Manual Del Programador

Leeynyker Montaño
20181020140

Wilmer Ramos
20181020171

Andrés Baquero
20181020124

Investigación De Operaciones I
Universidad Distrital Francisco José de Caldas
2020-1

Requerimientos Básicos

Para el desarrollo del software se utilizó python y algunas librerías de este lenguaje los cuales son:

- Python 3.7.1
- Tkinter
- Math
- Numpy
- Matplotlib

Tkinter: Nos ayuda en la interfaz que hay entre el usuario y el programa

Math: Esta biblioteca importa todas las operaciones aritméticas y funciones matemáticas

Numpy: Nos ayuda con el cálculo de operaciones a nivel matricial

Matplotlib: Esta biblioteca ayuda con la graficación de las restricciones

Desarrollo

Empezamos por importar cada librería

```
#Nos ayuda en la interfaz que hay entre el usuario y el programa
from tkinter import *
import tkinter as tk

#Esta biblioteca importa todas las operaciones aritmeticas y funciones matematicas
import math

#nos ayuda con el calculo de operaciones a nivel matricial
import numpy as np

#Esta biblioteca ayuda con la graficación de las restricciones
from matplotlib import pyplot as plt
```

Definimos una lista de los colores que se utilizaran para pintar cada restricción que se ingrese

```
14  colores = ('C0', 'C1', 'C2', 'C3', 'C4', 'C5', 'C6')
```

Para el desarrollo del software utilizamos varias clases las cuales procederemos a explicar una a una

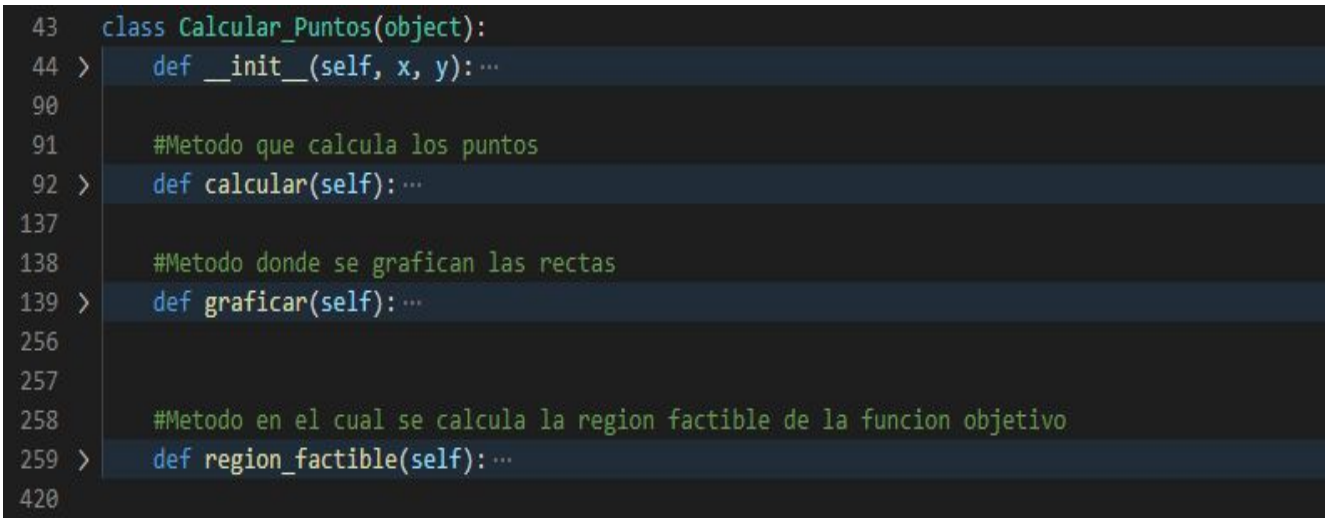
Clase punto: Describe cada punto con sus coordenadas en X y en Y

```
15  #Clase para describir cada punto
16  class punto:
17    def __init__(self,x,y):
18      self.px = x
19      self.py = y
20    def __str__(self):
21      return "("+str(self.px)+","+str(self.py)+")"
```

Clase Línea: Hace cada recta y halla cada punto de intersección entre estas

```
24  class linea:
25    def __init__(self, p0, p1):
26      self.p0 = p0
27      self.p1 = p1
28      self.A = p1.px - p0.px
29      self.B = p1.py - p0.py
30      self.C = p1.px*p0.py - p0.px*p1.py
31    def intersecta(self, otro):
32      det = self.A*otro.B - otro.A*self.B
33      cx = otro.A*self.C-self.A*otro.C
34      cy = otro.B*self.C-self.B*otro.C
35      if det != 0:
36          cordenadas=[cx/det,cy/det]
37      else:
38          cordenadas=[0,0]
39      return (cordenadas) #Devuelve las cordenadas de los puntos de interseccion entre las rectas
```

Calcular Puntos: Es nuestra clase principal, en la cual hallamos los puntos, y se realiza la gráfica de cada función ingresada, así como se halla la región factible



Inecuaciones: Se crean para los campos para las inecuaciones y se arman las mismas