Tecnológico de Monterrey Campus Guadalajara

Matemáticas III

Integrantes:

Oscar Isaac Pelayo Alvarez A01632183

Juan Pablo Rivera Ruíz A01636350

Agustín Salvador Quintanar de la Mora A01636142

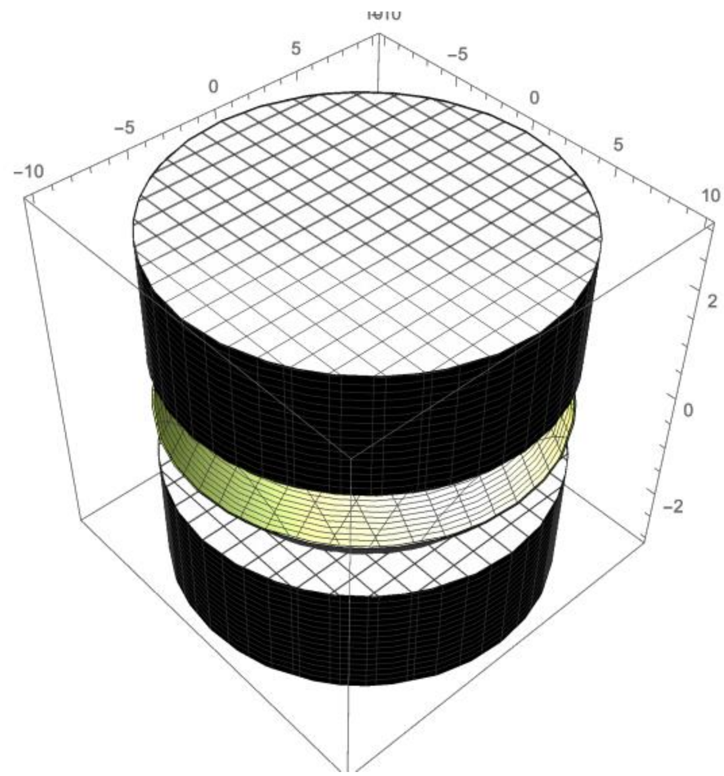
Proyecto: Modelado de un Galletero

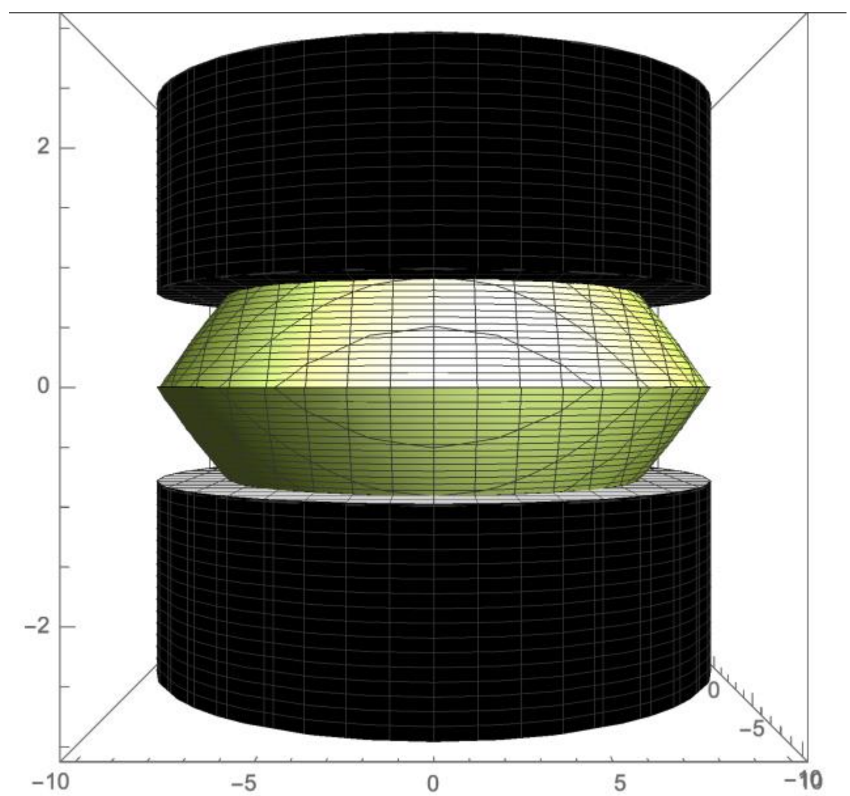
Etapas I-II

Modelado de un Galletero

En el presente proyecto se presentará el desarrollo y modelación de un galletero compuesto de superficies cuadráticas, de manera que, en su conjunto, se logre el objetivo de ser un diseño innovador, ergonómico y, además, capaz de soportar una capacidad mínima de 500 g. En adición a este prototipo, se modeló un contenedor del mismo galletero. El presente grupo de desarrolladores del proyecto, optó por hacer un modelo a escala de una galleta tipo “Oreo”, de manera que se pueda realizar una interacción donde se puedan tomar galletas, de una galleta, figuradamente.

El primer paso para poder modelar dicho galletero por medio de funciones cuadráticas fue determinar el volumen mínimo necesario para almacenar los 500 gramos de galletas, para lo cual se investigo el peso y volumen de una galleta para determinar cuantas galletas serian necesarias y el volumen necesario para contenerlas. Una vez determinadas las medidas, empezamos a modelar las ecuaciones mostradas debajo utilizando desigualdades para delimitar la superficie ocupada por cada de una de las funciones. Para obtener la función del paraboloide elíptico, se tuvieron que obtener sus trazas a través del conocimiento de 3 puntos, una vez conocidas las trazas, se obtuvo la ecuación de la superficie en el espacio. La parte media superior del galletero es ligeramente mas pequeña que la inferior de modo en que pueda embonar con la parte media inferior del galletero. Las ecuaciones y su representación en el espacio se muestran a continuación:


10
5
0
−5
−10
−5
0
5
10
2
0
−2

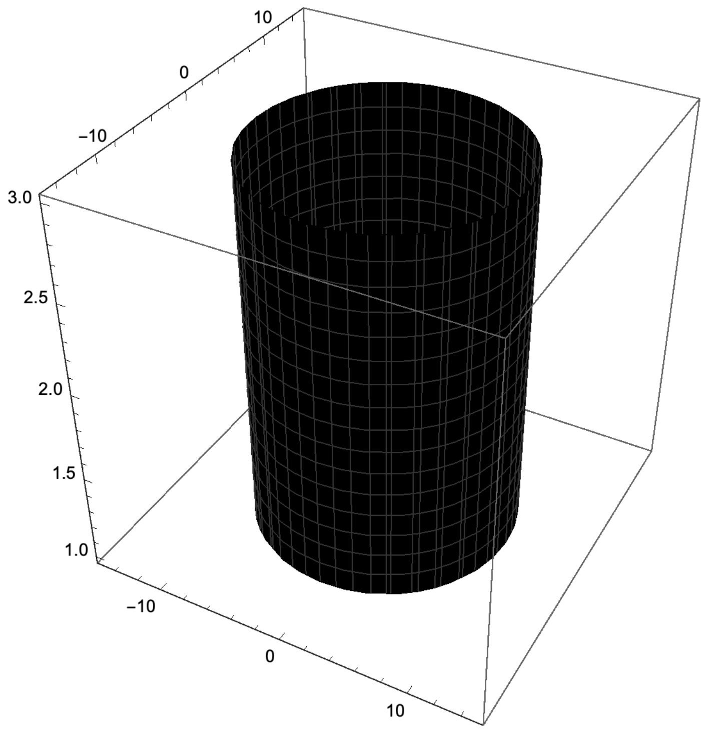

2
0
−2
−10
−5
0
5
−10
0
−5

En conclusión, el desarrollo del modelado de una galleta en 3D, presentó un nivel de reto considerado interesante por los desarrolladores del proyecto, de manera que pudieron expandir su interés por lo que es en sí, el modelado, además de, por el reto que fue presentado al tener que deducir una ecuación de una superficie cuadrática deseada y que cumpliera con las condiciones requeridas por el proyecto. De esta manera, se concluye en la relevancia de las funciones descritas, en la cotidianeidad del ser humano, donde a pesar de no hacer un detenimiento a observar y analizar estas funciones en el entorno, siempre están ahí presentes.

In[136]:= parteSup = ContourPlot3D[x ^ 2 + y ^ 2 == 100,
representación 3D de contornos

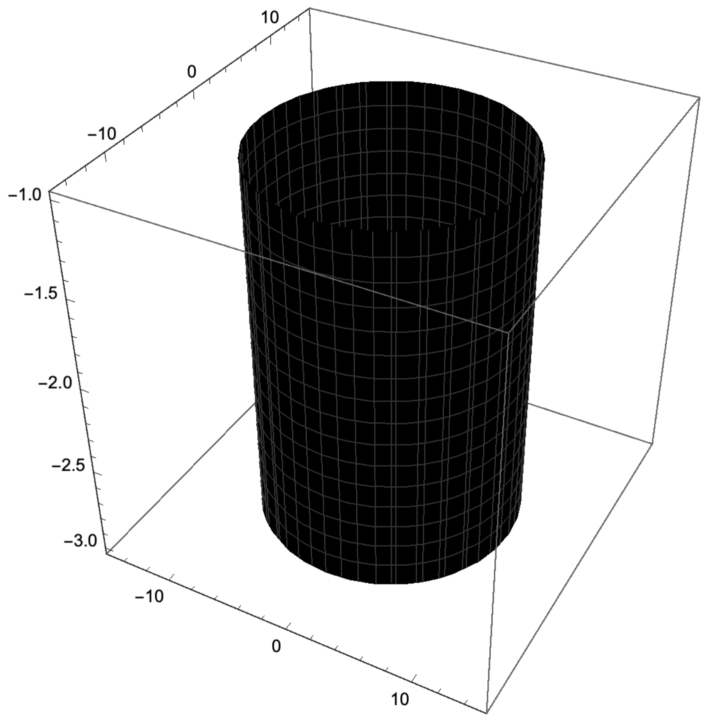
{x, -15, 15}, {y, -15, 15}, {z, 1, 3}, ColorFunction → GrayLevel]
función de color nivel de gris

Out[136]=



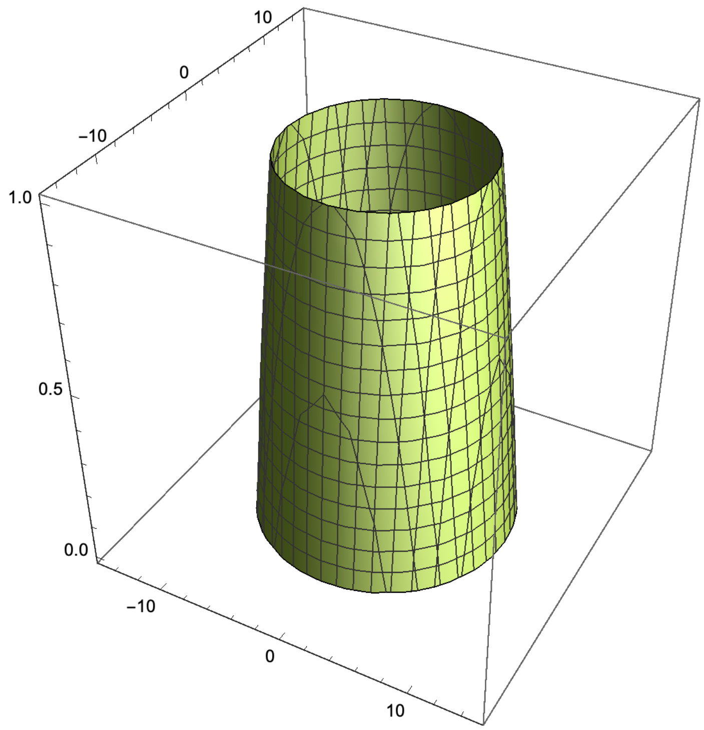
In[137]:=
```
parteInf = ContourPlot3D[x^2 + y^2 == 100,
  {x, -15, 15}, {y, -15, 15}, {z, -3, -1}, ColorFunction → GrayLevel]
```
representación 3D de contornos · función de color · nivel de gris

Out[137]=



In[138]:=
```
cremaAba = ContourPlot3D[-0.0239 x^2 - 0.0239 y^2 + 2.347 == z,
  {x, -15, 15}, {y, -15, 15}, {z, 0, 1}, ColorFunction → White]
```
representación 3D de contornos · función de color · blanco

Out[138]=

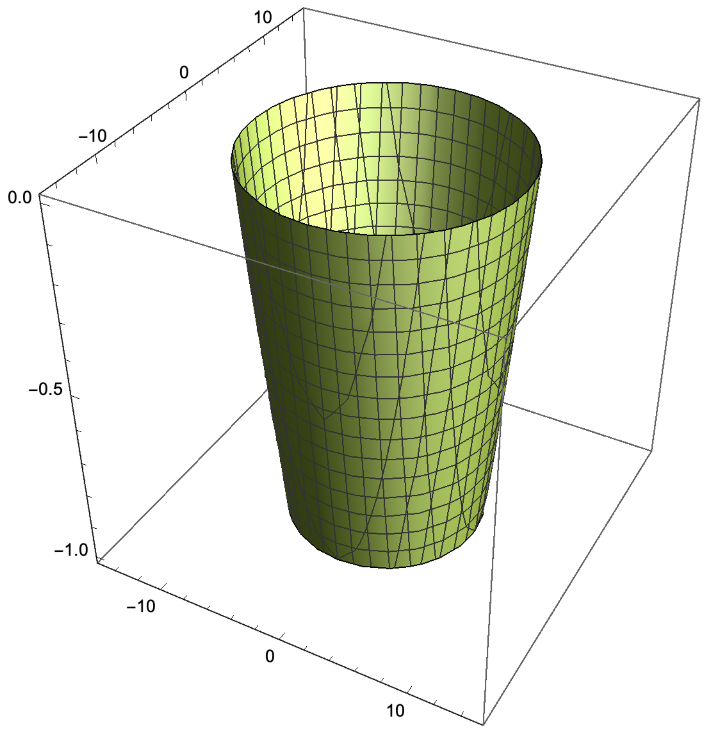


In[139]:=
```
cremaArr = ContourPlot3D[0.0228 x^2 + 0.0228 y^2 - 2.285 == z, {x, -15, 15},
  {y, -15, 15}, {z, -1, 0}, ColorFunction → RGBColor[255, 255, 255]]
```
representación 3D de contornos

función de color

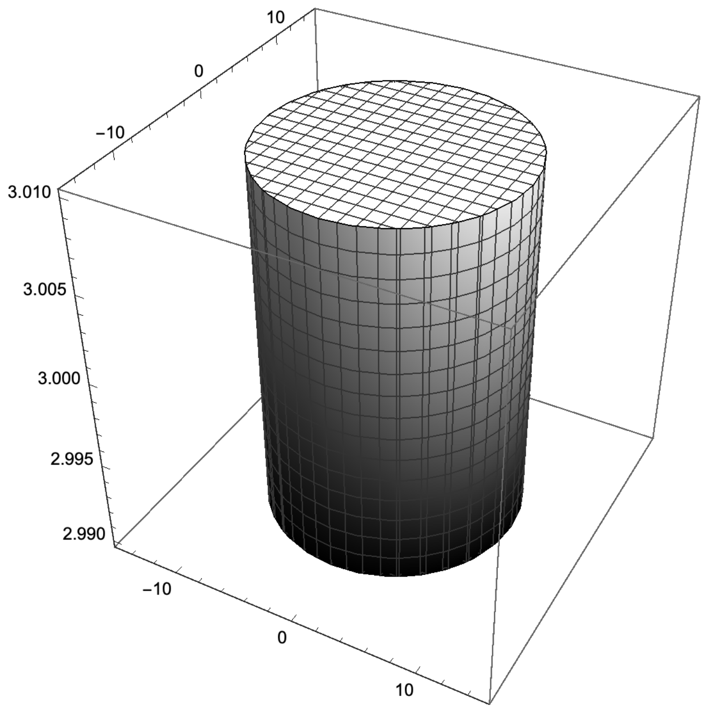
color RGB

Out[139]=



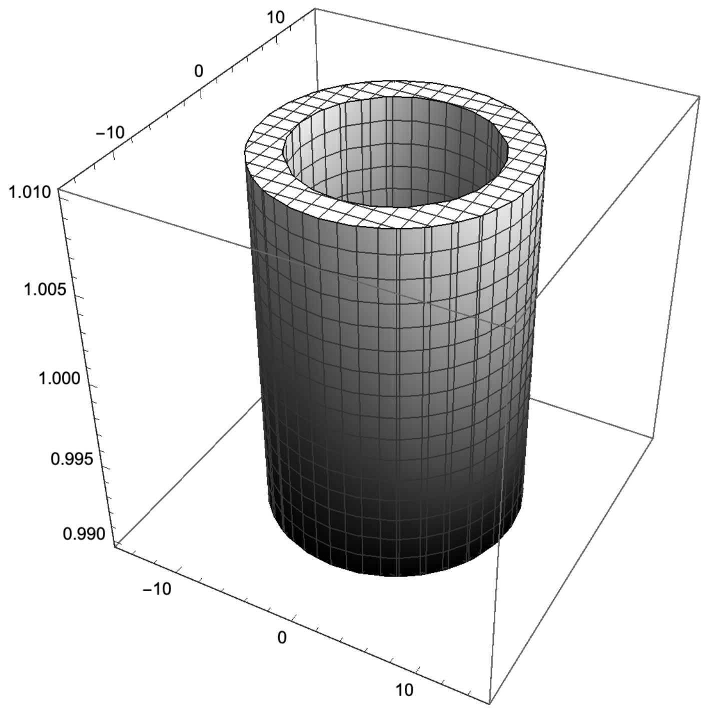
In[140]:=
```
tapaSup1 = RegionPlot3D[x^2 + y^2 ≤ 100, {x, -15, 15},
  {y, -15, 15}, {z, 2.99, 3.01}, ColorFunction → GrayLevel]
```
representación de región 3D · función de color · nivel de gris

Out[140]=



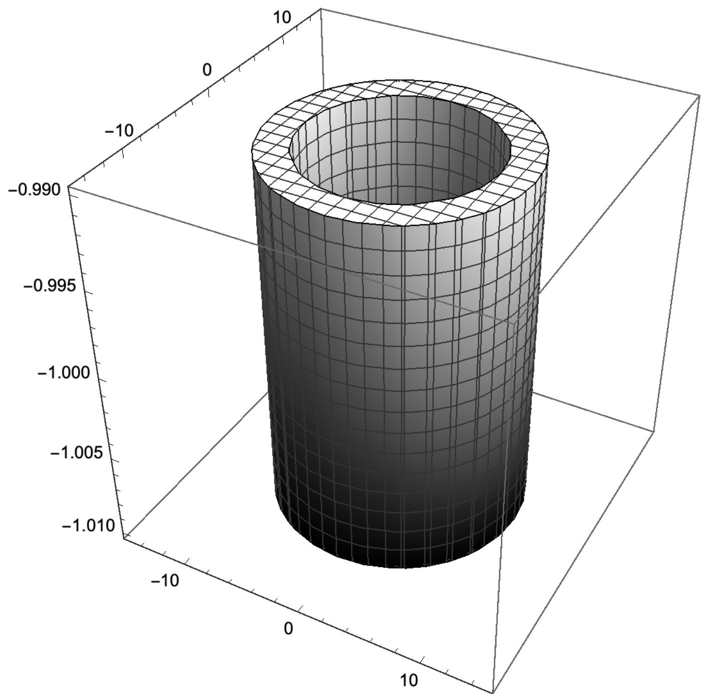
In[141]:=
```
tapaSup2 = RegionPlot3D[56.25 <= x^2 + y^2 ≤ 100, {x, -15, 15},
  {y, -15, 15}, {z, 0.99, 1.01}, ColorFunction → GrayLevel]
```
representación de región 3D · función de color · nivel de gris

Out[141]=



```
In[142]:= tapaInf2 = RegionPlot3D[56.25 ≤ x^2 + y^2 ≤ 100, {x, -15, 15},
                     representación de región 3D
  {y, -15, 15}, {z, -1.01, -0.99}, ColorFunction → GrayLevel]
                                   función de color  nivel de gris
```

Out[142]=



In[143]:=
```
Show[parteInf, parteSup, cremaAba, cremaArr ,
  tapaSup1, tapaSup2, tapaInf2, PlotRange -> All]
```
muestra — rango de repre⋯ — todo

Out[143]=