



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
Mecánica – Diseño mecánico

TRANSFORMACIÓN GRÁFICA DEL CORAZÓN HUMANO BASADO EN CATIA

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
ING. JOSÉ URIM SERRANO MARTÍNEZ

TUTOR PRINCIPAL
DR. ÁLVARO AYALA RUIZ,
FACULTAD DE INGENIERÍA

MÉXICO, D. F. ENERO 2016

JURADO ASIGNADO:

Presidente: Dr. Vicente Borja Ramírez

Secretario: Dr. Adrián Espinosa Bautista

Vocal: Dr. Álvaro Ayala Ruiz

1^{er.} Suplente: Dr. Ricardo Yáñez Valdez

2^{do.} Suplente: Dr. Fernando Velázquez Villegas

Lugar o lugares donde se realizó la tesis: Ciudad Universitaria, México

TUTOR DE TESIS:

Dr. Álvaro Ayala Ruiz

Dedicatoria.

A mí familia.

Resumen.

La formación del sistema cardiovascular humano es descrita mediante figuras y esquemas, mientras que el corazón adulto se encuentra caracterizado con modelos virtuales. Debido a que el corazón embrionario presenta una sucesión de cambios simultáneos durante los 9 meses de gestación no se tienen modelos virtuales, y por ende, se requiere de una serie de representaciones gráficas para complementar la información existente. Ya que en la etapa de tabicación se desarrollan las paredes interiores del corazón para la separación en cavidades y es la que conlleva a más cambios en poco tiempo, se usó como caso de estudio de este trabajo.

Se propone una metodología basada en la parametrización, graficación y transformación gráfica conocida como “*Morphing*”. De modo que para generar los modelos 3D se requiere el desarrollo de herramientas con técnicas especializadas que contiene la automatización de tareas iterativas. Estas herramientas proporcionarán la facilidad de creación y edición de las geometrías centradas en los cambios de forma del corazón embrionario.

Para la construcción de modelos se usan herramientas que realicen la parametrización de las entidades tomando de referencia la forma de modelos 3D generados a partir de cortes axiales por tomografía.

Usando un conjunto de superficies poligonales planas, las cuales son la base para generar y simular el cambio del modelo 3D, con el uso del sistema CAD/CAM/CAE de nombre comercial CATIA.

Abstract.

The formation of the human cardiovascular system is described in figures and diagrams, while the adult heart is characterized with virtual models, because the embryonic heart presents a series of simultaneous changes during the 9 months of pregnancy does not have virtual models, and therefore, it requires a series of models to complement existing information. Due to the stage of septal heart interior walls to separate cavities are developed and is leading to further changes in a short time, it was used as a case study of this work.

A methodology based on the parameterization, charting and graphics processing known as "Morphing" is proposed. So to generate 3D models development tools containing specialized technical automation of iterative tasks required. These tools provide the ease of creating and editing geometry changes focused on embryonic heart shape.

For modeling tools that perform the parametrization of the reference entities taking the form of 3D models generated from axial tomography cuts they are used.

Using a set of flat polygonal surfaces, which are the basis for generating and simulating the change of the 3D model, using the CAD / CAM / CAE system trademark CATIA.

Estructura de la tesis

La presente tesis está compuesta de 5 capítulos, en los que se exponen gradualmente desde los motivos para llevar a cabo este trabajo hasta las técnicas de parametrización desarrolladas a medida que se van describiendo las herramientas generadas y usadas. Esto anterior examinando un modelo que muestre la tabicación del corazón embrionario a partir de modelos de plastilina.

En el Capítulo 1 se realiza una introducción sobre el desarrollo del corazón embrionario, el cual es una parte del caso de estudio del trabajo.

El Capítulo 2 sirve como una base de conceptos importantes sobre los elementos para la construcción de las geometrías y las técnicas de morphing que se aplicarán sobre el modelo del caso de estudio.

El Capítulo 3 es el núcleo del trabajo, en el que se expone detalladamente la metodología propuesta para la construcción de modelos tridimensionales a partir de secciones divididas por planos.

Para validar la metodología en el Capítulo 4 se presenta el caso de estudio que describe el proceso de construcción del modelo embrionario del corazón en desarrollo desde el punto de vista de una sección coronal, con el uso de las herramientas que se generaron al desarrollar el trabajo.

En el Capítulo 5 y último se hace una recopilación de las conclusiones derivadas de la presente tesis, así como las principales líneas futuras de trabajo.

Mientras que en la Sección del Anexo se muestra la información complementaria al trabajo.

Contenido

i.	Introducción.....	1
ii.	Planteamiento del problema.....	2
iii.	Justificación del trabajo.....	3
iv.	Objetivos.....	4
1.	DESARROLLO DEL CORAZÓN EN EL EMBRIÓN.....	5
1.1	Formación de vasos sanguíneos (Angiogénesis).....	5
1.2	Desarrollo del corazón primitivo.....	6
1.3	Aspecto externo.....	6
1.3.1	Desarrollo del tubo cardiaco.....	6
1.3.2	Formación y evolución del asa cardiaca.....	6
1.3.3	Desarrollo del seno venoso.....	7
1.4	Circulación fetal y cambios en la circulación posnatal.....	8
2.	GRAFICACIÓN POR COMPUTADORA.....	9
2.1	Ecuaciones Paramétricas.....	9
2.2	Curvas.....	10
2.2.1	Curvas Spline.....	11
2.3	Panorama general.....	12
2.3.1	Ecuaciones de plano.....	13
2.4	Superficies.....	16
2.4.1	Ecuaciones Paramétricas en espacio tridimensional.....	16
2.4.2	Superficies Curvas.....	17
2.5	Morphing.....	19
2.5.1	Base del Morphing: Transformaciones y Distorsiones.....	19
2.5.2	Técnicas de Morphing.....	23
3.	DESARROLLO.....	27
3.1	Metodología propuesta para la parametrización de imágenes.....	28

3.1.1	Contar con imágenes de referencia	29
3.1.2	Definición de parámetros	29
3.1.3	Proceso de graficación.....	32
3.1.4	Imágenes intermedias.....	32
4.	CASO DE ESTUDIO.....	33
4.1	Contar con geometrías de referencia	34
4.1.1	Digitalización de geometrías de referencia.	34
4.2	Definición de parámetros	36
4.3	Proceso de graficación.....	36
4.3.1	Construcción del modelo intermedio	50
4.3.2	P_{int} en función de t	53
4.4	Archivo CAD resultante de la parametrización y Morphing.	55
4.5	Otras herramientas	56
5.	CONCLUSIONES	63
5.1	Conclusiones generales	63
5.2	Propuesta de Mejora	64
	Referencias	65
	Anexo A. Detalles de la tabicación del corazón humano embrionario.	68
	Anexo B. Otras curvas y superficies.....	72
	Anexo C. Automatización en CATIA	83
	Anexo D. Programas CATScript y CATVBA.....	94

i. Introducción.

El desarrollo del sistema cardiovascular se compone de varias etapas conocidas por los médicos, sin embargo, las formas que van tomando los órganos durante el estado embrionario son difíciles de entender debido a su forma y funcionamiento. En otras áreas de la medicina esto no ocurre, ya que con el estudio de órganos disponibles, el aprendizaje se construye a partir de la experiencia en el estudio de la anatomía y la disección, disciplina en la cual se guía al alumno de medicina para que asimile los conceptos estudiados, manipulando órganos para su estudio y así el alumno deduzca las posibles situaciones. Este tipo de aprendizaje no se puede extrapolar a la embriología humana, ya que existen limitaciones para obtener un órgano humano en plena formación, además que la legislación en el país no permite manipular fácilmente los órganos humanos en formación. (Constitución Política de los Estados Unidos Mexicanos, 1983)

En ingeniería se han desarrollado técnicas de representación, a partir de planos ortogonales, los cuales tienen información geométrica necesaria para construir modelos tridimensionales mediante software de Diseño Asistido por Computadora (CAD, por sus siglas en inglés), el cual es un entorno virtual. Estas representaciones son modelos virtuales que muestran los objetos antes de su fabricación los cuales enmarcan su función para su entendimiento, también facilitan la edición de los modelos.

Actualmente en la literatura no existen referencias publicadas acerca de modelos virtuales en el proceso de desarrollo de un corazón embrionario humano, mientras que sí hay modelos referentes al corazón adulto en (Chen & Guan, 2011).

Para caracterizar las múltiples etapas conocidas del corazón embrionario se requiere una serie de modelos virtuales tridimensionales, que muestren paso a paso los cambios entre cada una de las etapas. Con los beneficios de la visualización utilizando sistemas CAD, es posible crear dichos modelos. El hecho de obtener los modelos en formación proveerá una alternativa de gran ayuda para los académicos de la Facultad de medicina de la UNAM y a su vez dichos modelos permitirán ofrecer apoyo visual, para que alumno de medicina pueda consultar.

Es posible, que con los modelos virtuales del desarrollo del corazón embrionario humano se facilite el proceso de aprendizaje y revisión de manera auto dirigida por parte del alumno de medicina, el modelo le servirá al alumno para complementar sus conocimientos de la asignatura de embriología humana.

Para que dichos modelos sean creados se necesita de una gran cantidad de tiempo. Debido a que hay pequeñas variaciones entre las etapas del desarrollo del corazón embrionario, se requieren herramientas específicas que permitan automatizar la generación de entidades que permita la creación de dichos modelos, estas herramientas se crean mediante la programación de comandos, los cuales faciliten la edición de cada modelo.

ii. Planteamiento del problema

Durante el primer año de formación de un médico, éste debe estudiar la embriología humana, lo cual le permite conocer los mecanismos de la embriogénesis de modo general, las malformaciones congénitas y sus causas. Sin embargo, los estudios de embriología normalmente se realizan utilizando literatura que describe los mecanismos de formación, tales como dibujos (Carrillo Esper & Zárate Vega, 2011), esquemas, videos, etc., los cuales tienen limitantes de visualización ya que están realizados en dos dimensiones.

En la embriología humana, un caso importante en el desarrollo del feto es el sistema cardiovascular, el cual es el primero con actividad funcional durante la vida embrionaria, debido a que tiene como función principal el proveer los nutrientes y el oxígeno para el funcionamiento de las células del feto, mediante el intercambio de la madre y éste. (Arce Cedeño & Hernández Sosa, 2014)

El desarrollo del sistema cardiovascular está compuesto de varias etapas conocidas por los médicos, sin embargo, las formas que van tomando los órganos durante el estado embrionario son difíciles de asimilar para el alumno. Esto requiere de un gran esfuerzo para lograr su comprensión, al no tener los órganos en formación disponibles para su estudio en cada ciclo escolar, se requiere de una gran cantidad de información visual (fotografías, resonancias, o dibujos) en diferentes estados de desarrollo así como una cantidad de tiempo para la comprensión de este proceso. (Arce Cedeño & Hernández Sosa, 2014)

Lo anterior lleva a una aproximación aparente, por el hecho de basarse en representaciones bidimensionales.

En cuanto a la obtención de imágenes mediante radiografías y tomografías en los primeros días del desarrollo embrionario como recurso para este estudio, está contraindicado por razones de salud del feto, ya que al tener una exposición por radiación éste puede generar leucemia (Mondaca A., 2006).

Con base en lo anterior, para su estudio se reproducen fotografías tomadas por microscopía electrónica de corazones de embrión de pollo, ya que son aproximados en las primeras etapas, así como una gran cantidad de información visual, ya sea por dibujos o libros. Pese a estos recursos, es difícil entender la forma del corazón tridimensional a partir de formas planas.

iii. Justificación del trabajo

Debido a que no existen modelos virtuales de entorno didáctico ni de alguna otra índole en el proceso de desarrollo de un corazón embrionario humano, el presente trabajo tiene por objeto la creación de una serie de modelos en etapas representativas del proceso de tabicación; así como profundizar en el diseño paramétrico, el cual se basa en la definición de parámetros de una geometría, y su construcción mediante relaciones y fórmulas con un entorno que facilite el trabajo para la creación de estos modelos dentro de la paquetería de diseño.

El CAD otorga grandes ventajas de visualización y edición debido a que el modelo tridimensional puede rotarse, permite ver en el interior del mismo y hacer multitud de operaciones con él (cortes de plano, visualización de modelo en alambre, zoom, entre otros). Por lo tanto, un modelo virtual tridimensional del corazón embrionario, tendría los mismos beneficios de visualización que los modelos usados en ingeniería, sin embargo, en el desarrollo embrionario del corazón humano existen “etapas” definidas y es necesaria una cierta cantidad de modelos base para su construcción.

Para que dichos modelos sean creados se requiere una gran cantidad de tiempo, debido a que hay pequeñas variaciones entre las etapas del desarrollo del corazón embrionario. Es necesario el uso de herramientas específicas que permitan automatizar la generación de entidades, ya sean creación de: puntos, curvas o

superficies. Para facilitar la creación de dichos modelos con sus elementos, estas herramientas se crean mediante la programación de comandos en los cuales facilitan la edición de los elementos del modelo; esto con el uso de software disponible, CATIA V5, mediante la parametrización de tareas que faciliten la edición de las mismas.

El programa CATIA ofrece la capacidad de automatizar el proceso de diseño utilizando programación en Visual Basic, que facilita la creación de modelos geométricos sólidos. (Dassault Systèmes, 2009)

El programa comercial CATIA V5 provee un conjunto integrado de aplicaciones de CAD con un entorno de tratamiento de entidades y los datos generados tienen características definidas. Dependiendo de su uso para la creación de geometrías específicas, el desarrollo de aplicaciones propias dentro del entorno son muchas veces optimizadas in situ, debido a su programación amigable. Por lo anterior es uno de los programas más comerciales e influyente en este momento, armadoras de autos y aviones lo utilizan.

iv. Objetivos

El objetivo principal del trabajo es desarrollar herramientas para el modelado del corazón embrionario, que validen la metodología propuesta, con el caso de estudio en la etapa de tabicación en el interior de las cavidades, usando paquetería disponible (CATIA V5), mediante la automatización de tareas repetitivas que faciliten la edición de los modelos.

Además conviene tener una herramienta flexible que sea adaptable para minimizar los tiempos de edición sobre el modelo geométrico, dejando a discreción la cantidad de puntos para interpolar las curvas del modelo con el fin de usar menos elementos.

CAPÍTULO

1

1. DESARROLLO DEL CORAZÓN EN EL EMBRIÓN

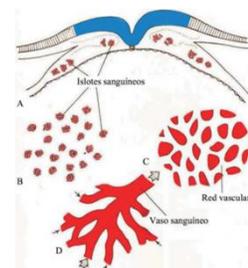
La biología del desarrollo humano tiene por estudio la formación y desarrollo del hombre, dentro de esta materia, se encuentra la rama de la embriología humana, la cual se especializa en el estudio y desarrollo del embrión y del feto humano, en donde su periodo de estudio abarca desde la gestación hasta el parto (los 9 meses de la gestación). Para el estudio de los fetos y los embriones, se agrupan una serie de características en periodos de tiempo que comparte el órgano en desarrollo, las cuales denominan “Etapas del desarrollo” (Matsumura, 1996).

A partir de células madre se producen interacciones moleculares y celulares, donde la influencia de células vecinas estimula la formación temprana del corazón, Siendo el sistema cardiovascular el primero con actividad funcional durante la vida embrionaria debido a las necesidades de oxígeno y nutrientes, de ahí su importancia. El estudio general se enfoca en el desarrollo del corazón en el embrión, donde los primeros componentes aparecen a finales de la tercera semana, estos comienzan a funcionar hacia el final de la cuarta y principios de la quinta con el intercambio materno-fetal que ocurre a nivel del corion vellosos.

Otro aspecto importante del sistema cardiovascular es el hecho de que en las etapas iniciales existe una gran diferencia de tamaño entre el embrión y la placenta. De ahí que el corazón embrionario primitivo tenga que impulsar la sangre.

1.1 Formación de vasos sanguíneos (Angiogénesis)

La formación de los primeros vasos sanguíneos ocurre durante la tercera semana del desarrollo, con células endoteliales planas que se unen formando largos tubos (vasos) que se fusionan para formar una red (plexo), que inicia la distribución del sistema circulatorio (Fig. 1.1)



*Fig 1.1. Formación de los vasos sanguíneos.
(Valdés Valdés, Armando et al. 2010)*

1.2 Desarrollo del corazón primitivo.

Formación del área cardiogénica

A mediados de la tercera semana. La formación del sistema vascular se observa en un área cardiogénica; En la cual los tubos (vasos) establecen conexiones, a través de los arcos aórticos, con el plexo en herradura para contribuir a la formación del tubo cardíaco.



Fig. 1.2. Formación del plexo en herradura en un embrión de tres semanas, vista dorsal. (Valdés Valdés, Armando et al. 2010)

1.3 Aspecto externo

1.3.1 Desarrollo del tubo cardíaco

Al final de la tercera semana, con un proceso de plegamiento lateral, los tubos endoteliales derecho e izquierdo se desplazan y fusionan en la línea media, el corazón se convierte en un tubo único continuo ensanchado o lo que se conoce como tubo cardíaco. Este tubo cardíaco va formando una protuberancia en la cavidad pericárdica y permanece unido dorsalmente en esta cavidad por medio del mesocardio dorsal. (Fig. 1.3).

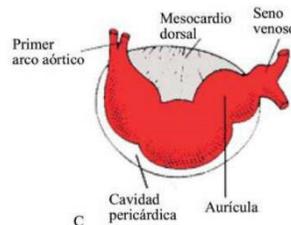


Fig. 1.3. Tubo cardíaco (endocárdico) y cavidad pericárdica en desarrollo (Valdés Valdés, Armando et al. 2010)

1.3.2 Formación y evolución del asa cardíaca

Durante la cuarta semana. La porción cefálica del tubo cardíaco se pliega en dirección ventral y caudal y hacia la derecha, mientras que la porción caudal (auricular) se desplaza en dirección dorsal y craneal y hacia la izquierda. Este plegamiento forma el asa cardíaca, que inicialmente tiene forma de C y luego de S y se completa a finales de la cuarta semana (Fig. 1.4. B y C).

Durante la formación del asa cardíaca van apareciendo cuatro dilataciones alineadas, delimitadas por acortamientos. El tubo cardíaco queda organizado por regiones a lo largo de su eje craneocaudal, desde la región troncoconal al ventrículo derecho, al ventrículo izquierdo y a la región auricular, respectivamente (Fig. 1.4).

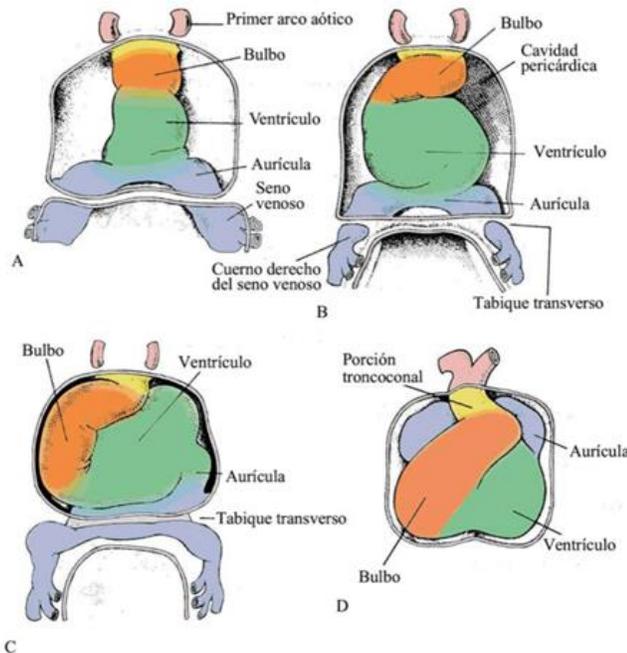


Fig. 1.4. Formación del asa cardíaca (Valdés Valdés, Armando et al. 2010)

Al completarse la formación del asa cardíaca, el tubo cardíaco comienza a formar trabéculas primitivas en dos zonas perfectamente definidas, proximales y distales al agujero interventricular primario. La porción auricular y las demás porciones del bulbo cardíaco conservan, por el momento, sus paredes lisas. El ventrículo primitivo recibe el nombre de ventrículo izquierdo primitivo. De la misma manera, el tercio proximal trabeculado del bulbo cardíaco se puede denominar ventrículo derecho primitivo.

La porción troncoconal del bulbo cardíaco, que primero se encuentra del lado derecho de la cavidad pericárdica, se desplaza gradualmente hacia una posición más medial.

1.3.3 Desarrollo del seno venoso

A mediados de la cuarta semana hasta la sexta semana. La vena umbilical derecha y la onfolomesentérica izquierda se obliteran durante la quinta semana de desarrollo, por lo que la prolongación izquierda del seno pierde importancia.

Finalmente, la vena cardinal común izquierda se oblitera a las 10 semanas, y queda sólo la prolongación izquierda del seno, la vena oblicua de la aurícula izquierda y el seno coronario.

La prolongación derecha, que representa entonces la única comunicación entre el seno venoso original y la aurícula, se incorpora a la aurícula derecha para formar la pared lisa de esta. Su desembocadura, el orificio senoauricular, está limitado de cada lado por un pliegue valvular, las válvulas venosas derecha e izquierda. (Fig. 1. 5).

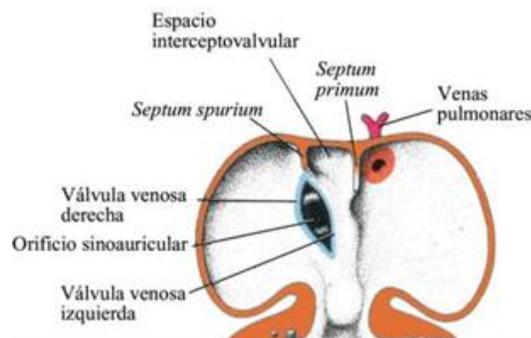


Fig. 1.5. Desarrollo de las válvulas venosas. (Valdés Valdés, Armando et al. 2010)

En un principio, las válvulas venosas son grandes, pero cuando la prolongación derecha del seno queda incorporada a la pared de la aurícula, la válvula venosa izquierda y el septum spurium se fusionan con el tabique interauricular en desarrollo.

1.4 Circulación fetal y cambios en la circulación posnatal

Durante el periodo prenatal, la sangre oxigenada viaja desde la placenta, el flujo se regula por contracciones oscilantes en el conducto venoso. La sangre va del ventrículo izquierdo al tronco aórtico y se distribuye principalmente en las arterias de la cabeza, cuello y extremidades superiores.

Uno de los cambios dramáticos al momento del nacimiento es la respiración, por lo que la sangre fluye por los vasos pulmonares, el caudal del sistema se incrementa, produciendo un cambio de presión sanguínea en la aurícula izquierda. Al mismo tiempo que disminuye la presión en la aurícula derecha, al interrumpirse la circulación placentaria, este incremento de presión en la aurícula izquierda hace que el septum primum se una fuertemente al septum secundum, produciendo el cierre funcional del agujero oval.

CAPÍTULO

2

2. GRAFICACIÓN POR COMPUTADORA

Frecuentemente las representaciones gráficas para un grupo de datos y procesos en ingeniería y de medicina se conocen como visualización científica. Existen muchas variedades de conjuntos de datos y las representaciones de visualización dependen de las características de los datos. Una compilación de datos contiene valores escalares, vectores, tensores de orden superior o cualquier composición de estos tipos de datos. Y los conjuntos de datos pueden ser bidimensionales o tridimensionales. Cualquier computadora de uso general puede utilizarse para hacer gráficas de los datos utilizando la programación o software especializado en el sistema con objeto de formar modelos, patrones o ecuaciones.

2.1 Ecuaciones Paramétricas.

Una ecuación paramétrica representa una o varias curvas o superficies en el plano o en el espacio, mediante valores arbitrarios o mediante una constante, llamada parámetro. En el caso de una curva arbitraria, puede ser difícil idear un conjunto de ecuaciones paramétricas que defina completamente la forma de la curva. Pero cualquier curva puede aproximarse utilizando diferentes conjuntos de funciones paramétricas en secciones de la curva. Estas aproximaciones se forman con funciones polinomiales. Dicha construcción por partes de una curva debe implantarse cuidadosamente para asegurar de que haya transición sencilla de una sección de la curva a la siguiente.

La uniformidad de una curva puede describirse a partir de la continuidad de la curva entre las secciones. La continuidad de orden cero (Fig. 2.1 a) se refiere simplemente a que las curvas se interceptan. Continuidad de primer orden (Fig. 2.1 b) significa que las líneas tangentes (primeras derivadas) de dos secciones

adyacentes de la curva son la misma en el punto de adyacencia. Continuidad de segundo orden (Fig. 2.1 c) quiere decir que las curvaturas (segundas derivadas) de las dos secciones de la curva son las mismas en la intersección. La Fig. 2.1 muestra ejemplos de los tres órdenes de continuidad. (Hearn, 2006)



Fig. 2.1 Especificación por partes de una curva mediante la conexión de dos segmentos de curva con órdenes de continuidad variantes (Hearn, 2006)

2.2 Curvas

Cuando la curva desplegada pasa a través de los puntos de control, como en la Fig. 2.2 se dice que interpola los puntos de control.



Fig. 2.2. Conjunto de seis puntos de control interpolados por una curva. (Hearn, 2006)

Por otro lado, se dice que los puntos de control se aproximan si la curva desplegada pasa cerca de ellos (Fig.2.2). Existen muchas técnicas para constituir ecuaciones paramétricas polinomiales de curvas y superficies, dadas las coordenadas de los puntos de control. Entre los métodos básicos para desplegar curvas específicas con punto de control se incluyen las formulaciones de Bézier y “spline”. (Hearn, 2006)

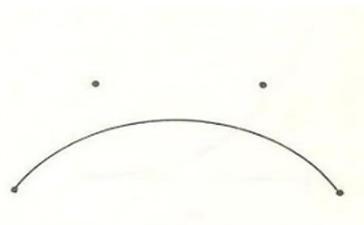


Fig. 2.3. Conjunto de cuatro puntos de control aproximados por una curva. (Hearn, 2006)

2.2.1 Curvas Spline

Un spline es una curva diferenciable definida en porciones mediante funciones que son utilizadas en aplicaciones que requieren la interpolación de datos, o un suavizado de curvas. Los splines se utilizan para aproximar formas complicadas. Debido a la simplicidad de la representación y la facilidad de cómputo, los splines son frecuentemente usados en la representación de curvas (Fig. 2.4), particularmente en el terreno de los gráficos por ordenador.

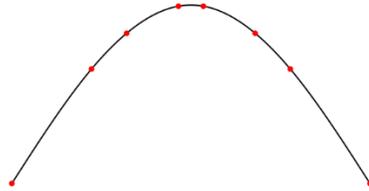


Fig. 2.4 Puntos interpolados por una curva Spline (Hearn, 2006)

Las curvas spline son los más utilizados en la práctica, las fuentes *True Type* es un ejemplo del uso de splines cuadráticos, mientras que las splines cúbicas se usan en programas de diseño gráfico.

En general, no pasan por ningún punto de control (ni siquiera los extremos), aunque se puede forzar que lo hagan.

La curva spline está representada por una función polinómica de grado n en x , y se define como (la ec. (i)):

$$y = \sum_{k=0}^n a_k x^k \quad . \text{ ec. (i)}$$

De forma desarrollada:

$$y = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} + a_n x^n \quad \text{ec. (ii)}$$

Se puede describir el polinomio cúbico como:

$$\left. \begin{aligned} x(u) &= a_x u^3 + b_x u^2 + c_x u + d_x \\ y(u) &= a_y u^3 + b_y u^2 + c_y u + d_y, \quad (0 \leq u \leq 1) \\ z(u) &= a_z u^3 + b_z u^2 + c_z u + d_z \end{aligned} \right\} \text{ ec. (iii)}$$

Ventajas de las curvas spline

- Ofrecen un compromiso razonable entre la flexibilidad y la velocidad de cálculo.
- Son útiles en número de aplicaciones gráficas de tendencias de los datos en un conjunto discreto de puntos de datos.
- Las splines son más flexibles para el modelado de formas curvas arbitrarias.
- Sobre todo, más apropiadas para el diseño interactivo: más "suaves", control local.

Desventajas de las curvas spline

- Se debe establecer la continuidad con respecto a otra spline.
- Son sensibles a cambios locales. El desplazamiento de un sólo punto de control produce efecto en la curva.
- Es necesario indicar la tensión de la curvatura.

2.3 Panorama general

Cualquier objeto tridimensional puede representarse como un conjunto de superficies poligonales planas. Para algunos objetos, como un polígono, esto define precisamente las características de la superficie. En otros casos, una representación de un polígono ofrece una descripción aproximada del objeto. En la figura 2.7 se despliega un objeto sólido modelado como una malla de superficies poligonales. Esta representación de la malla de polígonos puede desplegarse rápidamente para dar una indicación general de la estructura del objeto y la aproximación puede mejorarse dividiendo las superficies del objeto en caras poligonales menores.

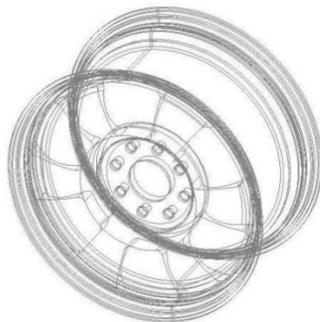


Figura 2.7 Objeto representado en alambre (Autor)

Cada polígono de un objeto puede especificarse en un paquete de gráficas mediante comando de líneas o de llenado de áreas para definir las coordenadas del vértice. Los paquetes CAD a menudo permiten a los usuarios introducir posiciones para el vértice junto con fronteras de polígonos con métodos interactivos. Estos vértices pueden representar el resultado de la digitalización de un trazo o bien pueden ser introducidos por un diseñador que esté creando una nueva figura.

2.3.1 Ecuaciones de plano

Los parámetros que especifican la orientación especial de cada polígono se obtienen de los valores coordenados de los vértices y las ecuaciones que definen los planos poligonales. Estos parámetros de planos se utilizan en transformaciones de visión, modelos de sombreado y algoritmos de superficies ocultas que determinan qué líneas y planos se traslapan a lo largo de la línea de visión.

La ecuación de una superficie plana puede expresarse así:

$$Ax + By + Cz + D = 0 \quad \text{ec. (iv)}$$

Donde (x, y, z) es cualquier punto del plano. Los coeficientes A, B, C, D son constantes que pueden calcularse utilizando los valores coordenados de tres puntos no colineales en el plano. Comúnmente se usan las coordenadas de tres vértices sucesivos en una frontera de un polígono para hallar valores de estos coeficientes. Al denotar las coordenadas de tres vértices de un polígono como (x_1, y_1, z_1) , (x_2, y_2, z_2) y (x_3, y_3, z_3) , se puede resolver el siguiente conjunto de ecuaciones planas simultáneas para las razones $A/D, B/D$ y C/D :

$$\left(\frac{A}{D}\right)xi + \left(\frac{B}{D}\right)yi + \left(\frac{C}{D}\right)zi = -1 \quad ; \quad i = 1, 2, 3 \quad \text{ec. (v)}$$

Utilizando un método de solución como la regla de Cramer, se puede escribir la solución de los parámetros del plano en forma de determinantes:

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_2 \end{vmatrix} ; \quad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_2 \end{vmatrix} ; \quad C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} ; \quad D = \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_2 \end{vmatrix} \quad \text{ec. (vi)}$$

Se pueden ampliar los determinantes y escribir los cálculos de los coeficientes del plano en la forma explícita:

$$\begin{aligned}
 A &= y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2) \\
 B &= z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2) \\
 C &= x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2) \\
 D &= -x_1(y_2z_3 - y_3z_2) - x_2(y_3z_1 - y_1z_3) - x_3(y_1z_2 - y_2z_1)
 \end{aligned}
 \quad \left. \vphantom{\begin{aligned} A \\ B \\ C \\ D \end{aligned}} \right\} \text{ec. (vii)}$$

Los valores de A, B, C y D se almacenan en la estructura de datos que contiene la información de coordenadas y atributos referente al polígono definido en este plano.

La orientación de una superficie plana se especifica por medio del vector normal al plano, como se muestra en la figura 2.8. Este vector normal tridimensional tiene las coordenadas cartesianas (A, B, C).

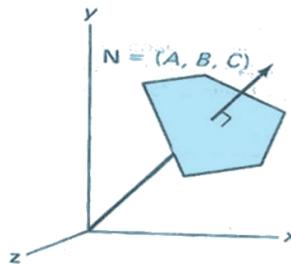


Figura 2.8 El vector N , normal a la Superficie de un plano descrito por la ecuación $Ax + By + Cz + D = 0$ tiene las coordenadas (A, B, C.) (Hearn, 2006)

Puesto que con frecuencia se trabaja con superficies poligonales que encierran un objeto interior, se necesita distinguir entre los lados de la superficie.

El lado del plano que da la cara al objeto interior se denomina “interior” y el lado visible o externo se llama “exterior”. Si se especifican vértices en un sentido igual al del reloj cuando se observa el lado externo del plano en su sistema coordenado por la derecha, la dirección del vector normal irá de adentro hacia afuera. Esto se demuestra para un plano de un cubo unitario de la figura 2.9.

Para determinar los componentes del vector normal de la superficie sombreada que se muestra en la figura 2.9, se seleccionan tres de los cuatros vértices situados a lo largo de la frontera del polígono.

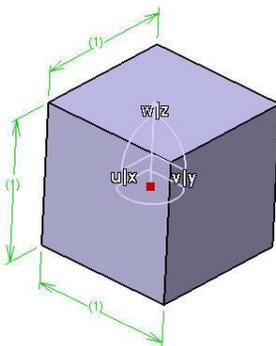


Fig. 2.9 La superficie poligonal sombreada del cubo unitario tiene la ecuación del plano $X-1=0$ y el vector normal $(1, 0, 0)$ (Autor)

Estos puntos se seleccionan en un sentido igual al del reloj cuando observamos el exterior del cubo hacia el origen. Las coordenadas de estos vértices, en el orden seleccionado, se utilizan en la ecuación a fin de obtener los coeficientes del plano:

$A = 1, B = 0, C = 0, D = -1$. El vector normal de este plano está en el sentido del eje x positivo. Las ecuaciones del plano se utilizan también para identificar puntos interiores y exteriores. Cualquier punto (x, y, z) exterior a un plano satisface la desigualdad.

$$Ax + By + Cz + D > 0 \quad \text{ec. (viii)}$$

Análogamente, cualquier punto situado en el interior del plano produce un valor negativo de la expresión $Ax + By + Cz + D$. Para la superficie sombreada de la figura 2.12, cualquier punto exterior al plano cumple la desigualdad $x - 1 < 0$, mientras que cualquier punto interior al plano tiene un valor de coordenadas x menor que 1.

2.4 Superficies

2.4.1 Ecuaciones Paramétricas en espacio tridimensional.

Mediante la introducción de un cuarto parámetro, u , en la descripción coordenada de una curva, se puede expresar cada una de las tres coordenadas cartesianas en forma paramétrica. Cualquier punto de la curva puede representarse entonces por medio de la función vectorial.

$$P(u) = (x(u), y(u), z(u)) \quad \text{ec. (x)}$$

Por lo general, las ecuaciones paramétricas se constituyen de manera que el parámetro se defina en el intervalo de 0 a 1. Por ejemplo, una circunferencia en el plano XY con centro en el origen coordenado podría definirse en forma paramétrica como:

$$x(u) = r \cos(2\pi u) \quad ; \quad y(u) = r \sin(2\pi u) \quad ; \quad z(u) = 0 \quad \text{ec. (ix)}$$

También son posibles otras formas paramétricas para describir circunferencias y arcos circulares.

Las ecuaciones paramétricas de las superficies se formulan con dos parámetros u y v . Una posición coordenada de una superficie se representa entonces por medio de la función vectorial paramétrica.

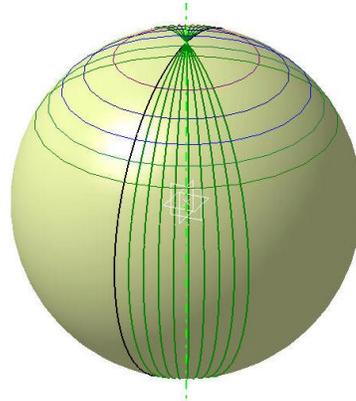
$$P(u, v) = (x(u, v), y(u, v), z(u, v)) \quad \text{ec. (xi)}$$

Las ecuaciones de las coordenadas x , y , z a menudo se acomodan de modo que los parámetros u y v están definidos dentro del intervalo de 0 a 1. Por ejemplo una superficie esférica puede describirse con las ecuaciones.

$$\left. \begin{aligned} x(u, v) &= r \sin(\pi u) \cos(2\pi v) \\ y(u, v) &= r \sin(\pi u) \sin(2\pi v) \\ z(u, v) &= r \cos(\pi u) \end{aligned} \right\} \text{ec. (xii)}$$

Donde r es el radio de la esfera. El parámetro u describe líneas de latitud constante sobre la superficie, mientras que el parámetro v describe líneas de

longitud constante. Al mantener uno de estos parámetros fijo mientras se varía el otro sobre cualquier valor dentro del intervalo de 0 a 1, se podrían trazar líneas de latitud y longitud de cualquier sección esférica (fig. 2.10).



*Fig. 2.10 Sección de una superficie esférica descrita por las líneas de u constante y líneas de v constantes en las ecuaciones [xii]
(Autor)*

En aplicaciones de diseño, una curva o superficie a menudo se define especificando interactivamente un conjunto de puntos de control, los cuales indican la forma de la curva. Estos puntos de control son usados por el paquete para formar ecuaciones paramétricas polinomiales para desplegar la curva definida.

2.4.2 Superficies Curvas

Los despliegues tridimensionales de las superficies curvas pueden generarse a partir de un conjunto de entrada de las funciones matemáticas que define las superficies o bien a partir de un conjunto de puntos de datos especificados por el usuario. Cuando se especifican funciones de curvas, un paquete puede emplear las ecuaciones definidoras para localizar y graficar posiciones de píxeles a lo largo de la trayectoria de la curva, casi igual como sucede con las curvas en dos dimensiones. Un ejemplo de la clase de superficies que pueden generarse a partir de una definición funcional se da en la figura 2.12. A partir de un conjunto de datos de entrada, un paquete determina las descripciones funcionales de la curva que mejor se ajusta a los puntos de datos según las restricciones de la aplicación.

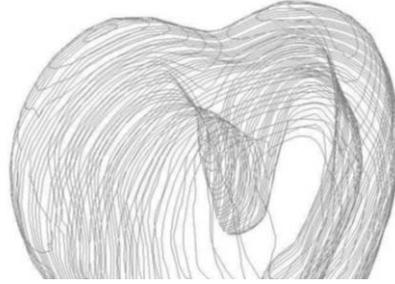


Fig. 2.12 Las superficies tridimensionales pueden generarse a partir de una definición funcional (Autor)

Podemos representar una línea curva tridimensional en forma analítica con la pareja de funciones.

$$y = f(x) \quad ; \quad z = g(x)$$

Con la coordenada x seleccionada como variable independiente, los valores de las variables dependientes “ y ” y “ z ” se determinan después a partir de las ecuaciones 1.6 a medida que se avanza a través de valores de x de un extremo de la línea al otro. Esta representación tiene algunas desventajas; si se desea una gráfica alisada, se debe cambiar la variable independiente siempre que la primera derivada (pendiente) de $f(x)$ o bien $g(x)$ se vuelve mayor que 1. Esto significa que se debe verificar continuamente los valores de las derivadas, que pueden volverse infinitas en algunos puntos. Así mismo, las ecuaciones anteriores ofrecen un formato desproporcionado para representar funciones con valores múltiples. Una representación más propicia de las curvas para aplicaciones de las gráficas es en términos de ecuaciones paramétricas.

En la figura 2.13 se muestra un objeto cuyas superficies curvas pueden ser definidas por un conjunto de entrada de punto de datos.

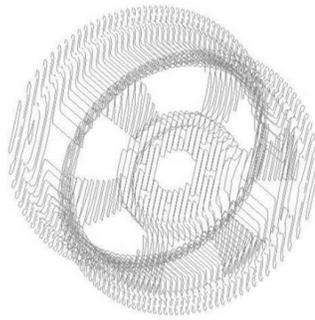


Fig. 2.13 Trazo con líneas de un objeto (Rin) con superficies curvas que puede generarse a partir de un conjunto de puntos de datos. (Autor)

2.5 Morphing

Morphing es un término utilizado en graficación por computadora, el cual se basa en el concepto de metamorfosis, de forma que al realizar el cambio de un estado de una geometría a otra mediante una interpolación, produce una transición entre las geometrías en un tiempo determinado. (Goldstein & Gotsman, 1995)

2.5.1 Base del Morphing: Transformaciones y Distorsiones

La transformación geométrica puede manifestarse mediante deformaciones, que van de una translación, rotación, escalamiento o una combinación de ellas, o una transformación completa no lineal.

Una distorsión (transformación en el plano) define una relación geométrica (Hearn, 2006) entre cada punto de una imagen de entrada y cada punto de una imagen de salida. En otras palabras, una deformación 2D es una transformación que distorsiona un espacio base 2D en un espacio final 2D, en el cual se asigna un punto de origen $[u, v]$ para producir un punto de destino $[x, y]$.

De forma general una distorsión relaciona un sistema de coordenadas de parámetros de entrada (u, v) con los parámetros de salida (x, y) , se expresa como:

$$[x, y] = [X(u, v), Y(u, v)] \quad \text{ec. (xiii)}$$

O referente al sistema de coordenadas del destino al origen y se denota de la siguiente manera:

$$[u, v] = [U(x, y), V(x, y)] \quad \text{ec. (xiv)}$$

Donde $[u, v]$ se refiere a la entrada de coordenadas para transformarlas a la nuevas coordenadas $[x, y]$ mediante funciones que especifican la transformación

espacial X , Y , U y V las cuales son distorsiones. Las funciones X y Y asignan la entrada a la salida y por lo tanto (xiii) se conoce como la deformación hacia adelante. Las funciones U y V asignan la salida a la entrada y (xiv) se conoce como deformación reversa.

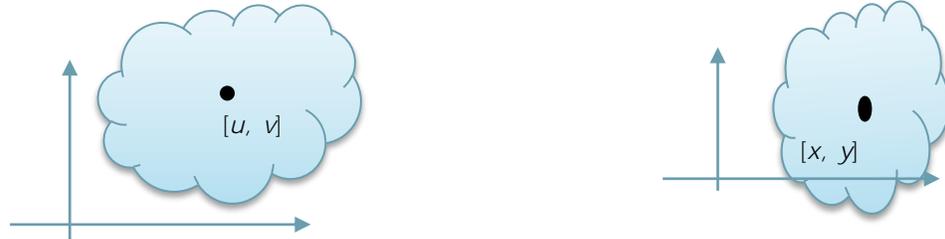


Figura 2.21 Punto distorsionado.

Considerando que las transformaciones geométricas están en términos de las funciones de deformación hacia adelante X y Y (transformar una imagen de origen en el sistema de coordenada uv la imagen de destino en el sistema de coordenada xy). Transformaciones similares se aplicarán para las funciones inversas U y V .

Formalmente una transformación $L(x)$ es lineal si y sólo si:

- $L(x + y) = L(x) + L(y)$
 - $L(\alpha x) = \alpha L(x)$ para cualquier escalar α
- } ec. (xv)

Una transformación $T(x)$ es afín si y sólo si existe una constante c y $L(x)$ tal que una transformación lineal:

$$T(x) = L(x) + c \quad \text{ec. (xvi)}$$

Están satisfechas para todo x . Todas las transformaciones lineales son transformaciones afines, las cuales no pasan sobre el origen. Para las transformaciones afines, funciones de transformación hacia delante (en coordenadas euclidianas) pueden definirse como:

$$\begin{aligned} x &= a_{11}u + a_{21}v + a_{31} \\ y &= a_{12}u + a_{22}v + a_{32} \end{aligned} \quad \text{ec. (xvii)}$$

La ecuación general para llevar a cabo una transformación afín (en coordenadas homogéneas) se puede escribir como:

$$[x \ y \ 1] = [u \ v \ 1] \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{23} & 1 \end{bmatrix} \quad \text{ec. (xviii)}$$

Las transformaciones afines son las transformaciones más comunes utilizadas en los gráficos por computadora. Ellas producen combinaciones de cuatro transformaciones fundamentales: escala, traslación, rotación, y deformación angular. Una sucesión de estas transformaciones afines se pueden combinar en una transformación afín en general.

Transformaciones de escala

Mediante la aplicación de los factores de escala (s_u y s_v) a la fuente de las coordenadas, todos los puntos se escalan. Este tipo de escala se conoce como la ampliación sobre el origen, porque el punto P se mueve s_u veces sobre el origen en la dirección “ U ” y s_v veces más lejos del origen en la dirección “ V ”.

$$[x \quad y \quad 1] = [u \quad v \quad 1] \begin{bmatrix} s_u & 0 & 0 \\ 0 & s_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{ec. (xix)}$$



Figura 2.22 Escala de un pentágono 1x a 2x

Los ensanchamientos se especifican con factores de escala positivos mayores que uno, reducciones con factores positivos. Si un factor de escala es negativo hay una reflexión alrededor de un eje de coordenadas. Es decir, hará que la imagen se refleje. Si los factores de escala son idénticos a la transformación, es una escala uniforme. Factores de escala no idénticos alterarán las proporciones de la imagen, esto se llama una escala diferencial.

Transformaciones de traslación

A menudo, una geometría debe ser trasladada a una posición diferente. Mediante el aumento de desplazamientos t_u y t_v para u y v , respectivamente, todos los puntos en el plano xy son trasladados a una nueva posición. La transformación es representada:

$$[x \quad y \quad 1] = [u \quad v \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_u & t_v & 1 \end{bmatrix} \quad \text{ec. (xx)}$$



Figura 2.23 Desplazamiento de un pentágono

Transformaciones de rotación

Una operación de gráficos fundamental es la rotación de una figura alrededor de un punto dado a través de un cierto ángulo (θ). Expresado por:

$$[x \ y \ 1] = [u \ v \ 1] \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{ec. (xxi)}$$



Figura 2.24 rotación con respecto al origen y un ángulo

En coordenadas homogéneas, formará una rotación en sentido horario (sobre el origen) para valores positivos de θ en todos los puntos en el plano uv .

Transformaciones de desplazamiento angular

Es una transformación que desplaza cada punto en una dirección fija (u), por una cantidad proporcional (h_u) a su distancia con signo a partir de una línea que es paralela a esa dirección. Un desplazamiento angular en la dirección v es dado por:

$$[x \ y \ 1] = [u \ v \ 1] \begin{bmatrix} 1 & h_u & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{ec. (xxii)}$$



Figura 2.21 desplazamiento angular (en dirección horizontal) de un pentágono

En la figura cualquier punto con coordenadas (u, v) al modificar el punto $(u+h_u, v)$. En este caso, el desplazamiento es horizontal, la línea fija es la del eje U .

De forma similar el desplazamiento angular en la dirección “ U ” es dado por:

$$[x \quad y \quad 1] = [u \quad v \quad 1] \begin{bmatrix} 1 & 0 & 0 \\ h_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{ec. (xxiii)}$$

2.5.2 Técnicas de Morphing

Las técnicas Morphing tienen como base las transformaciones de escala, rotación y translación. Estos inician con la selección de los puntos base (puntos de control) correspondientes en cada imagen. Los puntos de control se utilizan entonces para calcular una transformación y un mapeo que define la espacial relación entre todos los puntos en las dos imágenes. Las funciones de deformación se utilizan para interpolar las posiciones correspondientes de los puntos de control a través de la secuencia de Morphing.

Disolución cruzada.

Antes del desarrollo de Morphing, las transiciones entre dos imágenes fueron hechas habitualmente por disolución cruzada (por ejemplo, una interpolación lineal de color se desvanezca de una imagen a otra). Una disolución cruzada se aplica generalmente a toda la imagen y la textura de la imagen original se transforma a la imagen de destino mediante la mezcla el color de los píxeles. (Wolberg, Image morphing: a survey, 1998)

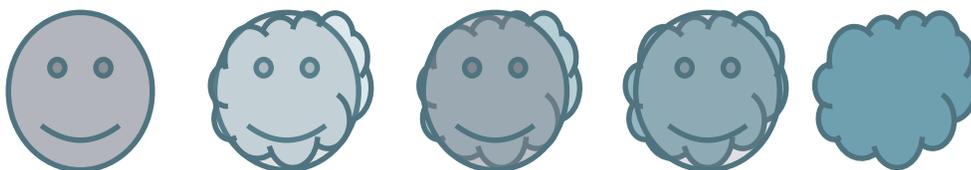


Figura 2.21 Disolución cruzada de imágenes

Deformación de malla

Usando un cuadrilátero no uniforme de mallas para especificar los puntos de control de una trama, se calcula a partir de los puntos de malla correspondientes a través de algo así como la interpolación spline correspondiente. (Wolberg, Digital image warping, 1990)



Figura. 2.22 Deformación por malla

Morphing de campo

El Morphing de campo utiliza un conjunto de segmentos de línea para especificar las características de una imagen. Un par de líneas en las dos imágenes determinarán una trama de su sistema de coordenadas local. Cuando se especifica más de un par de líneas de una media ponderada se utiliza para determinar la influencia de cada par de líneas en las imágenes. (Beier & Neely, 1992)

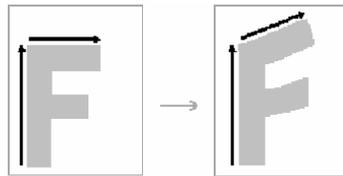


Figura 2.23 letra "F" segmentada por 2 líneas

Simplifica la tarea de especificación de una función. En lugar de utilizar mallas y splines para especificar características, esta técnica hace uso de segmentos de línea. Cada pixel de coordenadas se transforma por una rotación, traslación y escalado o transformando así toda la imagen. Sin embargo, algunas transformaciones afines, tales como escalas uniformes, no son posibles de realizar con este método.

Las "Snake's" y deformaciones multinivel de forma libre

Esta técnica "Snake" se utiliza para simplificar la especificación característica en las imágenes de origen y de destino. "Snake's" (Kass, Witkin, & Terzopoulos, 1988) son las estrías, minimizando la energía que se mueven bajo la influencia de la imagen y de restricción fuerzas.

El uso de las "Snake's" se basa en gran medida de las características de una imagen que está bien definida por sus bordes, es decir una superficie cerrada.

Para especificar una función de una "Snake" se inicializa mediante la colocación de una poli-línea (puntos de control conectados) cerca de una característica.

La correspondencia entre dos Snake's se puede derivar de las polilíneas que proporcionan sus posiciones iniciales. La correspondencia entre la función de imagen de origen y de destino se traduce en un conjunto de pares de puntos muestreados en las primitivas de funciones.

Morphing 3D

Por lo general, Morphing entre las imágenes con la misma forma 3D se traducirá en formas que son matemáticamente diferentes. Esto significa que las técnicas descritas anteriormente no pueden manejar los cambios en el punto de vista del objeto base.

- Mediante partículas.

Es posible realizar un Morphing mediante partículas (Corkum, 1994), las cuales son pequeños segmentos heterogéneos de planos orientados que se encuentran organizados como una superficie con interacciones entre ellas, mediante fuerzas asociadas a un vector normal.

En un sistema de partículas orientadas, se tiende a organizar como una superficie que favorece a un arreglo esférico o uno plano, Corkum desarrollo funciones viables para la interacción de las partículas con direcciones, con el fin de reducir el número de partículas en un modelo sin pérdida de detalle fino. Un sistema de partículas heterogénea proporciona una representación mínima de la superficie mediante el uso de partículas orientadas de diferentes tamaños, ofrece una buena aproximación a una superficie dada, al tiempo que reduce el número total de partículas necesario para la representación.

El proceso de Morphing, Corkum lo divide en dos etapas separadas que lo llama macro y micro *morph's*. El *macro morph* consiste en una transformación de cuerpo rígido estándar o generalizada que implica vectores propios que se trasladan y el modelo inicial gira de tal manera que se logra una nueva posición en el modelo de destino. Una vez que éste ajuste se ha logrado, se realiza un *micro morph*, se asume que los dos modelos deben ser superpuestos de la mejor manera en forma.

La transformación visual final se obtiene combinando la metamorfosis *macro morph* con *micro morph*. Debido a que no existe una solución única entre dos sistemas, el atractivo visual de cualquier metamorfosis es un juicio puramente subjetivo.

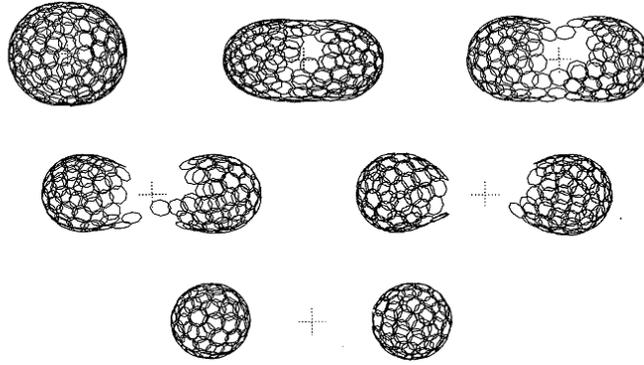


Figura 2.24 Morphing de una esfera a 2 esferas pequeñas. (Corkum, 1994)

- Mediante mallas

Este tipo de Morphing (Zhang, 2010) desarrolla su proceso de la metamorfosis en 4 bloques:

- Parametrizar una esfera para conseguir los puntos de control de una malla.
- Generar un esqueleto mediante un algoritmo de contracción y extracción de una superficie. (Au, Tai, Chu, Cohen-Or, & Lee, 2008)
- Realiza un remallado en la esfera.
- Mediante un algoritmo integra la esfera parametrizada a la superficie conjunto al esqueleto de la misma, los puntos de control de la esfera se interpolan para proceder con el Morphing.

CAPÍTULO

3

3. DESARROLLO

Para revisión de la bibliografía respecto al desarrollo embrionario se tomaron dos fuentes bibliográficas en una primera instancia, así como asesoría de doctores por parte de la Facultad de Medicina en la UNAM (Arce Cedeño & Hernández Sosa, 2014), con el objetivo de introducirse en el tema.

En el marco de referencia se encontró que las figuras del corazón embrionario son irregulares y que se pueden dibujar con curvas, las cuales tienen sus singularidades como crestas y valles al interior de los ventrículos. Los conceptos de parametrización que resultaron ser de mayor ventaja son los de puntos, curvas y superficies, mientras que la técnica de morphing se encontró esencial en el proceso de graficación. Se propone que utilizando los conceptos anteriores es posible construir modelos geométricos del corazón embrionario en un entorno virtual.

Durante la revisión de la literatura, se encontró que ya se tienen identificadas las características que comparte el corazón humano en desarrollo, es decir, los médicos han propuesto las geometrías iniciales, intermedias y finales del proceso embrionario, también han establecidos el tiempo de desarrollo.

Se consultó literatura en la cual la metodología y su implantación muestra que es posible usar como referencia la parametrización de partes mecánicas (Kirkire & Suryawanshi, 2009), donde por ejemplo, desarrollan la parametrización de una caja de engranes utilizando CATIA, mediante la programación de macros y el lenguaje de programación Visual BASIC, donde modifican los valores dimensionales, en general, cualquier geometría que tenga un número se puede cambiar con alguna función específica.

Al poseer propiedades dimensionales, una pieza se puede obtener con referencia de una imagen en un plano. En piezas mecánicas se usan 3 planos principales para modelar una pieza. Para el presente trabajo se estudiará el desarrollo del corazón embrionario agrupándolo en características, de donde se propone que estas sean forma y tiempo.

Mientras que para simular el cambio de forma que ocurre en el corazón embrionario al menos se requieren dos imágenes de referencia la inicial y la final. En

el mismo plano, al tener como objetivo una geometría intermedia, se requiere interpolarla respecto al cambio que se pretende obtener, como marco de referencia se considera 0% el estado inicial y 100% el estado final (Goldstein & Gotsman, 1995).

Con lo anterior en mente, el tiempo de la visualización requiere de una función, en la cual al incrementar el tiempo las geometrías cambien de forma proporcional, un aplicación donde el tiempo ha sido usado en Visual BASIC (Molina Martínez & Jiménez Buendía, 2010) es en la programación de alarmas de eventos.

Dado lo anterior es posible representar y simular los cambios del corazón embrionario, utilizando la información existente, con énfasis en el proceso de tabicación del corazón, así como mostrar las diferentes perspectivas y planos de corte donde se propone la siguiente metodología para crear modelos paramétricos en dos y tres dimensiones, basados en la técnica de Morphing.

3.1 Metodología propuesta para la parametrización de imágenes

Contar con imágenes de referencia

Se seleccionan dos imágenes de referencia, una de estado inicial y otra de estado final de cada etapa.

Definición de parámetros

Se delimitan los parámetros de la imagen y se elige una geometría apropiada para representar dicha imagen, las cuales se tomarán como referencia.

Proceso de graficación

Mediante relaciones paramétricas y razones de cambio se desarrollan programas que faciliten la edición y creación de geometrías intermedias basadas en la técnica del morphing.

Resultados

Se cuestiona sí era necesario contar con una mayor cantidad de imágenes, si no, se procede con los resultados

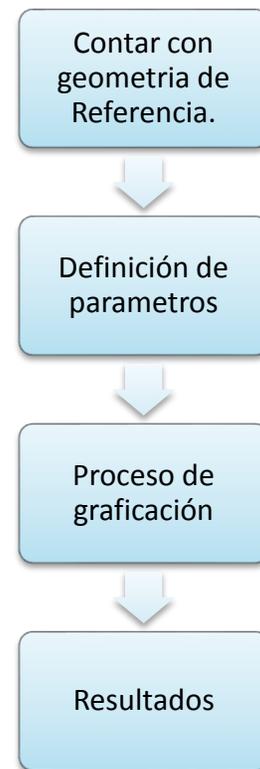


Figura 3. 1 Diagrama de la metodología.

3.1.1 Contar con imágenes de referencia

Una pieza, al poseer propiedades dimensionales, se puede obtener con referencia de una imagen en un plano. En piezas mecánicas se usan 3 planos principales para modelar una pieza, por lo que es necesario determinar el alcance del proyecto.

Para la reconstrucción de un modelo en tres dimensiones generalmente se ocupa la proyección de un sólido en los tres planos principales. Al tener que trabajar con una modelo irregular, para su reconstrucción en tercera dimensión se requiere otra alternativa de construcción. Se propone, una sucesión de *cortes* del modelo que pueda representar curvas discretas del modelo trimensional.

El ejemplo de las cónicas (Fig. 4.2) ayuda a comprender el concepto de *cortes*, en el cual en cada plano hay una intersección que caracteriza a la pieza. A partir de planos, y la unión de las intersecciones de los mismos se puede crear un modelo en tercera dimensión.

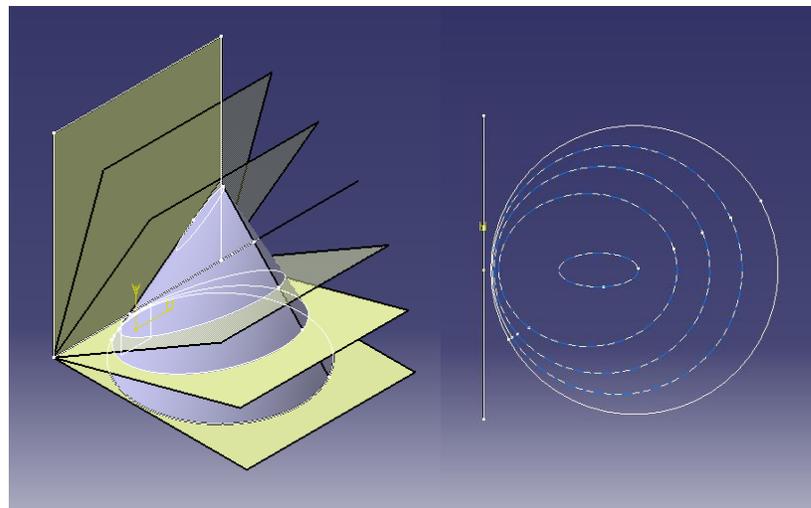


Figura 3.2 (a) Cono cortado por varios planos. (b) Vista superior de las intersecciones.

En el caso de las cónicas se parte de un eje de referencia donde inicia desde el plano en el que hay una área máxima, y finaliza en el plano con un área mínima, los cortes entre mayor es la cantidad de imágenes que se toman para los planos se puede obtener una aproximación más cercana a la referencia. La cantidad necesaria de planos depende de la complejidad de la geometría, esto debido a que entre menor sean los cambios entre un plano y otro, el modelo se puede representar de una forma más “fiel”.

3.1.2 Definición de parámetros

Para tener una aproximación con respecto a las geometrías de referencia, se proponen los siguientes parámetros que se requirieran tener como base.

- Angulo entre planos (separación entre planos)
- Cantidad de puntos
- Porcentaje de cambio
- Velocidad de cambio

El ángulo entre los planos Fig. 3.3, ya que al determinar el ángulo como parámetro, se controla la separación de los planos que se van a usar como referencia.

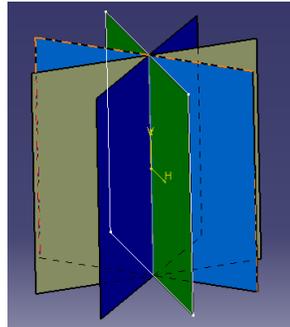


Figura 3.3 cuatro planos con un eje vertical común, con un ángulo entre planos de 45°

Otro parámetro es la cantidad de puntos, debido a que en un plano de corte, una silueta de la intersección se puede caracterizar por dicha cantidad de puntos que definen a una línea o curva, al tener complejidad en una silueta podemos establecer la cantidad de puntos necesarios para describir el perímetro.

Es necesario tener igual *cantidad de puntos* de dos siluetas en el plano de corte, debido a que se requiere una fórmula para interpolar, trabajando con las imágenes de referencia inicial y final.

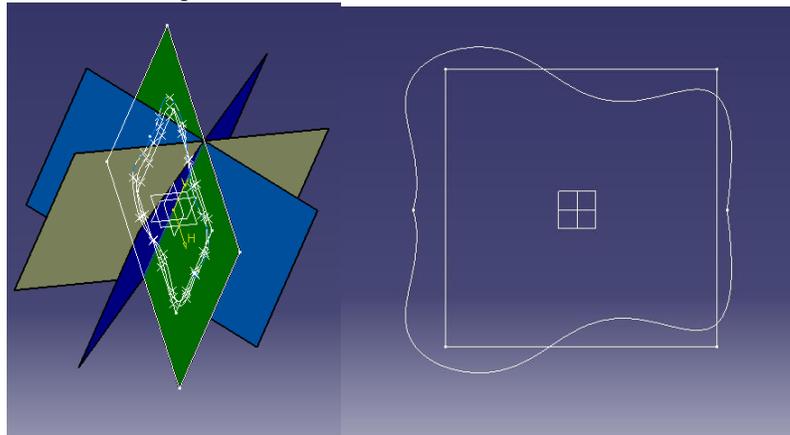


Figura 3.4. (a) Plano Verde isométrico, (b) vista normal al plano de la intersección.

En la figura 3.4, se muestra un caso general, donde un cuadrado sólo requerirá de 4 puntos para caracterizarlo, mientras que para la curva, se requieren más puntos. Para determinar los puntos mínimos de ambas figuras, y recordando el uso de la curva Spline, sus características indican que existen unos puntos singulares donde la derivada es cero y para el ejemplo sólo se ocuparían 18 puntos para caracterizar la curva.

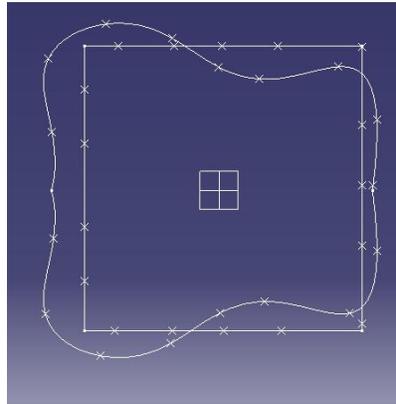


Figura 3.5. Plano con 18 puntos en cada intersección.

Para interpolar ambas siluetas es necesario incrementar o disminuir la cantidad de puntos, en este caso es recomendable el incrementar la cantidad de puntos para igualar a los de la spline, debido a que en la spline no se pueden reducir la cantidad, siendo los mínimos y necesarios para caracterizar de forma correcta la curva Fig. 3.5, sin embargo, para este caso se requiere una mayor cantidad de puntos en las esquinas, porque estamos ocupando una curva para caracterizar una recta, debido a que se corre el riesgo de visualizar un chaflán en lugar de una esquina cuando se requiera la transformación de cuadrado (silueta inicial) a curva (silueta final).

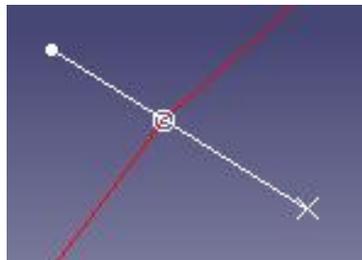


Figura 3.6 Punto inicial (x) intermedio (O) y Final (•)

Otro parámetro que se propone es el porcentaje de cambio de la geometría uno (Spline) a la geometría dos (cuadrado) esto debido a que el objetivo que se

busca es el de generar geometrías intermedias. Con esto en mente, este porcentaje caracteriza a cada punto intermedio, Figura 3.6, donde este valor, modifica a todos los puntos intermedios.

La velocidad de cambio del estado inicial al estado final, es un parámetro que se establece para ver la simulación de una forma más rápida o lenta, y este indica el tiempo de espera entre las interpolaciones de las geometrías intermedias.

3.1.3 Proceso de graficación.

Se propone utilizar el programa CATIA, principalmente porque es una solución con una amplitud de herramientas y uso extendido en la industria y rama de la ingeniería mecánica. Las relaciones paramétricas y razones de cambio que se aplicaron, fueron base para desarrollar unos programas que generó geometrías intermedias con respecto a los estados inicial y final basadas en la técnica del morphing, donde se construyeron elementos que facilitan al proceso de la generación de geometrías.

3.1.4 Imágenes intermedias

Para tener una mejor aproximación y evitar un cambio brusco en el modelo de la figura 3.2, se debe cuestionar la necesidad de incluir más planos intermedios, debido a que al disminuir el ángulo entre planos e incrementando la cantidad de los mismos, se tiene una mayor información para generar un modelo más fiel a las imágenes de referencia.

Al final de este proceso se obtendrá un modelo tridimensional con base en las geometrías de referencia, logrando simular el morphing dentro del entorno del software.

4. CASO DE ESTUDIO

Con la idea de evaluar la metodología propuesta se realiza el desarrollo de la transformación gráfica utilizando parametrización en el programa CATIA, para modelar el proceso de tabicación durante el desarrollo embrionario del sistema cardíaco. Aplicando la metodología propuesta aplicada al caso de estudio (Fig. 4.1).

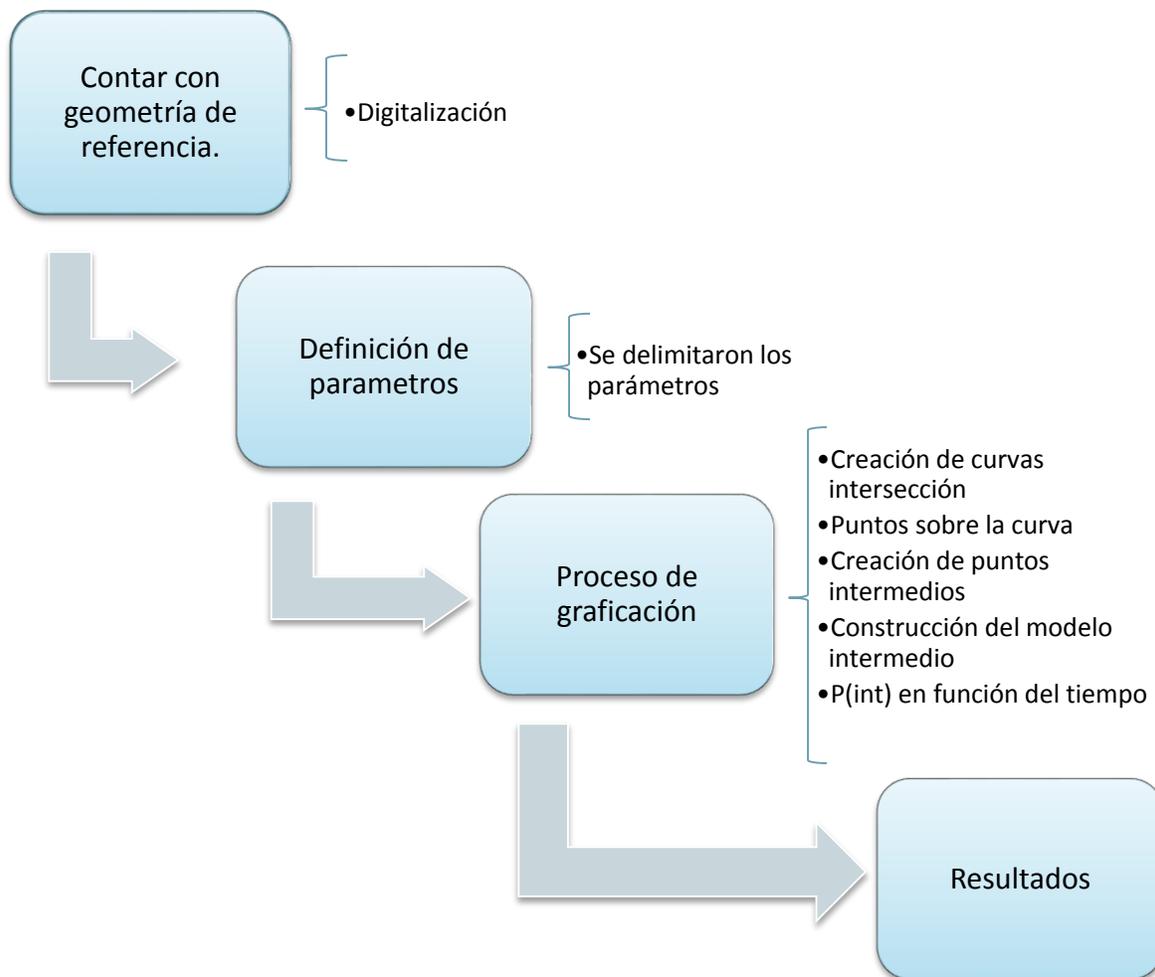


Figura 4.1 Proceso en bloques de desarrollo del caso de estudio.

4.1 Contar con geometrías de referencia

La tabicación del sistema cardiaco se encuentra dentro del periodo que abarca entre la quinta y sexta semana del desarrollo, al ser un proceso conocido por los médicos, la facultad de medicina (Arce Cedeño & Hernández Sosa, 2014) proporcionó 3 modelos (fig. 4.3) en plastilina (en su etapa inicial, intermedio y final) para usarlo como geometrías de referencia.

De los tres modelos se seleccionaron 2 geometrías de referencia, una de estado inicial y otra de estado final para la construcción de dos modelos intermedios uno entre el modelo 1-2 y otro entre los modelos 2-3, a continuación se muestran los modelos en plastilina Fig. 4.3.



Fig. 4.3. Modelos en plastilina 1 (Izquierda), 2 (Centro) y 3 (Derecha)

4.1.1 Digitalización de geometrías de referencia.

Según (Ayala R., et. al. 2015) es posible obtener modelos de superficie y sólidos a partir de tomografías, por lo que decidió partir de 3 modelos en tres etapas de formación del corazón, correspondientes a la tabicación del sistema cardiaco. Fig. 4.2. Cada uno de estos modelos en plastilina se digitalizaron a través de un tomógrafo y se obtuvieron los modelos de superficie.



Fig. 4.4 Modelos de superficie de los modelos 1 (Izquierda), 2 (Centro) y 3 (Derecha)

Dichos modelos de superficie (Fig. 4.4) fueron las geometrías de referencia. Para la visualización de los modelos intermedios con respecto al cambio de geometría de 1 a 2 y de 2 a 3, se consideraron las dimensiones volumétricas de los modelos de plastilina los cuales proporcionan un marco de las unidades en las cuales trabajamos en este caso los límites de cada una, las cuales son:

	Largo (mm)	Ancho (mm)	Alto (mm)
<i>Modelo 1</i>	110	80	51
<i>Modelo 2</i>	114	83	47
<i>Modelo 3</i>	120	84	50

Tabla 4.1 Dimensión de los modelos de plastilina Digitalizados.

4.2 Definición de parámetros

Se delimitaron los parámetros y variables, que definen los dominios de las geometrías, estos describen de manera general curvas que no tienen continuidad geométrica, y además son irregulares.

Planos de intersección

Para tener un mejor control de la superficie se generaron secciones discretas de la geometría utilizando planos de corte. Las secciones que se obtuvieron contienen las curvas intersección definidas por los planos de corte y la superficie.

Así se estableció que los planos de corte deberían construirse equidistantes con 5 mm de separación entre ellos.

Puntos de control

Los puntos de control se extraen a partir de las curvas intersección, y se determina la cantidad mínima de puntos necesarios para generar una curva parametrizada del tipo spline.

Porcentaje de cambio

Dado que se cuenta con los límites inferior y superior de cada sección fue posible generar una n cantidad de subsecciones, que permitió generar la simulación de la formación de la tabicación.

4.3 Proceso de graficación.

Mediante relaciones paramétricas se desarrollaron nuevas herramientas computacionales “*in process*” que se requirieron para la construcción y simulación de geometrías intermedias con respecto al dominio inicial y dominio final. El entorno de programación usado para implantar el código de parametrización, fue VBScripts basado en el lenguaje Visual Basic.

Para ejecutar los programas implementados dentro de CATIA, se decidió asignar a cada uno de ellos un icono, concentrándolos en una barra de herramientas (Figura 4.5) dentro del entorno de CATIA.



Fig.4.5. Barra de Herramientas creada con los programas personalizados

En la figura 4.5 se muestra el Toolbar que contiene los íconos asignados para las siguientes tareas, de izquierda a derecha son:

- *Creación de Curvas intersección.*
- *Cantidad de puntos.*
- *Creación de puntos intermedios.*
- *P_{int} en función (t)*
- *Otras herramientas (ver sección 4.8)*

Creación de Curvas intersección.

Con el programa *creación de curvas intersección*, se agrega un objeto del tipo *HybridShapeCurveExplicit*, utilizando el juego de parámetros de entrada: superficie de referencia, plano de referencia y separación entre planos.

```
mySuperficie 'Selección de la superficie  
myPlano 'Selección del plano de referencia  
input 'Separacion entre planos [mm]
```

A partir del plano de corte coronal se referencian planos paralelos equidistantes con el parámetro “input”, estos planos al intersectar la superficie del modelo, con lo que se crean objetos del tipo curva intersección.

El proceso se describe en el diagrama de flujo (Fig. 4.5) como:

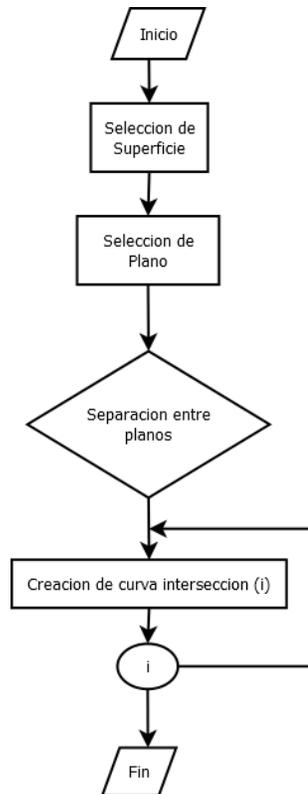


Fig. 4.6 Diagrama: Creación de curvas Intersección

Para ilustrar el proceso de esta sección se usa la geometría del Modelo 1.

- Inicio. Se llama al programa con el icono asignado (“Creación de Curvas intersección”).



Fig.4.7. Icono Creación de Curvas intersección. Recalcado en la barra de herramientas

- Selección de superficie. Dentro del entorno se selecciona la superficie (“Automatic Surface.1”, color verde) de referencia que será tratada con el macro.

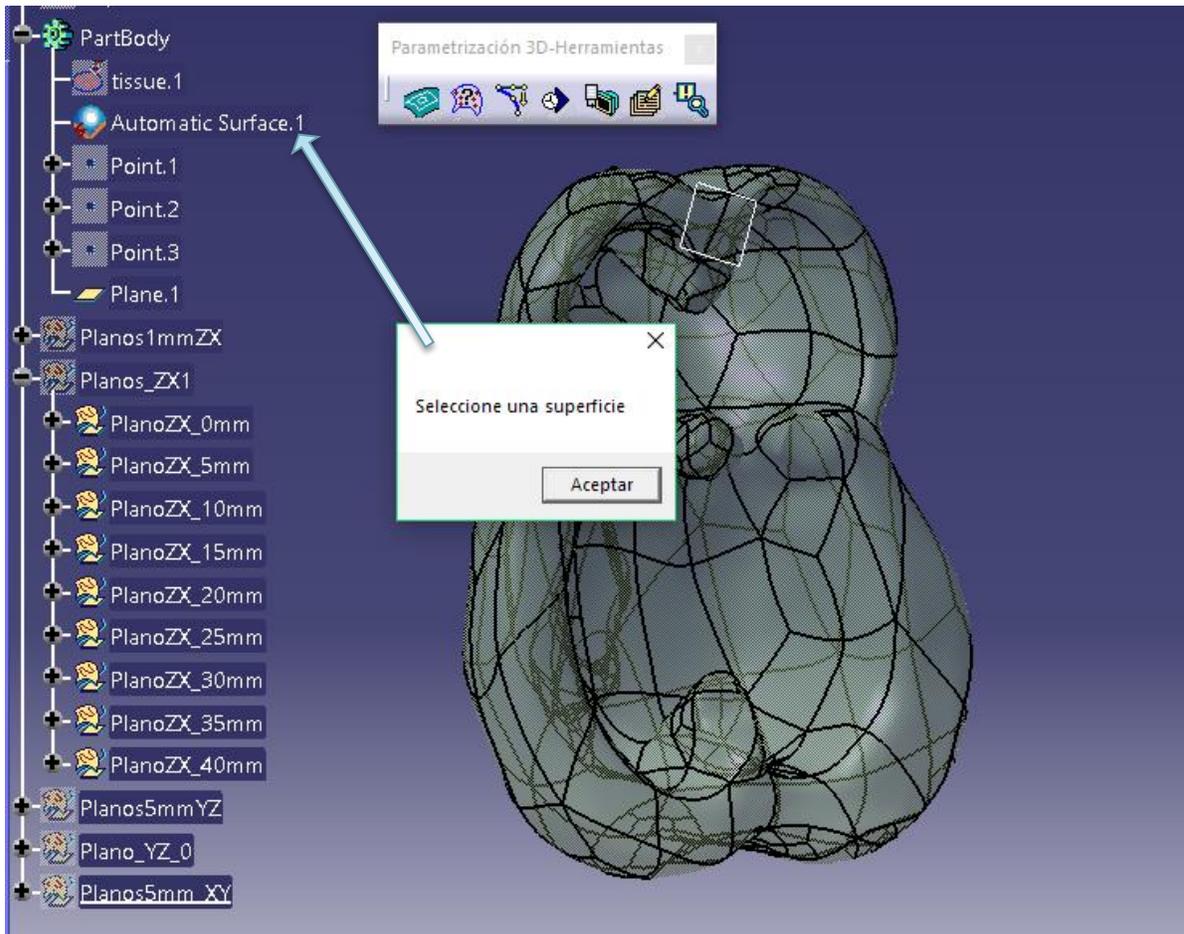


Fig. 4.8 Selección de superficie de referencia en el árbol dentro de CATIAV5 del modelo 1

- Selección de plano. Se seleccionó “plane.1” como un plano de referencia. Dicho plano se tomará para la creación de otros planos paralelos a este y al plano ZX.

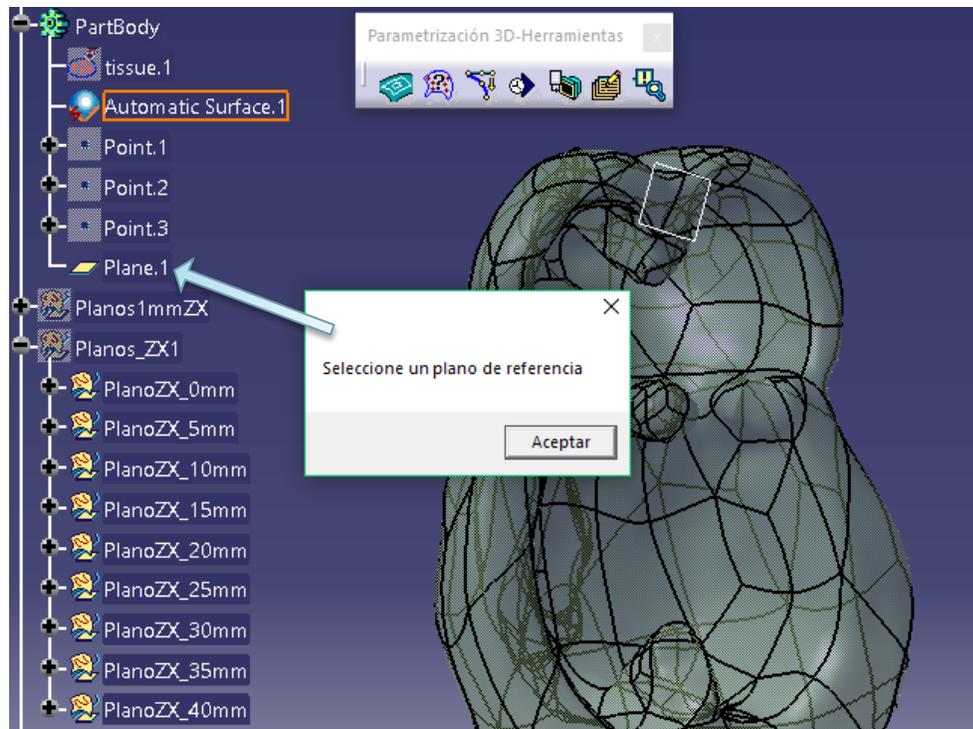


Fig. 4.9 Selección de Plano de referencia “plane.1”

- Separación entre planos. Entrada en la cual se debe indicar la longitud de la separación entre planos. la separación fue entre 5mm por cada capa.

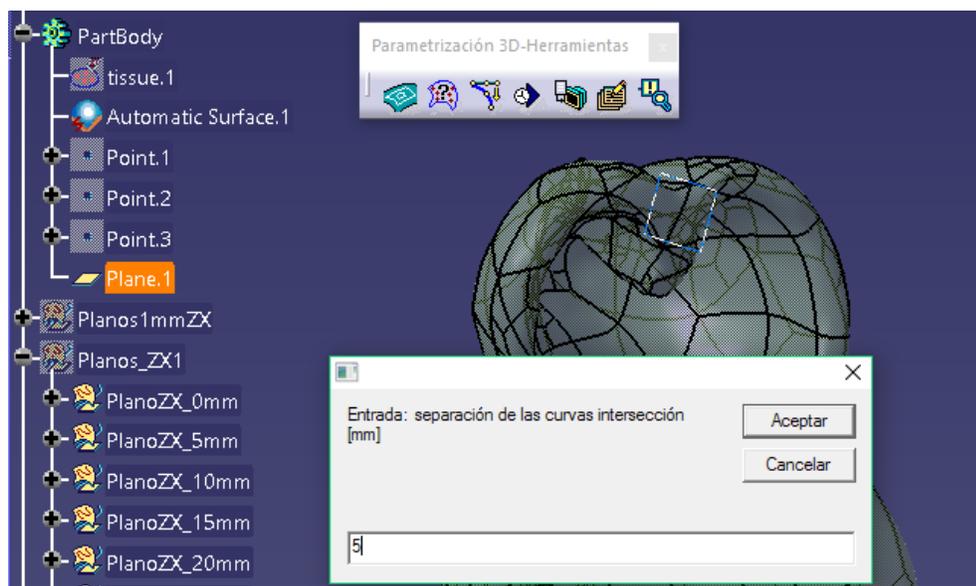


Fig. 4.10 Entrada de separación. (5 mm)

- Creación de curva intersección. Con la referencia de la superficie, y el plano de referencia, la intersección entre ellos corresponde al conjunto de datos que es la curva intersección.
- Ciclo i . Con la referencia del plano intersección y la separación entre planos se crea otro plano paralelo al de referencia con el incremento del valor de la separación entre planos. Dicho plano también interseca a la superficie, posteriormente se crea una curva intersección correspondiente al plano y la superficie, repitiéndose hasta que la superficie y el plano no se intersectan con un total de 9 capas



Fig. 4.11 Planos paralelos a ZX en el modelo 1 con referencia a "plane.1"

Debido a que para fines de visualización fue preferible truncar 2 mm del modelo uno menos del 5% de la geometría original y los correspondientes al modelo dos, los cortes de las capas no altera la visualización del resultado final.

Como se trabajó con modelos de superficies y planos equidistantes y paralelos entre sí, se obtuvieron sus curvas características que representen al modelo de referencia. Dichas curvas de nivel contienen a los puntos de control que caracterizan a la geometría en el espacio.

Se constituyó de la misma forma con el Modelo 2 (Fig. 4.12) y el Modelo 3 (Fig. 4.13), los cuales se superpusieron geoméricamente para tener los mismos planos de corte. En cada plano, al tener superpuestos los modelos, se pudieron apreciar los cambios generales que existen entre los modelos.

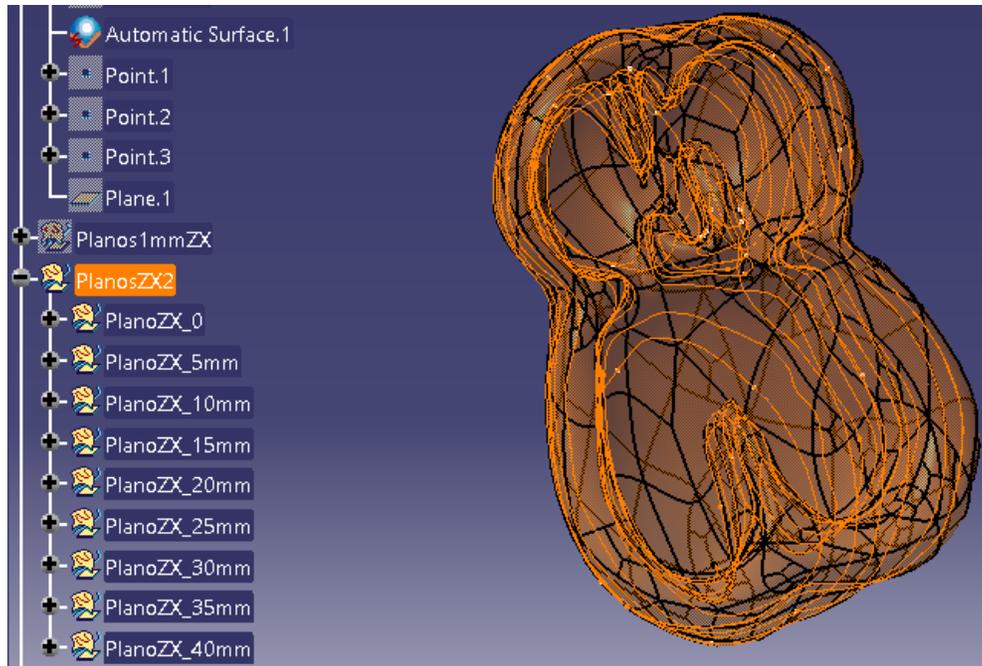


Fig. 4.12 Planos paralelos a ZX en el modelo 2 con referencia a plane.1

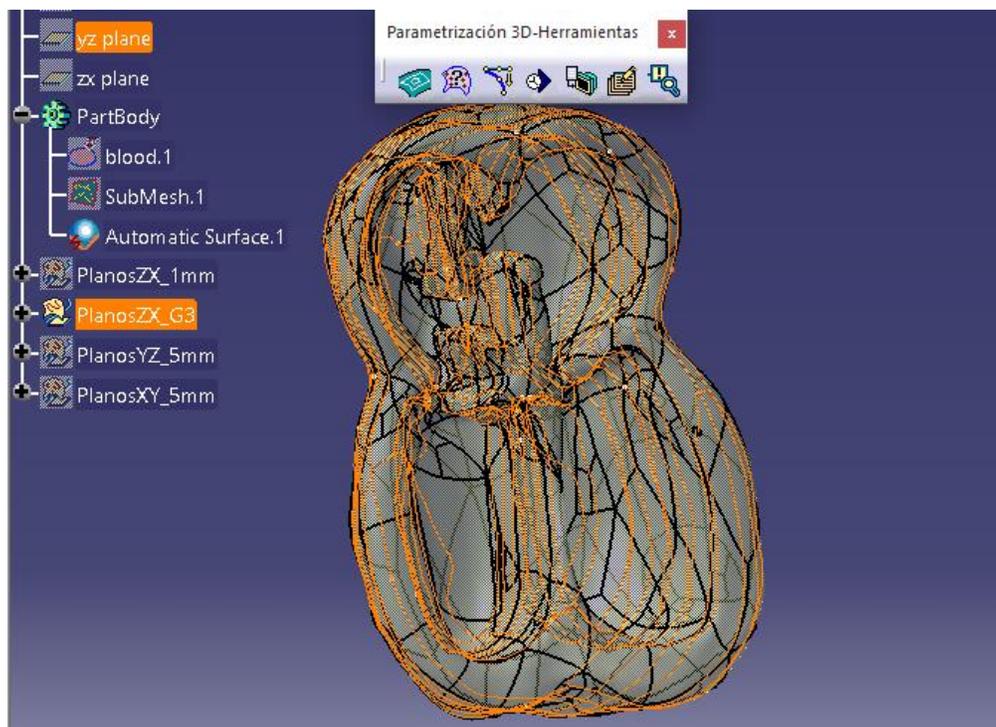


Fig. 4.13 Planos paralelos a ZX en el modelo 3 con referencia a plane.1

Al tener las curvas características de los modelos se prosiguió con la creación de los puntos de control de cada curva en los modelos.

Cantidad de puntos.

Para la creación adecuada de puntos de referencia se inició con el despliegue de una ventana de diálogo, la cual a partir de la selección de una curva intersección, indica un rango de “puntos equidistantes” apropiados con una distancia de separación de los puntos equidistantes entre 4 a 5 mm dependiendo de la longitud de la curva. Para la posterior construcción de los puntos de control intermedios en cada curva intersección.

Para asegurar la selección del objeto *HybridShapeCurveExplicit* se utilizó la clase “*edge*”, la cual sólo filtra a los objetos tipo curva, a su vez para manipular la curva seleccionada se le asignó a la variable *sCurvaRef*. La longitud de la curva se obtiene con el método “*Length*” del objeto *Mesurable*.

El proceso se describe en el diagrama de flujo como:

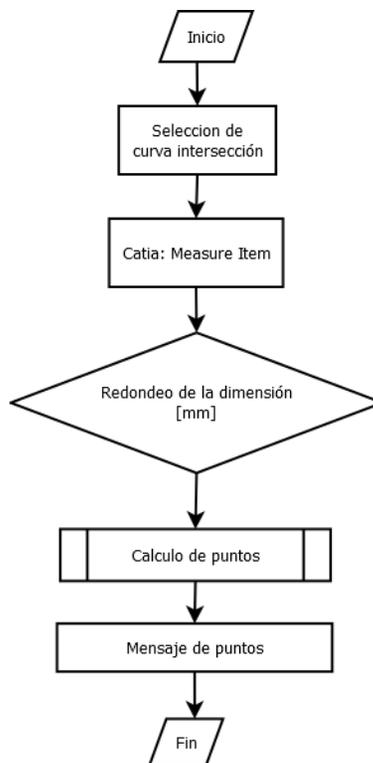


Fig. 4.14 Diagrama: Cantidad de puntos

- Inicio. Se inicia el programa, llamándolo con el icono asignado (“Cantidad de puntos”).



Fig. 4.15. Cantidad de Puntos. Icono recalcado en la barra de herramientas

- Selección de Curva Intersección. Despliega un mensaje de selección de la curva, en el cual únicamente se puede seleccionar los elementos tipo *HybridShapeCurveExplicit*, como se muestra en la figura 4.16.

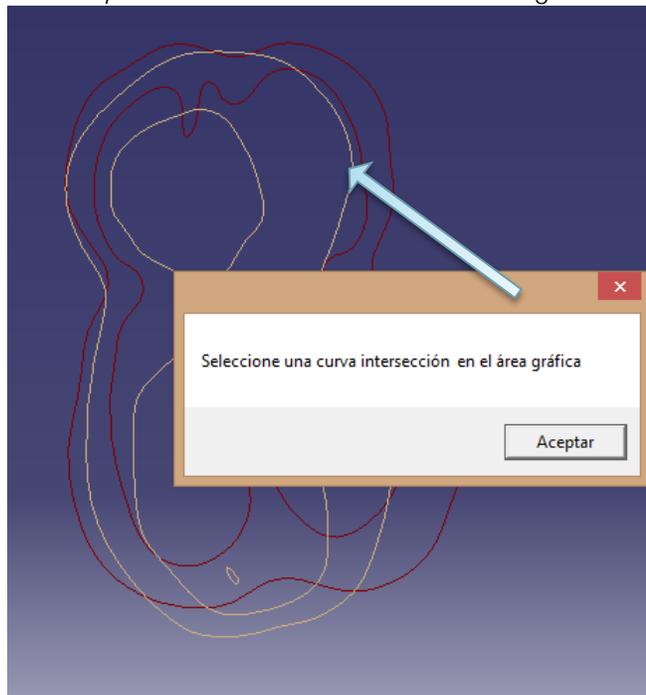


Fig. 4.16 Aviso, selección de la curva intersección

- Measure Item. Muestra el resultado del llamado del comando *Measure Item* de CATIA, el cual es una dimensión
- Redondeo. Pregunta al usuario mediante una caja de entrada cual es la dimensión que desea redondear (Fig. 4.17)

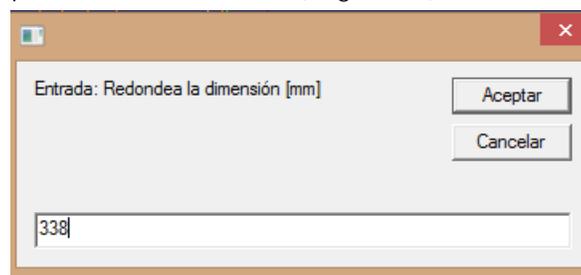


Fig. 4.17 Redondeo de dimensión de la curva

- Cálculo de puntos. Con la dimensión que se introdujo se calcula la cantidad de puntos con una relación de 4 y 5 mm de separación.
- Mensaje de puntos. Se muestra el resultado de la cantidad de puntos en un rango que es idónea para la curva seleccionada.

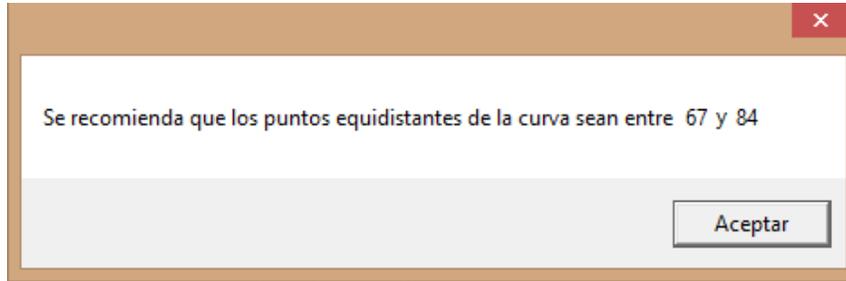


Fig. 4.18 Mensaje de puntos recomendados.

Con dicha cifra se utiliza la creación de puntos sobre la curva, por defecto del sistema, que toma como entrada la curva y la cantidad. De esta forma se construyen los grupos de puntos de referencia del primer plano ZX del Modelo 1 agrupado en “Geometrical Set” conjunto con los grupos de puntos de referencia del Modelo 2.

En la Figura 4.19 se muestran los puntos que componen a una curva en un plano de corte de los modelos 1 y 2. Estos puntos están agrupados en “Geometrical Set’s” para el modelo 1 “FG1_0_C1_pts” y para el modelo 2 “FG2_0_C1_pts”, con 64 puntos cada grupo.

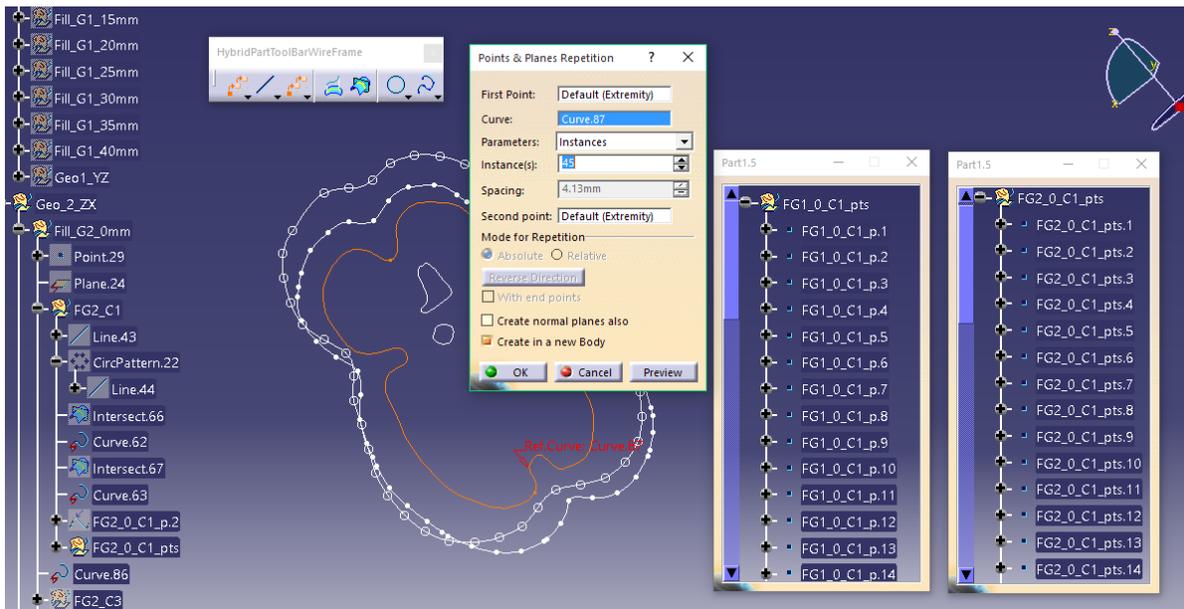


Fig. 4.19. Puntos de Referencia agrupados en “geometrical set” del modelo 1 y modelo 2, en el primer plano de corte y creación de nuevos puntos con Points&Planes.

Aplicando este proceso para cada curva de los 9 niveles creados anteriormente para los modelos 1 y 2. Al tener los puntos de Referencia de cada curva intersección se prosiguió con la creación de los “puntos intermedios”

Creación de puntos Intermedios (Pint)

Con los grupos de puntos de referencia agrupados en dominios, se realizó la interpolación entre los dominios inicial y final de las secciones, y así situar a los puntos de control intermedios.

Con el método *AddNewPointBetween* se toman los atributos de cada punto perteneciente a los dominios *ohybridbody1* y *ohybridbody2* de referencia, creando los objetos punto intermedios, a los cuales posteriormente se les asignó la relación de proporción.

ohybridbody1 ‘dominio inicial (Fig 4.12, árbol izquierdo)

ohybridbody2 ‘dominio final (Fig 4.12, árbol derecho)

F_cont ‘Porcentaje de cambio

Dado que se tienen *n* número de secciones, se creó un macro *in process*, para generar los puntos de control intermedios

El proceso se describe en el diagrama de flujo como:

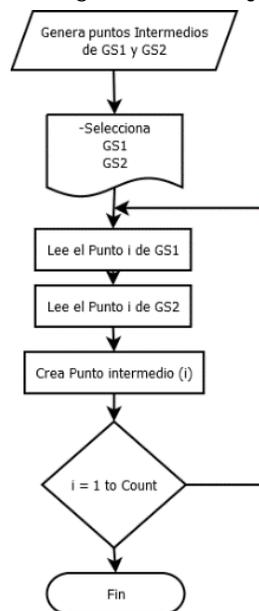


Fig. 4.20. Creación de puntos intermedios

- Inicio. Se inicia el programa, llamándolo con el icono asignado (“Creación de puntos Intermedios”).

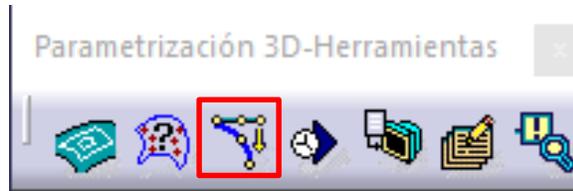


Fig.4.21. Creación de puntos Intermedios. Icono recalado en la barra de herramientas

- Selección GS1. Muestra un aviso donde se pide la selección de un “Geometrical Set” que contiene un grupo de puntos iniciales y bloquea cualquier otra selección con el filtro *HybridShapes* usado para asegurar la selección únicamente del grupo de puntos. Por ejemplo, la selección del Geometrical Set Inicial llamado FG1_0_C1_pts mostrado en la Fig. 4.22.

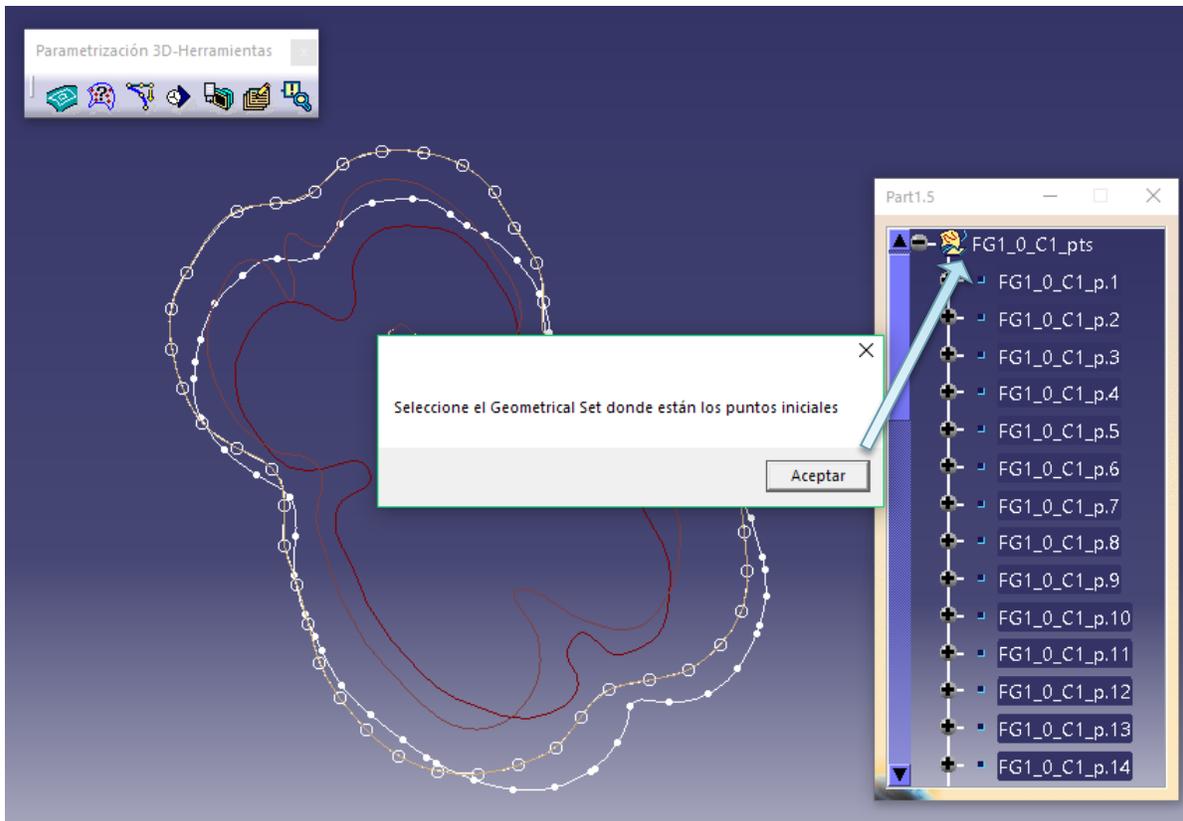


Fig. 4.22. Adquisición del dominio inicial.

- Selección GS2. Muestra un aviso donde se pide la selección de un “Geometrical Set” que contenga a un grupo de puntos finales y bloquea cualquier otra selección. La selección del Geometrical Set final Fig. 4.23, por ejemplo FG2_0_C1_pts.

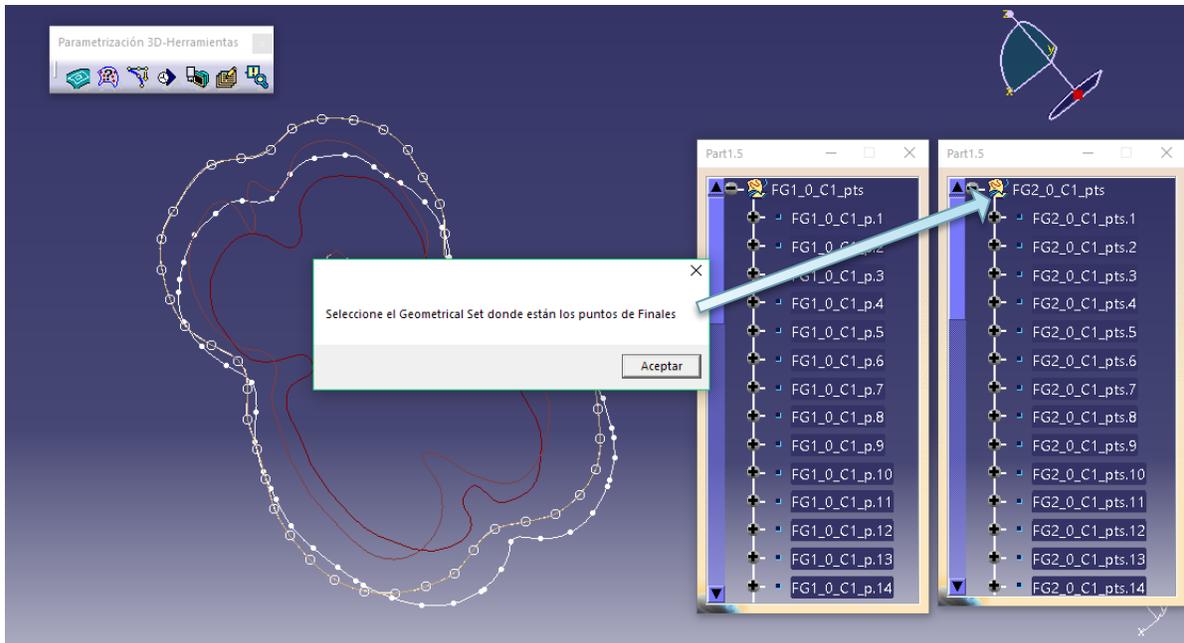


Fig. 4.23. Adquisición del dominio final.

- Selección GSInt. Avisa y pide el nombre del “Geometrical set” donde se encontrarán los Puntos Intermedios donde se alojarán los puntos intermedios resultantes de la interpolación, por ejemplo en el geometrical set: F12_0_C1 (Fig. 4.24).

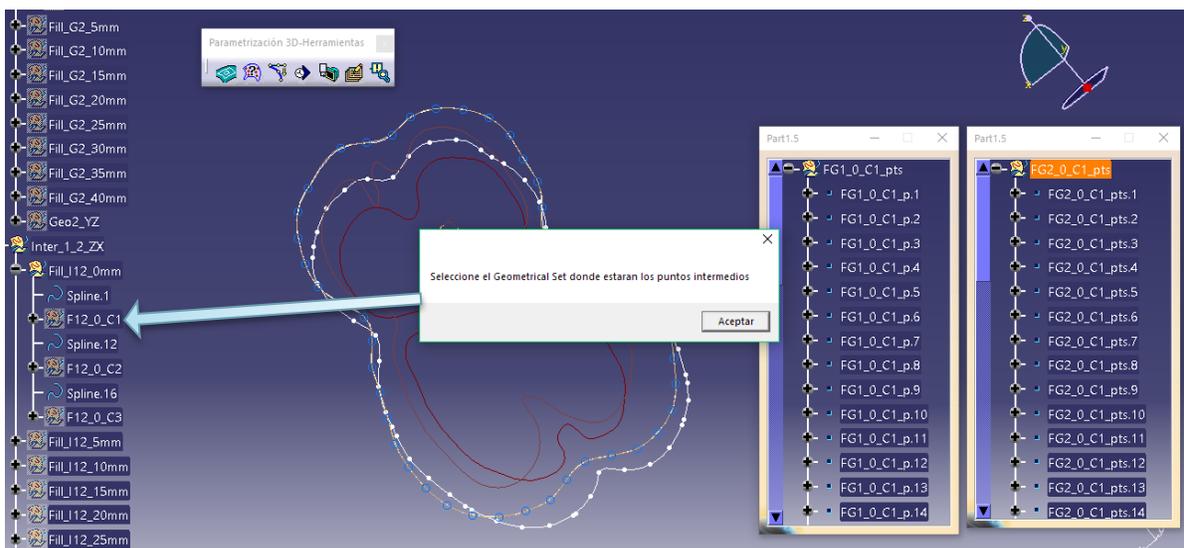


Fig. 4.24. Ubicación del dominio intermedio.

- Ciclo i. Se repite n veces las operaciones del ciclo desde $i=1$ hasta la cantidad de puntos de la selección GS1, las operaciones son:

- Lee el punto i de GS1. Lee las propiedades geométricas (x,y,z) del punto i de la selección del Geometrical set inicial. Para interactuar con el método *AddNewPointBetween* se utilizaron como parámetros dominio inicial y final, porcentaje de cambio y dirección. Siendo este último parámetro, la dirección de la normal de los dominios (eje Y), la cual es estática.
- Lee el punto j de GS2. Lee las propiedades geométricas (x,y,z) del punto j .
- Crea punto intermedio. Interpolando los valores del punto i , y el punto j . Esto generó cada punto intermedio resultando la figura 4.25.



Fig. 4.25. Puntos intermedios generados con el programa.

El programa se generó como un CATScript, debido a su sencillez visual y aparenta a una función nativa del programa. Con el proceso anterior (Fig. 4.26) se construyeron 3810 puntos intermedios formado una nube de puntos Fig.4.14 necesarios para construir el modelo intermedio entre los modelos “1” y “2”.

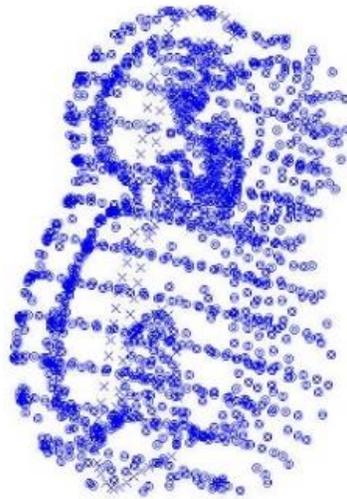


Fig. 4.26. Nube de puntos de control del corazón embrionario, corte coronal.

4.3.1 Construcción del modelo intermedio

Teniendo como base la estructura de datos de la nube de puntos, se crearon las curvas spline parametrizadas, para cada grupo de puntos de control (Fig. 4.27).



Fig. 4.27. 43 Spline's de la geometría intermedia en los planos paralelos a ZX

Las superficies paramétricas se generaron a partir de los perímetros definidos por las curvas spline paramétricas de cada plano de corte, como se muestra en la figura 4.27. Iniciando con el plano coronal y en niveles en dirección posterior separados a cada 5 mm.

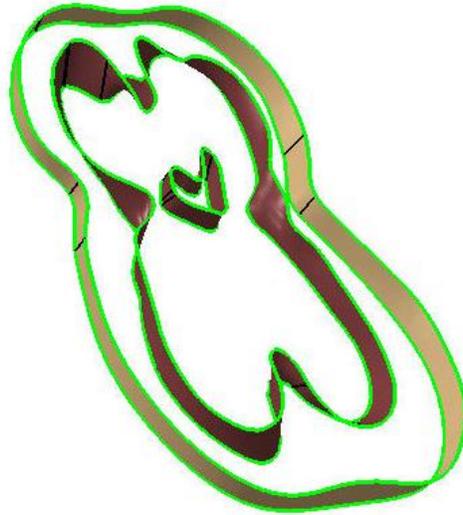


Fig. 4.28. Superficies generadas a partir de curvas spline parametrizadas.

De esta forma se realizó una construcción secuencial de superficies con los métodos estándar de CATIA (*AddNewLof*), por lo que no se requirió la programación en esta etapa. Aún con las 43 splines paramétricas el proceso de creación de la superficie del modelo intermedio, se requiere sólo de un comando nativo de CATIA para obtener el modelo de superficies de la figura 4.29.

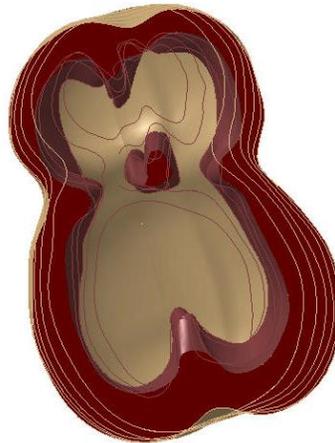


Fig. 4.29. Modelo intermedio de superficies.

Debido a la geometría del corazón, algunas de las curvas paramétricas generadas resultaron ser incompatibles con el método de defecto *AddNewLof*, para formación de superficies, ya que presentan discontinuidades y singularidades.

La discontinuidad entre la curva generatriz y las curvas directrices (curvas spline paramétricas), se presenta cuando hay puntos singulares que no concuerdan

con el “perfil de interpolación” y las curvas directrices, impidiendo la formación de superficies, y generándose huecos.

Para complementar el modelo intermedio fue necesario crear nuevas superficies referidas a planos perpendiculares al plano canal auriculoventricular, ya que se presentaron huecos en el modelo.

Al tener nuevos planos intersección, se aplicó la metodología usando los macros creados, comenzado con las curvas intersección perpendiculares, de las cuales se obtuvieron los puntos de referencia de los dominios iniciales y finales, con dichos puntos de referencia se generaron los puntos intermedios para construir las curvas paramétricas que definen a las superficies complementando la geometría intermedia.

Con los nuevos puntos de control perpendiculares se restringieron a segmentos en las curvas spline parametrizadas, con esta segmentación de las curvas interiores fue posible crear las nuevas superficies para formar la geometría (Fig. 4.30) de la tabicación, al interior del corazón.

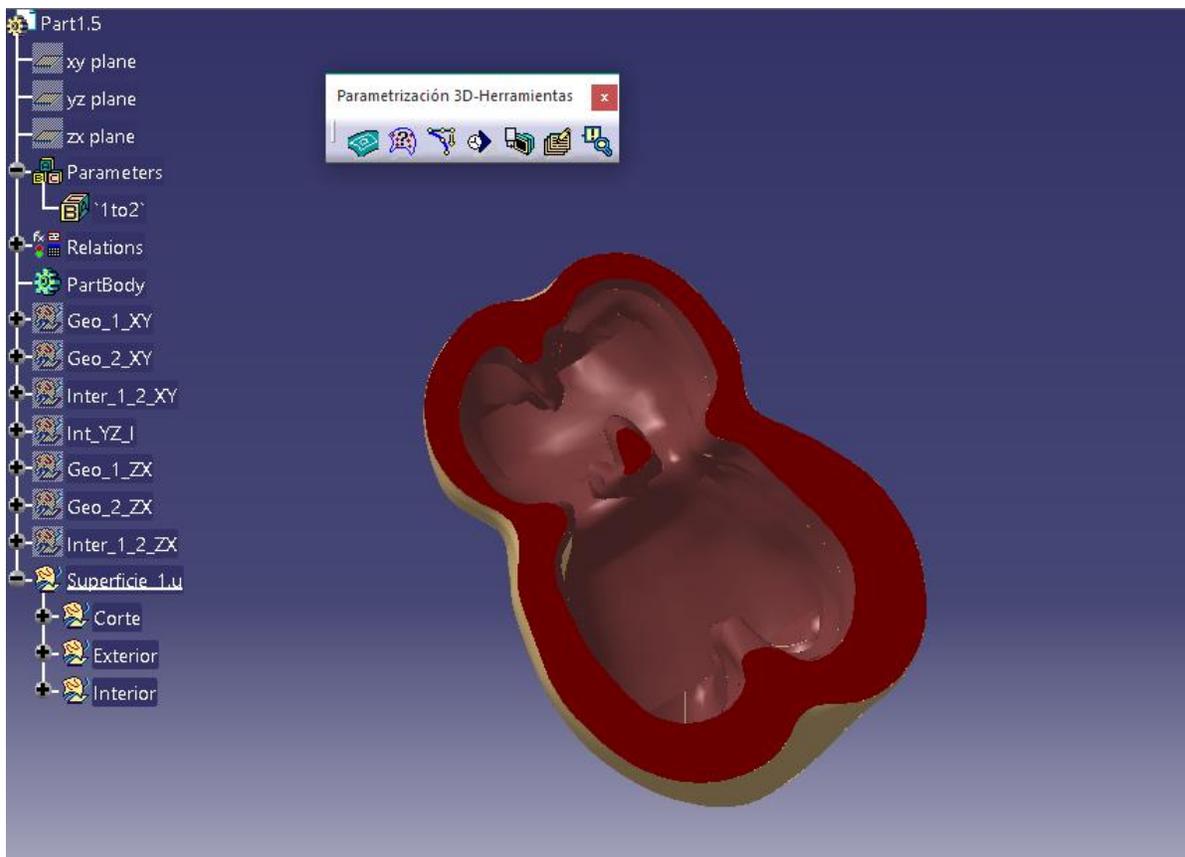


Fig. 4.30. Modelo de superficies intermedio entre el Modelo 1 y el Modelo 2.

4.3.2 P_{int} en función de t .

Se creó un macro que toma los datos del dominio intermedio y los modifica en función del *porcentaje de cambio* de estado inicial a final, mediante las funciones del punto intermedio que se crearon anteriormente en el proceso de *creación de puntos intermedios*.

El programa tomó como parámetro de entrada el Porcentaje de cambio.

F_{cont} 'Porcentaje de cambio' t_1 to t_2

La ejecución del programa “ P_{int} en función de t ” muestra la simulación del cambio entre la geometría del modelo 1 y el modelo 2, el cual modifica los valores de F_{cont} , en los elementos primitivos (puntos intermedios), por ende actualizando el modelo de superficies.

El proceso se describe en el diagrama de flujo (Fig. 4.31) como:

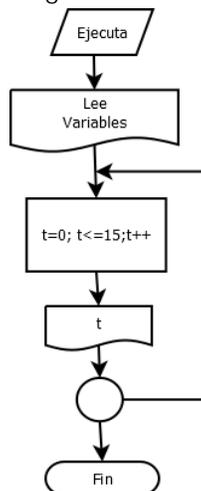


Fig. 4.31. Diagrama, Puntos int en función de t

- Ejecuta. Un botón generado dentro de CATIA, llama a las instrucciones.

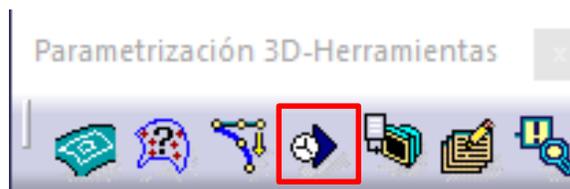


Fig.4.32. Creación de puntos Intermedios. Icono recalcado en la barra de herramientas

- Lee Variables. Se lee el documento con las funciones y los parámetros: porcentaje de cambio (Fig. 4.33).

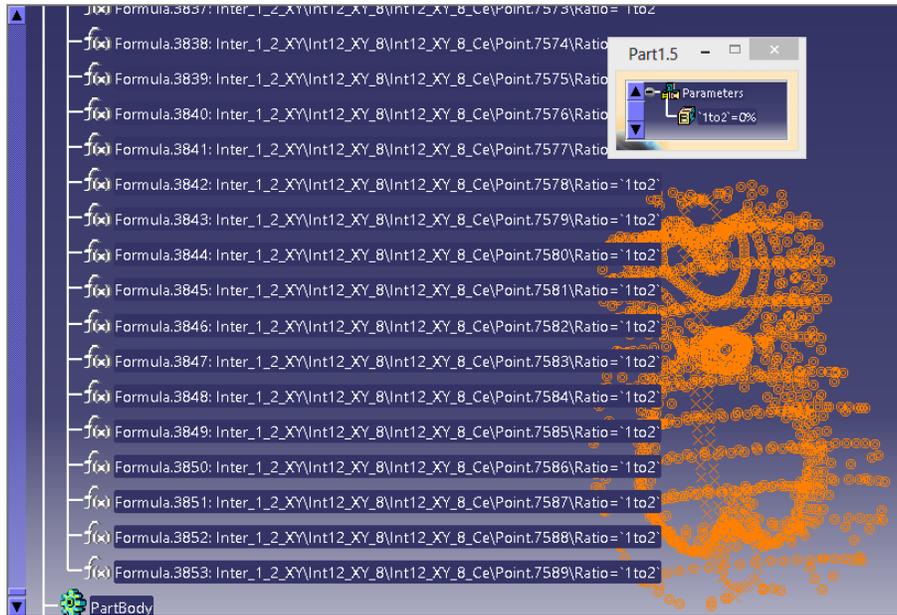


Fig. 4.33. Funciones asignadas a los puntos intermedios, y porcentaje de cambio ('1to2') del modelo intermedio.

- $t=0; t \leq 15; t++$ Es un ciclo en el cual inicia con el tiempo cero hasta que el valor máximo sea de 15.
- t . Se actualiza el modelo con un nuevo valor de porcentaje de cambio modificado.

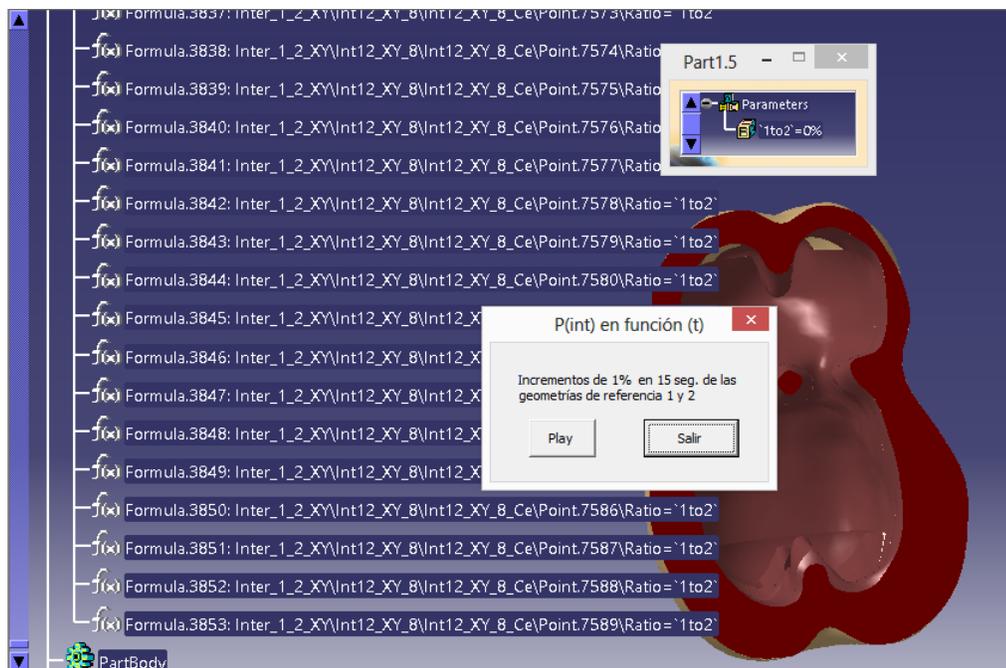


Fig. 4.34. Ventana de inicio de la simulación en incrementos de 1% durante 15 seg.

Se visualiza el cambio de la geometría después de ejecutar el comando hasta llegar al 100%.

4.4 Archivo CAD resultante de la parametrización y Morphing.

El modelo generado partiendo de las geometrías 1 y 2, tiene la información necesaria para visualizar en cualquier momento el cambio entre dichas geometrías, simulando el “Morphing” (Goldstein & Gotsman, 1995) mediante la construcción de las etapas intermedias. A continuación se muestra en las figuras 4.35 y 4.36 los cambios de 0%, 25% 50%, 75% y 100% de la geometría generada.

Para la visualización del Morphing de una forma fluida durante la simulación en el transcurso de 15 segundos, se consideró que la transición debería de cumplir los 30 frames por segundo (fps), de tal manera que se generen 450 geometrías diferentes con un incremento entre ellas de 0.22% y esto depende de las propiedades de la computadora.

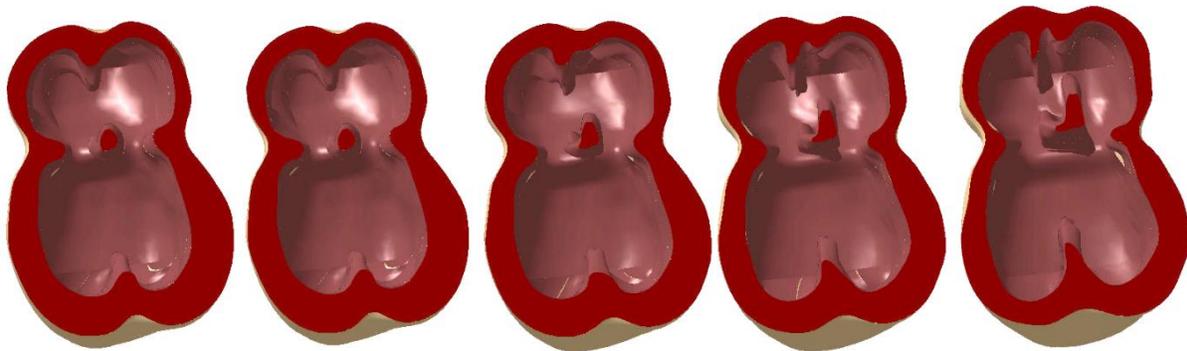


Fig. 4.35 Captura con la función programada de la geometría intermedia del modelo 1 al modelo 2 resultante con incrementos de 25% (razón de cambio)

El modelo generado partiendo de las geometrías 2 y 3, tiene la información necesaria para visualizar en cualquier momento el cambio entre dichas geometrías.

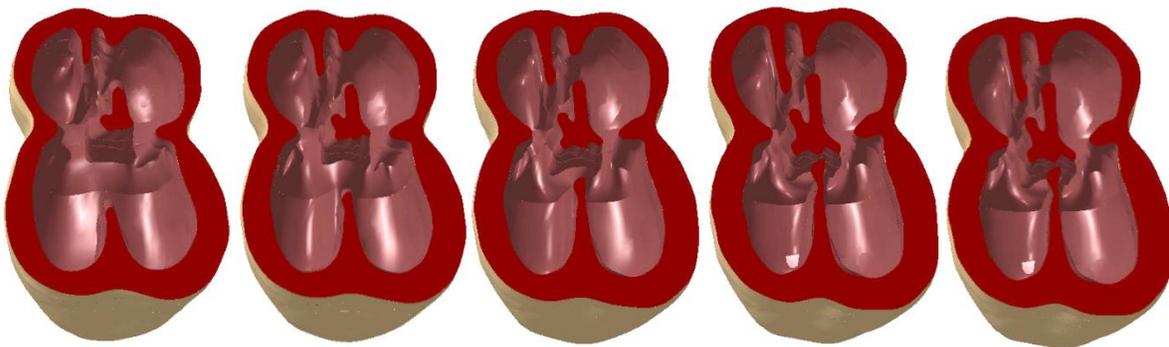


Fig. 4.36 Captura con la función programada de la geometría intermedia del modelo 2 al modelo 3 resultante con incrementos de 25% (razón de cambio)

4.5 Otras herramientas

Con el fin de facilitar el trabajo de creación se generaron otros programas que suministraron ayuda, para el desarrollo de este trabajo.

Captura de pantalla

El programa de CATIA tiene por defecto un “background” azul degradado, el cual es útil para dar una sensación visual de profundidad, sin embargo el hacer un ScreenShot desde Windows captura todo lo que muestra el monitor, con el fin de tener una buena imagen, sin necesidad de hacer un render del modelo, este programa quita el degradado y solo hace una captura dentro de CATIA mostrando solo la parte visible de la pieza o modelo, para guardarlo como un documento *.jpg.

El proceso se describe en el diagrama de flujo como:

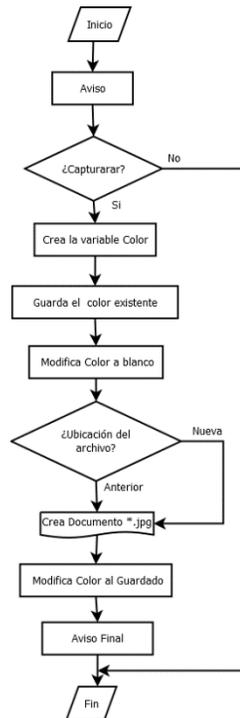


Fig. 4.37 Diagrama, captura de pantalla

- Inicio. Se inicia el programa, llamándolo con el icono asignado (“Creación de puntos Intermedios”).



Fig. 4.38 Icono para llamar la función Captura de pantalla

- Aviso. Muestra el inicio del programa
- Capturar. Pregunta si se desea hacer la captura de pantalla. En caso de que no sea así, cancela la captura

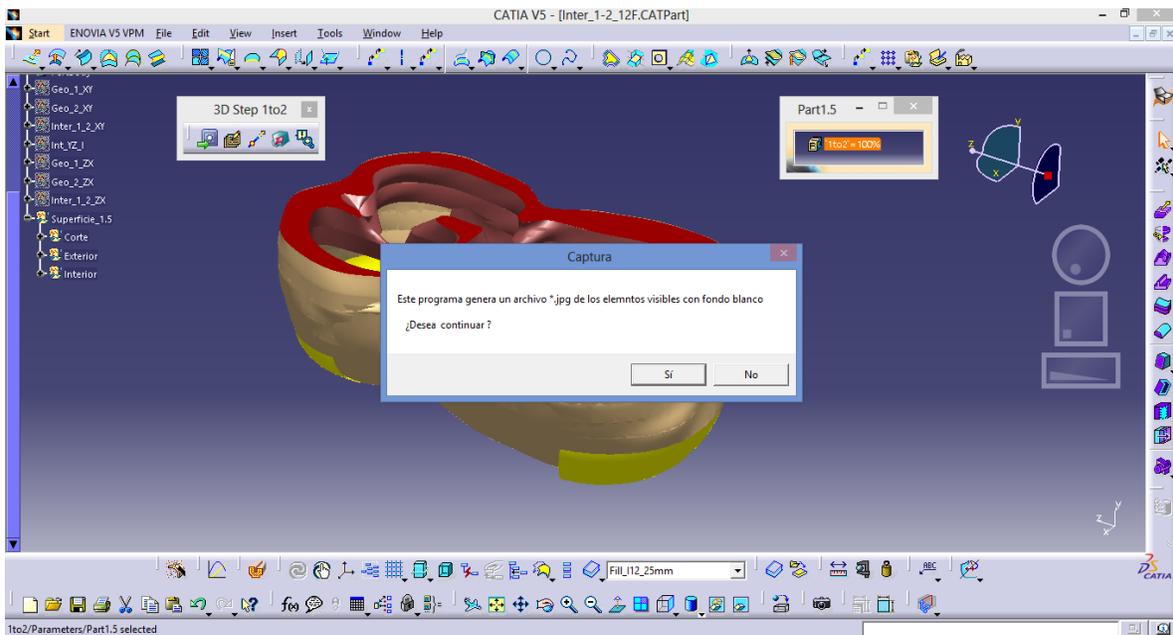
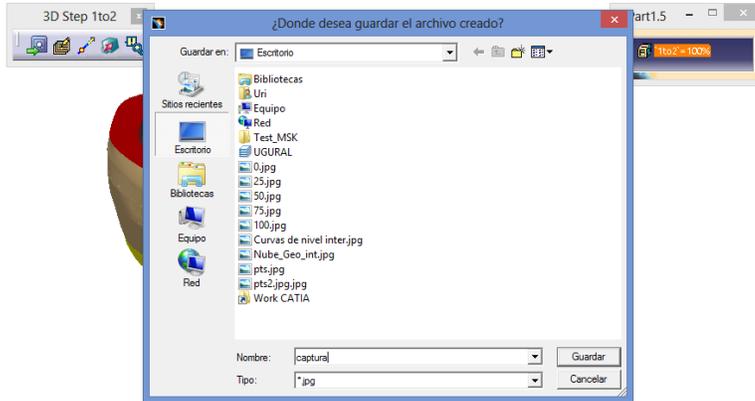


Fig. 4.39 Captura de pantalla del programa en ejecución mostrando mensaje

- Crea variable color. Para almacenar el color actual en formato (R,G,B)
- Guarda el color existente. Escribe en la variable actual para poder restaurar la configuración por defecto.
- Modifica el color. Cambia la variable al color blanco
- Ubicación del archivo. Pregunta donde se va a guardar el nuevo documento con la captura.



. Fig. 4.40 Captura de pantalla donde se muestra el fondo blanco y el menú de guardado

- Crea documento .jpg. Guarda el nuevo archivo de imagen en extensión *jpg*
- Modifica el color. Restaura a el color de defecto
- Aviso final. Mensaje de que el programa ha terminado.

Renombrado de puntos

Este programa renombra todos los puntos dentro de un “*Geometrical Set*” facilitando su identificación.

El proceso se describe en el diagrama de flujo como:

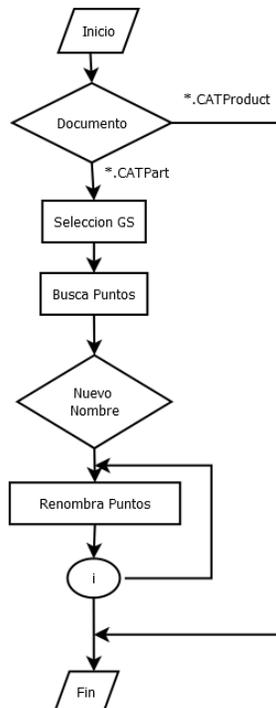


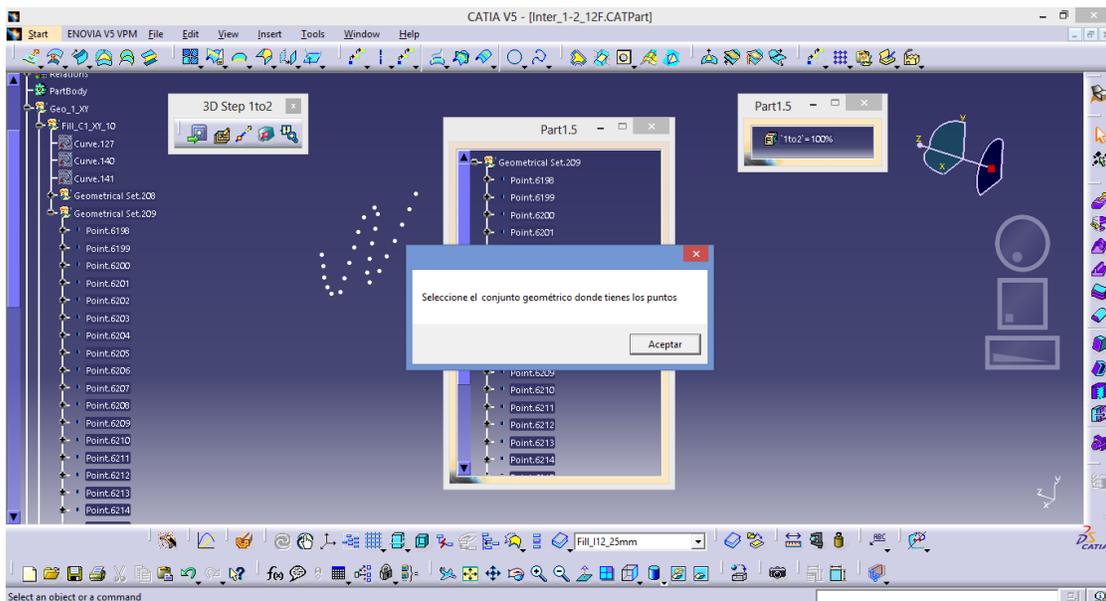
Fig. 4.41 Diagrama de flujo de Renombrado de puntos

- Inicio. Se inicia el programa, llamándolo con el icono asignado (“Renombrado de puntos”).



Fig.4.42. Icono Renombrado de puntos recalcado en la barra de herramientas

- Documento. Revisa el documento activo, en el caso de trabajar con un “product” muestra un mensaje y finaliza el programa. En caso de trabajar con un archivo “part” continúa con el siguiente paso.
- Selección GS. Desactiva lo seleccionado con anterioridad y pide la selección únicamente de un “Geometrical Set” que contenga los puntos.



4.43 Captura Para la selección del grupo de puntos en un Geometrical Set

- Busca Puntos. Del GS seleccionado, busca únicamente los puntos y los selecciona, esto debido a que en el GS puede haber líneas y otros elementos.
- Nuevo nombre. Muestra una caja de texto en el cual se debe ingresar el nuevo nombre del grupo de puntos, en caso de que este esté en blanco, reescribe por defecto al nombre Point.# guardándolo como una cadena de texto

- Renombra puntos. Renombra el punto i seleccionado con la cadena guardada.

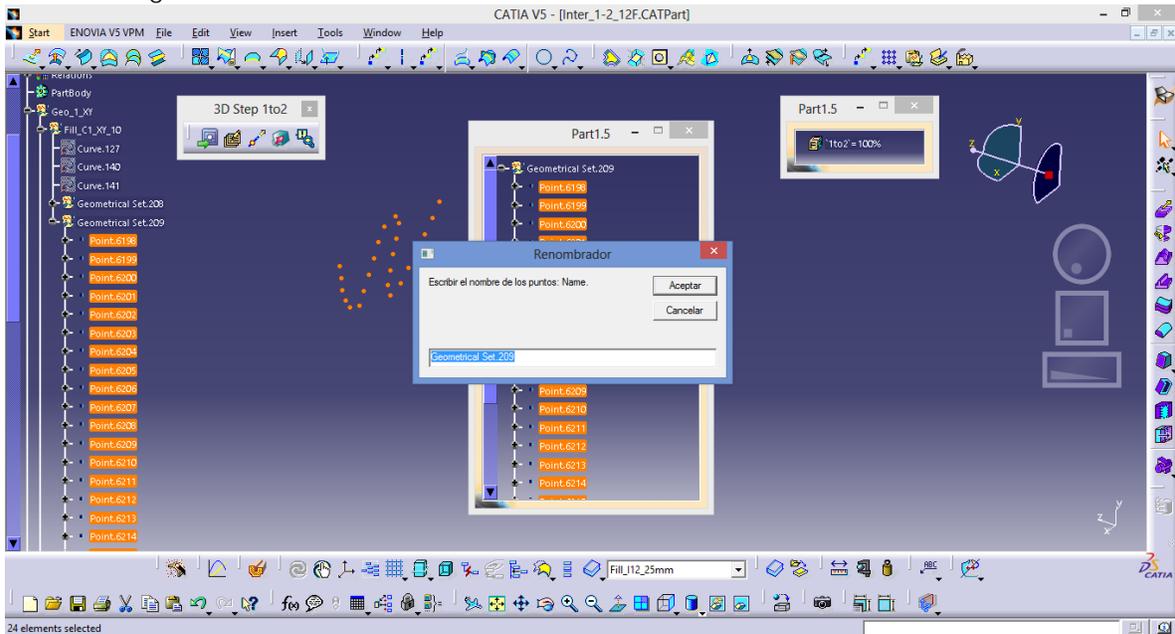


Fig. 4.44 Captura. Caja de texto para introducir nuevo nombre

- i. de la selección se extrae la cantidad de puntos para ejecutar un ciclo de 1 a la cantidad contada de puntos seleccionados.

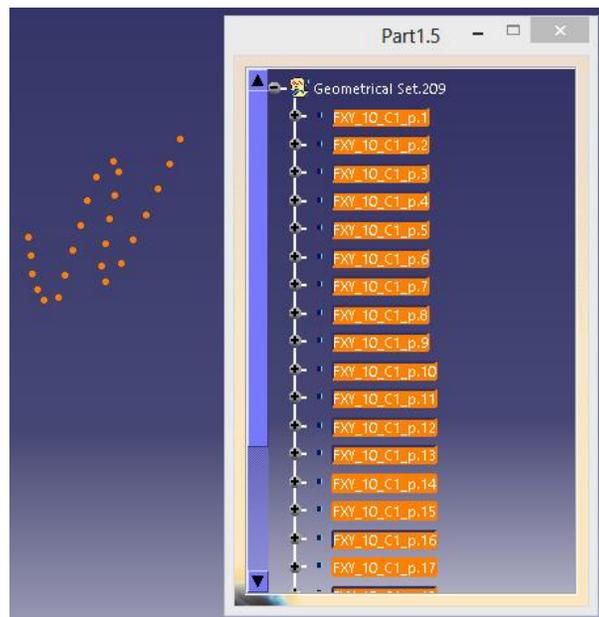


Fig. 4.45 Captura. Nuevos nombres de puntos

Búsqueda por nombre

En un archivo abierto con gran cantidad de elementos, puntos, líneas, curvas, etc. Es difícil encontrar un objeto en el árbol de elementos, el cual se tiene como referencia y se conoce su nombre, este programa muestra una caja de texto donde se introduce el elemento que se desea buscar y si está oculto lo muestra, también expande el árbol de elementos de CATIA centrando el resultado.

El proceso se describe en el diagrama de flujo como:

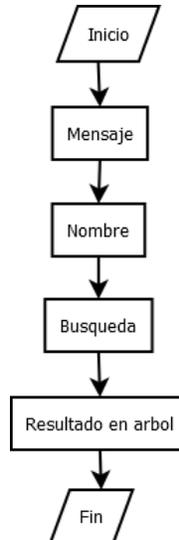


Fig. 4.46 Diagrama, búsqueda por nombre

- Inicio. Se inicia el programa, llamándolo con el icono asignado (“Creación de puntos Intermedios”).



Fig.4.47. Creación de puntos Intermedios. Icono recalcado en la barra de herramientas

- Mensaje. Muestra y pide la selección de un elemento para buscarlo por su nombre, mediante una caja de texto.

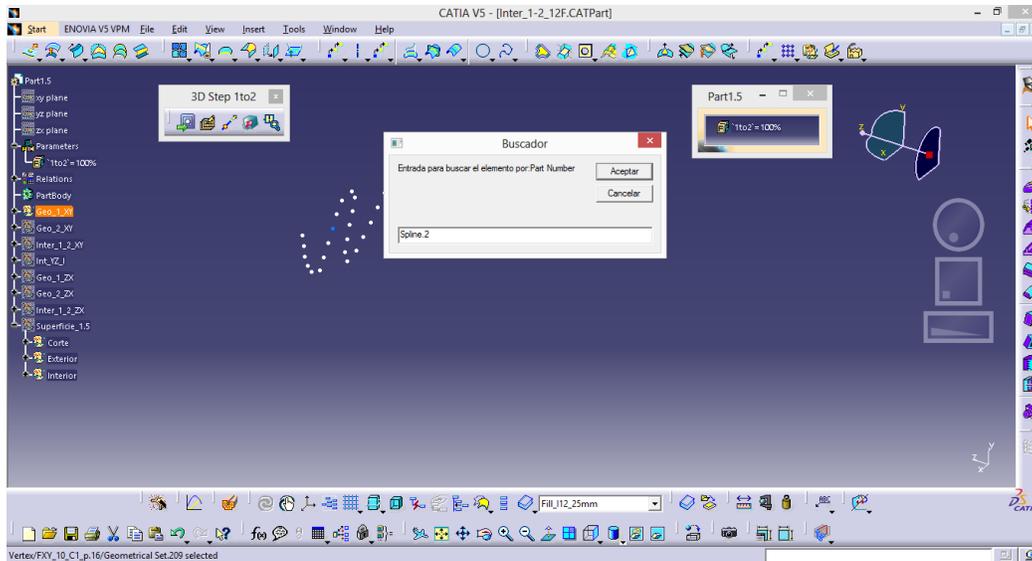


Fig. 4.48 Captura. Caja de texto para buscar el elemento dentro de la pieza en el árbol

- Nombre. Guarda el nombre como una cadena de caracteres para su búsqueda.
- Búsqueda. Hace una comparación de caracteres de los elementos existentes.
- Resultado en árbol. Expande el árbol con el elemento buscado, y muestra un mensaje de la cantidad de elementos encontrados, en caso de no encontrar algún elemento, muestra otro mensaje indicándolo.

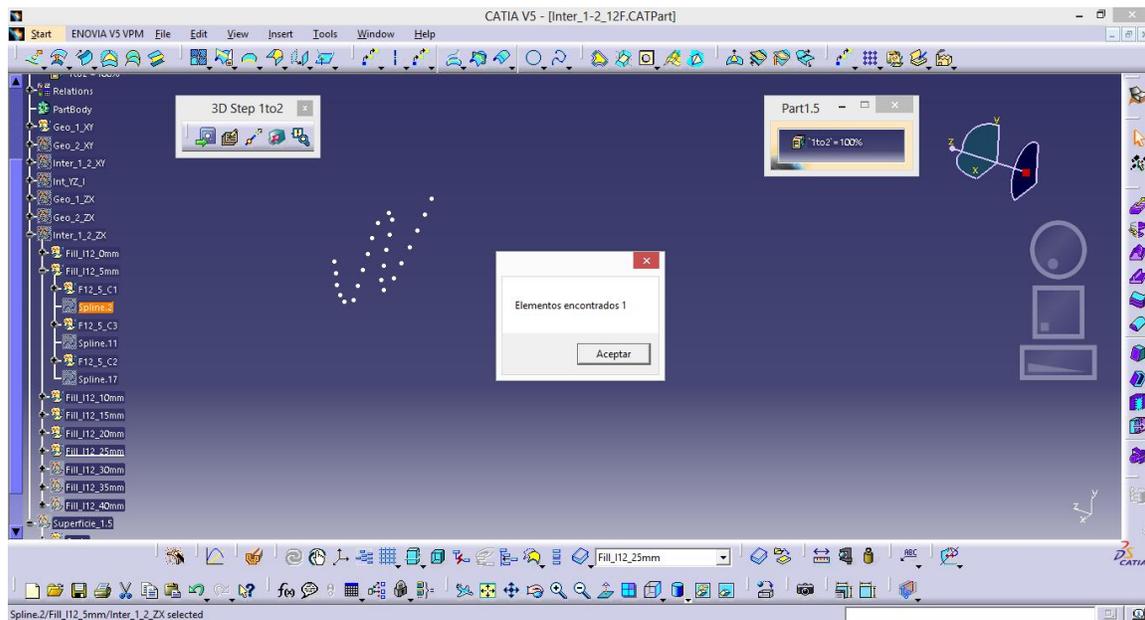


Fig. 4.49 Captura. Resultado de la búsqueda

CAPÍTULO

5

5. CONCLUSIONES

5.1 Conclusiones generales

La metodología propuesta fue funcional y automatizable debido a que el modo de empleo como herramienta resultó factible para la creación de geometrías irregulares con geometrías de referencia, validada por el caso de estudio, lo anterior dentro del entorno CATIA.

El principal objeto de estudio y problemática de este trabajo fue la transformación gráfica en el desarrollo del corazón embrionario, el cual se situó en la creación de una serie de modelos virtuales basados en CAD, llevado a cabo en el contexto de visualización gráfica. Para lograr la transformación gráfica entre los modelos se recurrió a la técnica de parametrización descrita en la metodología. Se construyeron los modelos que se enfocaron en mostrar el cambio de las paredes internas del corazón embrionario, el cual ocurre en pocas semanas del desarrollo del embrión, con el conocimiento de los académicos en medicina estos cambios podrán ser ubicados en el tiempo de gestación. (Aburto, 2014)

Debido a que el órgano en formación (el corazón) no se puede digitalizar con una tomografía (Mondaca A., 2006), no es posible realizar un estudio comparativo de desviación de las superficies entre órgano y modelo, sin embargo, cuando se pueda disponer de algún órgano se podría continuar con trabajos derivados.

Al desarrollar la parametrización del proceso de tabicación del corazón humano se programaron herramientas que redujeron el tiempo de creación y modificación de geometrías específicas al centrar los cambios en una variable, que es el porcentaje de cambio. De forma que al modificar el porcentaje de cambio se alteran todas las superficies del modelo y al modificar esta variable se aprecia la transformación gráfica.

5.2 Propuesta de Mejora

Si bien la programación de tareas y procesos, en cualquier software, reduce los tiempos, es necesario recordar que la programación implantada en éste trabajo es únicamente para tareas y procesos similares, teniendo la necesidad de programar otras funciones para nuevas tareas.

La programación realizada para este trabajo principalmente se orientó a la creación de modelos de superficies en el espacio basadas en alambre, en consecuencia una mejora sería el enfoque a la creación de sólidos, para cualquier geometría con la adaptación a un algoritmo que trabaje con las nubes de puntos y el proceso de parametrización lo realice de forma automática a pesar de la pérdida de flexibilidad en la toma de decisiones.

Referencias

- Aburto, M. (2014). Platicas sobre el desarrollo del sistema cardico embrionario humano. México: Jefa del departamento de embriología de la Facultad de Medicina de la UNAM.
- Arce Cedeño, A., & Hernández Sosa, C. H. (Agosto de 2014). Curso: Introduccion a embriología. Ciudad Universitaria: Facultad de Medicina, UNAM.
- Au, O. K., Tai, C. L., Chu, H. K., Cohen-Or, D., & Lee, T. Y. (2008). Skeleton extraction by mesh contraction. *ACM Transactions on Graphics (TOG)*, (Vol. 27, No. 3, p. 44). ACM.
- Ayala, A., & et.al. (2015). Generación de modelos 3D a partir de cortes axiales por tomografía. *SOMIM*.
- Baojian, D., Xiang, W., Yuling, F., Dongping, Y., & Gang, X. (2014). Intelligent Assembly Technology Based on Standard Parts feature of CATIA.
- Beier, T., & Neely, S. (1992). Feature-based image metamorphosis. *ACM SIGGRAPH Computer Graphics*, Vol. 26, No. 2, pp. 35-42). ACM.
- Carrillo Esper, R., & Zárate Vega, V. (2011). Comunicación interauricular tipo ostium secundumcon tratamiento percutáneo mediante amplatzer. *Med Int Mex*, 27(5):485-491.
- Chen, S. Y., & Guan, Q. (2011). Parametric shape representation by a deformable NURBS model for cardiac functional measurements. . *Biomedical Engineering, IEEE Transactions* , 58(3), 480-487.
- Constitución Política de los Estados Unidos Mexicanos. (Febrero de 1983). *Ley General de Salud*. Recuperado el 5 de Diciembre de 2014, de REGLAMENTO de la Ley General de Salud en materia de control sanitario de la disposición de órganos, tejidos y cadáveres de seres humanos.: <http://www.salud.gob.mx/unidades/cdi/nom/compi/rlgsmcsdotcsh.html>
- Corkum, M. (1994). *Three Dimensional Morphing Using an Adaptive Oriented Particle System*. Providence, R.I. EE.UU.: Brown University.
- Dassault Systèmes. (2009). CATIA V5 Automation. Student Guide.
- Goldstein, E., & Gotsman, C. (1995). Polygon Morphing Using a Multiresolution Representation.
- Hearn, D. (2006). Graficas por computadora. En D. Hearn , & M. Baker, *Graficas por computadora* (págs. 204-211).
- Hussein, H. (2014). Computeraided Blanking Die Desing using CATIA. *Procedia CIRP*, 18, 96-101.

- Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. . *International journal of computer vision*, 1(4), 321-331.
- Kirkire, M. S., & Suryawanshi, V. (2009). Knowledge based Parametric Approach For Desing Of Gearbox. *IJAEA*, 28-34.
- Lamousin , H., & Waggenspack. , W. (1994). NURBS-based free-from deformations. *IEEE Computer Graphics and Applications*, 14(6):59-65.
- Ledermann, C., Hanske, C., Wenzel, J., Ermanni, P., & Kelm, R. (2008). Associative parametric CAEmethods in teh aircraft pre-design. *journal of Aerospace Science and Technology*, 9(7), 641-651.
- Matsumura, G. (1996). Embriología Representaciones Graficas.
- Molina Martinez, J. M., & Jiménez Buendía, M. (2010). *Programación Gráfica para Ingenieros*. Marcombo.
- Mondaca A., R. (2006). POR QUE REDUCIR LAS DOSIS DE RADIACION EN PEDIATRIA. *Revista chilena de radiología*. doi:10.4067/S0717-93082006000100008.
- Mukherjee, U. (2014). Self-overlapping curves: Analysis and applications. *Computer-Aided Design*, 46, 227-232.
- Piegl, L., & Tiller, W. (1997). *The NURBS Book. Monographs in visual communications*. Springer.
- Rahul, K., Sanjay, P., Dattatray, J., & Narendra, D. (2014). Customization of Catia V5 for Crating Different types of Holes on Disc Wheel.
- Río, M. G., Martínez, C. J., Ortuño, P. F., & Ortiz. (2007). Task automation for modelling solids with Catia V5. *Aircraft Engineering and Aerospace Technology*, 53 – 59.
- Ross, E. (2013). *VB Scripting for CATIA V5: How to Program CATIA Macros*. CreateSpace Independent Publishing Platform.
- Sederberg, T. W., Zheng, J., Bakenov, A., & Nasri, A. (2003). T-splines and t-nurccs. *ACM Transactions on Graphics*,, 22(3):477-484.
- Spiteri, L., Daynes, S., & Watkins, S. (2015). Design of a Morphing Bi-Stable Composite Air Intake. *SAE Technical Paper*.
- Valdés Valdés, A., & et. al. (2010). *Embriologia Humana*. La Habana: Ciencias Médicas.
- Vince, J. (2008). *Geometric Algebra for Computer Graphics*. Springer.
- Wolberg, G. (1990). Digital image warping. *IEEE computer society press*.
- Wolberg, G. (1998). Image morphing: a survey. *The visual computer*, 14(8), 360-372.

- Yang, W., Wang, X., & Wang, G. (2014). Part-to-part morphing for planar curves. *The Visual Computer*, 30(6-8), 919-928.
- Zhang, R. (2010). *3D mesh metamorphosis from spherical parameterization for conceptual design*. Iowa State University.
- Ziethen, D. (2013). *CATIA V5: Macro Programming with Visual Basic Script*. McGraw-Hill.

Anexo A. Detalles de la tabicación del corazón humano embrionario.

Tabicamiento cardíaco

Este implica un complicado proceso de remodelamiento y división de las cavidades cardíacas primitivas, lo que da origen al corazón definitivo de cuatro cavidades (dos aurículas o atrios y dos ventrículos) que funcionan durante toda la vida fetal y luego, con pequeñas modificaciones, durante la vida posnatal.

Debido a su complejidad, se analizan por separado el tabicamiento en las distintas porciones del corazón pero precisando que todos estos procesos ocurren de manera más o menos sincrónica y en un periodo que abarca la quinta y sexta semana del desarrollo, cuando el embrión crece unos 10 mm en longitud, desde 5 mm, hasta 16 mm.

Mecanismos de tabicamiento

En el tabicamiento cardíaco se observan dos mecanismos básicos diferentes, estos son:

1. Dos estructuras de tejido conectivo en el centro de la luz tubular (como por ejemplo, canal auriculoventricular y troncoconal), que crecen activamente y se fusionan dividiendo la luz de forma total en dos canales (Fig. 1.6. A).
2. Formación de pliegues. En este caso, si una franja estrecha de tejido en la pared de la aurícula o del ventrículo deja de crecer, mientras las áreas a cada lado de esta se expanden muy rápido, una cresta estrecha se forma entre las dos porciones en expansión (Fig. 1.6. C y D)

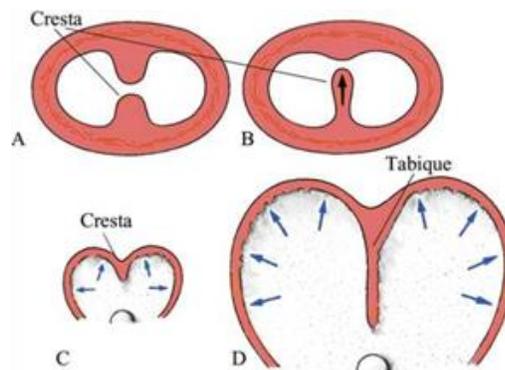


Fig. 1.6. Formación del tabique por la presencia de crestas de crecimiento activo. A. Dos crestas opuestas. B. Una sola. C y D. Formación de la cresta entre dos porciones de la pared cardíaca en expansión. (Valdés Valdés & et. al., 2010)

Tabicamiento auricular

El primer signo de tabicamiento auricular es, cuando en la cuarta semana aparece una cresta o tabique desde el techo de la aurícula que crece hacia su luz. Esta cresta es el septum primum (Fig. 1.7), el cual se extiende hacia las almohadillas endocárdicas situadas en el canal auriculoventricular.

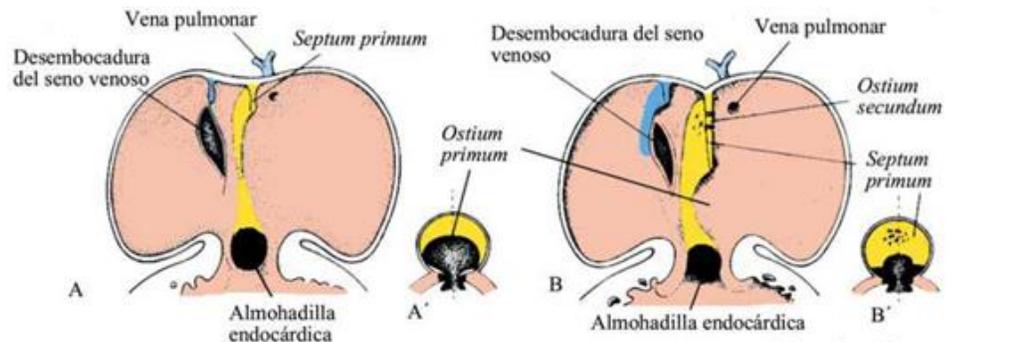


Fig. 1.7. Diversas etapas del desarrollo del tabique interauricular (Valdés Valdés & et. al., 2010)

El orificio situado, entre el borde inferior del septum primum y las almohadillas endocárdicas, recibe el nombre de ostium primum. Al aumentar la cavidad auricular derecha por la incorporación de la prolongación sinusal, aparece un nuevo tabique de forma semilunar, el septum secundum, que por su forma nunca llega a la separación completa de las aurículas. Cuando la parte superior del septum primum desaparece de forma gradual, la parte remanente se convierte en la válvula del agujero oval, a través del cual la sangre fluye desde la aurícula derecha a la izquierda.

Mientras la aurícula derecha primitiva se agranda por la incorporación del cuerno derecho del seno venoso, la aurícula izquierda primitiva también se expande. Inicialmente, una vena pulmonar embrionaria se desarrolla como un brote posterior de la pared auricular izquierda, a la izquierda del *septum primum*. Durante el desarrollo ulterior, la vena pulmonar y sus ramas se incorporan en la aurícula izquierda, las cuales forman la mayor parte de la pared lisa de la aurícula del adulto. Aunque al principio, una sola vena pulmonar entra a la aurícula izquierda, al final lo hacen cuatro venas pulmonares que, de manera progresiva, son incorporadas en la pared auricular en expansión. Al completarse el desarrollo del corazón, la aurícula izquierda embrionaria original está representada por la orejuela auricular trabeculada izquierda, mientras que su pared lisa se origina de las venas pulmonares. En el lado derecho, la aurícula derecha embrionaria original se convierte en la orejuela auricular trabeculada derecha que contiene los músculos

ANEXO

pectíneos, y la pared lisa del *sinus venarum* se origina del cuerno derecho del seno venoso.

Tabicamiento del canal auriculoventricular

Al final de la cuarta semana, el canal auriculoventricular desemboca sólo en el ventrículo izquierdo primitivo, separado del bulbo cardiaco por el reborde bulboventricular (Fig. 1.8).

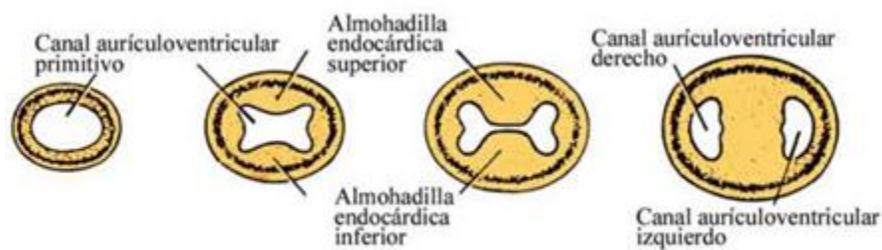


Fig. 1.8. Formación del tabique en el canal auriculoventricular (Valdés Valdés & et. al., 2010)

Válvulas auriculoventriculares

Se originan como proyecciones hacia el interior de los orificios auriculoventriculares derecho e izquierdo, por el aporte, tanto de las almohadillas endocárdicas fusionadas, como del tejido mesenquimatoso subyacente del endocardio (Fig. 1.9).

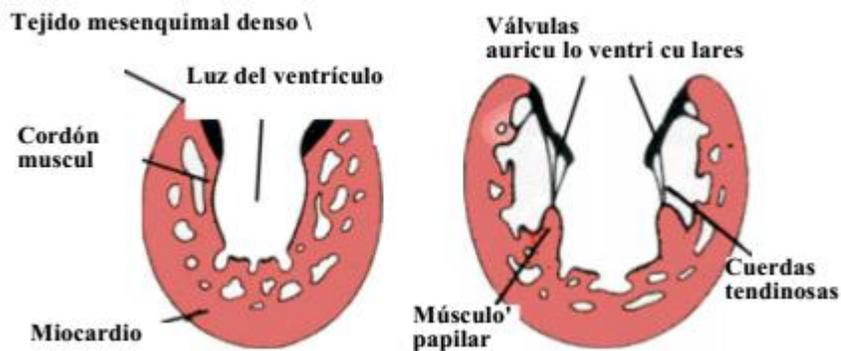


Fig 1.9. Formación de las válvulas auriculoventriculares y de las cuerdas tendinosas. (Valdés Valdés & et. al., 2010)

Por esto, dos valvas en forma de hojas conforman la válvula bicúspide o mitral (por su semejanza con la mitra) en el orificio auriculoventricular izquierdo, y tres valvas en el orificio auriculoventricular derecho forman la válvula tricúspide.

Tabicamiento ventricular

La porción proximal dilatada del bulbo y el ventrículo primitivo participan en el origen de los ventrículos definitivos (Fig. 1.10. A). El crecimiento ventricular se produce por un crecimiento continuo del miocardio en su parte externa y por una formación de divertículos y trabéculas en su parte interna (Fig. 1.10. B y C).

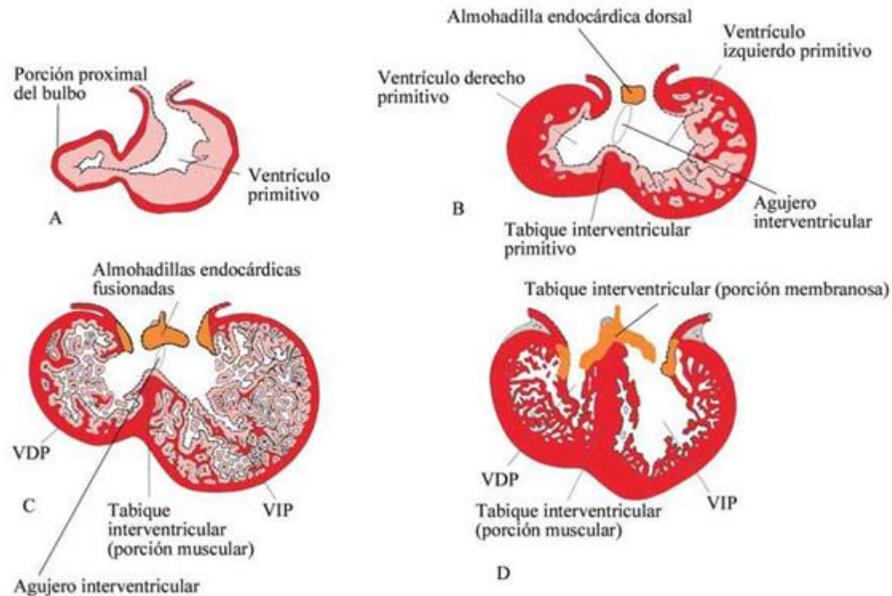


Fig. 1.10. Tabicamiento interventricular (Valdés Valdés & et. al., 2010)

El tabique interventricular muscular comienza a desarrollarse por plegamiento del piso ventricular, que crece en dirección cefálica hacia las almohadillas endocárdicas; sin embargo, no llega a unirse y deja un orificio en la porción superior, denominado agujero interventricular. El orificio interventricular se cierra más tarde por la llamada porción membranosa del tabique interventricular.

Anexo B. Otras curvas y superficies.

Curvas de Bézier

En general, es posible ajustar una curva de Bézier para cualquier número de puntos de control. El número de puntos de control que se debe aproximar y su posición relativa determinan el grado del polinomio de Bézier.

La idea de definir geoméricamente las formas no es demasiado compleja: un punto del plano puede definirse por coordenadas. Por ejemplo, un punto A tiene unas coordenadas (x_1, y_1) y a un punto B le corresponde (x_2, y_2) . Para trazar una recta entre ambos basta con conocer su posición.



Fig. B.1 Curva Bézier con 7 puntos de control

Si en lugar de unir dos puntos con una recta se unen con una curva, surgen los elementos esenciales de una curva Bézier: los puntos se denominan puntos de anclaje o nodos. La forma de la curva se define por unos puntos invisibles en el dibujo, denominados puntos de control, manejadores o manecillas.

Curvas Cubicas (Grado 3)

La curva comienza en el punto P_0 y se dirige hacia P_1 y llega a P_3 viniendo de la dirección del punto P_2 . (Figura B.2) Usualmente, no pasará ni por P_1 ni por P_2 . Estos puntos sólo están ahí para proporcionar información direccional.

- Cuatro puntos de control (P_0, P_1, P_2 y P_3).
- Son las más utilizadas.
- Se aplica el mismo principio de inducción, partiendo inicialmente de tres curvas de Bézier lineales.
- Así se pueden generar curvas de cualquier grado.

ANEXO

La distancia entre P_0 y P_1 determina "qué longitud" tiene la curva cuando se mueve hacia la dirección de P_2 antes de dirigirse hacia P_3 .

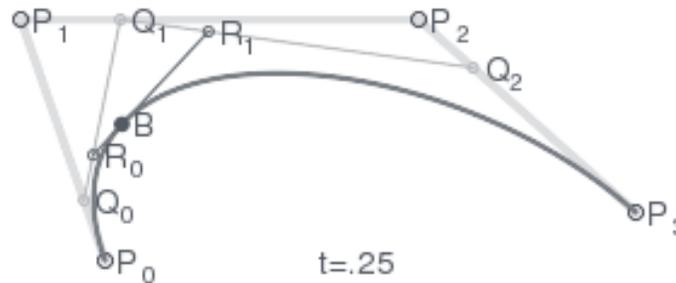


Figura B.2 Interpolación entre P_0 , P_1 , P_2 y P_3 con el tiempo igual a 0.25

La forma paramétrica de la curva es:

$$B(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, t \in [0,1]$$

Ecuación (i)

Propiedades de Bézier

- El grado de la base de polinomios es uno menos que la cantidad de puntos de control.
- El primer y último punto de la curva coincide con el primer y último punto del grafo de control.
- El vector tangente en los extremos de la curva tiene la misma dirección que el primer y último segmento del grafo de control respectivamente.
- Tienen control global.

Desventajas de las Curvas de Bézier

- Para grafos de control complejos (formados por muchos puntos). El grado de la base es elevado.
- Tienden a suavizar demasiado la geometría del grafo de control.
- Se tornan insensibles a pequeños cambios locales. El desplazamiento de un sólo punto de control casi no produce efecto en la curva.
- El control global provoca que el desplazamiento de un sólo punto de control modifique a toda la curva.

ANEXO

Superficies planas.

Una vez que el usuario haya definido cada superficie del polígono, el paquete de gráficas organiza los datos de entrada en las tablas que utilizarán en el procesamiento y despliegue de las superficies. Las tablas de datos contienen las propiedades geométricas y de atributos del objeto, organizadas para facilitar el procesamiento. Las tablas de datos geométricos contienen coordenadas y parámetros de fronteras para identificar la orientación en el espacio de las superficies poligonales. La información de atributo del objeto incluye designaciones de cualquier modelo de color y sombreado que se aplicará a las superficies.

Los datos geométricos pueden organizarse de varias maneras. Un método adecuado para almacenar información de coordenadas consiste en crear tres listas: una tabla de vértices, una de aristas y una de polígonos. Los valores coordinados de cada vértice del objeto se almacenan en la tabla de vértices. La tabla de aristas enlista los vértices extremos que definen a cada arista. Cada polígono se define en la tabla de polígonos como una lista de aristas componentes. Este esquema se ilustra en la figura 2.11 para un objeto que consta de dos superficies poligonales. La tabla de polígonos contiene apuntadores que indican la tabla de aristas la cual, a su vez, contiene apuntadores que señalan los valores coordinados en la tabla de vértices. Cuando en una escena se va a representar más de un objeto, cada uno puede identificarse en una tabla de objetos por medio del conjunto de superficies poligonales en la tabla de polígonos que definen a ese objeto.

El listado de los datos geométricos en tres tablas, como se muestra en la figura 2.11, ofrece una referencia adecuada de los componentes (vértices, aristas y polígonos) de un objeto. Así mismo, el objeto puede desplegarse eficazmente mediante el uso de datos de la tabla de aristas para trazar las líneas componentes. Sin la tabla de aristas, el objeto se desplegaría utilizando datos de la tabla de polígonos y esto quiere decir que algunas líneas se trazarían dos veces. Si tampoco se dispusiera de la tabla de vértices, la tabla de polígonos tendría que enlistar coordenadas explícitas de cada uno de los vértices de cada polígono. Toda la información referente a vértices y aristas tendría que reconstruirse a partir de la tabla de polígonos y las posiciones coordinadas se duplicarían para los vértices situados en la frontera de dos o más polígonos.

ANEXO

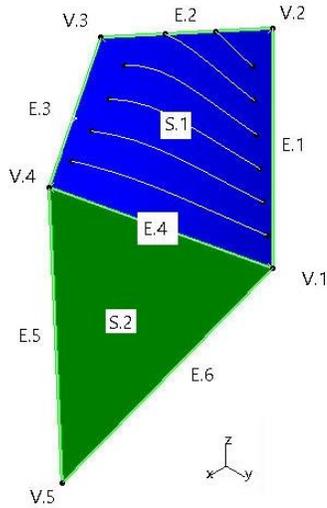


Tabla de Vértices	Tabla de Aristas	Tabla de Polígonos
$V_1: x_1, y_1, z_1$	$E_1: V_1, V_2$	$S_1: E_1, E_2, E_3, E_4$ $S_2: E_4, E_5, E_6$
$V_2: x_2, y_2, z_2$	$E_2: V_2, V_3$	
$V_3: x_3, y_3, z_3$	$E_3: V_3, V_4$	
$V_4: x_4, y_4, z_4$	$E_4: V_4, V_2$	
$V_5: x_5, y_5, z_5$	$E_5: V_4, V_5$	
$V_6: x_6, y_6, z_6$	$E_6: V_5, V_1$	

Fig. 2.11 Objeto tridimensional representado por dos superficies poligonales planas, con seis aristas cinco vértices.

Podría incorporarse información adicional en las tablas de datos de la figura 2.11 para lograr una extracción más rápida de la información. Por ejemplo, podría extenderse la tabla de aristas para incluir apuntes en la tabla de polígonos de modo que pudieran identificarse aristas comunes entre polígonos de mayor rapidez. Esto es particularmente útil cuando se aplican modelos de sombreado a las superficies, con los patrones de sombreado que varían ligeramente de un polígono al siguiente. También podría ampliarse la tabla de vértices haciendo referencia cruzada de vértices para las aristas correspondientes.

Ya que las tablas de datos geométricos pueden contener listados extensos de vértices y de aristas de objetos complejos, es importante que los datos se verifiquen en consistencia y completitud. Cuando se especifican definiciones de vértices, líneas y polígonos, es posible que pudieran cometerse ciertos errores de entrada que distorsionarían el despliegue del objeto. Cuanta más información se incluya en las tablas de datos, más fácil será verificarla. Algunas de las pruebas que podrían realizar un paquete de gráficas son:

- Que todos y cada uno de los vértices se enliste como un extremo de cuando menos dos líneas.
- Que toda línea sea parte cuando menos de un polígono.

ANEXO

- Que todo polígono sea cerrado.
- Que cada polígono tenga al menos una arista compartida.
- Si la tabla de aristas contiene apuntadores a polígonos, que toda arista referenciada por un apuntador de polígono tenga un apuntador recíproco hacia el polígono.

Superficies de Bézier

Pierre Bézier, ingeniero francés desarrolló este método de aproximación de spline para utilizarlo en el diseño de carrocerías de los automóviles Renault. Las spline de Bézier tienen varias propiedades que hacen que sean muy útiles y convenientes para el diseño de curvas y superficies; además de ser de fácil implementación. Por estos motivos, las spline de Bézier están disponibles en forma común en varios sistemas de CAD, en paquetes generales de gráficos y en paquetes seleccionados de dibujo y pintura.

Se pueden utilizar dos conjuntos de curvas de Bézier para representar superficies de objetos especificados por punto de control de entrada. La función vectorial paramétrica de la superficie de Bézier se forma como el producto cartesiano de las funciones de combinación de Bézier:

$$P(u, v) = \sum_{j=0}^m \sum_{k=0}^n p_{j,k} B_{j,m}(u) B_{k,n}(v) \quad \text{Ecuación (ii)}$$

Con $P_{j,k}$ que especifica las localidades de los $(m + 1)$ por $(n + 1)$ puntos de control.

A continuación se ilustran dos trazos de superficies de Bézier. Los puntos de control están conectados por líneas más delgadas y más gruesas que muestran curvas con constantes u y v .

Cada curva de u constante se gráfica variando u con respecto al intervalo 0 a 1, con u fija en uno de los valores en este intervalo unitario. Las curvas de u constante se trazan en forma análoga.

Las superficies de Bézier tienen las mismas propiedades que las curvas de Bézier y ofrecen un método adecuado para aplicaciones de diseño interactivas. En

ANEXO

la figura 2.14 las líneas punteadas conectan los puntos de control de las superficies de Bézier.

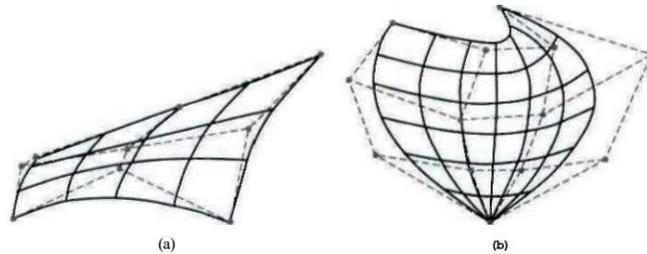


Fig. 2.14 Se ilustra una superficie formada con dos secciones de una superficie de Bézier. (a) Para $M=3$, $N=3$ y para (b) $M= 4$, $N= 4$

Como sucede con las curvas, una transición sencilla de una sección a la otra se asegura estableciendo continuidad de orden cero y de segundo orden en la “línea de frontera”.

La continuidad de primer orden se obtiene eligiendo puntos de control a lo largo de una “línea recta” que atraviesa la frontera y manteniendo una razón constante de segmentos de la “línea colineal”, para cada una de estas líneas que cruzan la “línea de frontera” de la superficie.

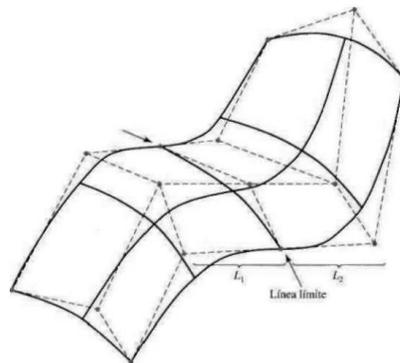


Fig. 2.15 En esta figura las superficies de Bézier son construidas con dos secciones de Bézier, unidas en la línea de frontera.

Las líneas punteadas conectan puntos de control específicos.

ANEXO

Se establece continuidad de primer orden haciendo la razón de la longitud “ a ” a la longitud “ b ” constante para cada “línea colineal” de puntos de control a través de la “línea de frontera”.

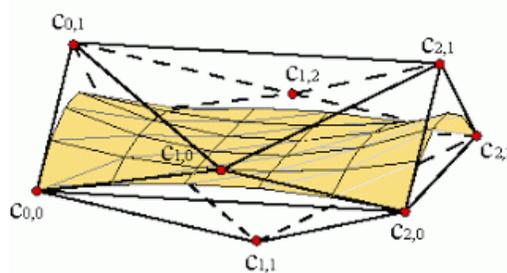


Fig. 2.16 Superficie de Bézier (amarilla) con 9 puntos de control ($C_{i,j}$)

Propiedades de las superficies de Bézier

Las superficies de Bézier son una generalización natural de las curvas polinómicas, muchas de las propiedades de estas se mantienen, dichas propiedades son:

- La superficie sigue estando comprendida, si los pesos son todos positivos, en la envolvente convexa de la malla de control.
- En el caso de superficies a trozos, cada tramo de superficie está además comprendida en la envolvente convexa de su malla de control.
- Sin embargo, no hay una generalización sencilla de la propiedad de disminución de la variación. Es fácil, por ejemplo, comprobar que una recta puede cortar a la superficie en más puntos que los que corta al poliedro descrito por la malla de control.

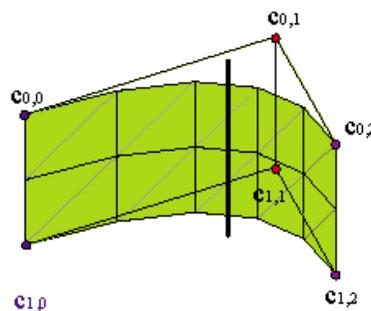


Fig. 2.17 Superficie de Bézier (verde) con 6 puntos de control ($C_{i,j}$)

Al definir la parametrización en un recinto rectangular, $[a,b] \times [c,d]$ distinto del que sea empleado $[0,1] \times [0,1]$, no precisamos más que realizar una re-

ANEXO

parametrización afín de los parámetros u, v , que no afecta ni a la malla de control ni a la gráfica de la superficies.

$$(u') = \frac{u'-a}{b-a}, u' \in [a, b], v(v') = \frac{v'-c}{d-c}, v' \in [c, d] \quad \text{Ecuación (iii)}$$

De manera que la nueva parametrización sea:

$$c'(c', v') = c(u(u), v(v')) := c\left(\frac{u'-a}{b-a}, \frac{v'-c}{d-c}\right), u' \in [a, b], v' \in [c, d] \quad \text{Ecuación (iv)}$$

Del mismo modo que las curvas de Bézier sólo pasan por los vértices extremos, correspondientes a $t = 0, t = 1$, las superficies sólo pasan por cuatro vértices, las esquinas de la malla de control

$$c_{0,0} = c(0,0) \quad ; \quad c_{m,0} = c(1,0) \quad ; \quad c_{0,n} = c(0,1) \quad ; \quad c_{m,n} = c(1,1) \quad \text{Ecuación (v)}$$

Aunque la superficie pasa por las cuatro curvas de Bézier que describen los bordes de la malla de control.

$$c(0, v) = S_{j=0} \cdot n \cdot c_{0,j} \cdot Bn_j(v) \quad ; \quad v \in [0,1] \quad \text{Ecuación (vi)}$$

Por ejemplo, para $u = 0$, la curva correspondiente, $c(0, v)$, tiene por polígono de control $\{c_{0,0}, \dots, c_{0,n}\}$, la primera fila de la malla de control de la superficie. Del mismo modo, la última fila, $\{c_{m,0}, \dots, c_{m,n}\}$, es el polígono de control de la curva $c(1, v)$. Y, para $v = 0$, el polígono de control de la curva $c(u, 0)$ es la primera columna de la malla de control, $\{c_{0,0}, \dots, c_{m,0}\}$. De manera análoga, el polígono de control de $c(u, 1)$ es la última columna, $\{c_{0,n}, \dots, c_{m,n}\}$

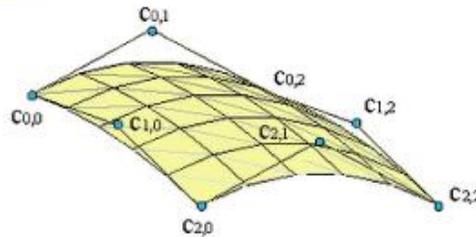


Fig. 2.18 Superficie de Bézier (amarilla) con 8 puntos de control ($C_{i,j}$)

Por tanto, el borde de la malla se corresponde con el borde de la superficie.

$$c(u, 0) = S_{i=0} \cdot m S_{=0} \cdot n c_{i,j} \cdot Bm_i(u) \cdot Bn_j(0) = S_{i=0} \cdot m c_{i,0} \cdot Bm_i(u) \quad ; \quad u \in [0,1]$$

Esta propiedad se mantiene en las superficies racionales.

NURBS (Non-Uniform Rational B-Spline)

Las superficies paramétricas *No Uniformes Racionales B-splines* o *NURBS* son uno de los modelos matemáticos más significativos para representar geometrías complejas mayormente utilizado en la gráfica por computadoras y en el Diseño Asistido tales como maquinas industriales, aviones, piezas, etc. Unos de los motivos de su interés son:

- Aportan unas bases matemáticas estables para constituir formas canónicas y arbitrarias.
- Los algoritmos matemáticos de las superficies NURBS son rápidos y numéricamente estables.
- Las superficies mantienen la propiedad de invariancia ante diversas operaciones geométricas, como la translación o la rotación.
- Sus propiedades matemáticas y algorítmicas, así como su sencillez las hacen altamente eficaces para cualquier tipo de estudio.
- Ofrecen una estructura matemática común para figuras analíticas estándar (por ejemplo cónicas) y figuras de forma libre.
- Proporcionan flexibilidad para diseñar una gran variedad de figuras.
- Reducen el consumo de memoria al almacenar figuras (en comparación con métodos más sencillos).

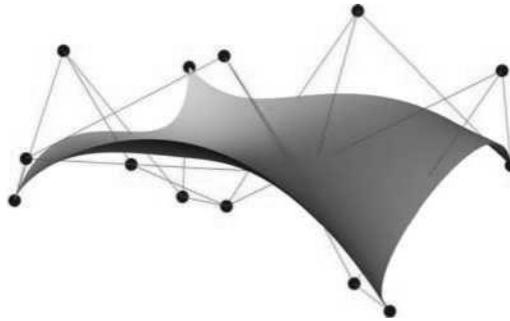


Figura 2.19: Una superficie NURBS, los puntos de control son los puntos, conectados por rectas

Aunque las ventajas que representa trabajar con NURBS son muchas, este tipo de elementos tiene una serie de inconvenientes:

- La parametrización depende de los pesos asignados y la forma depende de una aplicación adecuada de los pesos.
- Para determinar los parámetros correspondientes a un punto de la superficie, algunos algoritmos son inestables numéricamente y se requieren incrementar los cálculos computacionales., (Piegl & Tiller, 1997)

ANEXO

- Algunas representaciones elementales (círculos, cuadrados, etc.) requieren una mayor cantidad de datos utilizando NURBS que utilizando otras formas de representación.

Este modelo ha sido ampliamente utilizado en diversas áreas, por ejemplo, para la obtención de formas a partir de imágenes (Lamousin & Waggenspack, 1994), deformaciones libres de forma y en muchas otras más, incluso en el ámbito de las imágenes biomédicas. Por ejemplo Chen & Guan (2011), proponen un método de representación paramétrica y de medidas funcionales de formas cardíacas 3D basadas en NURBS deformables.

T-SPLINE

Cuando se unen dos superficies NURBS, surge el problema de la combinación de las mallas de control de ambas. Mediante NURBS, es necesario añadir filas y columnas de puntos de control en la malla. La nueva malla obtenida permite representar la forma de las dos superficies unidas, aunque se incrementan los cálculos en la computadora y con frecuencia requiere un esfuerzo extra por parte del usuario. Además, surge otro problema en el momento en el que se desea añadir un punto de control. En este caso se tiene que añadir a la malla de control una fila o una columna entera debido a la regularidad requerida en dicha malla.

El aspecto fundamental es que cada punto de control en una malla de control NURBS, exceptuando los puntos de los bordes de dicho mallado, tiene exactamente dos vecinos en cada dirección de la malla, tanto horizontal como verticalmente. Sin embargo, en una malla de control *T-mesh* es posible tener un punto de control con solamente uno adyacente horizontal o vertical, este punto es conocido como unión en T, *T-junction*.

El modelo de *T-Splines* (Sederberg, Zheng, Bakenov, & Nasri, 2003) permite constituir superficies *B-Spline* no uniforme con *T-junctions*. Con este modelo es posible insertar puntos en la malla de control sin que esto provoque el hecho de añadir toda una fila o columna. Además es posible combinar dos *T-Splines* sin ningún esfuerzo adicional.

La *T-Splines* son *Spline's* con una malla de control *T-mesh*, que impone algún tipo de organización a los puntos de control. La malla de control permite manipular más fácilmente a los puntos de control de lo que el planteamiento totalmente libre de las *PB-Splines* permite. Esta malla de control permite además tener un mecanismo de cálculo de los nudos de control s_i y t_i para cada función de base.

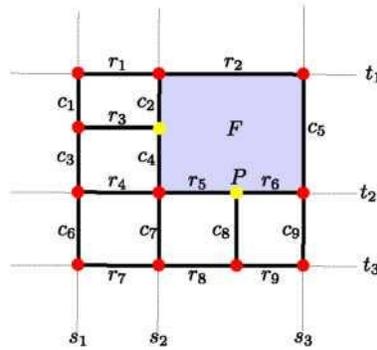


Figura 2.20: Una malla T-mesh. Las líneas punteadas representan los s-edges y los T-edges. Los puntos son los puntos de control, de los que los de color rojo representan los puntos de control regulares y los amarillos son los T-junctions.

Una malla T-mesh (Figura 2.20) es básicamente una malla rectangular que admite T-junctions.

Cada arista en un T-mesh está etiquetada en un intervalo de nudos, según las siguientes reglas:

- La suma de los intervalos de aristas opuestas en cualquier cara debe ser igual. De esta forma, para la cara F de la figura 2.20: $r_2 = r_5 + r_6$ y $C_2 + C_4 = C_5$.
- Si una T-junction en una arista de una cara puede ser conectada a una arista opuesta de la cara (dividiéndose de esta forma la cara en dos) sin que se fracione la primera regla, entonces dicha arista puede ser incluida en el T-mesh.

En definitiva, las superficies T-Spline tienen similares ventajas que las NURBS, pero requieren un modelo de representación más variable.

Anexo C. Automatización en CATIA

El software de Computer Aided Design (CAD) que se utilizó en este trabajo fue CATIA V5 de Dassault System, el cual es un entorno de gama alta para modelar geometrías, proporciona una variedad de herramientas y opciones para su uso.

Software CAD

El programa comercial CATIA V5 fue elegido porque representa uno de los sistemas de CAD más moderno disponible en relación con los conceptos de programación y estructuras de datos. Otra de las ventajas que CATIA ofrece es la capacidad para automatizar el diseño CAD mediante la introducción de restricciones en el modelo a través de herramientas integradas o de programación.

Al tener compatibilidad con la Programación Orientada a Objetos (POO) incluida: el poder tratar a un grupo de propiedades, métodos y otros miembros relacionados se tratan como si fuese una sola unidad (encapsulamiento); además de crear nuevas clases basadas en una clase existente (herencia); y poseer múltiples clases que se pueden utilizar de forma intercambiable (polimorfismo) con Visual Basic. En el cual el código interactúa con los objetos, construidos dentro del entorno formando una geometría.

La razón de la elección de este software es porque ofrece tantas capacidades como la generación de macros sofisticados, con características útiles, de forma que se utilizan considerablemente para crear un modelo virtual, en el presente caso, el corazón primitivo debido a que se puede conseguir dentro del software (CATIA V5) evitando una programación del entorno ya existente.

Modelado paramétrico en CATIA

El software permite diferentes niveles de parametrización, ofreciendo una amplia variedad de capacidades de automatización y representación esquemática de los diferentes niveles de parametrización, estos son:

Modelos fijos, parámetros, patrones, fórmulas, normas y reacciones.

- Modelos fijos

La base para la parametrización son los modelos geométricos ya construidos dentro del entorno, un ejemplo de ello sería una caja, con sus respectivas dimensiones (10 x 20 x 30 [mm]), para modificar este objeto, se tiene que acceder a la cota correspondiente y su propiedad puede entonces ser modificada.

ANEXO

- Parámetros

Los parámetros representan un mayor nivel de parametrización que los modelos fijos. En este caso, los parámetros pueden ser asociados para definir la propiedad geométrica fuera de las dimensiones del objeto, de esta forma cuando se cambia el valor del parámetro asociado con la geometría, la geometría cambiará también.

- Patrones

Los patrones son copias geométricas de una cierta geometría la cual se repite de la misma geometría una y otra vez. Los elementos del patrón pueden ser modificados individualmente. Una desventaja es que todos los elementos del patrón son una entidad única, la ventaja es que se pueden crear instancias dinámicamente mediante el uso de parámetros. Los objetos permiten la instanciación dinámica de la geometría tantas veces como sea necesario.

- Fórmulas

Las fórmulas se utilizan para definir la relación matemática entre los parámetros o propiedades geométricas. Si se cambia el valor de cierta dimensión, las dimensiones asociadas se alterarán.

- Reglas y Reacciones

Las reglas y las reacciones son los niveles más altos de parametrización que pueden ser utilizados para introducir el conocimiento del diseño en la geometría, siendo herramientas útiles en la automatización y la modificación de la geometría.

Las *reglas* están siempre activas en la geometría y se pueden utilizar para describir las relaciones entre los objetos geométricos, parámetros, propiedades, etc.

Por otro lado las *reacciones* se activan para definir la respuesta de la geometría sobre la base de un determinado suceso o evento.

Estructura de los objetos en CATIA.

CATIA al tener afinidad con la programación orientada a objetos, los estructura de tal manera que son el resultado de la instanciación de una clase. Un objeto individual es un contenedor de geometría u otra información. Esto es, un miembro de una clase que tiene propiedades en lugar de variables. De forma que interactúa con métodos.

ANEXO

Un método es la instrucción utilizada para modificar un objeto existente o crear un nuevo objeto. Un método puede tener varios parámetros de entrada y salida.

Los objetos en la aplicación se organizan en una estructura jerárquica de padres e hijos. Mientras que un objeto de la lista es una “colección de objetos” individuales. Cada ruta de objeto de las entidades de la lista comienza con la “colección de clase” base, la colección proporciona métodos básicos para cada objeto de la lista. El “objeto raíz” para todas las macros en CATIA es de “*Application*”. Con esta entidad se puede crear y operar documentos y gráficos. (Figura A.1)

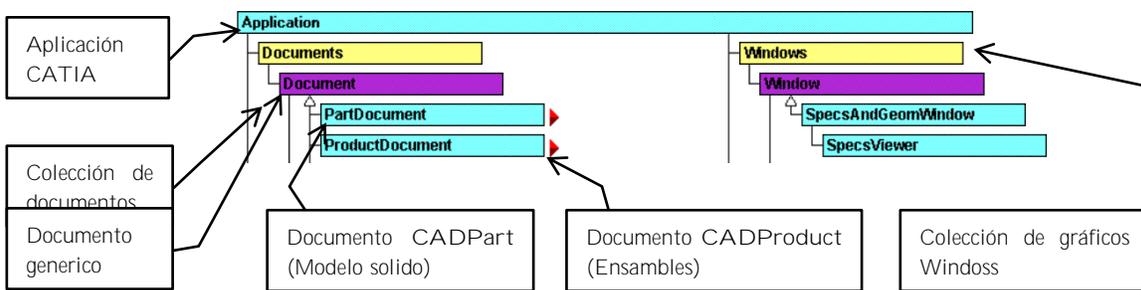


Figura A.1 Estructura Clases y objetos en Catia (Dassault Systèmes, 2009)

Estructura de *PartDocument*

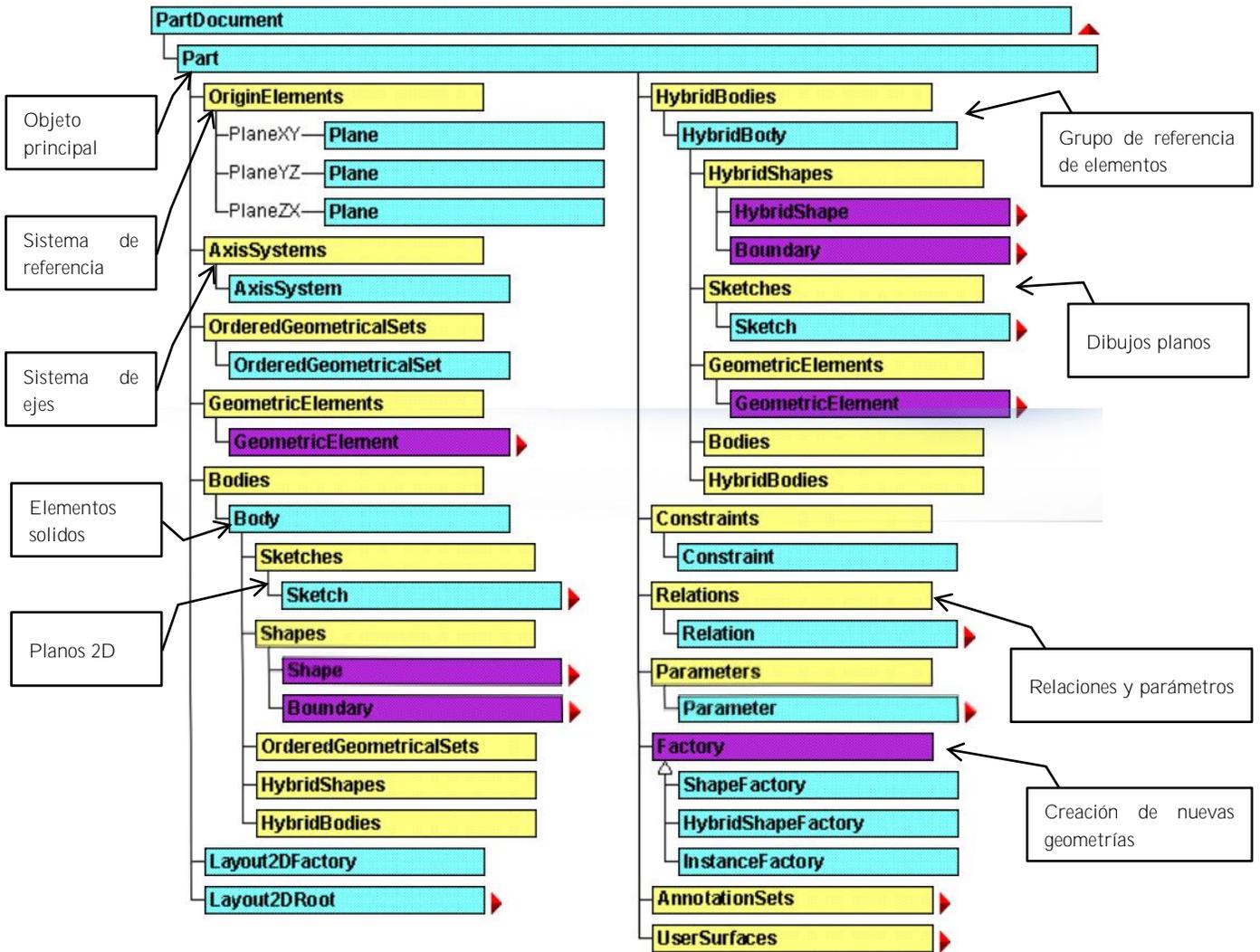
Los descendientes del objeto *PartDocument*, tienen la estructura del árbol comenzando con el objeto *Part* (Figura A.2). Entre estos objetos, encontramos primeros objetos y colecciones. Estos objetos son: líneas, planos, superficies, etc.

Las entidades que se crean, empezando por el cuerpo principal se almacenan en la colección *Body*. Estos cuerpos contiene la geometría de la parte, los cuerpos híbridos que se crean son almacenados en la colección *HybridBodies*.

Las restricciones se pueden establecer para objetos y almacenados en la colección de restricciones que pueden recuperar el uso de restricciones de propiedad de la *Part*.

Mientras que las relaciones entre los parámetros almacenados en una colección de relaciones se pueden recuperar mediante la propiedad de Relaciones de la Parte. Los parámetros son los únicos que se pueden leer sin la propiedad de la *Part*.

Part Document Automation Objects



Legend

- Collection
- Abstract object
- Object

Figura 3.2 Estructura de los objetos. (Dassault Systèmes, 2009)

Las propiedades de *CATParts* se dividen en estándar y personalizadas. Las propiedades estándar están presentes en cada *CATPart*. Estos son: el número de parte, la revisión, la definición, la nomenclatura, la fuente, y la descripción. Mientras que las personalizadas sólo cuando son creados por un usuario.

Personalización y el procesamiento de datos en CATIA

Para automatizar tareas repetitivas, así como para desarrollar nuevas operaciones (funciones) que faciliten el proceso de modelado y que no se incluyen de forma predeterminada en el menú, se requiere agrupar un conjunto de operaciones que definen una tarea específica. Las herramientas para la automatización de tareas en el modelado con CATIA son: *PowerCopy*, *User Defined Feature* (UDF) y Macros. El objetivo de estas tres herramientas es la generación automática de resultados utilizando diferentes entradas que deben ser totalmente genéricas.

PowerCopy

Permite la automatización de las tareas que han sido previamente ejecutadas manualmente en un modelo sólido (documento tipo *CATPart*). Para usar la tarea *PowerCopy* se debe: definir los elementos del modelo (geometría, parámetros, fórmulas, etc.) que se tomarán como componentes referentes en el proceso de automatización, así como generalizar los parámetros donde los valores pueden ser modificados por el usuario y por último seleccionar los elementos resultantes.

Una vez guardado, el modelo que contiene un *PowerCopy* puede ejecutarse en cualquier otro documento *CATPart*, en el cual se especifican los valores de los parámetros que se establecieron al crear el *PowerCopy* procesando las características de los elementos junto con las especificaciones de diseño en el nuevo documento.

User Defined Feature (UDF)

El *UDF* es una plantilla que trabaja en el nivel de la pieza la cual ayuda a ejecutar tareas automáticamente dentro de un documento *CATPart* en forma muy similar a *PowerCopy*, al igual que esta, a partir de un conjunto de características (geometría, parámetros, fórmulas, etc.). El resultado es una característica del diseño de la forma que se pueden volver a utilizar en el diseño en otra parte, pero sin sus especificaciones de diseño.

Las principales diferencias entre una *UDF* y una *PowerCopy* son las siguientes:

- Es posible modificar un *PowerCopy* después que es creado, pero es imposible modificar una *UDF* después de su creación.
 - Es imposible ver el funcionamiento interno de *UDF*, que se considera como un "caja negra".
 - Los mecanismos utilizados en el momento de instanciación son diferentes.
-

ANEXO

Macros

Una macro es un código escrito en un lenguaje de programación determinado, utilizado para agrupar un conjunto de operaciones que definen la tarea de ser automatizado. Las ventajas esenciales de esta herramienta son las siguientes:

- Automatización de todo tipo de tareas creando o editando objetos
- Máxima reducción en el número de entradas, por lo tanto, minimizar la intervención del usuario en el proceso
- La posibilidad de trabajar junto con los documentos *CATPart* y *CATDrawing*, generando por lo tanto todas las posibles vistas del modelo sólido.

Funciones especiales: Macros

Una macro se compone de una serie de funciones agrupadas en un sólo comando, estas están escritas en algún lenguaje de programación, para realizar una serie de tareas definidas de manera automática. Cuando una tarea se realiza repetidamente es conveniente usar una macro para automatizar la tarea, ahorrando tiempo y reduciendo la posibilidad de error humano en los procesos iterativos.

Otras ventajas es la estandarización de procesos así como ampliar las capacidades de los programas. Una forma de creación de macros es con *Component Application Architecture* (CAA, CAA V5, or CNext), que es la interfaz de programación de aplicaciones (API) dentro del entorno de CATIA.

Cada programa de software tiene su propio lenguaje de programación, muchos de ellos comparten algunos elementos, en el caso de CATIA V5 automatización, fue diseñado originalmente para VisualBasic 6 (VB6), Visual Basic for Applications (VBA) y Visual Basic Script (VBScript). En el cual VB6 es más compleja que VBA y que VBScript, así como derivados de Visual Basic utilizando secuencias de comandos como CATScript.

CATScript es la versión interna de Dassault System. Un CATScript es un lenguaje de programación secuencial y contiene una interface de edición. Los editores de texto (Como Notepad++) pueden utilizarse para la codificación. Una ventaja de escribir macros de CATScript es el hecho de estar inmerso en el entorno de CATIA, con la grabación de macros. Por otra parte, existen desventajas, como: difícil de depurar y de flexibilidad limitada.

ANEXO

VBScript (VBScript MS) y CATScript son muy similares con la gran diferencia de la declaración de variables. Al declarar variables como *As String*, *As Integer*, etc. Se puede dar un mejor seguimiento de cada tipo de variable.

Visual Basic (VB6 o VB) es la versión completa. Derivado de BASIC, los programas VB6 pueden generar otro tipo de programas independientes, pueden crear servidores y ActiveX y pueden ser compilados. Los programas VB se ejecutan en su propio espacio de memoria.

VBA (Visual Basic for Applications) es un subconjunto de Visual Basic y se aloja en aplicaciones tales como CATIA (a partir de la versión V5R8), Microsoft Word, Excel, etc. VBA proporciona un entorno de programación completo con un editor, depurador y visor de objetos de ayuda. Se permite declarar la biblioteca de objetos usada. En CATIA, VBA tiene la sintaxis completa de VB6 y de IDE, que es similar a VBA de Excel. Las ventajas de utilizar las macros CATvba incluyen el uso de la GUI, la construcción de formas y la capacidad de depuración del editor de macros. La desventaja es que los programas de VBA no pueden compilarse en archivos ejecutables o archivos dll y no opere en su propio espacio de memoria.

Existen dos métodos principales de comunicación de una macro con CATIA: dentro del entorno o fuera del entorno. (Ziethen, 2013)

Dentro del entorno: El primer método de comunicación de una macro con CATIA es cuando la aplicación programada se ejecuta dentro del entorno. CATIA esencialmente cede a las instrucciones contenidas en la macro y la memoria asignada se limpia después de cada ejecución del programa. Las variables del macro se borran una vez que ha finalizado el macro. Para crear y acceder a macros que estén en el entorno de CATIA hay que ir a herramientas>Macros, teniendo en cuenta las únicas opciones son VBScript, CATScript o VBA.

Fuera del entorno: El otro método de comunicación ocurre cuando el programa se ejecuta independiente. La aplicación puede ejecutarse desde Excel, Word, Windows Explorer, etc. CATIA es plenamente activa mientras se ejecuta el programa. VB.net o VB6 también puede ser utilizado.

Sintaxis de los Macros en CATIA

A continuación se presentan algunas de las características de la sintaxis de programación macro (Ross, 2013) en CATIA:

- **Minúsculas:** De forma predeterminada, VBA no es sensible a mayúsculas y no distingue entre mayúsculas y minúsculas ni la ortografía de las palabras.
 - **Comentario:** Añadir comentarios a sus declaraciones con un apóstrofo ('), al principio de una línea separada, o al final de una declaración. Se recomienda agregar comentarios siempre que sea posible para facilitar sus scripts comprender y mantener, especialmente si otro usuario tiene que realizar cambios en este, más adelante en el camino.
 - **Sangría:** Sangría o fuera abolladura script para reflejar la estructura lógica y anidamiento de las declaraciones para hacerlo más fácil de leer.
 - **Paréntesis:** Es importante utilizar paréntesis correctamente en cualquier declaración para lograr el resultado deseado y evitar errores.
 - **Punto y coma (:):** Insertar un punto y coma le permite escribir varios comandos en la misma línea de código.
 - **Apostrofe (') simple:** Para devolver una comilla simple que no indica un Comentario (necesario en una fórmula, por ejemplo), se utiliza la función Chr. Chr() es una función incorporada de VBA que devuelve el carácter que se corresponde con el valor numérico de su argumento usando el esquema de codificación ASCII. Al proporcionar un valor entero (por ejemplo, 39) Chr informará atrás el carácter que corresponde a ese valor. El valor ASCII de 39 es la marca de comilla simple. Chr(34) es la marca de comillas dobles.
 - **Espacios:** Los espacios son ignorados por VBA. Sin embargo agregar espacios en blanco adicionales al script puede mejorar la claridad de lectura para el programador.
 - **Las cadenas de texto:** Cuando se introduce un valor como una cadena de texto debe agregar comillas antes y después de la cadena. Se puede concatenar, o combinar, varias cadenas utilizando un plus (+) o ampersand (&) signo: txtString = "este valor es" + Valor_X. Para obtener la longitud completa de la cadena de len(string). Para averiguar si una cadena contiene otro uso cadena Instr(). O convertir cadenas numéricas y viceversa mediante str(num) o val(string).
 - **Guión Bajo (_):** Pasa a la siguiente línea (símbolo de la concatenación de línea).
-

Creación de Macros

CATIA V5 permite la creación y ejecución de macros mediante el uso de distintos lenguajes de programación basados en objetos. Cualquier entidad de aplicación (documentos, ventanas gráficas, geometría, parámetros, fórmulas, etc.) representado y será operada por estos idiomas como un "objeto".

Bibliotecas de los Macros

Las macros CATIA están almacenadas en bibliotecas en uno de tres lugares: carpetas (vbscript y CATScript), archivos de proyecto (catvba) o documentos tales como CATParts, CATProducts y CATDrawings. Sólo una de estas bibliotecas macro puede utilizarse a la vez. Al crear una nueva biblioteca de macros, ya debe existir la ubicación de la carpeta o ruta de acceso.

Establecimiento de nombres de las variables en CATIA

Las variables conforman la columna vertebral de cualquier lenguaje de programación. Básicamente, las variables almacenan la información que se encontrará más adelante al referirse a un nombre o "variable".

Se puede declarar variables "tipo" para orientarnos acerca de lo que es posible realizar con esa información. El valor de una variable no necesariamente será constante a lo largo de la duración de la ejecución de un programa.

Hay algunas pautas que debemos seguir al nombrar variables; deben ser menos de 255 caracteres, no pueden empezar con un número y caracteres especiales como espacios y símbolos. Para evitar conflictos de nombres. Dos variables no pueden tener el mismo nombre.

- Buenos nombres de Variable: *dimensionUno, dimension1, Dim_1*
- Nombres de variables malos: *dimensión uno, 1dimension*

Existen diferentes tipos de variables. Para guardar su nombre en una variable, se utilizaría una variable de cadena. Para guardar tu edad entonces ese tipo de variable es un entero. Declarar variables como un tipo específico dice a CATIA la forma que se almacena la información será "una cierta forma".

Para almacenar un número como un número entero quiere decir que será un número entero sin decimal, como 5, 66 o -777. Si se intenta almacenar \$69.96 como un entero, obtendrá un error.

Dos variables que son enteros pueden añadir o restar uno al otro ($50-5 = 45$) pero no tendría sentido intentar sumar dos cadenas (Nombre+Apellido =???). Declarar las variables como un tipo nos da un sentido de lo que puede y no se puede hacer con ellas.

Tipo de variable	Valor predeterminado	Descripción
String	Null	Texto (letras, números, espacios, signos de puntuación)
integer	0	Número entero
Double	0	Cualquier número, positivo o negativo
Long	0	Un entero de 4 bytes
Boolean	FALSO	Declaración lógica como verdadero o falso, sí o no, etc..
CATVariant	Null	Índice de una lista de objetos
CATBStr	Null	Cadena de expresiones de CATIA

Un ejemplo de esto sería el famoso “Hola mundo!”, que se inicia en cualquier tutorial de programación. El siguiente código pertenece a un CATScript. (Ross, 2013)

```
CODE:CATScript, 'Exercise.CATScript
Option Explicit Sub CATMain()
'Declara la variable como tipo String
Dim strMessage As String
'Define la variable String como el texto Hola Mundo
strMessage = "Hola Mundo!"
'Muestra el mensaje dentro de CATIA
Msgbox strMessage
End Sub
```

Asignar un ícono a un Macro

En lugar de tener que abrir la ventana macro para ejecutar su programa cada vez, puede ser mucho más rápido asignar un ícono a macro y colocarlo en una barra de herramientas de acceso rápido. (Ziethen, 2013) Para el código anterior es necesario hacer lo siguiente:

- 1) Vaya a herramientas > Personalizar > comandos.
- 2) Desplácese hacia abajo hasta Macros y haga clic en "Mostrar propiedades". debajo de la opción Comandos, seleccione la macro de Exercise1.CATScript y haga clic en la casilla "...". El navegador de íconos aparece. Hay más de 9.000 íconos únicos para elegir (o se puede crear uno)

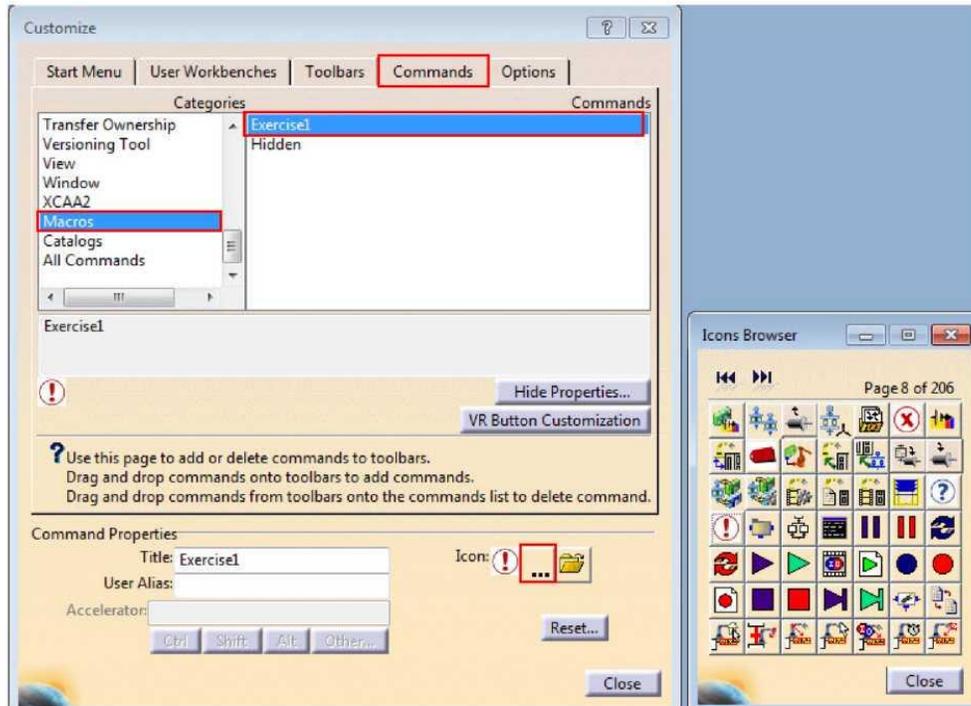


Fig. A.1 "Customize" en CATIA

3. Finalmente, arrastrar y soltar el archivo macro desde la ventana de comandos a cualquier barra de herramientas te gustaría que aparezca en el icono (como "Propiedades gráficas" o crear uno nuevo). También se puede configurar atajos de teclado personalizados desde dentro de la misma ventana también o crear una barra de herramientas personalizada por ir a *herramientas > Personalizar > barras de herramientas > nueva*.

```

CODE:CATScript, • Creación de curvas intersección.
'Funcion Esperar
Sub PauseSeconds(NumSeconds As Double)

Dim PauseTime, Start, Finish, TotalTime
PauseTime = NumSeconds
Start = Timer
Do While Timer < Start + PauseTime
    DoEvents
Loop
Finish = Timer
TotalTime = Finish - Start
End Sub

Sub CATMain()
'CREA CURVAS INTERSECCION
'Documento Activo
Dim oPartDoc As Part
On Error Resume Next
Set oPartDoc = CATIA.ActiveDocument.Part
If Err.Number <> 0 Then
Message = MsgBox("Este script sólo funciona con un CATPart como documento activo",
vbCritical, "Error")
Exit Sub
End If
'=====
'-----Seleccion de Superficie o Volumen-----
msgbox "Seleccione una superficie"
Set Select_sup = CATIA.ActiveDocument.Selection
Select_sup.Clear
Dim InputObjectType(0)
InputObjectType(0)= "Edge" ' "Surface" no selecciona la superficie
Status=Select_sup.SelectElement2(InputObjectType,"Seleccione un
elemento",false)
if (Status="Cancel") then
Select_sup.Clear : Exit Sub
Else
Set mySuperficie = Select_sup.Item(1).Value
MsgBox "Sup.name: "& mySuperficie.Name
end if
'-----Seleccion de Plano de Referencia-----
'Selecciona Plano Ref.
msgbox "Seleccione un Plano de referencia"
Set Select_plane = CATIA.ActiveDocument.Selection
Select_plane.Clear
Dim oInputType(0)
oInputType(0)="Plane"
Status=Select_plane.SelectElement2(oInputType,"Seleccione un elemento",false)
if (Status="Cancel") then
Select_plane.Clear : Exit Sub
Else
Set myPlano = Select_plane.Item(1).Value
MsgBox "Plane.name: "& myPlano.Name
end if

```

ANEXO

```
'-----Entrada de separacion entre planos-----
Dim input
Dim sLF
Dim arrayOfVariantOfBSTR1(0)
sLF = Chr(10)
input = InputBox ("Entrada: Separación de las curvas intersección [mm]")
If input = "" Then
Exit Sub
End If
'-----Planar Sections-----Op (1)-----
Set oSelection = CATIA.ActiveDocument.Selection
oSelection.add(mySuperficie)
oSelection.add(myPlano)
REM oSelection.add(input)
'Selecciona Plano Ref.
CATIA.StartCommand "Planar Sections"
    Call PauseSeconds(200)
    oSelection.Search "Name=" & mySuperficie.Name ' nombre del elemento a
aplicar la sección. (superficie o solido)
    Call PauseSeconds(200)
    oSelection.Search "Name=" & myPlano.Name 'selección del primer plano con la
que uno se aplicará la sección
    Call PauseSeconds(200)
    REM for i=0 to 7 ' Ciclo para alcanzar la entrada de separación [mm]
    REM call CATIA.SystemService.ExecuteBackGroundProcessus("WScript.exe
d:\Macros\SendTAB.vbs")
    REM oSelection.Search "Name=" & input.Value '¿Como puedo enviar los
dijitos??
    REM Call PauseSeconds(200)
    REM Next
    'oSelection.ValuateFromString(input) ' clic en el boton APPLY
    REM Call PauseSeconds(200)

    for i=0 to 14 ' Ciclo para alcanzar el botón de aplicar
    call CATIA.SystemService.ExecuteBackGroundProcessus("WScript.exe
d:\Macros\SendTAB.vbs")
    REM oSelection.Search "Name=" & input.Value
    Call PauseSeconds(200)
    Next
    call CATIA.SystemService.ExecuteBackGroundProcessus("WScript.exe
d:\Macros\SendEnter.vbs")'
    Call PauseSeconds(200)
    for i=0 to 15 'Ciclo para alcanzar el botón OK
    call CATIA.SystemService.ExecuteBackGroundProcessus("WScript.exe
d:\Macros\SendTAB.vbs")
    Call PauseSeconds(200)
    Next
    call CATIA.SystemService.ExecuteBackGroundProcessus("WScript.exe
d:\Macros\SendEnter.vbs")' clic en el boton OK
    Call PauseSeconds(200)
    oSelection.Search "Name=" & "Planar Sections." & "*", all"
'seleccionar todas las secciones "Planar Sections." creadas para ocultarlas
    Call PauseSeconds(200)
    Set VisPropertySet1 = oSelection.visProperties
    VisPropertySet1.SetShow 1 ' ocultando la selección
    PauseSeconds(200).Sleep 200
```

ANEXO

```
oSelection.Clear

'=====
oPartDoc.Update

End Sub
```

CODE:CATScript, • Cantidad de puntos.

```
Sub CATMain()
'Documento Activo
Dim oPartDoc As Part
On Error Resume Next
Set oPartDoc = CATIA.ActiveDocument.Part
If Err.Number <> 0 Then
Message = MsgBox("Este script solo funciona con un CATPart como documento activo",
vbCritical, "Error")
Exit Sub
End If
'=====
'Selecciona la curva intersección sobre la cual se crearan los puntos
msgbox "Seleccione una curva intersección en el área gráfica"
Set Selection = CATIA.ActiveDocument.Selection
Dim InputObjectType(0)
InputObjectType(0)="Edge"
Status=Selection.SelectElement2(InputObjectType,"Seleccione un elemento",false)
if (Status="Cancel") then
Selection.Clear : Exit Sub
Else
Set myselection = Selection.Item(1).Value
' MsgBox "Sel.Name: "& myselection.Name
end if
'Valor de la disancia de la curva
CATIA.StartCommand "Measure Item"
' SendKeys "{ENTER}", TRUE ' clic on OK button
Dim input
Dim sLF
Dim arrayOfVariantOfBSTR1(0)
sLF = Chr(10)
input = InputBox ("Entrada: Redondea la dimensión [mm]")
If input = "" Then
Exit Sub
End If
'---- Begin resolution script for object : Part2\MeasureEdge.3\Length
REM Dim oParams As Parameters
REM Dim Length_C, Puntos_5, Puntos_4 As integer
REM Set oParams = CATIA.ActiveDocument.Part.Parameters
REM Set Sel_param = CATIA.ActiveDocument.Selection
REM Set oCount = oParams.count
REM Sel_param.add(oParams(oCount))
REM 'Length_C = 200
REM MsgBox "Length_C = " & Sel_param.Value
REM MsgBox "Length_C.n = " & Length_C.Name 'No
REM MsgBox "Length_C.v = " & Length_C.Value 'No
'Calcula los numeros de puntos
Length_C = input
Puntos_5 = Length_C / 5 'La curva entre 5[mm]
```

ANEXO

```
Puntos_4 = Length_C / 4 'La curva entre 4[mm]
MsgBox "Se recomienda que los puntos equidistantes de la curva sean entre " &
Puntos_5 & " y " & Puntos_4
REM Selection.Clear
oPartDoc.Update
End Sub
```

CODE:CATScript, • Creación de puntos intermedios.

```
Sub CATMain()
'CREA UN PUNTO INTERMEDIO DE DOS GRUPOS DE GS
'Documento Activo
Dim oPartDoc As Part
On Error Resume Next
Set oPartDoc = CATIA.ActiveDocument.Part
If Err.Number <> 0 Then
Message = MsgBox("Este script sólo funciona con un CATPart como documento activo",
vbCritical, "Error")
Exit Sub
End If
'-----Declaracion de Variables -----
Dim oHyShaF_1 As Factory
Set oHyShaF_1 = oPartDoc.HybridShapeFactory
Dim oGS1, oGS2 As HybridShape
Dim ref1, ref2 As Reference
Dim hSPB_int As HybridShapePointBetween

'-----GS1-----
Dim EnableSelectionFor(0)
EnableSelectionFor(0) = "HybridBody"
Set sSEL = CATIA.ActiveDocument.Selection
' sSEL.Clear' Reset the Selection
' Define Selection
MsgBox "Seleccione el 'Geometrical Set' dónde están los puntos iniciales"
UserSelection = sSEL.SelectElement2(EnableSelectionFor, "Seleccione otro
Geometrical Set", False)
' Evalua si la seleccion es correcta o no-----
If UserSelection <> "Normal" Then
MsgBox "Error con la seleccion"
Exit Sub
Else
'--Guarda nombre t1.N-----
Set ohybridbody1 = sSEL.Item(1).Value
REM MsgBox ohybridbody1.Name
Dim hybridShapes1 As HybridShapes
Set hybridShapes1 = ohybridbody1.HybridShapes
Dim Sel_count As Selection
Set Sel_count = CATIA.ActiveDocument.Selection
'Busca los puntos en la seleccion-----
Sel_count.Search "CATPrtSearch.Point,sel"
Dim oCount
oCount = Sel_count.Count2
REM MsgBox "# de rep " & oCount
End If

'-----GS2-----
EnableSelectionFor(0) = "HybridBody"
```

ANEXO

```
Set sSEL2 = CATIA.ActiveDocument.Selection
    sSEL2.Clear ' Reset the Selection
' Define Selection
MsgBox "Seleccione el 'Geometrical Set' dónde están los puntos de finales"
UserSelection = sSEL2.SelectElement2(EnableSelectionFor, "Seleccione otro
Geometrical Set", False)

' Evalua si la seleccion es correcta o no-----
If UserSelection <> "Normal" Then
    MsgBox "Error con la selección"
    Exit Sub
Else
'--Guarda nombre t2.N
    Set ohybridbody2 = sSEL2.Item(1).Value
    ' sSEL2.Clear
    End If
    REM MsgBox ohybridbody2.Name
    Dim hybridShapes2 As HybridShapes
    Set hybridShapes2 = ohybridbody2.HybridShapes

'-----GS_int-----
'Selecciona GS_Intermedio
EnableSelectionFor(0) = "HybridBody"
Set sSEL4 = CATIA.ActiveDocument.Selection
    sSEL4.Clear' Reset the Selection
' Define Selection
MsgBox "Seleccione el Geometrical Set dónde estarán los puntos intermedios"
UserSelection = sSEL4.SelectElement2(EnableSelectionFor, "Seleccione otro
Geometrical Set", False)
' Evalua si la seleccion es correcta o no-----
If UserSelection <> "Normal" Then
    MsgBox "Error con la seleccion"
    Exit Sub
Else
    Set ohybridbody3 = sSEL4.Item(1).Value
' sSEL4.Clear
End If
    Dim oGS_int As HybridBody
    Set oGS_int = ohybridbody3

=====Creacion_ P_ intermedio =====
Dim i
For i = 1 to oCount
    Set oGS1 = hybridShapes1.Item(i)
    Set oGS2 = hybridShapes2.Item(i)
    Set ref1 = oPartDoc.CreateReferenceFromObject(oGS1)
    Set ref2 = oPartDoc.CreateReferenceFromObject(oGS2)
    Set hSPB_int = oHyShaF_1.AddNewPointBetween(ref1, ref2, 0.555, 1)
    oGS_int.AppendHybridShape hSPB_int
    oPartDoc.InWorkObject = hSPB_int
Next

'---Busca-los-parametros-de-cada-Punto-(Point.Ratio)-----
Dim P1
CATIA.ActiveDocument.Selection.clear
CATIA.ActiveDocument.Selection.add(ohybridbody3)
```

ANEXO

```

Set oSelec_1 = CATIA.ActiveDocument.Selection 'CATIA.ActiveDocument.Selection
Dim realParam1 As Parameter
Dim formula1 As Formula
Dim oParams As Parameters
Set oParams = CATIA.ActiveDocument.Part.Parameters
Dim oDim1, oDim2 As Dimension
Dim Rels As Relations
Set Rels = CATIA.ActiveDocument.Part.Relations
Dim FName, F_comment, F_cont As CATBSTR

For i = 1 To oCount
'---Busca los Puntos del GS Intermedio
  oSelec_1.Search "CATPrtSearch.Point,sel"
  Set P1 = oSelec_1.Item(i).Value
  REM MsgBox "Point.Name: " & P1.Name
  REM MsgBox "Point.Ratio.Name: " & P1.Ratio.Name 'Set realParam1 =
parameters1.Item("Part6\Inter\Point.255\Ratio")
  REM MsgBox "Point.Ratio.Value: " & P1.Ratio.Value
'===Crea-la-Formula=====
  'Creacion de Parametros-----
  Set oDim1 = oRatio1.Value
  Set oDim2 = P1.Ratio 'oParams.CreateReal ("oDim2", 0) en caso de que no exista
  REM MsgBox "Ratio.Name: " & oDim2.Name
  REM MsgBox "Ratio.Value: " & oDim2.Value
  ' Creacion de Formulas -----
  FName = "" 'Nombre de la Formula, lo genera CATIA
  F_comment = "Comentario" ' Comentario oDim2.Name & " Es = a " & oDim1.Name
  F_cont = "`1to2`" 'nombre del parametro en catia
  Dim Formel As Formula
  Set Formel = Rels.CreateFormula (FName, F_comment, oDim2, F_cont)
  REM          Set          realParam1          =
CATIA.ActiveDocument.Part.Relations.Parameters.P1.Ratio
Next
'=====
oPartDoc.Update

End Sub

```

CODE:CATvbs, • Puntos intermedios en función del tiempo.

```

'Funcion Esperar =====
Sub PauseSeconds(NumSeconds As Double)
Dim PauseTime, Start, Finish, TotalTime
PauseTime = NumSeconds
Start = Timer
Do While Timer < Start + PauseTime
  DoEvents
Loop
Finish = Timer
TotalTime = Finish - Start
End Sub
'fin de Funcion Esperar =====
'CATScript PLAY o a 100%
Sub CATMain()
'Documento Activo
Dim oPartDoc As Part
On Error Resume Next

```

ANEXO

```
Set oPartDoc = CATIA.ActiveDocument.Part
If Err.Number <> 0 Then
Message = MsgBox("Este script solo funciona con un CATPart como documento activo",
vbCritical, "Error")
Exit Sub
End If

Dim parameters1 As Parameters
Dim realParam1 As RealParam
Set parameters1 = oPartDoc.Parameters
Set realParam1 = parameters1.Item(1)
'MsgBox "Variable.N:      " & realParam1.Name & "      y      valor      :" &
realParam1.ValueAsString
Ves = 0
For Ves = 0 To 100 Step 10 'incrementos de 10%
    realParam1.ValuateFromString(Ves & "%")
    'MsgBox "Param.Valor      :" & realParam1.Value
    oPartDoc.Update 'Actualiza
    Call PauseSeconds(500) 'en milisegundos
Next
End Sub
```