Reto 2 Interpolación- Análisis Numérico

Sandra Isabel Chávez Alcalde, Santiago Romero Pineda, David Ricardo Bernal Alfonso
Ingeniería de sistemas, Pontificia Universidad Javeriana
Bogotá D.C., Colombia
sandra.chavez@javeriana.edu.co
santiagoromero@javeriana.edu.co
bernal-david@javeriana.edu.co

13 de Noviembre de 2020

I. INTRODUCCIÓN

El presente documento contiene el proceso de realización del reto número dos de la materia de Análisis numérico, el cual consiste en la interpolación de una figura , el problema se expone a continuación.

II. RETO DE INTERPOLACIÓN

III. METODOLOGÍA Y MÉTODOS UTILIZADOS

A. ***Método de Bézier:*** *Es un método de interpolación el cual permite mediante algunos puntos de control dados, el modelamiento de curvas y superficies sin la necesidad de utilizar por completo los puntos que componen dicha curva.*

B. ***Curvas de nivel:*** *Es aquella línea que en un mapa une todos los puntos que tienen igualdad de condiciones dentro del espacio.*

IV. MARCO TEÓRICO

Blender: Programa informático gratis y de código libre orientado al modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales. El programa antes descrito se utilizó para la construcción de la figura en 3D.

Curvas de Bézier: Es un sistema que se desarrolló hacia los años 1960, fue pensado para el trazado de dibujos técnicos, se usa especialmente en el diseño aeronáutico y de carros [1]. Fue propuesto por Pierre Etienne Bézier, quien fue jefe de diseño y producción de automóviles Renault durante la mayor parte de su vida profesional. Su investigación comenzó sobre el diseño y la fabricación asistidos por computadora en 1960, desarrollando así herramientas interactivas para el diseño de curvas y superficies, e inició el fresado generado por computadora para el modelado de automóviles [2]. Actualmente en el diseño computacional, las curvas de Bézier son de gran importancia. Al trazar una línea que conecte todos los puntos de control, se obtiene el polígono de control, tiene el nombre de polígono debido a que es una versión poligonal de la curva deseada.

Propiedades de la curva de Bézier:

- La curva de Bézier se encuentra en el interior de la envolvente convexa de los puntos de control.
- La curva de Bézier es infinitamente derivable.
- El control de la curva es global. Modificar un punto de control implica modificar completamente la curva.
- Para efectuar una transformación afín de la curva es suficiente efectuar la transformación sobre todos los puntos de control.
- La curva comienza en el punto P_0 y termina en el P_n Esta peculiaridad es llamada interpolación del punto final.
- La curva es un segmento recto si, y solo si, todos los puntos de control están alineados.

Casos de curvas de Bézier: Existen tres grados para las curvas de Bézier, los cuales se encuentran a continuación:

- **Curvas lineales de Bézier:** Dados los puntos P_0 y P_n la curva de Bézier es una línea recta entre los dos puntos:
  $B(t) = P_0 + (P_1 - P_0)t = (1-t)P_0 + tP_1, t \in [0,1]$
- **Curvas cuadráticas de Bézier:** Es el camino trazado por la función B(t), dados los puntos: $P_0, P_1\ y\ \ P_2$:
  $B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2\ P_2, t \in [0,1]$
- **Curvas cúbicas de Bézier:** Cuatro puntos del plano o del espacio tridimensional, $P_0, P_{1,} P_2$, P_3 definen una curva cúbica de Bézier. 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. La forma paramétrica de la curva es:
  $B(t) = (1-t)^3 P_0 + 3P_1 t(1-t)^2 + 3\ P_2 t^2 (1-t) + P_3 t^3, t \in [0,1]$

Curvas de nivel: Según la teoría cuando se tiene una función $z = f(x, y)$ de dos variables reales y valor real, la gráfica de dicha función corresponde al conjunto:

$$gr(f) \coloneqq \{(x, y, f(x, y)) : (x, y) \in\ Dom\ (f).$$

Al ubicar dichos puntos en el espacio R3, se obtiene una superficie en dicho espacio. Una forma de estudiar dicha superficie, aunque en dos dimensiones, es considerar la intersección de dicha superficie con el plano $z = k$, donde $k\ \in Recorrido(f)$. De esta manera, se tiene el conjunto {(x, y, k): f (x, y) = k},el cual corresponde a la curva de nivel de la superficie z = f(x, y) con z = k.

Así al proyectar dicha intersección en el plano x, y, se obtiene lo que se denomina curva de nivel.

V. ANÁLISIS Y DESARROLLO

Para las curvas de Bézier, es necesario hacer uso de unos puntos de control que tendrán la finalidad de orientar la curva deseada, entre mayor sea el orden de la curva, mayor deberán de ser los puntos de control para tener una precisión acorde a la misma. Las líneas de Bézier pueden ser rectas o curvas dependiendo de la posición que tengan sus nodos y puntos de control. Un nodo es un punto en el espacio que sirve para unir el principio y el final de un trayecto, el trayecto es precisamente la unión entre dichos nodos y por los que al menos dos se forma la línea de Bézier, por último, los puntos de control son aquello que definen el trayecto de la línea. A continuación, se muestra en la figura un ejemplo de una curva de Bézier junto con sus componentes.

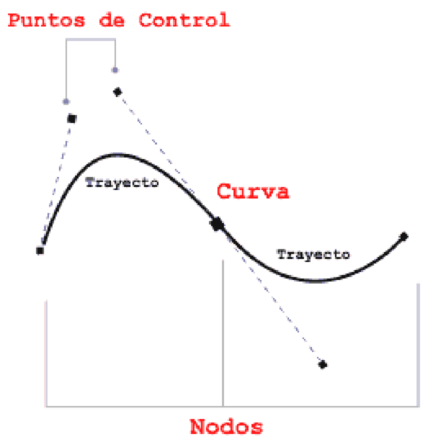

Figura 1: Curva de Bézier con sus componentes

Para la solución del reto se realizó una reconstrucción en 3D de la figura del jarrón utilizando el programa Blender, una vez obtenido el modelado se extrajeron de este la menor cantidad de puntos posibles con coordenadas x,y,z que permitieran graficar la figura fuera del programa de modelado, en la figura número dos se describe dicho proceso por medio de un diagrama de flujo, y en la siguiente sección se puede encontrar pantallazos del proceso de modelado del jarrón en tercera dimensión.

Por parte de la implementación se realizaron dos versiones, ambas desarrolladas en el lenguaje de programación Python.

En el caso de la primera versión el algoritmo desarrollado tomo en cuenta los valores obtenidos en el modelado de Blender se dispuso s dividir la figura en tres subfiguras para así obtener una más sencillo obtención del gráfico, se dividió en un cilindro, un esfera y un paraboloide, se graficaron uno a uno por separado a través de curvas de nivel y luego por medio de la interpolación del método de Bézier de unieron completando así el jarrón , además se le agregó también un pico realizado por completo utilizando el método antes descrito, se siguieron pasos similares a lo del modelado para la obtención del gráfico, en la figura 3 se puede apreciar un diagrama de flujo que explica más a fondo este proceso.

Para la versión dos del Jarrón se tomaron tres puntos de control basados en las coordenadas exportadas desde Blender, se dividió la figura en tres partes como las de la anterior versión y además se tuvo en cuenta el dividir también la figura entre cuatro cuadrantes que posibilitaran un mejor modelamiento, esta versión se desarrolló en su totalidad a través de curvas de Bézier, en la figura Número 4 se aprecia el diagrama de flujo de proceso. Además dentro de la carpeta Imágenes en los anexos se podrá encuentran las figuras en caso de no ser percibidas con comodidad en el presente documento.

Librerías usadas:

- Bezier
- Visvis
- Scipy.interpolate - Interp1d
- Matplotlib
- Numpy
- Math
- Mpl_toolkits.mplot3d -Axes3D, art3d

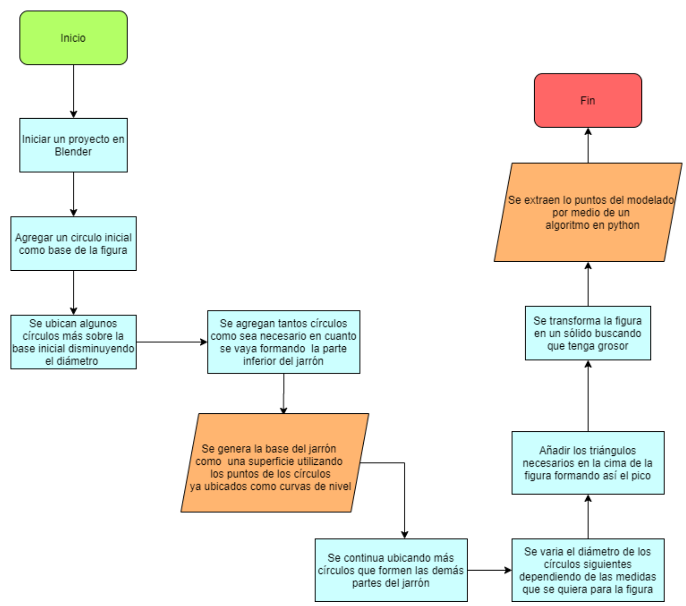

Figura 2: Diagrama de flujo Modelado en Blender

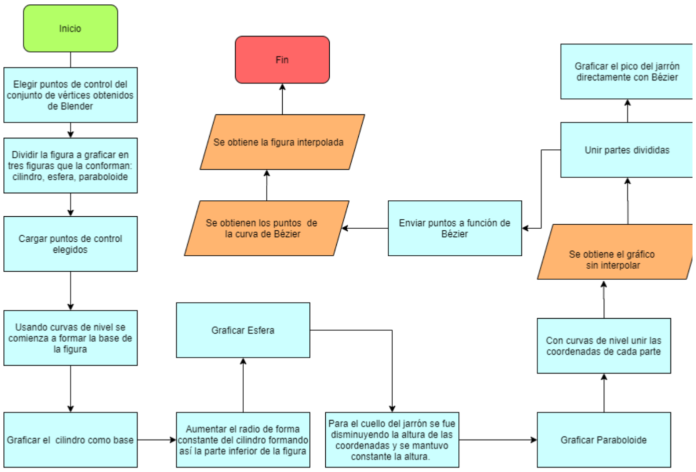

Figura 3: Diagrama de flujo Jarrón versión 1

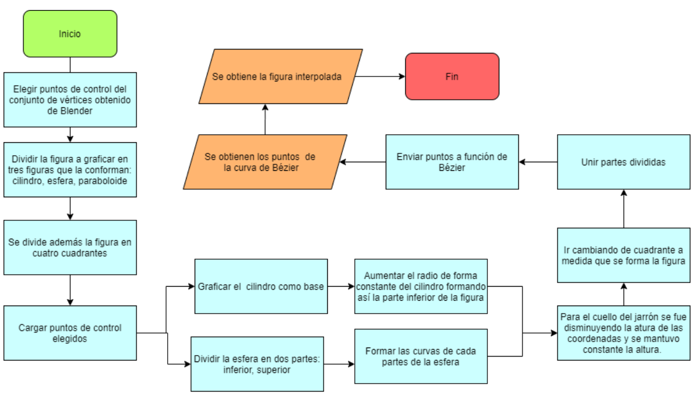

Figura 4: Diagrama de flujo Jarrón versión 2

VI. RESULTADOS

Del modelamiento realizado en Blender se obtuvieron un total de 1824 vértices en un principio y los siguientes son algunos de estos:

TABLA I

TAMAÑOS DE FUENTE PARA ARTÍCULOS

Puntos extraidos de figura generada en Blender			
No.	**X**	**Y**	**Z**
1	-1,1536E-07	0,225858509	-2,88E-09
2	-0,04406288	0,221518755	-2,88E-09
3	-0,08643234	0,208666086	-2,88E-09
4	-0,12548029	0,187794506	-2,88E-09
5	-0,15970606	0,159706175	-2,88E-09
6	-0,18779445	0,125480354	-2,88E-09
7	-0,20866597	0,086432427	-2,88E-09
8	-0,22151864	0,044062987	-2,88E-09
9	-0,22585845	2,21809E-07	-2,88E-09
10	-0,22151864	-0,044062540	-2,88E-09
11	-0,20866603	-0,086431980	-2,88E-09
12	-0,18779445	-0,125479937	-2,88E-09
13	-0,15970606	-0,159705698	-2,88E-09
14	-0,12548023	-0,187794089	-2,88E-09
15	-0,08643231	-0,208665669	-2,88E-09
16	-0,04406284	-0,221518338	-2,88E-09
17	-4,1762E-08	-0,225858092	-2,88E-09
18	0,04406275	-0,221518278	-2,88E-09
19	0,08643222	-0,208665609	-2,88E-09
20	0,12548015	-0,187794030	-2,88E-09

Luego de la interpolación realizada durante el reto, se pudieron reducir a 896 las coordenadas del jarrón.

En la siguiente tabla se encuentran los puntos de control tomados para la realización de la versión dos del jarrón.

TABLA II

PUNTOS DE CONTROL JARRA VERSIÓN 1

Puntos de Control Bezier Jarra Version 1		
X	**Y**	**Z**
0	1	0
0,5	0,8	0
0,8	0,5	0
1	0	0

En la siguiente tabla se encuentran los puntos de control tomados para la realización de la versión dos del jarrón.

TABLA III

PUNTOS DE CONTROL JARRA VERSIÓN 2

Puntos de Control Bezier Jarra Version 2		
X	**Y**	**Z**
0	1	0
0,7	0,7	0
1	0	1

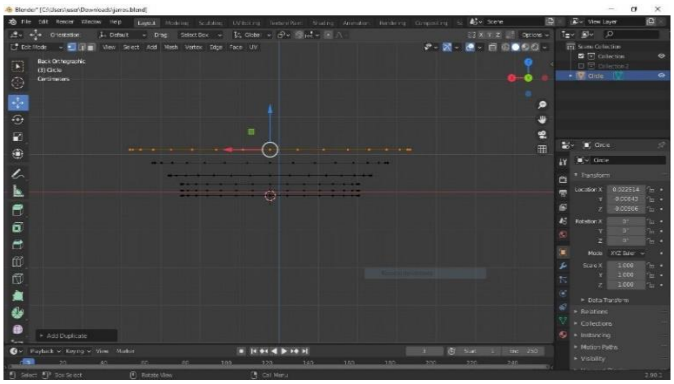

Figura 5: Inicio de modelado en Blender

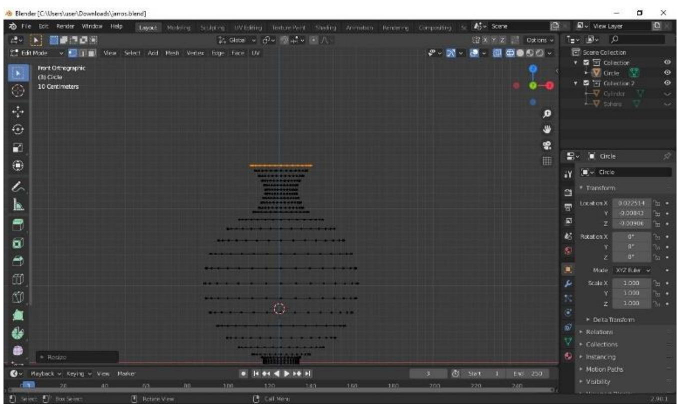

Figura 6: Modelado en Blender Jarrón base chata, cuello corto

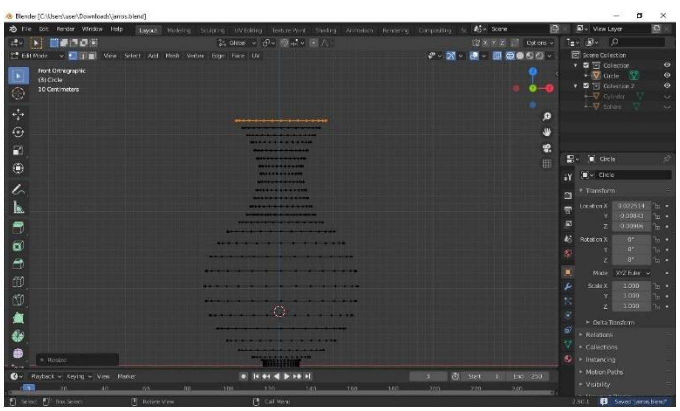

Figura 7: Modelado en Blender Jarrón cuello largo

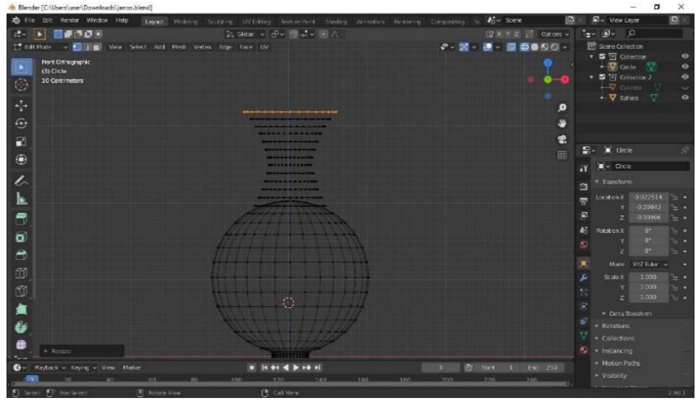

Figura 8: Modelado en Blender Jarrón cuello largo base esférica

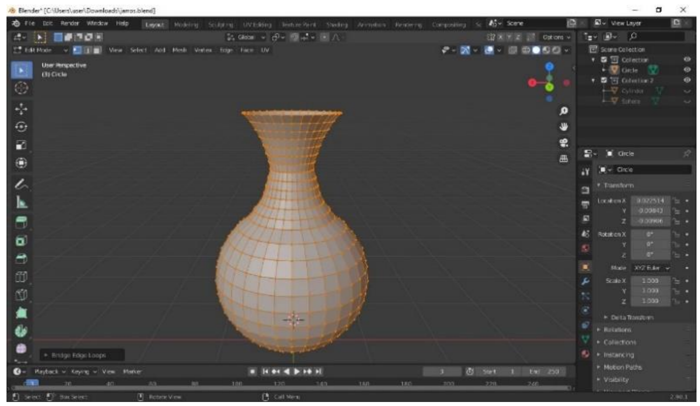

Figura 9: Modelado en Blender Jarrón cuello largo

Figura 10: Modelado en Blender Jarrón sólido con pico

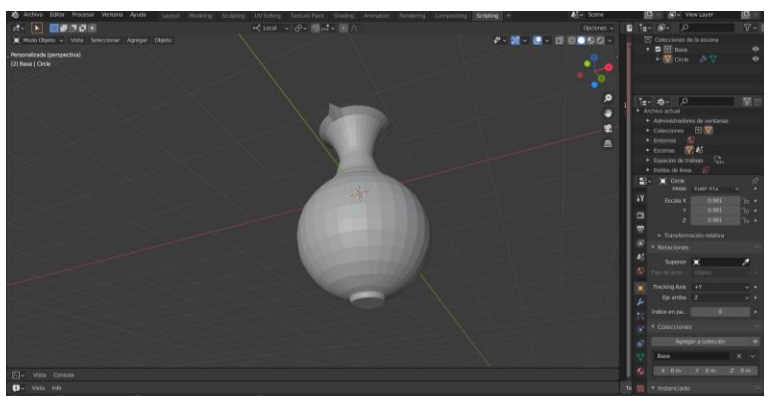

Figura 11: Modelado en Blender Jarrón sólido con pico, visión desde cuadrantes Versión 1 Python

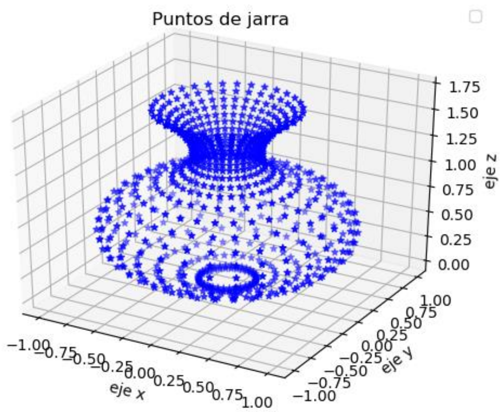

Figura 12: Gráfico Jarrón de puntos

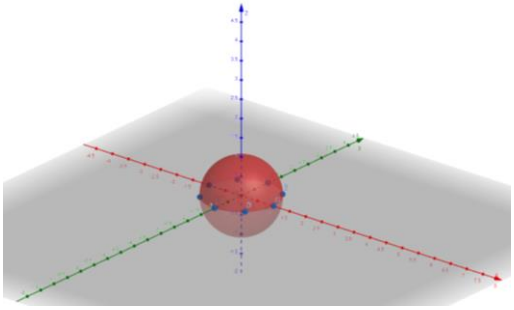

Figura 13: Gráfico 1, puntos de control Jarrón versión 1

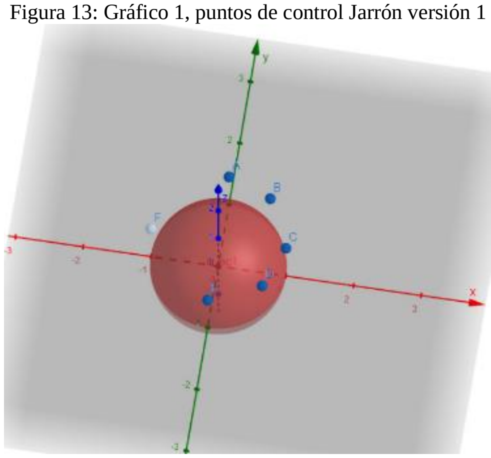

Figura 14: Gráfico 1, puntos de control Jarrón versión 1 GeoGebra

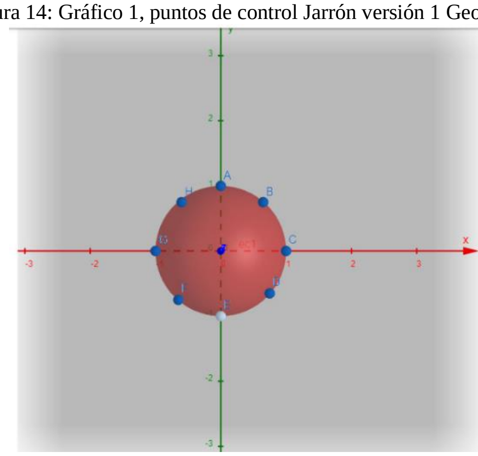

Figura 15: Gráfico 2, puntos de control Jarrón versión 1 GeoGebra

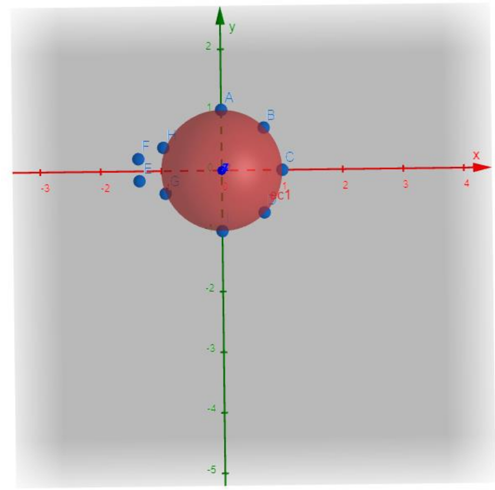

Figura 16: Gráfico puntos de control Pico Jarrón versión 1 GeoGebra

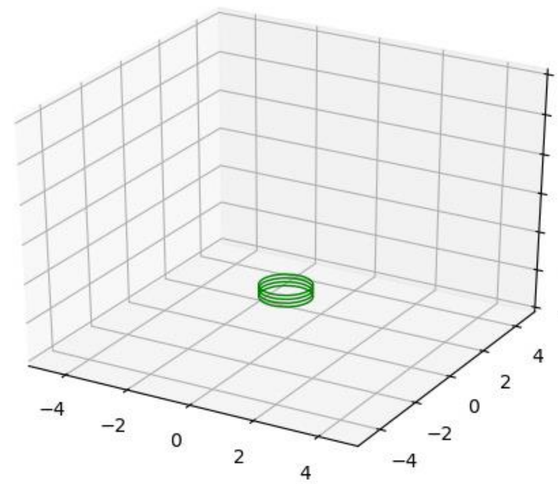

Figura 17: Gráfico de Cilindro Jarrón con curvas de nivel

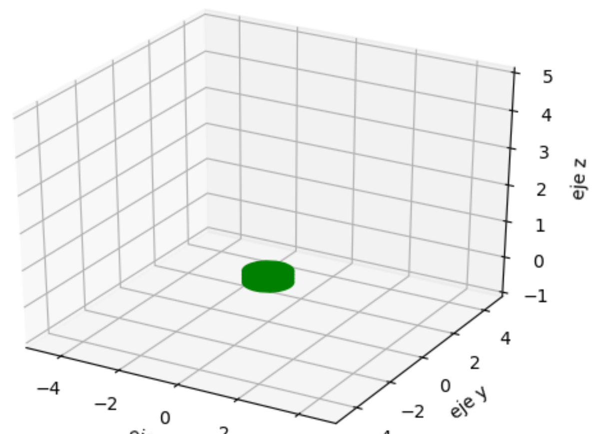

Figura 18: Gráfico de Cilindro interpolado
Jarrón con curvas de nivel

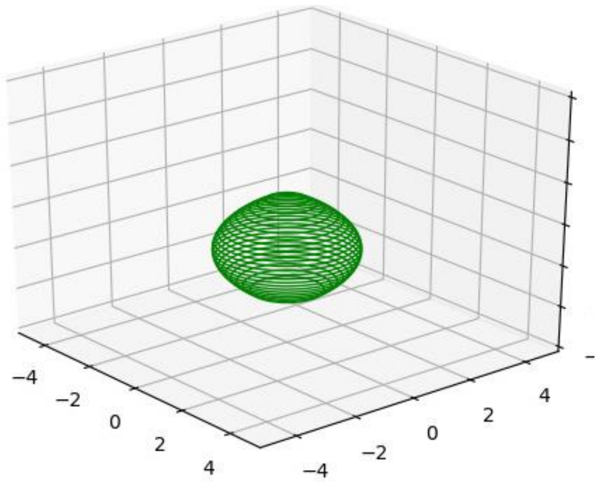

Figura 19: Gráfico de Esfera
Jarrón con curvas de nivel

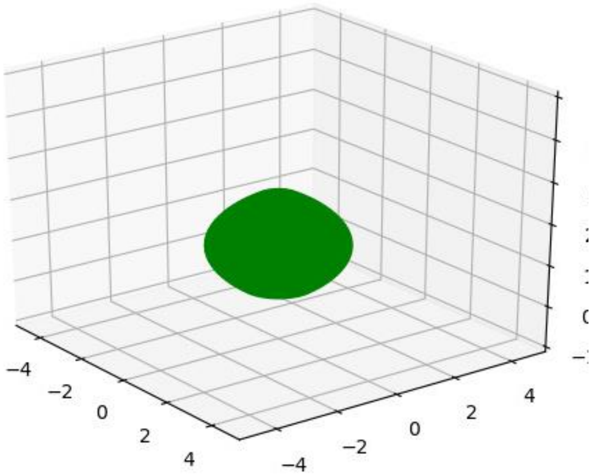

Figura 20: Gráfico de Esfera interpolada
Jarrón con curvas de nivel

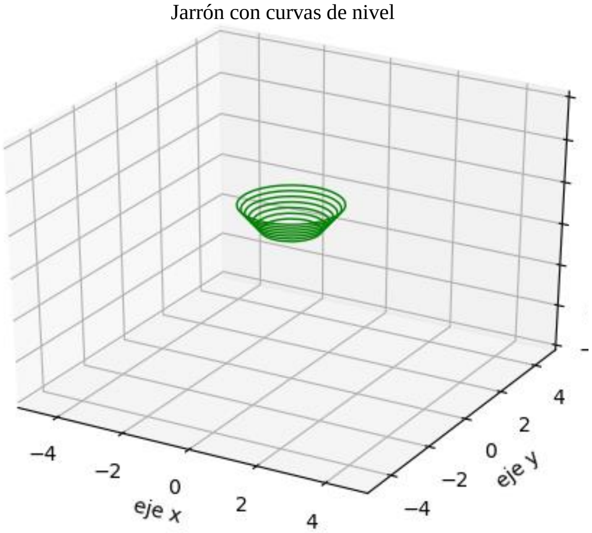

Figura 21: Gráfico de Paraboloide
Jarrón con curvas de nivel

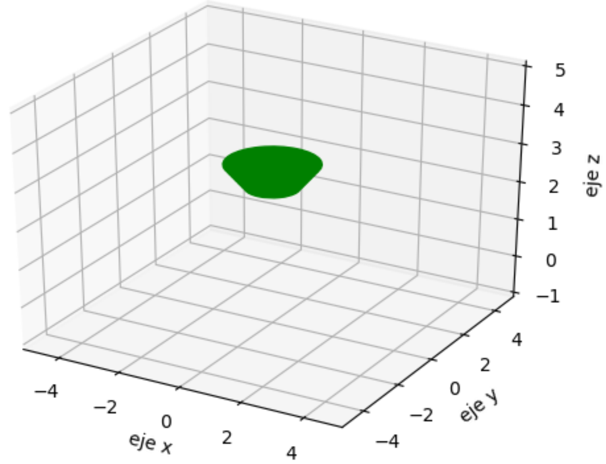

Figura 22: Gráfico de Paraboloide Interpolado
Jarrón con curvas de nivel

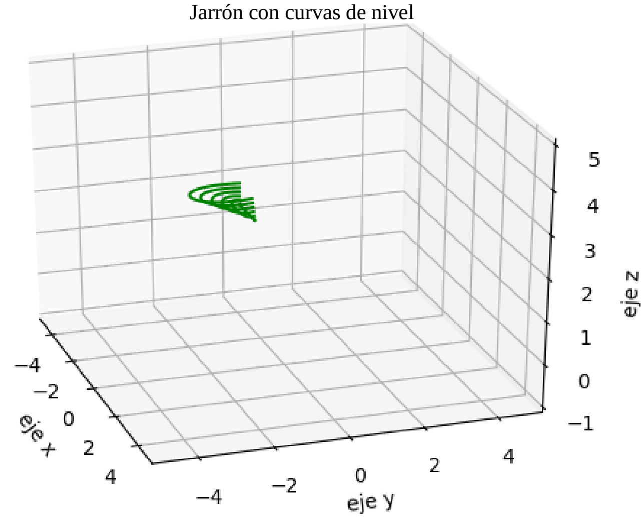

Figura 23: Gráfico de Pico Bézier
Jarrón con curvas de nivel

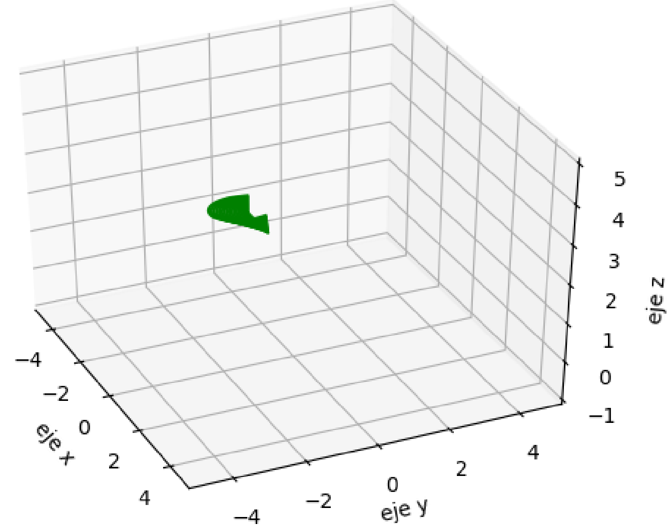

Figura 24: Gráfico de Pico Bézier Interpolado
Jarrón con curvas de nivel

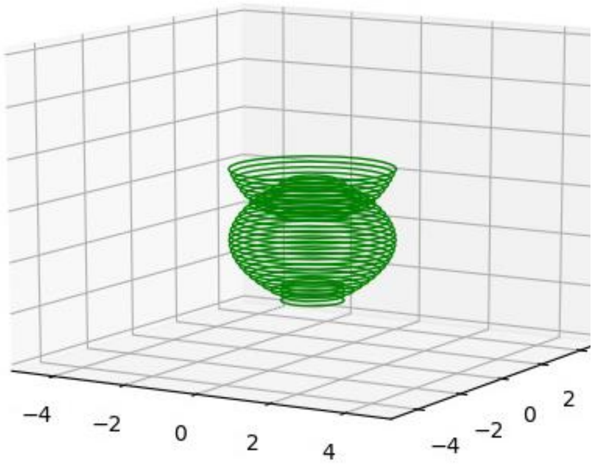

Figura 25: Gráfico Jarrón con curvas de nivel

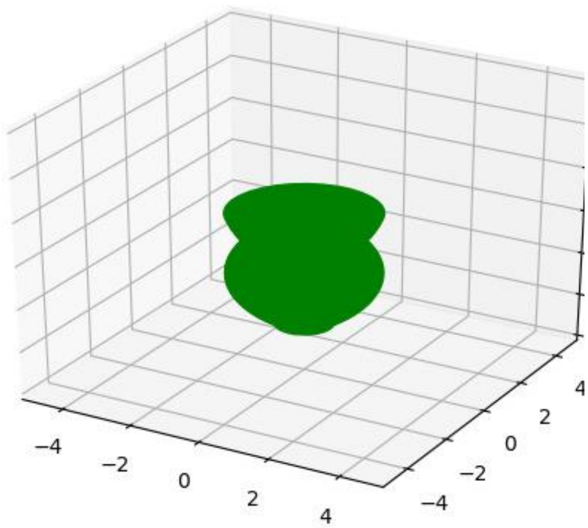

Figura 26: Gráfico Jarrón con curvas de nivel interpolado Bézier

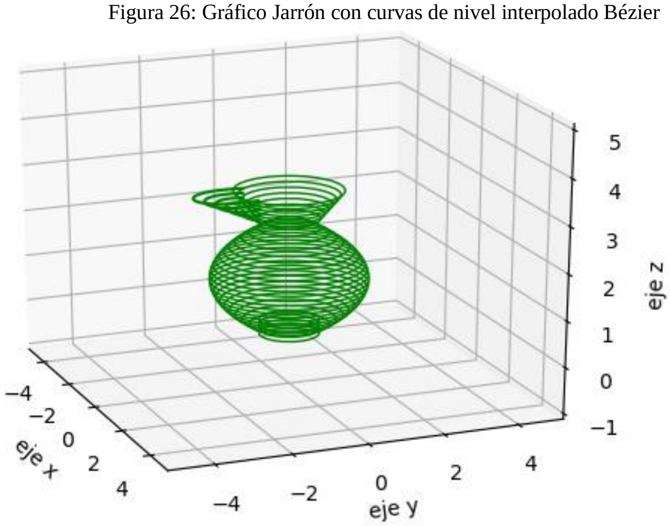

Figura 27: Gráfico Jarrón con curvas de nivel
y Pico Bézier

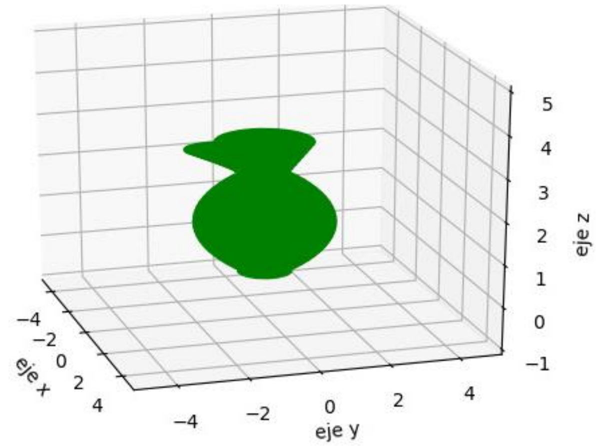

Figura 28: Gráfico Jarrón con curvas de nivel
y Pico Bézier

Version 2 Python

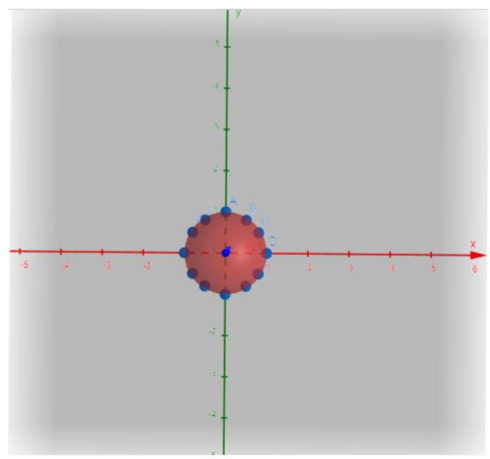

Figura 29: Gráfico puntos de control Jarrón Bézier GeoGebra

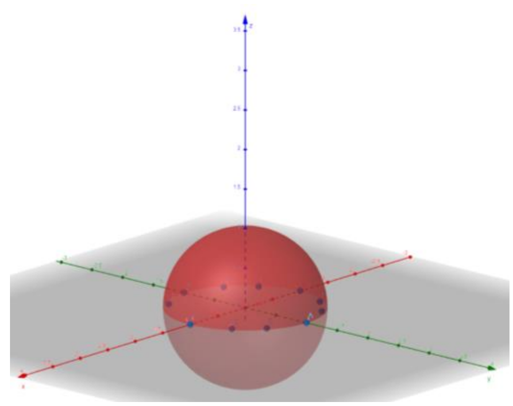

Figura 30: Gráfico 2, puntos de control Jarrón Bézier GeoGebr

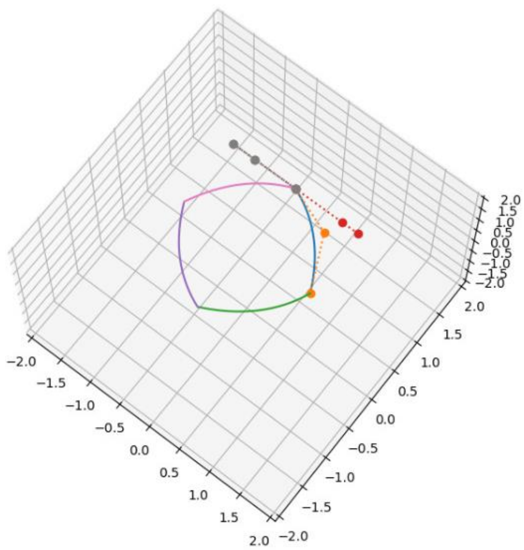

Figura 31: Gráfico puntos de control Jarrón Bézier Python

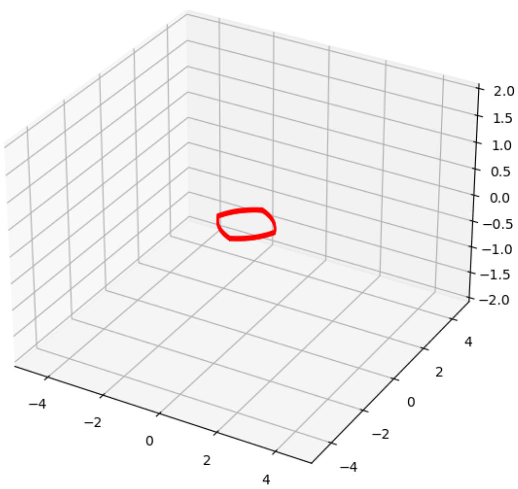

Figura 32: Gráfico cilindro Jarrón Bézier

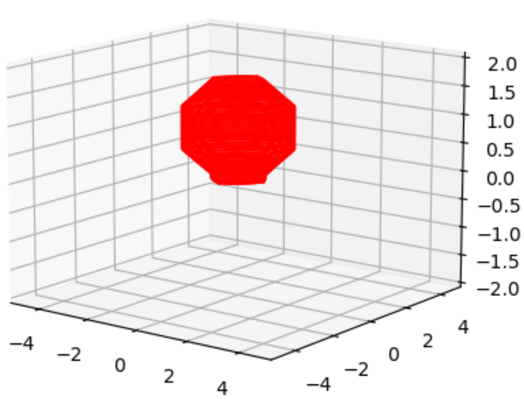

Figura 33: Gráfico esfera Jarrón Bézier

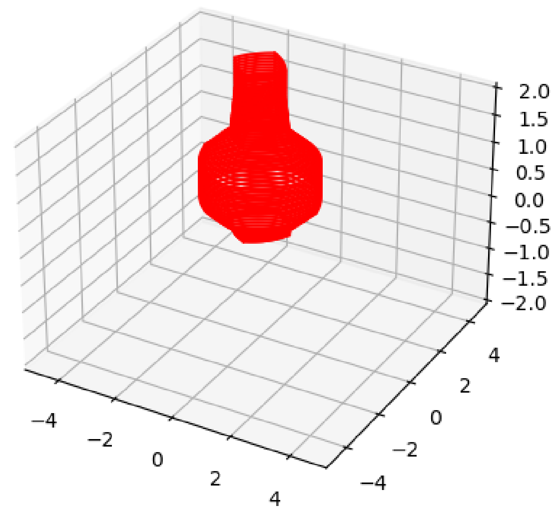

Figura 34: Gráfico Jarrón Bézier

Para determinar el área del jarrón se hizo uso de las fórmulas básicas para calcular el área del paraboloide, el cilindro, la esfera y el pico que lo conforman. Esta se calculó teniendo en cuenta la altura, el radio y el ancho del jarrón que se obtuvo gracias al modelado que se hizo en Blender:

$$\acute{A}rea\ del\ paraboloide: 0.5236\left(\frac{r}{h^2}\right) * \left((r^2+4h^2)\frac{3}{2}\right) - r^3$$

$$\acute{A}rea\ del\ cilindro: pi * r^2 h$$

$$\acute{A}rea\ de\ la\ esfera: 4r^2 * pi$$

$$\acute{A}rea\ del\ pico: \frac{1*1}{2} * 3$$

$$\acute{A}rea\ del\ Jarr\acute{o}n: \acute{a}rea\ pico + \acute{a}rea\ cilindro + \acute{a}rea\ esfera + \acute{a}rea\ paraboloide - (|\acute{a}rea\ cilindro - \acute{a}rea\ esfera|) - (|\acute{a}rea\ paaboloide - \acute{a}rea\ esfera|)$$

$$\acute{A}rea\ del\ Jarr\acute{o}n: 9.587213264652412\ m^2$$

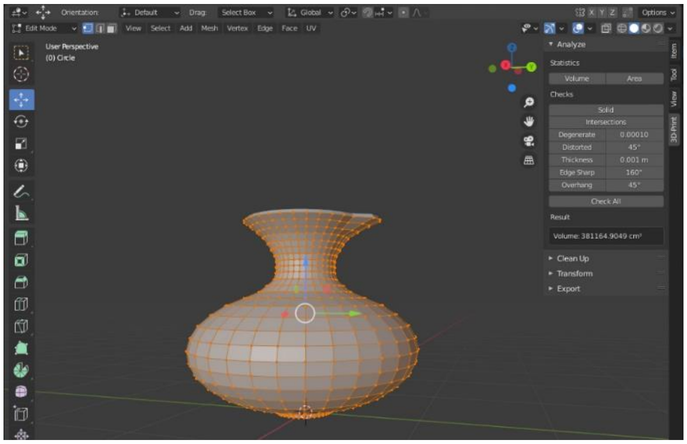

Figura 35: Volumen del Jarrón Modelado en Blender

Para determinar el volumen del jarrón se hizo uso de las fórmulas básicas para calcular el volumen del paraboloide, el cilindro y la esfera que lo conforman. Esta se calculó teniendo en cuenta la altura y el radio del jarrón que se obtuvo gracias al modelado que se hizo en Blender:

$$Volumen\ de\ la\ esfera: \frac{4}{3}r^3 * pi$$

$$Volumen\ del\ paraboloide: \frac{pi * r^2 h}{2}$$

$$Volumen\ del\ cilindro: (pi * r^2 h)h$$

$$Volumen\ del\ Jarrón: volumen\ cilindro + volumen\ esfera + volumen\ paraboloide - (|volumen\ cilindro - volumen\ esfera|) - (|volumen\ paraboloide - volumen\ esfera|)$$

$$Volumen\ del\ Jarrón: 2.5874026205603187\ \ m^3$$

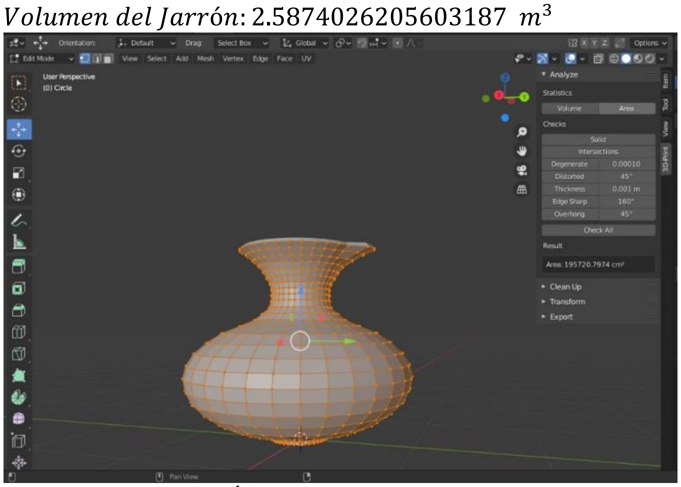

Figura 36: Área del Jarrón Modelado en Blender

Para determinar el error que tienen los cálculos realizados se llevó a cabo una comparación entre los resultados obtenidos anteriormente y el resultado que propone Blender, además es de tener en cuenta de que el área y volumen calculados experimentalmente son de una figura hueca, en cambio los cálculos obtenidos por medio de Blender son para un sólido en 3D.

Área Blender: 19.57207974 m^2
Volumen Blender: 0.3811649049 m^3

Error del Área:

$$E_a: \frac{19.57207974 - 9.587213264652412}{19.57207974}$$

E_a:0,51015869

Error del Volumen:

$$E_v: \frac{0.3811649049 - 2.5874026205603187}{0.3811649049}$$

E_v: |-5,78814494|

VII. CONCLUSIONES

Dado el reto descrito anteriormente se pudo llegar a algunas conclusiones como las siguientes:

- Al utilizar el método de modelación por medio de curvas de nivel se pudieron obtener gráficas mucho más suaves, en comparación con las obtenidas al usar el método de Bézier completamente para obtener la figura. Esto indica que para el caso de replicar la figura deseada se acerco mucho más la versión 1 del jarrón dadas las características de esta.
- Se intentó realizar la interpolación de la figura por otros métodos como el de Spline pero no se logró obtener una figura favorable como resultado, en cambio con el método de interpolación Bézier se obtuvieron dos versiones bastante acordes con lo que se requería con el reto documentado aquí.

VIII. ANEXOS

[1] Tablas_Puntos_Reto2.xlsx: Archivo Excel que contiene las tablas con algunos de los puntos iniciales de la jarra extraídos de la construcción en Blender , y también las tablas de los puntos de control de las dos versiones implementadas.

[2] retor_jarron_curvas_nivel.py: : Implementación en el lenguaje de programación Python de la solución del reto planteado, esta solución se dio por el método de Bézier para el pico de la figura y por curvas de nivel para el resto de esta, dentro del documento es llamada como versión 1.

[3] reto_jarron_bezier.py: Implementación en el lenguaje de programación Python de la solución del reto planteado, esta solución se dio completamente por el método de Bézier y dentro del documento es llamada como versión 2.

[4] esfera_4puntos.ggb: Archivo exportado de Geogebra el cual contiene una gráfica en 3D de la esfera usada como base para realizar la reconstrucción del jarrón para la versión 1, esta esfera se realizó con 4 puntos de control.

[5] esfera_3_puntosdecontrol.ggb: Archivo exportado de Geogebra el cual contiene una gráfica en 3D de la esfera usada como base para realizar la reconstrucción del jarrón para la versión 2, , esta esfera se realizó con 3 puntos de control.

[6]Pico_4puntoscontrol.ggb: Archivo exportado de Geogebra el cual contiene una gráfica en 3D de la esfera con los puntos de control usada como base para realizar la reconstrucción del pico del jarrón para la versión 1.

[7] Imagenes: Carpeta la cual contiene todos los pantallazos e imágenes generadas de las implementaciones y modelamiento para la solución del reto.

[8] Modelado Jarra en Blender: Dentro de esta carpeta se encuentran todos los archivos exportados desde el programa Blender , en los cuales se puede apreciar el proceso de modelado de la figura del jarrón .

[9] Vertices:Dentro de esta carpeta se encuentran los archivos txt extraídos de Blender, los cuales contienen los puntos que forman la figura modelada del Jarrón con distintas versiones entre reducciones en el número de puntos .

[10] Bezier.py: Implementación en el lenguaje de programación Python de la interpolación por el método de Bézier, este archivo fue obtenido de la referencia [4].

[11] puntosJarra.py: Implementación en el lenguaje de programación Python en la cual se logra graficar la figura a partir de los puntos obtenidos del modelado.

IX. REFERENCIAS

[1]Bourke, P. Bézier Curves. Disponible En: Http://Astronomy.Swin.Edu.Au/Pbourke/Curves/Bezier/

[2] Richard, L. Douglas, J. Numerical Analysis 9th Edition. Youngstown State University. 2010

[3] James, C. Multi-Variable Curve Interpolation, Acm, 1964.

[4] Torresjrjr/Bezier.Py: Create Bezier Curves In Python Https://Github.Com/Torresjrjr/Bezier.Py Accessed:2020-11-13

[5]Blender.Org - Home Of The Blender Project - Free And Open 3d Creation Software Https://Www.Blender.Org/ Accessed: 2020-11-13

[6] Curvas De Nivel Pedro H. Zambrano R, Departamento De Matemáticas, Universidad Nacional De Colombia, Bogotá, Disponible En: Https://Docplayer.Es/34213119-Funcion-De-Dos-Variables.Html