# UNIVERSIDAD POLITÉCNICA SALESIANA El VECINO - CUENCA

Estudiante: Gustavo Guallpa

Profesor: Ing. Diego Quisi

Asignatura: Sistemas Expertos

Tema: Informe de un Sistema Basado en Casos de Análisis de Vinos.

#### 1.CODIGO

```
from tkinter import ttk
from tkinter import messagebox
import pandas as pd
import operator
from tkinter import *
root = Tk()
def process():
    reportWindow = Tk()
    reportWindow.title("Report")
    df = pd.read csv('winequality-red.csv')
    lista = [list(row) for row in df.values]
    similares = {}
    cn = [float(fixedAciditySpinner.get()), float(volatileAciditySpinner.get()),
float(citricAciditySpinner.get()), float(residualSugarSpinner.get()),
float(chloridesSpinner.get()), float(freeSulfurDioxideSpinner.get()),
          float(toalcoholSpinnerlSulfurDioxideSpinner.get()),
float(densitySpinner.get()), float(phSpinner.get()), float(sulphatesSpinner.get()),
float(alcoholSpinner.get())]
    mini = [4.6, 0.12, 0, 0.9, 0.012, 1, 6, 0.99, 2.74, 0.33, 8.4]
    maxi = [15.9, 1.58, 1.0, 13.9, 0.611, 72.0, 289.0, 1.0, 4.01, 2.0, 14.9]
    weight = [float(fixedAcidityWeighchloridesSpinnerbox.get()),
float(volatileAcidityWeighchloridesSpinnerbox.get()),
float(citricAcidityWeighchloridesSpinnerbox.get()),
float(residualSugarWeighchloridesSpinnerbox.get()),
float(chloridesWeighchloridesSpinnerbox.get()),
              float(freeSulfurDioxideWeightCbox.get()),
              float(toalcoholSpinnerlSulfurDioxideWeightCbox.get()),
float(densityWeightCbox.get()), float(phWeightCbox.get()),
float(sulphatesWeightCbox.get()), float(alcoholWeightCbox.get())]
    def similarity(ce):
       valor = 0
        for i in range(len(mini)):
            valor += weight[i] * (1 - ((abs(ce[i] - cn[i])) / (maxi[i] - mini[i])))
```

```
return valor / sum(weight)
    for i in range(len(lista)):
        fila = []
        fila = lista[i]
        x = similarity(fila)
        similares.update({str(i): round(x, 3)})
    ordLista = dict(sorted(similares.items(), key=operator.itemgetter(1)))
    cols = (
    "#Wine", "Fixed Acidity", "Volatile Acidity", "Citric Acid", "Residual Sugar",
"Chlorides", "Free Sulfure Dioxide",
    "ToalcoholSpinnerl Sulfure Dioxide", "Density", "pH", "Sulphates", "Alcohol",
"Quality", "Similarity")
    tree = ttk.Treeview(reportWindow, columns=cols, show='headings')
    vsb = ttk.Scrollbar(reportWindow, orient="vertical", command=tree.yview)
    vsb.pack(side=RIGHT, fill=BOTH)
    tree.configure(yscrollcommand=vsb.set)
    for i in range(len(cols)):
        tree.heading(cols[i], text=cols[i])
        tree.column(cols[i], minwidth=0, width=50)
    tree.pack(expand=YES, fill=BOTH)
    alcoholSpinnerm = len(ordLista)
    for i in range(alcoholSpinnerm):
        pos = int(list(ordLista.items())[i][0])
        v1 = lista[int(pos)][0]
        v2 = lista[int(pos)][1]
        v3 = lista[int(pos)][2]
        v4 = lista[int(pos)][3]
        v5 = lista[int(pos)][4]
        v6 = lista[int(pos)][5]
       v7 = lista[int(pos)][6]
        v8 = lista[int(pos)][7]
       v9 = lista[int(pos)][8]
        v10 = lista[int(pos)][9]
        v11 = lista[int(pos)][10]
        v12 = lista[int(pos)][11]
        sim = str(list(ordLista.items())[i][1])
        tree.insert("", 0, i, values=(str(pos), v1, v2, v3, v4, v5, v6, v7, v8, v9,
v10, v11, v12, sim))
    # print("Item Mas Similar")
    fpos = list(ordLista.items())[alcoholSpinnerm - 1][0]
    fval = list(ordLista.items())[alcoholSpinnerm - 1][1]
    res = lista[int(fpos)][11]
    messagebox.showinfo(message="Calificacion: " + res)
lista = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
#Creacion de la venta principal
root.geometry('700x300') # anchura x altura
root.config(bg="lightblue")
Label(root, text="WINE QUALITY",fg="red",font=("Comic Sans MC",10,
'bold')).place(x=200, y=0)
Label(root, text="Fixed Acidy",font=('bold')).place(x=0, y=25)
fixedAciditySpinner = Spinbox(root, from =4.6, to=15.9, width=5, increment=0.1,
font='Helvetica 12')
fixedAciditySpinner.place(x=150, y=25)
fixedAcidityWeighchloridesSpinnerbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
fixedAcidityWeighchloridesSpinnerbox.place(x=225, y=25)
fixedAcidityWeighchloridesSpinnerbox.current(3)
Label(root, text="Volatily Acidy",font=('bold')).place(x=308, y=25)
volatileAciditySpinner = Spinbox(root, from =0.12, to=1.58, width=5, increment=0.01,
font='Helvetica 12')
volatileAciditySpinner.place(x=450, y=25)
volatileAcidityWeighchloridesSpinnerbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
volatileAcidityWeighchloridesSpinnerbox.place(x=525, y=25)
volatileAcidityWeighchloridesSpinnerbox.current(3)
Label(root, text="Citric Acid",font=('bold')).place(x=0, y=50)
citricAciditySpinner = Spinbox(root, from =0.0, to=1.0, width=5, increment=0.1,
font='Helvetica 12')
citricAciditySpinner.place(x=150, y=50)
citricAcidityWeighchloridesSpinnerbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
citricAcidityWeighchloridesSpinnerbox.place(x=225, y=50)
citricAcidityWeighchloridesSpinnerbox.current(3)
Label(root, text="Residual Sugar",font=('bold')).place(x=308, y=50)
residualSugarSpinner = Spinbox(root, from_=0.9, to=13.9, width=5, increment=0.1,
font='Helvetica 12')
residualSugarSpinner.place(x=450, y=50)
residualSugarWeighchloridesSpinnerbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
residualSugarWeighchloridesSpinnerbox.place(x=525, y=50)
residualSugarWeighchloridesSpinnerbox.current(5)
Label(root, text="Chlorides",font=('bold')).place(x=0, y=75)
chloridesSpinner = Spinbox(root, from =0.012, to=0.611, width=5, increment=0.001,
font='Helvetica 12')
chloridesSpinner.place(x=150, y=75)
chloridesWeighchloridesSpinnerbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
chloridesWeighchloridesSpinnerbox.place(x=225, y=75)
```

```
chloridesWeighchloridesSpinnerbox.current(1)
Label(root, text="Free Sulfur Dioxide",font=('bold')).place(x=308, y=75)
freeSulfurDioxideSpinner = Spinbox(root, from_=1.0, to=72.0, width=5, increment=1.0,
font='Helvetica 12')
freeSulfurDioxideSpinner.place(x=450, y=75)
freeSulfurDioxideWeightCbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
freeSulfurDioxideWeightCbox.place(x=525, v=75)
freeSulfurDioxideWeightCbox.current(1)
Label(root, text="Sulfure Dioxide",font=('bold')).place(x=0, y=100)
toalcoholSpinnerlSulfurDioxideSpinner = Spinbox(root, from =6.0, to=289.0, width=5,
increment=1, font='Helvetica 12')
toalcoholSpinnerlSulfurDioxideSpinner.place(x=150, y=100)
toalcoholSpinnerlSulfurDioxideWeightCbox = ttk.Combobox(root, values=lista, width=5,
font='Helvetica 12')
toalcoholSpinnerlSulfurDioxideWeightCbox.place(x=225, y=100)
toalcoholSpinnerlSulfurDioxideWeightCbox.current(1)
Label(root, text="Density",font=('bold')).place(x=308, y=100)
densitySpinner = Spinbox(root, from =0.9900, to=1.0000, width=6, increment=0.0001,
font='Helvetica 12')
densitySpinner.place(x=450, y=100)
densityWeightCbox = ttk.Combobox(root, values=lista, width=5, font='Helvetica 12')
densityWeightCbox.place(x=525, y=100)
densityWeightCbox.current(1)
Label(root, text="pH",font=('bold')).place(x=0, y=125)
phSpinner = Spinbox(root, from =2.74, to=4.01, width=5, increment=0.01,
font='Helvetica 12')
phSpinner.place(x=150, y=125)
phWeightCbox = ttk.Combobox(root, values=lista, width=5, font='Helvetica 12')
phWeightCbox.place(x=225, y=125)
phWeightCbox.current(6)
Label(root, text="Sulphates",font=('bold')).place(x=308, y=125)
sulphatesSpinner = Spinbox(root, from =0.33, to=2.0, width=5, increment=0.01,
font='Helvetica 12')
sulphatesSpinner.place(x=450, y=125)
sulphatesWeightCbox = ttk.Combobox(root, values=lista, width=5, font='Helvetica 12')
sulphatesWeightCbox.place(x=525, y=125)
sulphatesWeightCbox.current(1)
Label(root, text="Alcohol",font=('bold')).place(x=0, y=150)
alcoholSpinner = Spinbox(root, from =8.4, to=14.9, width=5, increment=0.1,
font='Helvetica 12')
alcoholSpinner.place(x=150, y=150)
alcoholWeightCbox = ttk.Combobox(root, values=lista, width=5, font='Helvetica 12')
alcoholWeightCbox.place(x=225, y=150)
alcoholWeightCbox.current(5)
```

```
btnProcess=Button(root, fg="red" ,font=("Comic Sans MC",20,
'bold'),text="Process",command=process)
btnProcess.place(x=275, y=200)
root.mainloop()
```

## 2.EXPLICACION DE LA REALIZACION

Para la implementación de este sistema, se utilizó la fórmula que nos proporciona **wguillen**, la misma que se encuentra en las diapositivas que proporciona el autor.

Para ello debemos ir al repositorio del autor:

https://github.com/wguilen/red-wine-quality-cbr/tree/master/presentation

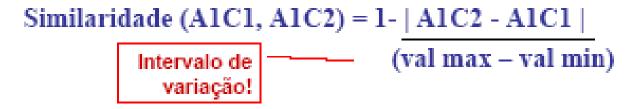


Fig. 1 Fórmula proporcionada por el autor.

Esta misma fórmula se utilizó en Python.

```
def similarity(ce):
    valor = 0
    for i in range(len(mini)):
        valor += weight[i] * (1 - ((abs(ce[i] - cn[i])) / (maxi[i] - mini[i])))
    return valor / sum(weight)
```

Cabe mencionar que para la implementación también me fui basando en el programa realizado en JAVA.

### 3.EJECUCION

Para la ejecución tomamos como base los valores por defecto que se presenta en la aplicación de java.

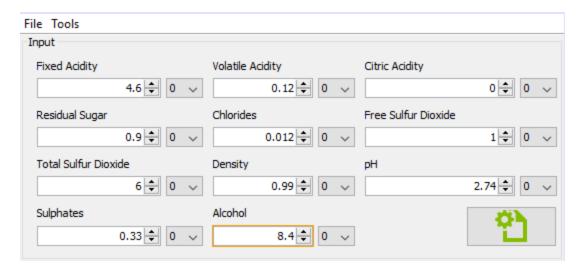
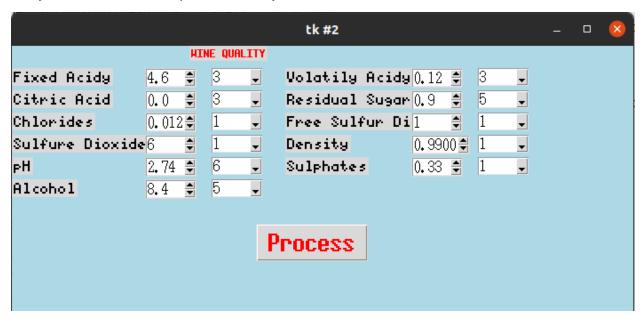


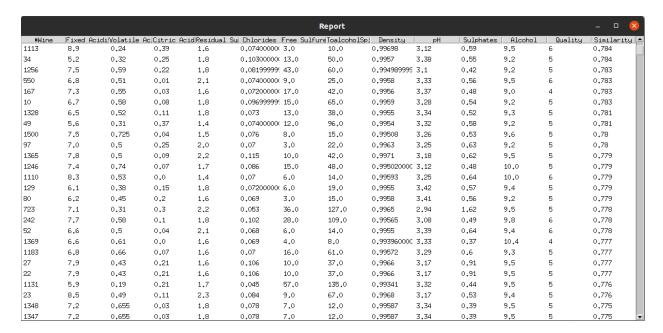
Fig. 2 Ventana de la aplicación realizada en Java.

La ejecución de nuestra aplicación en Python.

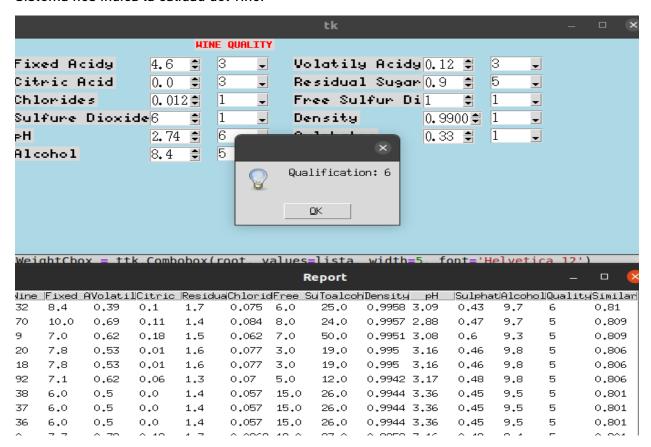


## **RESULTADOS:**

Los datos procesados de archivo **csv**, conjunto con los atributos que en base a los datos de ingreso también nos proporciona la información de la calidad y la similitud con el nuevo caso.



Sistema nos indica la calidad del vino.



### **4.ANALISI DE RESULTADOS**

Desde mi punto de vista realmente no sé con qué exactitud es tan importante cada una de las características para ver la calidad de un vino, pero revisando algunas fuentes de empresa que

| se dedican a la explotación de este campo, puedo decir que el sistema tiene una gran similito a la forma en la que realmente se realiza este proceso. | ıd |
|---|----|
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |
|   |    |