UNIVERSIDAD DEL ATLÁNTICO

Lainer Ortiz

Sebastián Bolaños

Aylin Rodríguez

Gisselle Herrera

Gresia Martínez

Leisy Sánchez

Angelly Cervantes Orozco

Amy Mejía

Elly Pérez

Juan Camilo Pérez

Software sobre cálculo de emisiones en tanques de hidrocarburos

PhD. Pedro Orozco

Barranquilla – Atlántico

2021

GUÍA DE INSTALACIÓN Y USUARIO

El programa de Emisiones y Tanques realizado en Python, utiliza librerías para su correcto funcionamiento y ejecución. Inicialmente se hará la salvedad de que el programa se llevó a cabo en Visual Studio Code, un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS.

Inicialmente al abrir nuestro programa como se muestra en la Figura 1, es necesario dirigirse a la barra de menú en la pestaña terminal y seleccionar la opción new terminal.

Texto

Descripción generada automáticamente

**Figura 1.** *Interfaz de VScode*

Posteriormente en nuestra ventana de comando escribimos el comando: pip install reportlab, presionamos enter y esperamos que se descargue la librería, como se muestra en la Figura 2:

Rectángulo

Descripción generada automáticamente con confianza media

**Figura 2.** *Instalación de librerías de Python*

Este paso también se puede llevar a cabo en la ventana de comando “CMD” de su ordenador.

Una vez hecho los pasos anteriores procedemos a ejecutar el programa y en la ventana de comando nos pide introducir algunos datos (Si alguno de estos datos es decimal, debe utilizarse el punto como decimal (.)) como se muestra en la Figura 3:

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente**Figura 3.** *Funcionamiento del programa*

En la carpeta donde esté alojado nuestro programa, debe guardarse automáticamente un archivo en formato en pdf como el mostrado en el Excel de Emisiones y Tanques de Ecopetrol, con un resumen de los datos obtenidos en nuestro programa, como se observa en la Figura 4.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Figura 4.** *Documento en pdf que corresponde al nombre de Reporte\_Ecopetrol.pdf*

El documento de pdf que se genera tendrá el siguiente esquema físico que se muestra a continuación en la Figura 5.

Texto

Descripción generada automáticamente con confianza media

**Figura 5.** *Esquema físico de Reporte\_Ecopetrol.pdf*

El código utilizado para el funcionamiento de este programa fue el siguiente:

import math

from math import log

from datetime import datetime, timedelta

# DV = Densidad de vapor

# DC = Diámetro del cilindro del tanque vertical (ft)

print("\nHIDROCARBUROS\n")

#Fecha inicial

def comprobar\_fechainicial(text):

    try:

        datetime.datetime.strptime(text, '%Y/%m/%d')

    except:

        return "El formato debe ser  YYYY/MM/DD"

    return datetime.datetime.strptime(text, '%Y/%m/%d')

fechainicial = input("Ingresa una fecha inicial en formato YYYY/MM/DD: ")

comprobar\_fechainicial(fechainicial)

#Fecha final

def comprobar\_fechafinal(text):

    try:

        datetime.datetime.strptime(text, '%Y/%m/%d')

    except:

        return "El formato debe ser  YYYY/MM/DD"

    return datetime.datetime.strptime(text, '%Y/%m/%d')

fechafinal = input("Ingresa una fecha final en formato YYYY/MM/DD: ")

comprobar\_fechafinal(fechafinal)

# Datos conocidos.

TAGTANQUE=input("Ingrese el TAG DEL TANQUE= ")

PERIODO=input("Ingrese el PERIODO (Si no aplica entonces: N/A)= ")

ESTACION=input("Ingrese el ESTACION (Si no aplica entonces: N/A)= ")

DV = float(input("Diámetro cilíndrico de tanque vertical (ft)= "))

VD = float(input("Densidad del vapor (lb/ft^3) = "))

HS = float(input("Altura de la carcasa (ft) = "))

HLX = float(input("Ingrese altura máxima del líquido (ft) = "))

HLN = float(input("Ingrese la altura mínima del líquido (ft) = "))

HL = ""

print("¿Posee el valor de la altura promedio del líquido? (yes/no):")

while HL != "yes" and HL != "no":

    HL = input().lower()

if HL == "yes":

    HL = float(input("HL (ft) = "))

else:

    HL = (HLX + HLN) / 2

    print("Altura promedio del líquido (ft) = ", HL)

print("Tipo de techo:")

print("1. Cubierta plana.")

print("2. Techo cónico.")

print("3. Techo en forma de domo.")

techo = 0

while techo != 1 and techo != 2 and techo != 3:

    techo = int(input())

HRO = 0

if techo == "1":

    HRO = 0

elif techo == "2":

    pendiente = float(input("Pendiente del techo (0 si el valor es desconocido) = "))

    if pendiente > 0:

        HRO = (pendiente \* DV / 2) / 3

    elif pendiente == 0:

        HRO = DV / 96

else:

    HR = float(input("Altura del techo (0 si el valor es desconocido) = "))

    if HR > 0:

        HRO = (HR / 2) + (2 \* HR \*\* 3) / (3 \* DV \*\* 2)

    elif HR == 0:

        HRO = 0.0686 \* DV

print("Corte del techo (ft) = ", HRO,)

# HVO = Interrupción del espacio de vapor

HVO = HS + HL + HRO

print("Interrupción del espacio de vapor = {} ft".format(HVO))

color = ""

print("Color de la pintura (blanco/negro): ")

while color != "blanco" and color != "negro":

    color = input().lower()

calidad = ""

print("Calidad de la pintura (buena/mala): ")

while calidad != "buena" and calidad != "mala":

    calidad = input().lower()

lambd = 0.0

if color == "blanco" and calidad == "buena":

    lambd = 0.17

elif color == "blanco" and calidad == "mala":

    lambd = 0.34

elif color == "negro" and calidad == "buena":

    lambd = 0.97

else:

    lambd = 0.97

MAX = float(input("Temperatura máxima (°C) = "))

MIN = float(input("Temperatura mínima (°C) = "))

MAX = (MAX \* 1.8 + 32) + 459.67

MIN = (MIN \* 1.8 + 32) + 459.67

TAA = (MAX + MIN) / 2

RVP = float(input("Presión de vapor de Stock Reid (RVP)(psi)= "))

hidrocarburo = ""

print("¿Crudo o Refinado? (crudo = c; refinado = r):")

while hidrocarburo != "c" and hidrocarburo != "r":

    hidrocarburo = input().lower()

KC = 0

if hidrocarburo == "r":

    S = float(input("S = "))

    A = 15.64 - 1.854 \* math.sqrt(S) - \

        (0.8742-0.3280 \* math.sqrt(S)) \* log(RVP)

    B = 8742 - 1042 \* math.sqrt(S) - (1049 - 179.4 \* math.sqrt(S)) \* log(RVP)

    KC = 1

    Producto='REFINADO'

    print("Producto=", Producto)

else:

    A = 12.82 - 0.972 \* log(RVP)

    B = 7261 - 1216 \* log(RVP)

    KC = 0.75

    Producto='CRUDO'

    print("Producto=", Producto)

# KS = Factor de saturación de vapor ventilado

radia = float(input("Radiación solar diaria (cal/cm^2\*día) = "))

I = radia / 0.27125

TLA = round(TAA + 0.56 \* (6 \* lambd - 1) + 0.0079 \* lambd \* I)

PVA = float(math.exp(A - (B / TLA)))

KS = 1 / (1 + 0.053 \* PVA \* HVO)

print("Temperatura media diaria de la superficie del líquido (°R) = ", TLA,)

print("La presión de vapor real de stock a la temperatura media de la superficie del líquidol (psia) = ", PVA)

print("Factor de producto = ", KC)

print("Factor de saturación de vapor ventilado = ", KS)

# Factor de expansión del espacio de vapor "KE, WV y LS"

PBX = float(input(

    "Presión máxima del respiradero en psig (0 si el valor es desconocido) = "))

if PBX == 0:

    PBX = 0.03

PBN = float(input("Presión mínima del respiradero en psig (0 si el valor es desconocido) = "))

if PBN == 0:

    PBN = -0.03

DPB = PBX - PBN

print("Rango de presión de vapor diario = {} psi".format(DPB))

KB = 1

print("Factor de corrección del ajuste de la ventilación = ", KB)

print("Tipo de tanque:")

print("1. Tanque aéreo no aislado.")

print("2. Tanque subterraneo no aislado.")

print("3. Tanque subterraneo o aéreo completamente aíslado.")

LS = 0

while LS != 1 and LS != 2 and LS != 3:

    LS = int(input())

DTV = 0

KE = 0

TBX=0

TBN=0

if LS == 1:

    if PVA <= 0.1 and DPB <= 0.063:

        DTV = (0.7 \* (MAX - MIN) + (0.02 \* I \* lambd))

        KE = 0.0018 \* DTV

    elif PVA > 0.1 and DPB > 0.063:

        DTV = (0.7 \* (MAX - MIN) + (0.02 \* I \* lambd))

        TBX = TLA + (0.25 \* DTV)

        TBN = TLA - (0.25 \* DTV)

elif LS == 2:

    LS = 0

    WV=0

    TBX = float(input("Temperatura máxima del líquido a granel (°C) = "))

    TBN = float(input("Temperatura mínima del líquido a granel (°C) = "))

    DTV = (TBX - TBN)

else:

    TBX = float(input("Temperatura máxima del líquido a granel (°C) = "))

    TBN = float(input("Temperatura mínima del líquido a granel (°C) = "))

    DTV = (TBX - TBN)

print("Rango de temperatura de vapor diario °R = ",DTV)

print("Temperatura máxima del líquido a granelTemperatura máxima del líquido a granel (°C)=",TBX)

print("Temperatura mínima del líquido a granelTemperatura mínima del líquido a granel (°C)=",TBN)

print("Factor de expansión del espacio de vaporFactor de expansión del espacio de vapor",KE)

if LS == 1 or LS == 3:

    PVX = float(math.exp(A - (B / TBX)))

    PVN = float(math.exp(A - (B / TBN)))

    DPV = PVX - PVN

    PA = 14.7

    KE = (DTV / TLA) + ((DPV - DPB)/(PA - PVA))

    if KE < 0:

        KE = 0

    elif KE > 1:

        KE = 1

    if techo == 1:

        WV = 0

    elif techo == 2:

        WV = (VD \* PVA) / (10.731 \* TLA)

    else:

        WV = (VD \* PVA) / (10.731 \* TLA)

    LS = 365 \* ((math.pi \* DV \*\* 2) / 4) \* HVO \* KS \* KE \* WV

print("Perdida permanente, LS = {} lb/año".format(LS))

# Rendimiento de la pérdida neta de trabajo (VQ)

print("¿Posee el valor de la suma anual de los aumentos en el nivel de líquido? (yes/no): ")

HQ = ""

while HQ != "yes" and HQ != "no":

    HQ = input().lower()

# Índice de rotación de existencias (N)

if HQ == "yes":

    HQ = float(

        input("Suma anual de los aumentos en el nivel de líquido HQ (ft/años) = "))

    N = HQ / (HLX - HLN)

    VQ = -HQ \* (math.pi \* (DV \*\* 2) / 4)

else:

    Q = float(input("Ingrese el rendimiento de stock (bbl/año) = "))

    VQ = 5.614 \* Q

    r= (math.pi \* DV \*\* 2)\*(HLX - HLN) / 4

    N = (VQ) / r

print("Rendimiento de la pérdida de trabajo neto (ft^3/año)= ", VQ)

# Factor de rotación (KN)

if N <= 36:

    KN = 1

else:

    KN = (180 + N) / (6 \* N)

# PERDIDA LABORAL (LW) "Lb/año"

LW = VQ \* KN \* KC \* KB \* WV

print("Pérdida laboral (Lw) = {} lb/año".format(LW))

LT = LS + LW

print("Pérdida total (Lt) = {} lb/año".format(LT))

from reportlab.lib.pagesizes import A4

from reportlab.pdfgen import canvas

w, h = A4

c = canvas.Canvas("Reporte\_Ecopetrol.pdf", pagesize=A4)

#Línea de arriba logo ecopetrol

x = 30

y = h - 20

c.line(x, y, x + 540, y)

#Título formato

text = c.beginText(170, h - 35)

text.setFont("Times-Roman", 9)

text.textLines("FORMATO PARA CÁLCULO DE EMISIONES EN TANQUES DE ALMACENAMIENTO")

c.drawText(text)

#Línea de abajo título formato

x = 150

y = h - 38

c.line(x, y, x + 420, y)

#Título vicepresidencia

text = c.beginText(155, h - 51)

text.setFont("Times-Roman", 9)

text.textLines("VICEPRESIDENCIA DE INNOVACIÓN Y TECNOLOGÍA CORPORATIVO DE NORMAS Y ESTÁNDARES")

c.drawText(text)

#Línea de abajo título vicepresidencia

x = 150

y = h - 54

c.line(x, y, x + 420, y)

#Título Código

text = c.beginText(152, h - 66)

text.setFont("Times-Roman", 8)

text.textLines("CÓDIGO CNE ECP-VIN-P-MBC-FT-038")

c.drawText(text)

#Línea vertical 1 debajo del título de vicepresidencia

x = 290

y = h-70

c.line(x, y, x, y + 15)

#Título Elaborado

text = c.beginText(312, h - 66)

text.setFont("Times-Roman", 9)

text.textLines("ELABORADO 16/05/2014")

c.drawText(text)

#Línea vertical 2 debajo del título de vicepresidencia

x = 430

y = h-70

c.line(x, y, x, y + 15)

#Título Versión

text = c.beginText(470, h - 66)

text.setFont("Times-Roman", 9)

text.textLines("VERSIÓN: 1")

c.drawText(text)

#Línea vertical al lado de ecopetrol

x = 150

y = h-70

c.line(x, y, x, y + 50)

#Imagen logo de ecopetrol

c.drawImage("logoecopetrol.jpg", 40, h - 70, width=100, height=40)

#Línea de abajo logo ecopetrol

x = 30

y = h-70

c.line(x, y, x + 540, y)

#Línea de abajo de la línea de debajo del logo ecopetrol

x = 30

y = h-72

c.line(x, y, x + 540, y)

#Línea de arriba de las etiquetas del tanque y la fecha

x = 30

y = h-80

c.line(x, y, x + 540, y)

#Título TAG TANQUE

text = c.beginText(40, h - 105)

text.setFont("Times-Roman", 9)

text.textLines("TAG TANQUE:")

c.drawText(text)

#Rectángulo TAG TANQUE

x = 105

y = h - 30

c.rect(x, h - 113, 60, 20)

#Texto dentro de TAG TANQUE

text = c.beginText(115, h - 105)

text.setFont("Times-Roman", 9)

text.textLines(str(TAGTANQUE))

c.drawText(text)

#Título PRODUCTO

text = c.beginText(170, h - 105)

text.setFont("Times-Roman", 9)

text.textLines("PRODUCTO:")

c.drawText(text)

#Rectángulo PRODUCTO

x = 230

y = h - 30

c.rect(x, h - 113, 60, 20)

#Texto dentro de RECTÁNGULO DE PRODUCTO

text = c.beginText(240, h - 105)

text.setFont("Times-Roman", 9)

text.textLines(str(Producto))

c.drawText(text)

#Título ESTACIÓN

text = c.beginText(170, h - 132)

text.setFont("Times-Roman", 9)

text.textLines("ESTACIÓN:")

c.drawText(text)

#Rectángulo ESTACIÓN

x = 230

y = h - 30

c.rect(x, h - 140, 60, 20)

#Texto dentro de RECTÁNGULO DE ESTACIÓN

text = c.beginText(240, h - 134)

text.setFont("Times-Roman", 9)

text.textLines(str(ESTACION))

c.drawText(text)

#Título FECHA INICIAL

text = c.beginText(305, h - 105)

text.setFont("Times-Roman", 9)

text.textLines("FECHA INICIAL:")

c.drawText(text)

#Rectángulo FECHA INICIAL

x = 385

y = h - 30

c.rect(x, h - 113, 60, 20)

#Texto dentro de FECHA INICIAL

text = c.beginText(400, h - 105)

text.setFont("Times-Roman", 9)

text.textLines(str(fechainicial))

c.drawText(text)

#Título FECHA FINAL

text = c.beginText(305, h - 135)

text.setFont("Times-Roman", 9)

text.textLines("FECHA FINAL:")

c.drawText(text)

#Rectángulo FECHA FINAL

x = 385

y = h - 30

c.rect(x, h - 140, 60, 20)

#Texto dentro de FECHA FINAL

text = c.beginText(400, h - 134)

text.setFont("Times-Roman", 9)

text.textLines(str(fechafinal))

c.drawText(text)

#Título PERÍODO

text = c.beginText(460, h - 135)

text.setFont("Times-Roman", 9)

text.textLines("PERÍODO:")

c.drawText(text)

#Rectángulo PERÍODO

x = 510

y = h - 30

c.rect(x, h - 140, 60, 20)

#Texto dentro de PERÍODO

text = c.beginText(520, h - 134)

text.setFont("Times-Roman", 9)

text.textLines(str(PERIODO))

c.drawText(text)

#Línea 1 de abajo de las etiquetas del tanque y la fecha

x = 30

y = h-150

c.line(x, y, x + 250, y)

#Línea 1 vertical de abajo de las etiquetas del tanque y la fecha

x = 30

y = h-170

c.line(x, y, x, y+20)

#Línea 2 de abajo de las etiquetas del tanque y la fecha

x = 300

y = h-150

c.line(x, y, x + 270, y)

#Línea 2 vertical de abajo de las etiquetas del tanque y la fecha

x = 280

y = h-170

c.line(x, y, x, y+20)

#Título DATOS DEL TANQUE

text = c.beginText(120, h - 162)

text.setFont("Times-Roman", 9)

text.textLines("DATOS DEL TANQUE")

c.drawText(text)

#Título DATOS DEL LÍQUIDO

text = c.beginText(320, h - 162)

text.setFont("Times-Roman", 9)

text.textLines("DATOS DEL LÍQUIDO ALMACENADO EN EL TANQUE")

c.drawText(text)

#Línea 1 de abajo de DATOS DEL LÍQUIDO

x = 30

y = h-170

c.line(x, y, x + 250, y)

#Línea 3 vertical de abajo de las etiquetas del tanque y la fecha

x = 300

y = h-170

c.line(x, y, x, y+20)

#Línea 2 de abajo de DATOS DEL LÍQUIDO

x = 300

y = h-170

c.line(x, y, x + 270, y)

#Línea 4 vertical de abajo de las etiquetas del tanque y la fecha

x = 570

y = h-170

c.line(x, y, x, y+20)

#Rectángulo DATOS DEL LÍQUIDO

x = 300

y = h - 300

c.rect(x, h - 320, 270, 170)

#Subtítulos del rectángulo LÍQUIDO

c.drawString(310, h - 220, "Densidad del vapor (lb/ft^3) =")

c.drawString(310, h - 240, "Presión de vapor de Stock Reid (RVP)(psi) =")

c.drawString(310, h - 260, "Temperatura media diaria de la superficie del líquido (°R) =")

#Resultados del rectángulo DATOS DEL LÍQUIDO

#DV

text = c.beginText(425, h - 220)

text.setFont("Times-Roman", 9)

text.textLines(str(VD))

c.drawText(text)

#RVP

text = c.beginText(480, h - 240)

text.setFont("Times-Roman", 9)

text.textLines(str(RVP))

c.drawText(text)

#TLA

text = c.beginText(530, h - 260)

text.setFont("Times-Roman", 9)

text.textLines(str(TLA))

c.drawText(text)

#Rectángulo DATOS DEL TANQUE

x = 30

y = h - 300

c.rect(x, h - 320, 250, 170)

#Subtítulos del rectángulo DATOS DEL TANQUE

c.drawString(33, h - 200, "Diámetro del tanque (ft) =")

c.drawString(33, h - 220, "Altura de la carcasa (ft) =")

c.drawString(33, h - 240, "Ingrese altura máxima del líquido (ft) =")

c.drawString(33, h - 260, "Ingrese altura mínima del líquido (ft) =")

c.drawString(33, h - 280, "Color de la pintura (blanco/negro) =")

c.drawString(33, h - 300, "Calidad de la pintura (buena/mala) =")

#Resultados del rectángulo DATOS DEL TANQUE

#DV

text = c.beginText(130, h - 200)

text.setFont("Times-Roman", 9)

text.textLines(str(DV))

c.drawText(text)

#HS

text = c.beginText(130, h - 220)

text.setFont("Times-Roman", 9)

text.textLines(str(HS))

c.drawText(text)

#HLX

text = c.beginText(180, h - 240)

text.setFont("Times-Roman", 9)

text.textLines(str(HLX))

c.drawText(text)

#HLN

text = c.beginText(180, h - 260)

text.setFont("Times-Roman", 9)

text.textLines(str(HLN))

c.drawText(text)

#color

text = c.beginText(170, h - 280)

text.setFont("Times-Roman", 9)

text.textLines(str(color))

c.drawText(text)

#calidad

text = c.beginText(170, h - 300)

text.setFont("Times-Roman", 9)

text.textLines(str(calidad))

c.drawText(text)

#Rectángulo EMISIONES TANQUE DE TECHO FIJO

x = 150

y = h - 500

c.rect(x, h - 520, 250, 140)

#Rectángulo título EMISIONES TANQUE DE TECHO FIJO

x = 150

y = h - 330

c.rect(x, h - 370, 250, 20)

#Subítulos del rectángulo EMISIONES TANQUE DE TECHO FIJO

c.drawString(160, h - 450, "Pérdida laboral (Lw) (lb/año) =")

c.drawString(160, h - 470, "Pérdida total (LT) (lb/año) =")

c.drawString(160, h - 430, "Pérdida permanente (LS) (lb/año) =")

#Resultados del rectángulo DATOS DEL TANQUE

#LS

text = c.beginText(290, h - 430)

text.setFont("Times-Roman", 9)

text.textLines(str(LS))

c.drawText(text)

#LW

text = c.beginText(280, h - 450)

text.setFont("Times-Roman", 9)

text.textLines(str(LW))

c.drawText(text)

#Lt

text = c.beginText(270, h - 470)

text.setFont("Times-Roman", 9)

text.textLines(str(LT))

c.drawText(text)

#Título EMISIONES TANQUE DE TECHO FIJO

text = c.beginText(190, h - 365)

text.setFont("Times-Roman", 9)

text.textLines("EMISIONES DE TANQUE DE TECHO FIJO")

c.drawText(text)

#Título DILIGENCIADO POR:

text = c.beginText(30, h - 700)

text.setFont("Times-Roman", 9)

text.textLines("Diligenciado por:")

c.drawText(text)

#Línea de Nombre

x = 30

y = h - 750

c.line(x, y, x + 250, y)

#Texto de NOMBRE

text = c.beginText(120, h - 800)

text.setFont("Times-Roman", 9)

text.textLines("NOMBRE")

c.drawText(text)

#Línea de Firma

x = 300

y = h - 750

c.line(x, y, x + 250, y)

#Texto de FIRMA

text = c.beginText(400, h - 800)

text.setFont("Times-Roman", 9)

text.textLines("FIRMA - REGISTRO")

c.drawText(text)

c.showPage()

#Guardar pdf

c.save()