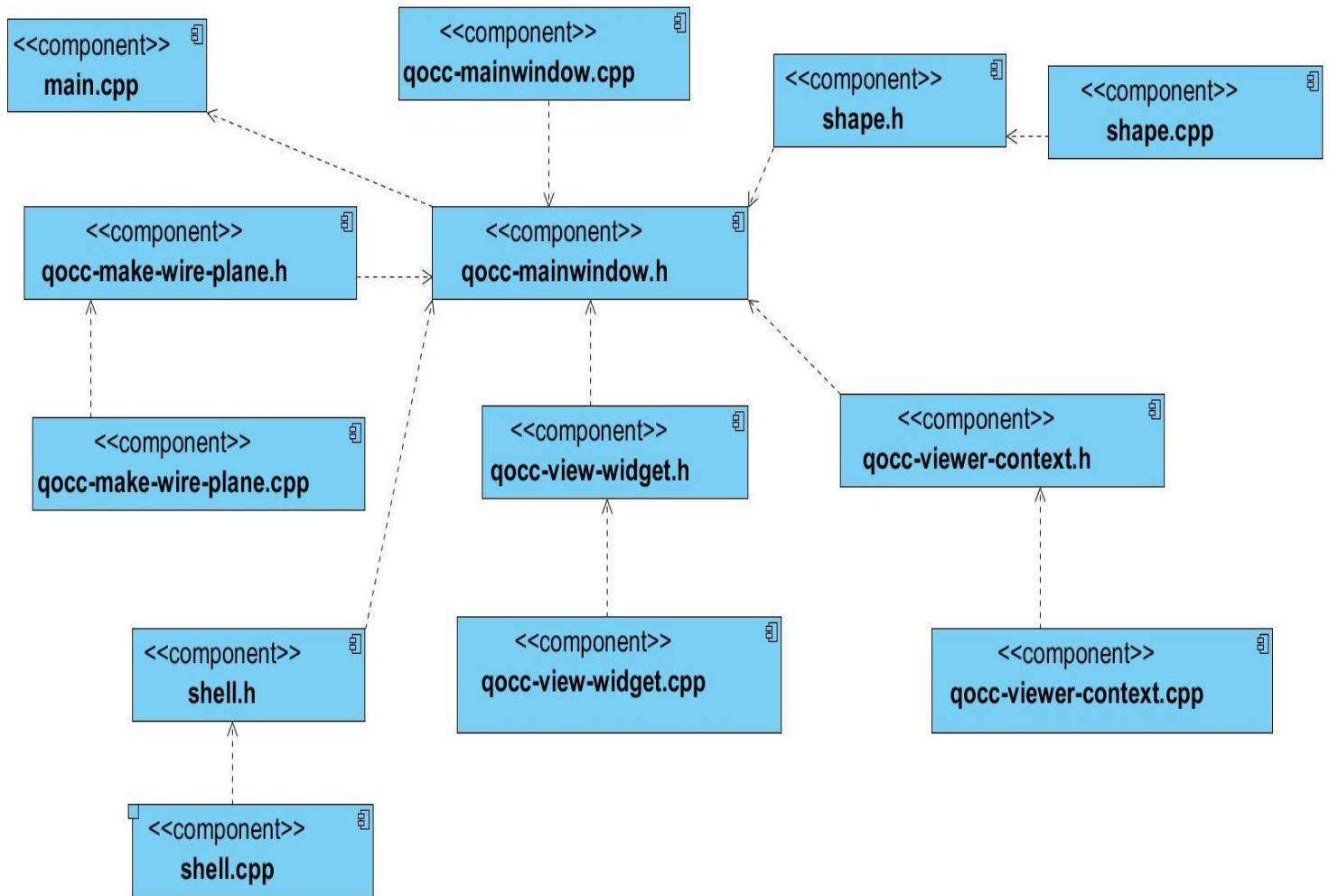


Figura 14 Diagrama de componentes del módulo desarrollado.

## 3.6 Validación de los resultados

---

En esta sección se muestran las pruebas realizadas al módulo propuesto a partir de casos de prueba, así como, los principales resultados alcanzados en el desarrollo del mismo, y la validación de estos.

**RUP** define como uno de sus flujos de trabajo el proceso de pruebas. Estas constituyen el proceso de ejecución de un programa con el fin de encontrar errores.

Siguiendo la guía que proporciona **RUP** para el desarrollo de proyectos se realizan pruebas al módulo desarrollado. Dichas pruebas tienen dos enfoques fundamentales: el enfoque estructural o de caja blanca y el enfoque funcional o de caja negra. Las pruebas de caja blanca se decidió realizarlas después de integrado el módulo a la herramienta GALBA-CAD ya que hasta el momento es solo un prototipo.

Para la realización de las pruebas de caja negra se definieron tres casos de prueba los cuales se encargan de verificar las funcionalidades del módulo así como los errores de interfaz que pueda tener.

### 3.6.1 CASOS DE PRUEBA

#### Caso de prueba basado en el caso de uso aplicar *Shell* interior

##### Descripción general.

El caso de uso aplicar **Shell** interior, realiza un vaciado eliminando las superficies seleccionadas por el ingeniero mecánico y proporciona un grosor a las paredes del vaciado, añadiendo caras hacia el interior de las mismas conectadas a través de aristas.

##### Condiciones de ejecución.

Debe existir al menos una estructura mecánica en el visor.

##### Caso de uso aplicar **Shell** interior.

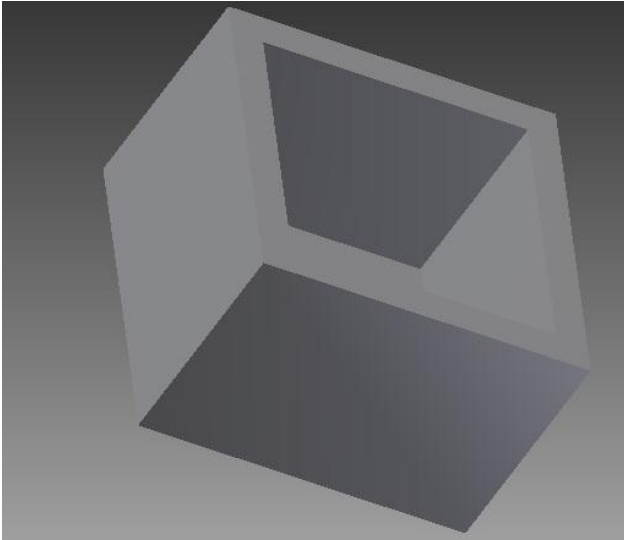
Escenario	Descripción	Caras seleccionadas	Cantidad de caras interiores	Respuesta que se espera del sistema	Respuesta del sistema	Flujo central
EC1 Introducir los valores permitidos.	Todos los campos poseen valores permitidos por el sistema.	V	V	El sistema realiza el vaciado y añade caras hacia el interior de las paredes del mismo.	La esperada	Ícono “ <b>Shell</b> ”/Selecciona superficies /introducir el valor de las caras interiores.
EC2Introducir algún valor no permitido	Alguno de los campos posee un valor que no es permitido por el sistema	V	I	El sistema muestra un mensaje de error “ la cantidad de caras no puede ser 0”	La esperada	Ícono “ <b>Shell</b> ”/Selecciona superficies/introducir el valor 0 como valor de las caras interiores.
		I	N/A	El sistema muestra un mensaje de error “Debe seleccionar al menos una cara.”	La esperada	Ícono “ <b>Shell</b> ”

**Tabla 5 Caso de prueba aplicar *Shell* interior**

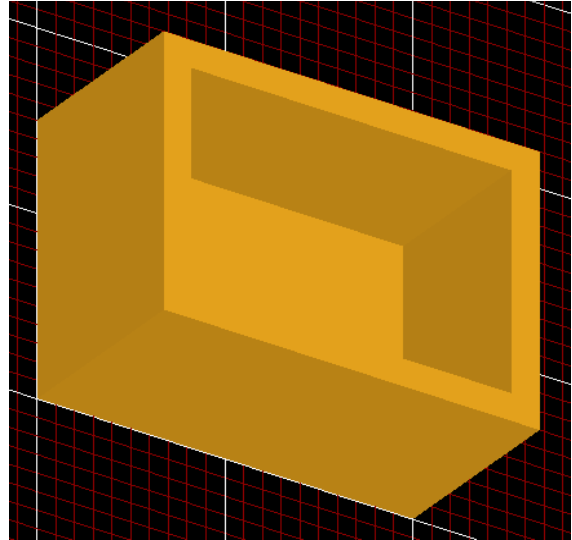
Teniendo en cuenta el concepto de **Shell** interior referido en el capítulo 1 y luego de desarrollado el caso de prueba se realizó en la herramienta **Inventor** la aplicación de dicho **Shell** a una estructura mecánica,



obteniendo resultados similares, por lo que se puede concluir que la prueba fue satisfactoria. Dicho resultado se observa en la figura 15.



**Shell** interior herramienta **Inventor**



**Shell** interior módulo desarrollado

**Figura 15** Resultado de aplicar **Shell** interior.

**Caso de prueba basado en el caso de uso aplicar *Shell* exterior.**

#### **Descripción general.**

El caso de uso aplicar **Shell** exterior, realiza un vaciado eliminando las superficies seleccionadas por el ingeniero mecánico y proporciona un grosor a las paredes del vaciado, añadiendo caras hacia el exterior de las mismas conectadas a través de aristas.

#### **Condiciones de ejecución.**

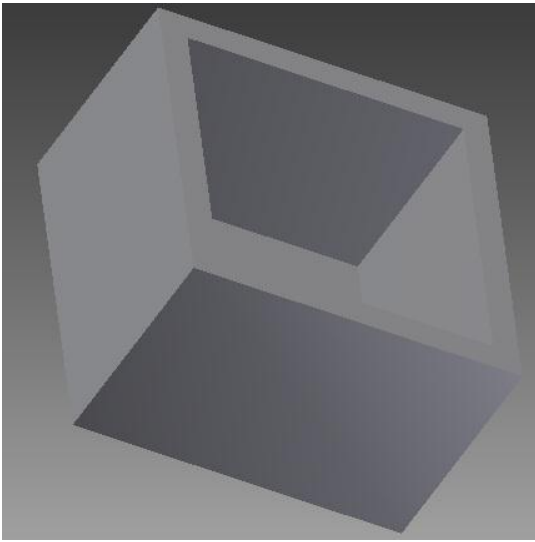
Debe existir al menos una estructura mecánica en el visor.

**Caso de uso aplicar Shell exterior.**

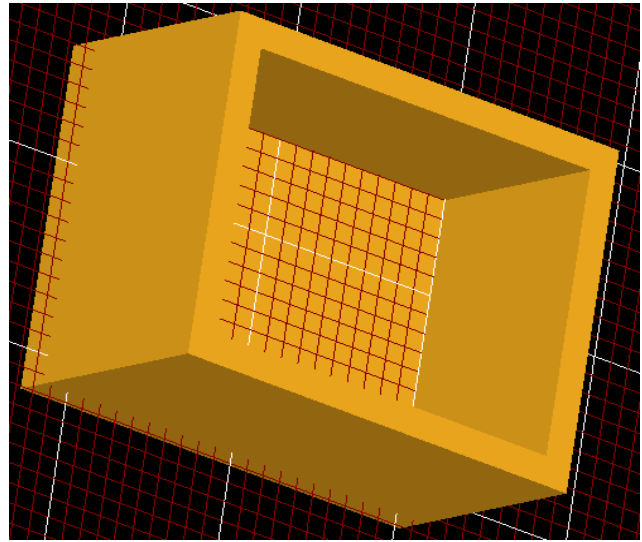
Escenario	Descripción	Caras seleccionadas	Cantidad de caras exteriores	Respuesta que se espera del sistema	Respuesta del sistema	Flujo central
EC1 Introducir los valores permitidos.	Todos los campos poseen valores permitidos por el sistema.	V	V	El sistema realiza el vaciado y añade caras hacia el exterior de las paredes del mismo.	La esperada	Ícono " <b>Shell</b> "/Selecciona superficies/introducir el valor de las caras exteriores.
EC2 Introducir algún valor no permitido	Alguno de los campos posee un valor que no es permitido por el sistema	V	I	El sistema muestra un mensaje de error " la cantidad de caras no puede ser 0"	La esperada	Ícono " <b>Shell</b> "/Selecciona superficies/introducir el valor 0 como valor de las caras exteriores.
		I	N/A	El sistema muestra un mensaje de error "Debe seleccionar al menos una cara."	La esperada	Ícono " <b>Shell</b> "

**Tabla 6 Caso de prueba aplicar *Shell* exterior**

Teniendo en cuenta el concepto de **Shell** exterior referido en el capítulo 1 y luego de desarrollado el caso de prueba se realizó en la herramienta **Inventor** la aplicación de dicho **Shell** a una estructura mecánica, obteniendo resultados similares, por lo que se puede concluir que la prueba fue satisfactoria. Dicho resultado se observa en la figura16.



**Shell** exterior herramienta **Inventor**



**Shell** exterior módulo desarrollado

**Figura 16** Resultado de aplicar **Shell** exterior.

### **Caso de prueba basado en el caso de uso aplicar *Shell* híbrido.**

#### **Descripción general.**

El caso de uso aplicar **Shell** híbrido realiza un vaciado eliminando las superficies seleccionadas por el ingeniero mecánico y proporciona un grosor a las paredes del vaciado añadiendo caras hacia el interior y el exterior de las mismas conectadas a través de aristas.

#### **Condiciones de ejecución.**

Debe existir al menos una estructura mecánica en el visor.

**Caso de uso aplicar Shell híbrido.**

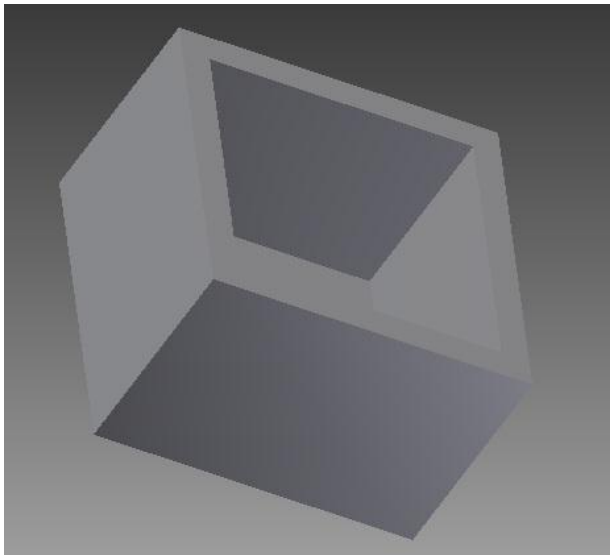
Escenario	Descripción	Caras seleccionadas	Cantidad de caras	Respuesta que se espera del sistema	Respuesta del sistema	Flujo central
EC1 Introducir los valores permitidos.	Todos los campos poseen valores permitidos por el sistema.	V	V	El sistema realiza el vaciado y añade caras hacia el interior y exterior de las paredes del mismo.	La esperada	Ícono <b>"Shell"</b> /Selecciona o no superficies /introducir el valor de la cantidad de caras.
EC2 Introducir algún valor no permitido	Alguno de los campos posee un valor que no es permitido por el sistema	V	I	El sistema muestra un mensaje de error " la cantidad de caras no puede ser 0"	La esperada	Ícono <b>"Shell"</b> /Selecciona o no superficies/ introducir el valor 0 como valor de la cantidad de caras.

**Tabla 7 caso de prueba aplicar Shell híbrido**

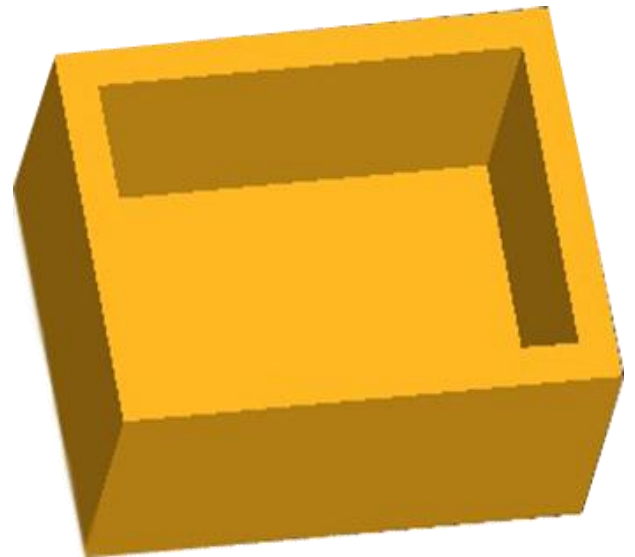
No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Caras seleccionadas	pushButton	No	Debe ser ejecutado antes de aplicar el <b>Shell</b> interior o el <b>Shell</b> exterior
2	Cantidad de caras interiores	SpinBox	No	No puede tomar valor 0.
3	Cantidad de caras exteriores	SpinBox	No	No puede tomar valor 0.

Tabla 8 Descripción de las variables de los casos de prueba

Teniendo en cuenta el concepto de **Shell** híbrido referido en el capítulo 1 y luego de desarrollado el caso de prueba se realizó en la herramienta **Inventor** la aplicación de dicho **Shell** a una estructura mecánica, obteniendo resultados similares por lo que se puede concluir que la prueba fue satisfactoria. Dicho resultado se observa en la figura 17.



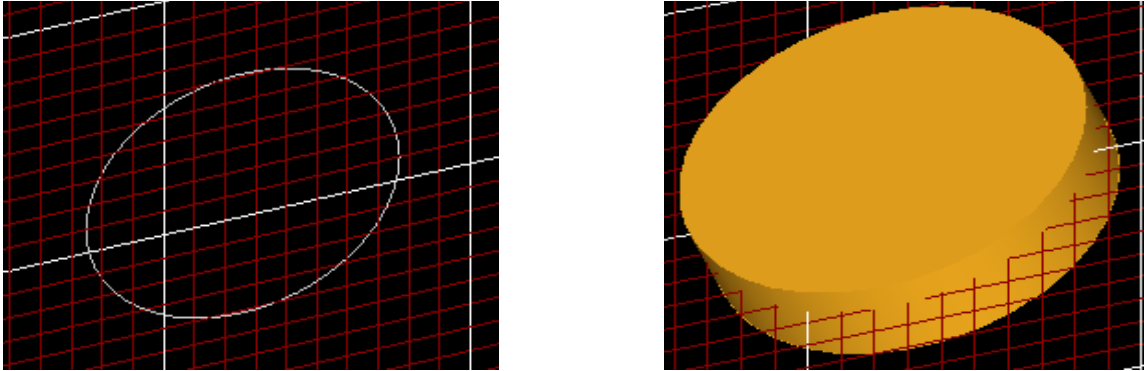
**Shell** híbrido herramienta **Inventor**.



**Shell** híbrido módulo desarrollado

Figura 17 Resultado de aplicar **Shell** híbrido.

Otro de los resultados obtenidos con el desarrollo del módulo reside en la posibilidad de aplicar **Shell** a una malla. Como se muestra en la figura 18.



**Figura 18** Aplicar Shell híbrido a una malla.

Después de descrita la solución propuesta como respuesta a la problemática planteada en este trabajo se procede a establecer las conclusiones y recomendaciones del mismo.

## Conclusiones

---

Al finalizar la presente investigación y llevado a cabo el desarrollo del módulo se establece como conclusión que la aplicación del módulo mejora la productividad en el proceso de construcción de estructuras mecánicas.

## Recomendaciones

---

Al término del trabajo se definen las siguientes recomendaciones:

- Integrar el módulo desarrollado a la herramienta GALBA-CAD.
- Permitir la aplicación simultánea a varias estructuras mecánicas de los distintos tipos de Shell en el módulo desarrollado.



## BIBLIOGRAFÍA

---

1. **azcona, gracias Juan pedro.** petroleo. *petroleo*. [En línea] 2000. [Citado el: 8 de Febrero de 2012.] <http://www.taringa.net/posts/apuntes/3287519/Perforacion-y-terminacion-de-pozos-petroleros.html>.
2. **Andrés García higuera, Francisco J castillo.** *El computador en la automatización de la solución*. s.l. : ciencia y técnica.
3. **autodesk.** AutoCAD. *AutoCAD*. [En línea] autodesk, 1982. [Citado el: 8 de Febrero de 2012.] <http://es.scribd.com/doc/60511914/Auto-Cad>.
4. **Ing. Qswaldo Rojas Lazo, Ing. Julio Salas Bacalla.** fabrica y diseño. *fabrica y diseño*. [En línea] industrial data, 1999. [Citado el: 8 de Febrero de 2012.] [http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v02\\_n1/produccion.htm](http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/indata/v02_n1/produccion.htm).
5. **Siemenes PLM.** ventajas CAD. *ventajas CAD*. [En línea] Siemenes PLM. [Citado el: 8 de febrero de 2012.] <http://admon-dana.blogspot.com/2011/11/sistemas-cad-cae-y-cam.html>.
6. **Dessaut Systems.** Catia. *Catia*. [En línea] Dessaut Systems. [Citado el: 8 de Febrero de 2012.] <http://www.3ds.com/es/products/catia/welcome/>.
7. **autodesk.** ArchicAD. *ArchicAD*. [En línea] autodesk, 1982. [Citado el: 8 de Febrero de 2012.] <http://aplicaciones.org/archicad-15-aplicacion-de-modelado-3d-alternativa-a-autocad/>.
8. **QCAD comunity edition.** QCAD. *QCAD*. [En línea] QCAD comunity edition. [Citado el: 8 de Febrero de 2012.] <http://www.ribbonsoft.com/qcad.html>.
9. **GstarCAD.** GstarCAD. *GstarCAD*. [En línea] Grafix- Gestión Informática, 2011. [Citado el: 8 de Febrero de 2012.] <http://www.datamicra.net/>.
10. **autodesk.** AutoDesk Inventor. *AutoDesk Inventor*. [En línea] autodesk, 1999. [Citado el: 8 de Febrero de 2012.] <http://usa.autodesk.com/autodesk-inventor/>.
11. **Dessaut Systems solidworks corp.** SolidWorks. *SolidWorks*. [En línea] Dessaut Systems solidworks corp, 95. [Citado el: 8 de Febrero de 2012.] <http://www.solidworks.com/sw/support/CustomBulletins.html>.
12. **Open Cascade SA.** Salome Meca. *Salome Meca*. [En línea] 2005. [Citado el: 8 de Febrero de 2012.] <http://www.salome-platform.org/>.
13. **spatial corporation.** Portal ACIS. *Portal ACIS*. [En línea] spatial corporation. [Citado el: 8 de febrero de 2012.] <http://doc.spatial.com/index.php/Portal:ACIS>.
14. **Siemens PLM software.** Parasolid. *parasolid*. [En línea] Siemens PLM software, 2008. [Citado el: 8 de febrero de 2012.] <http://www.interempresas.net/MetalMecanica/FeriaVirtual/Producto-Software-de-modelado-geometrico-3D-Parasolid-version-21-46493.html>.

15. **PTC corporation.** Pro/Engineer Wildfire. *Pro/Engineer Wildfire*. [En línea] PTC corporation, 1991. [Citado el: 8 de Febrero de 2012.] <http://www.gratisprogramas.org/descargar/proengineer-wildfire-5-m060-32-64-bits-hf/>.
16. **Leu, Pathiban Delli/ming.** *Unigraphics-NX3 Para Diseño Ingenieril*.
17. **Siemens PLM Software.** *Solid Edge*.
18. **kitware.** VTK. *VTK*. [En línea] kitware. [Citado el: 18 de junio de 2012.] [www.vtk.org](http://www.vtk.org).
19. **open cascade SA.** open cascade. *open cascade*. [En línea] 2000. [Citado el: 6 de junio de 2012.] <http://www.opencascade.org/>.
20. **autodesk.** 3dmax. *3dmax*. [En línea] autodesk, 1982. [Citado el: 9 de Febrero de 2012.] <http://usa.autodesk.com/3ds-max/>.
21. **Autodesk.** Operacion Shell. *Operacion Shell*. [En línea] Autodesk, 1999. [Citado el: 9 de Febrero de 2012.] <http://wikihelp.autodesk.com/Inventor/esp/2012/Help/0073-Autodesk73/0308-Piezas308/0353-Operacio353/0355-Creación355/0367-Operacio367>.
22. **IBM Corporation.** *Rational Unified Process*. 2006.
23. **visual paradigm international.** Visual\_Paradigm. *Visual\_Paradigm*. [En línea] visual paradigm international, 2010. [Citado el: 15 de marzo de 2012.] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(MÍ\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(MÍ)_14720_p/).
24. **Booch, Grady, Jacobson, Ivar y Rumbaugh, James.** *El proceso Unificado de desarrollo de software*. s.l. : Addison Wesley. 2000.0-201-57169-2.
25. **Universidad de alcala.** *FEA Consolidated Reference Model Document*. 2005.
26. **Royo, Javier.** *Diseño Digital*. s.l. : Ediciones Paidós Ibérica, 2004.
27. **microsoft.** visual studio. *visual studio*. [En línea] microsoft, 1998. [Citado el: 4 de abril de 2012.] <http://www.cursosporinternet.info/index.php/the-news/43-programacion/167-los-ides-de-programacion.html>.
28. **Open Cascade SA.** OCC. *OCC*. [En línea] Open Cascade SA, marzo de 2003. [Citado el: 15 de marzo de 2012.] <file:///G:/Adrian/OpenCASCADE6.3.0/doc/ReferenceDocumentation/index.html>.
29. **Serope, Kalpakjian.** *Manufactura, ingeniería y tecnología (4ª edición)*. s.l. : Pearson Educación, 2001.
30. **R, Claudia Marcela Sánchez R/ Carlos Julio Córtes.** *Rapid Casting y Nuevas Tecnologías en el proceso de Microfundición*. 2006.
31. **torres, J.C.** *Diseño Asistido por Ordenador*.

32. **Martin Fowler, Kendall Scott.** *"UML Gota a Gota"*. 1999.

33. **Helios.** PTK 2D Game Engine for Mac os X and Win32. *PTK 2D Game Engine for Mac os X and Win32*. [En línea] 2003. [Citado el: 5 de junio de 2012.] <http://www.phelios.com/ptk/index.html>.

## ANEXOS

### ANEXO A: IMÁGENES DE LA APLICACIÓN

A continuación se muestran diferentes imágenes obtenidas del sistema desarrollado a partir de la investigación realizada y otras imágenes de cómo es trabajada la operación Shell en otras herramientas.

**Figura 19** Aplicar *Shell* híbrido a una estructura mecánica en el módulo desarrollado

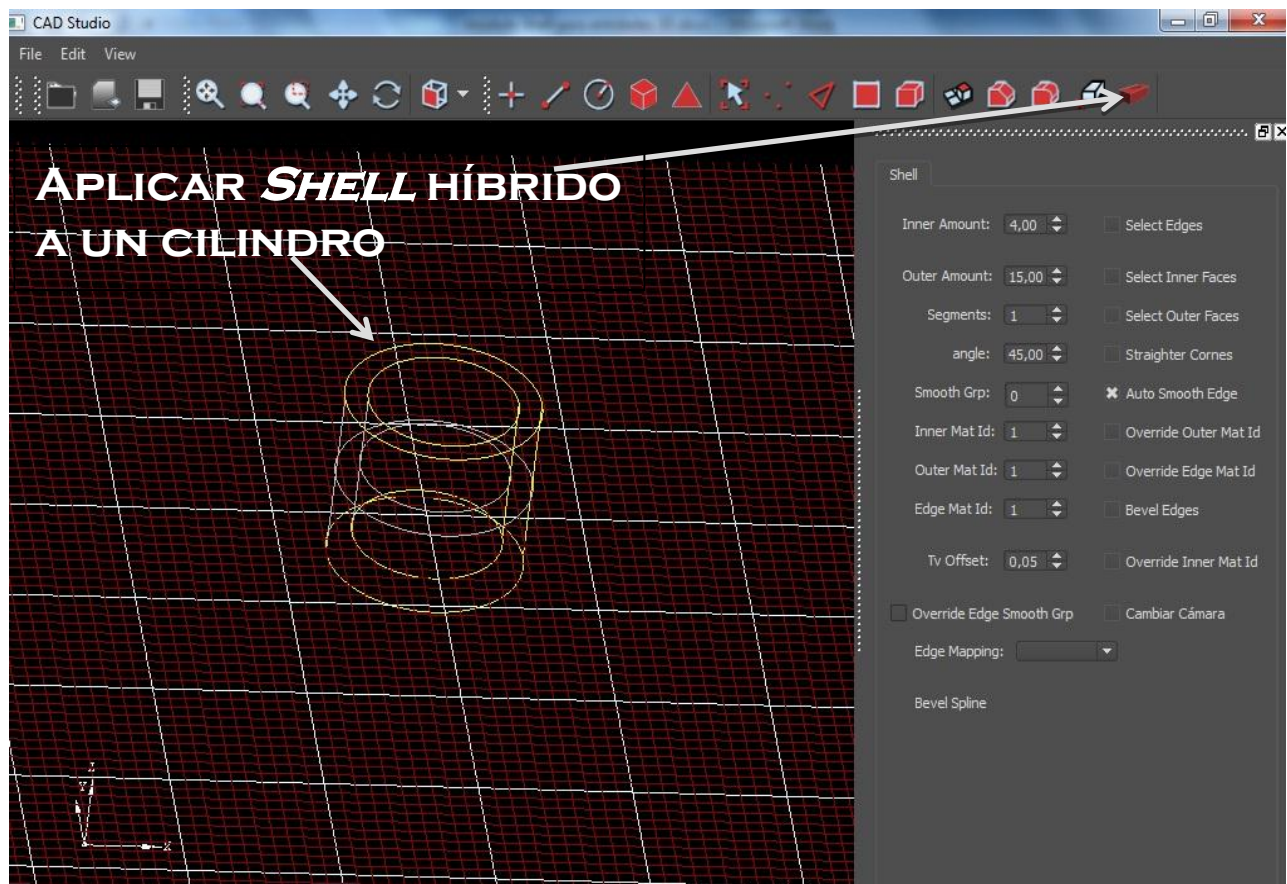


Figura 20 Aplicar *Shell* exterior a una estructura mecánica en el módulo desarrollado

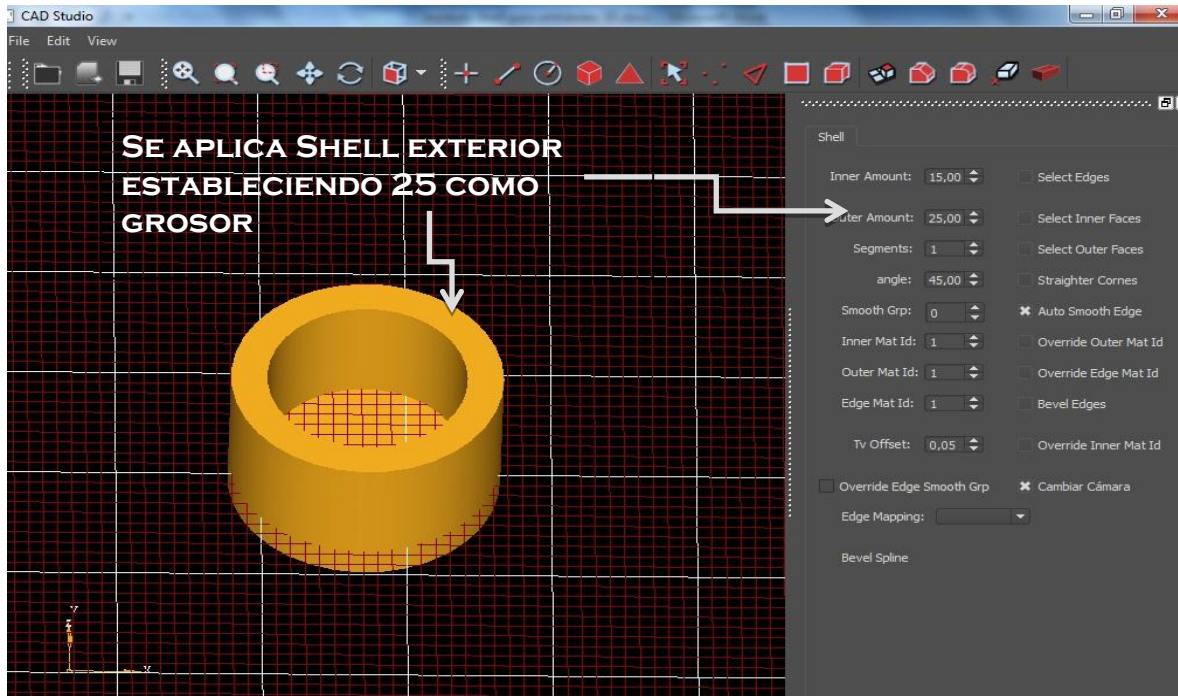


Figura 21 Aplicar *Shell* interior a una estructura mecánica en el módulo desarrollado

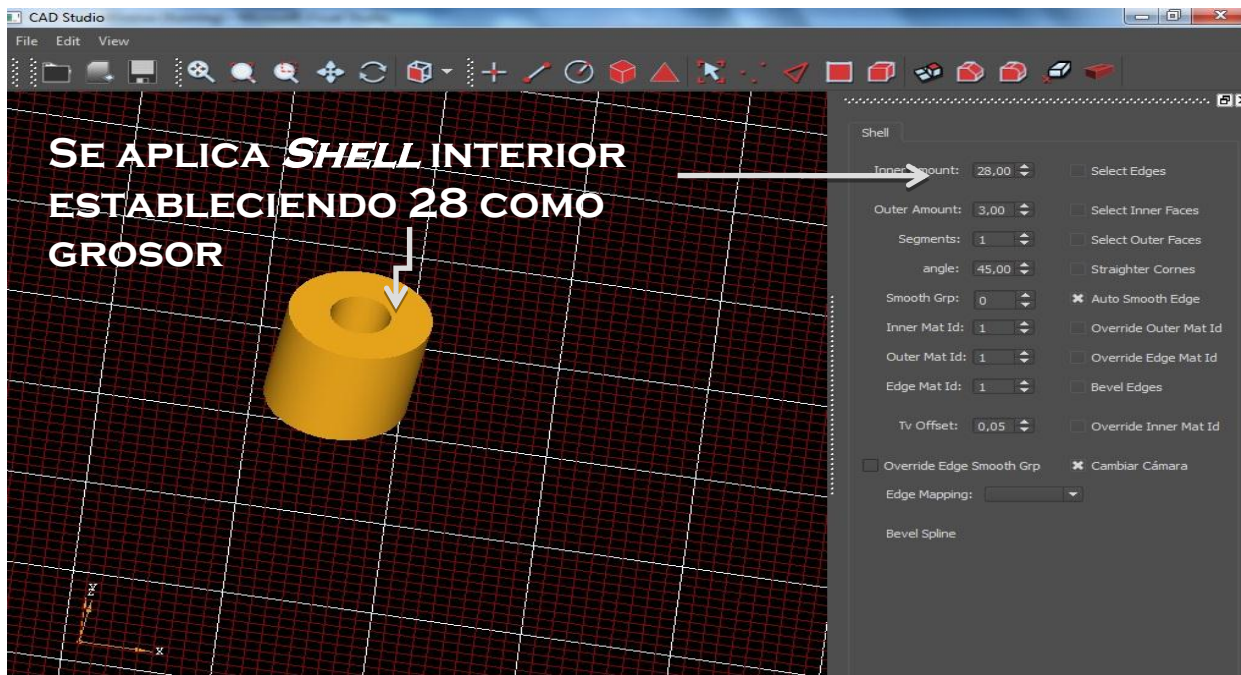


Figura 22 Aplicar *Shell* híbrido a una estructura mecánica en la herramienta inventor.

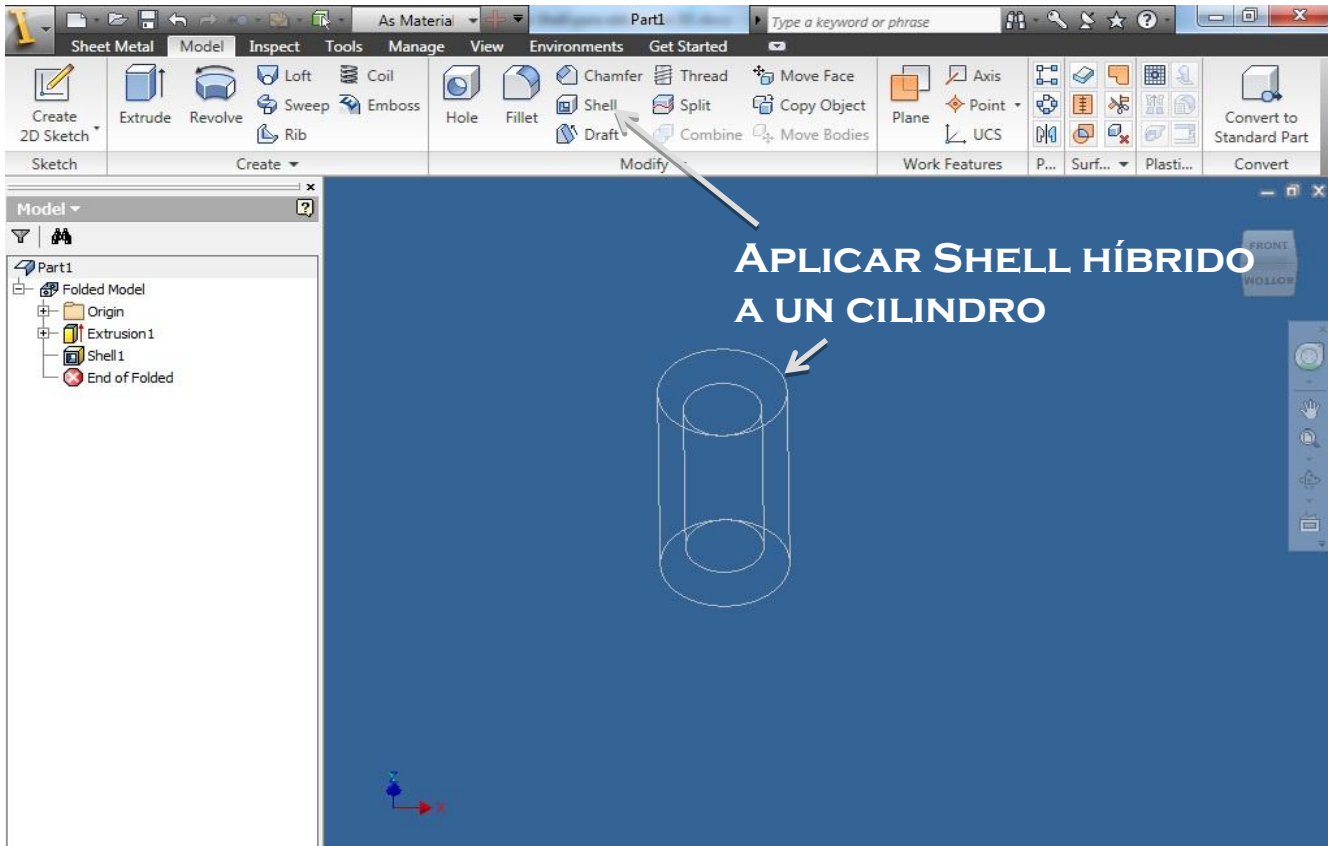
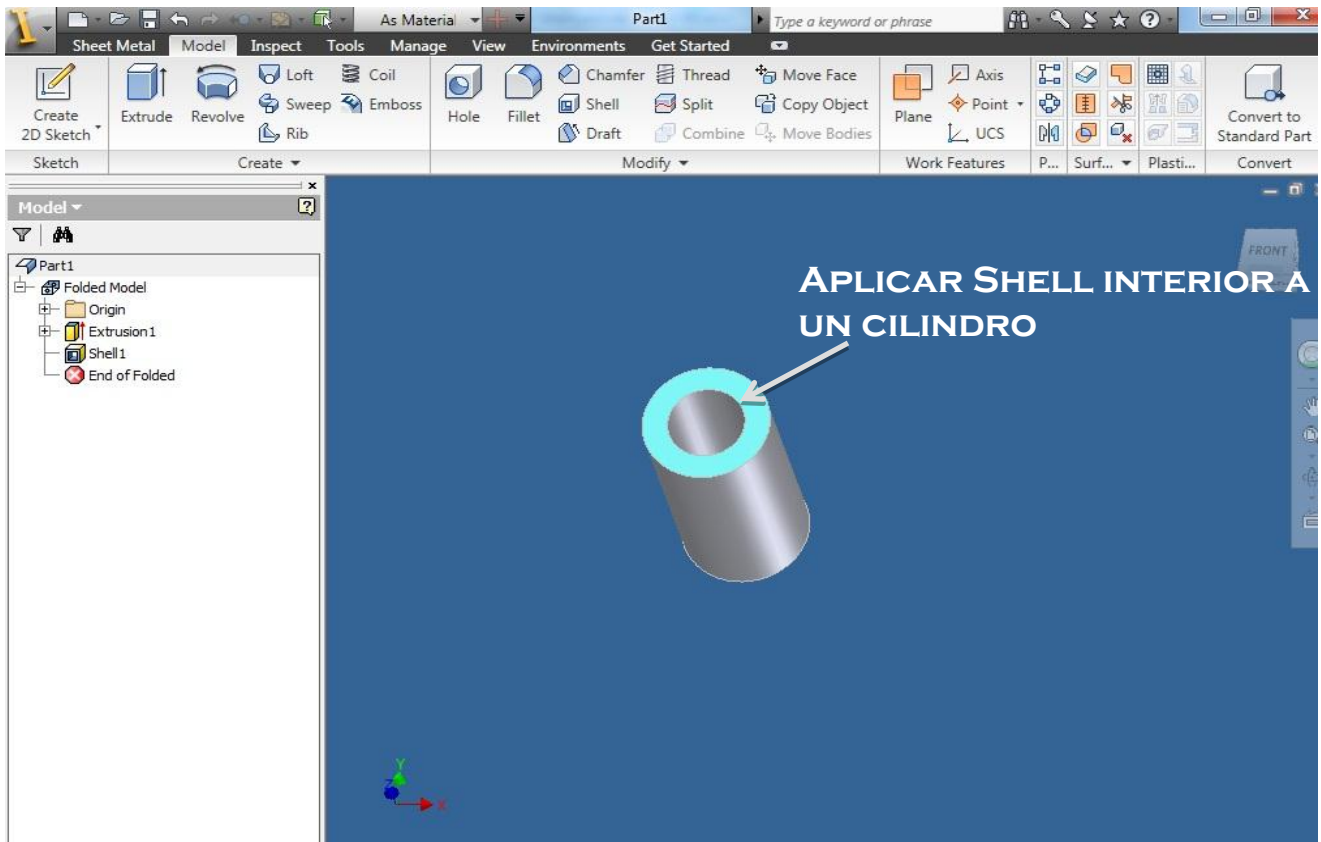
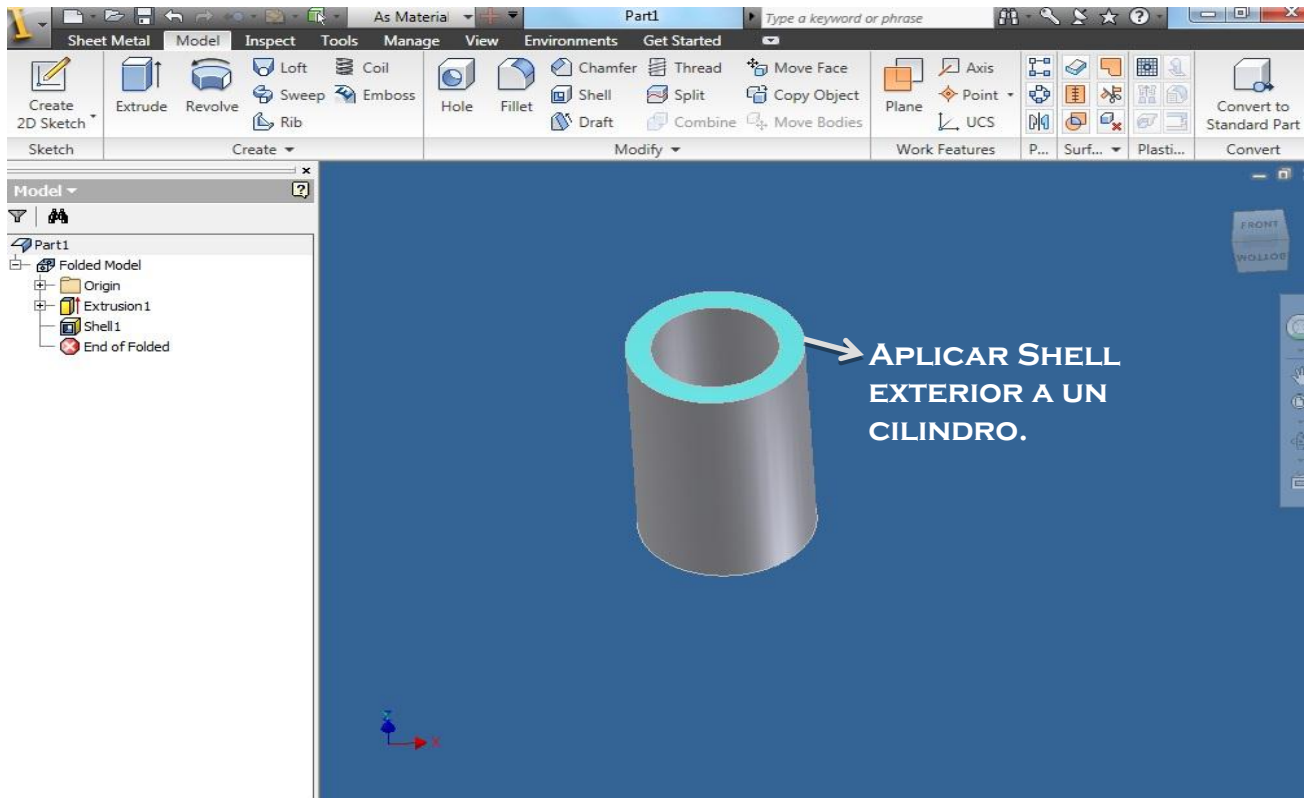


Figura 23 Aplicar *Shell* exterior a una estructura mecánica en la herramienta Inventor.



**Figura 24 Aplicar *Shell* exterior a una estructura mecánica en la herramienta Inventor.**





## GLOSARIO DE TÉRMINOS

---

### A

**Algoritmo:** Es una lista que, dado un estado inicial y una entrada, propone pasos sucesivos para arribar a un estado final obteniendo una solución.

### C

**Clase:** Una clase es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y el comportamiento que todos los objetos de la clase comparten.

### H

**Herramienta:** Una herramienta es un objeto elaborado a fin de facilitar la realización de una tarea. Se diseñan y fabrican para cumplir uno o más propósitos específicos.

### M

**Modelador:** Es una herramienta que permite como bien su nombre lo indica modelar cualquier espacio u objeto.

**Módulo:** es una parte autónoma de un programa de ordenador.