

“El usuario deberá llenar el espacio, con el nombre del archivo que se va a crear, el texto que se va a escribir en el archivo y el número de veces que se van a escribir el texto indicado en el archivo. Después de llenar los espacios, el usuario podrá dar click en el botón analizar y el programa entregará como resultado el número de vocales y consonantes que contiene el archivo creado Como evidencia de la actividad, deberán entregar un reporte en github de todo el proceso que siguieron para desarrollar esta app en python y su ejecución en su Raspberry”

Python(Código para el funcionamiento de la interfaz gráfica)

from PyQt5 import QtCore, QtGui, QtWidgets

```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(800, 600)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.lineEdit = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit.setGeometry(QtCore.QRect(210, 130, 113, 20))
        self.lineEdit.setObjectName("lineEdit")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(350, 130, 100, 13))
        self.label.setObjectName("label")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(370, 270, 75, 23))
        self.pushButton.setObjectName("pushButton")
        self.lineEdit_2 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_2.setGeometry(QtCore.QRect(210, 170, 113, 20))
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(350, 170, 100, 13))
        self.label_2.setObjectName("label_2")
        self.lineEdit_3 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_3.setGeometry(QtCore.QRect(210, 210, 113, 20))
        self.lineEdit_3.setObjectName("lineEdit_3")
        self.label_3 = QtWidgets.QLabel(self.centralwidget)
        self.label_3.setGeometry(QtCore.QRect(350, 210, 150, 13))
        self.label_3.setObjectName("label_3")
        self.lineEdit_4 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_4.setGeometry(QtCore.QRect(170, 350, 113, 20))
        self.lineEdit_4.setObjectName("lineEdit_4")
        self.label_4 = QtWidgets.QLabel(self.centralwidget)
        self.label_4.setGeometry(QtCore.QRect(170, 320, 150, 13))
        self.label_4.setObjectName("label_4")
        self.lineEdit_5 = QtWidgets.QLineEdit(self.centralwidget)
        self.lineEdit_5.setGeometry(QtCore.QRect(470, 350, 113, 20))
        self.lineEdit_5.setObjectName("lineEdit_5")
        self.label_5 = QtWidgets.QLabel(self.centralwidget)
        self.label_5.setGeometry(QtCore.QRect(470, 320, 150, 13))
        self.label_5.setObjectName("label_5")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
```

```

self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.label.setText(_translate("MainWindow", "TextLabel"))
    self.pushButton.setText(_translate("MainWindow", "PushButton"))
    self.label_2.setText(_translate("MainWindow", "TextLabel"))
    self.label_3.setText(_translate("MainWindow", "TextLabel"))
    self.label_4.setText(_translate("MainWindow", "TextLabel"))
    self.label_5.setText(_translate("MainWindow", "TextLabel"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```

Python(Código de funcionamiento del programa)

```

import os
from PyQt5 import QtGui, QtCore
#from PyQt5.QtCore import *
#from PyQt5.QtGui import *
from Mod4Act4ui import *

class Ui_MainWindow(QtWidgets.QMainWindow,Ui_MainWindow):
    def __init__(self, *args, **kwargs):
        QtWidgets.QMainWindow.__init__(self, *args, **kwargs)
        self.setupUi(self)

#-----
#Se nombran los elementos
self.label.setText("Nombre el archivo")
self.label_2.setText("Texto a escribir")
self.label_3.setText("Numero de lineas en el archivo")
self.label_4.setText("Numero de vocales")
self.label_5.setText("Numero de consonantes")
self.pushButton.setText("Actualizar")

#-----
#Acciones de los elementos
self.pushButton.clicked.connect(self.detectData)

def detectData(self, bool_1):
    f = open(self.lineEdit.text(), "w")
    data = self.lineEdit_2.text()
    lines = int(self.lineEdit_3.text())
    [f.write(data + "\n") for _ in range(lines)]
    con = 0
    for i in data:
        if(i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u'):

```

```

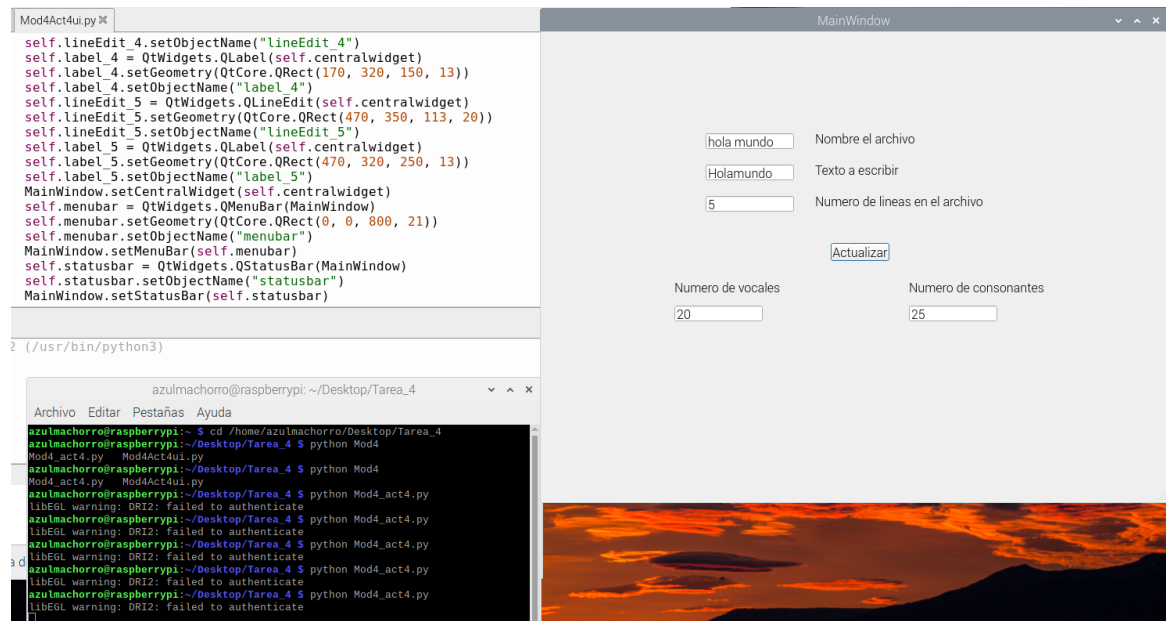
con += 1
self.lineEdit_4.setText(str(con*lines))
self.lineEdit_5.setText(str((len(data) - con)*lines))

def Actualizar(self):
    self.lineEdit_5.setText("")
    self.lineEdit_5.setText("Hola")

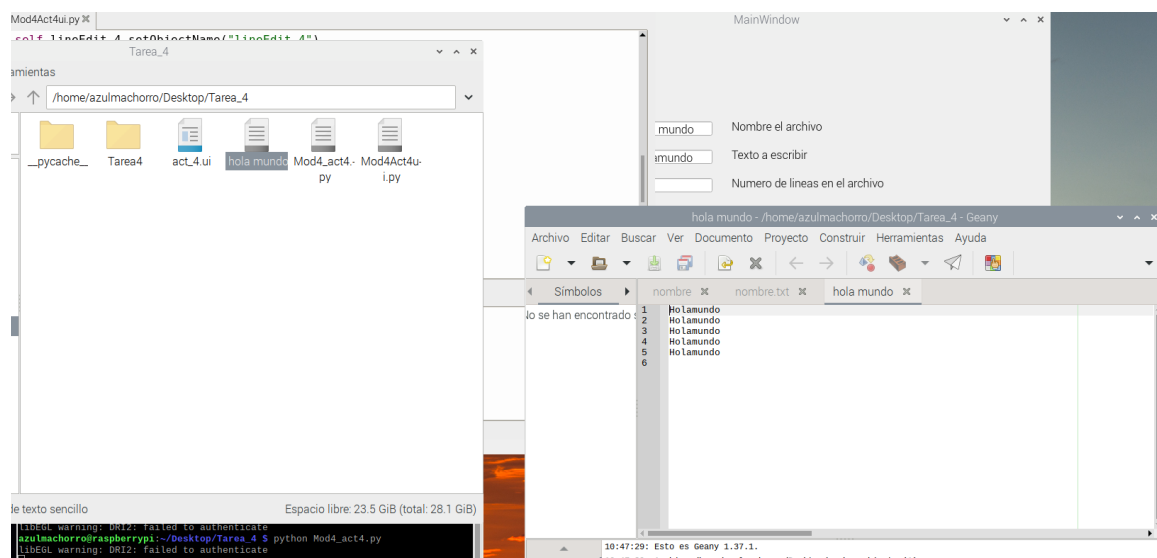
if __name__ == "__main__":
    app = QtWidgets.QApplication([])
    window = Ui_MainWindow()
    window.show()
    app.exec_()

```

Funcionamiento de la interfaz gráfica:



Creación del documento:



Funcionamiento del código:

Para el desarrollo de la interfaz gráfica hemos utilizado la aplicación QTdesigner la cual nos permitió el poder diseñar la interfaz gráfica con los recursos que nosotros necesitamos, siendo que para la interfaz gráfica utilizamos etiquetas, lineedit y push button. Para el correcto funcionamiento de la interfaz gráfica hemos creado dos códigos en los cuales el primero contiene los objetos que se mostrarán en la interfaz gráfica mientras que el segundo código se encarga de controlar el funcionamiento de los objetos de la interfaz gráfica, para que el programa pueda funcionar correctamente es necesario haber instalado la librería PyQt5 ya que esta es una librería gráfica la cual nos permitirá desplegar que hayamos diseñado.

Por otra parte se importará al programa el código con los objetos que hemos colocado para la interfaz gráfica, esto para poder asignar las funciones que necesitamos siendo que se colocarán los nombres para las etiquetas, se configurará el botón y se habilitará la lectura y guardado de los caracteres que los usuarios coloque en los line edit. Para poder trabajar con los objetos antes mencionados se necesita crear una clase en la cual se ponen las acciones de los elementos y posteriormente en esta misma clase realizamos el conteo de las vocales y consonantes tal y como lo pide la actividad.

Para finalizar el código crea un archivo el cual cuenta con el texto introducido por el usuario al igual que el número de líneas que este nos proporciona.

El funcionamiento de nuestro código en la raspberry es similar al que podemos ver en la máquina que se realizó el código, siendo que este debe ser ejecutado desde la terminal y los dos archivos deben de estar en la misma carpeta ya que de lo contrario no funcionará el programa que se ha realizado.

Conclusiones:

El código que hemos realizado es bastante fácil de entender, sin embargo el funcionamiento de este podría mejorarse ya que se podría introducirse todo en un mismo código haciendo todo más compacto, sin embargo el uso de la aplicación Qt designer fue de mucha utilidad ya que nos permite el poder manipular los elementos a nuestra voluntad y así podemos realizar un mejor diseño en un menor tiempo y esfuerzo.