A red sign with white text

Description automatically generated

**P4. Cadenas y Archivos en Python**

Alumna: Daniela Mendez Ramirez Número de Cuenta: 258331-9

**Aplicaciones de Redes**

Profesor: Omar Vázquez González

Fecha de Entrega: 3 de febrero de 2025

Abstract:

Durante esta práctica, se realizarán diversos ejercicios enfocados en la manipulación de cadenas y archivos en Python. Inicialmente, se trabajará con ejercicios prácticos para familiarizarse con diferentes métodos de manipulación de texto y gestión de archivos. Posteriormente, se desarrollará un proyecto final en el que se diseñará un programa automatizado para agilizar la creación de cartas personalizadas. Este programa tomará los nombres y las cualidades de las personas a quienes estarán dirigidas las cartas, generando un archivo individualizado para cada destinatario con contenido dinámico basado en una plantilla. El objetivo es comprender cómo manejar la lectura, modificación y escritura de archivos en Python, así como cómo personalizar el contenido de manera eficiente.

Introducción:

En esta práctica, se abordarán conceptos fundamentales de programación en Python, centrados en el manejo de cadenas de texto (String), archivos y diccionarios. A lo largo de los ejercicios, exploraremos diversas técnicas para manipular cadenas, que son esenciales para cualquier tipo de procesamiento de texto. Aprenderemos a leer y escribir archivos, una habilidad crucial en muchos proyectos de programación, y a utilizar diccionarios para almacenar y organizar información de manera eficiente. Estos conceptos se aplicarán en un proyecto final donde automatizaremos la creación de cartas personalizadas, utilizando plantillas y datos dinámicos como nombres y cualidades de personas, para generar archivos personalizados para cada destinatario. Este enfoque no solo nos ayudará a comprender la teoría detrás de estas estructuras de datos, sino también a implementarlas en escenarios prácticos.

Ejercicios de Cadenas

1. Cadena en Python, con comilla doble y con comilla simple.

A white text with green and blue text

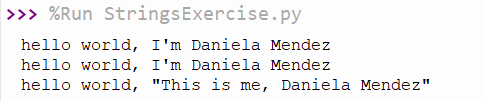
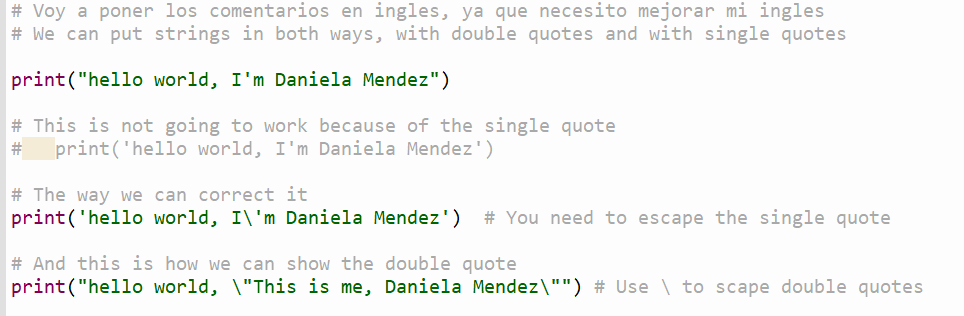
Description automatically generated

1. Imprimir la comilla simple

A screenshot of a computer

Description automatically generated

1. Imprimir comilla doble



1. Asignar texto o palabras a una variable.

A screenshot of a computer code

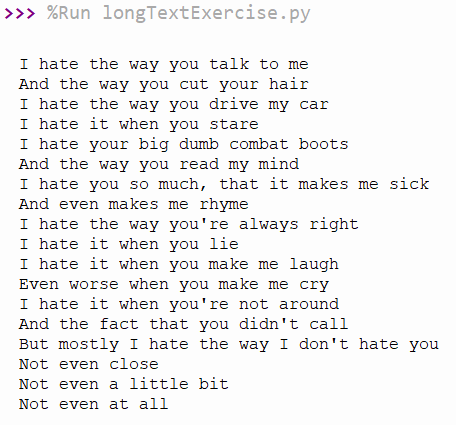
Description automatically generatedA close up of text

Description automatically generated

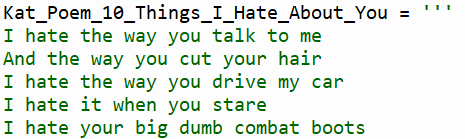
1. Imprimir un poema

A screenshot of a computer screen

Description automatically generated



Nota: Podemos hacerlo tanto con 3 comillas simples como con comillas dobles.



1. Descomponiendo una cadena.

A screen shot of a computer program

Description automatically generated

A white background with black text

Description automatically generated

1. Diferencia entre imprimir con coma e imprimir con +

A black and green text

Description automatically generated



1. Crear un archivo y escribir en el. Nota: Esto solo se puede hacer siempre y cuando el archivo recién se haya creado.

A screenshot of a computer program

Description automatically generated

Ejercicios de Archivos

1. Instrucciones en general

A screenshot of a computer program

Description automatically generated

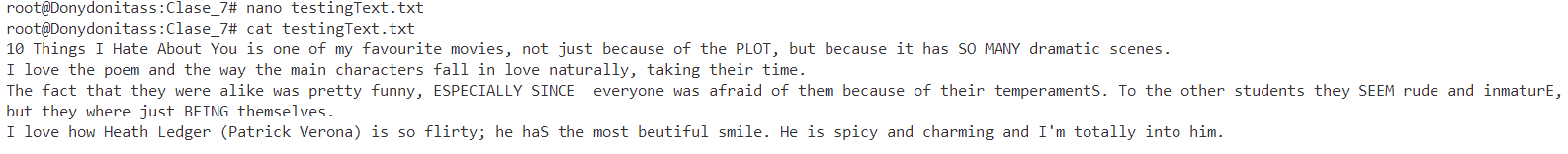
1. Crear y leer un archivo

A screen shot of a computer program

Description automatically generated



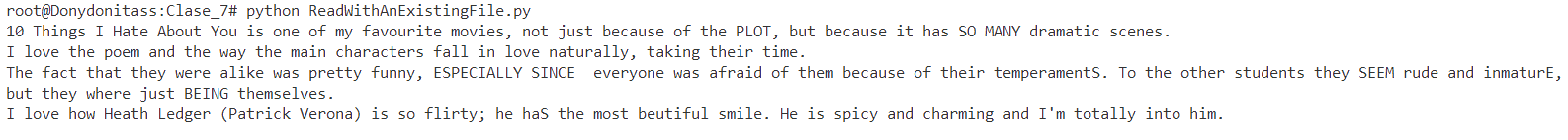
1. Ejercicios usando un archivo ya existente
2. Creamos el archivo con nano



1. Leer el archivo

A close-up of a text

Description automatically generated



1. Imprimir los primeros 5 caracteres

A close up of a text

Description automatically generated



1. Diferentes formas de Impresión

A screenshot of a computer code

Description automatically generated

A white background with black text

Description automatically generated

1. Recorrer todas las líneas de texto e imprimirlas

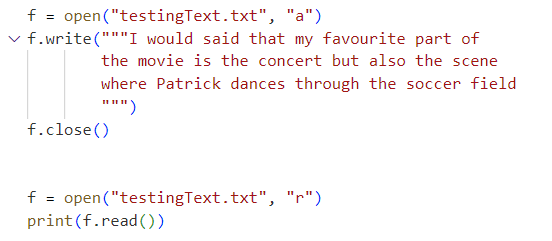
A screen shot of a computer

Description automatically generated

A white background with text

Description automatically generated

1. Anadir texto en el archivo



A close up of text

Description automatically generated

1. Eliminar un archivo

A screenshot of a computer code

Description automatically generated



Programa "San Valentín"

Código:

# San Valentin Program

import os

import re

if os.path.exists("Personitas.txt"):

    os.remove("Personitas.txt")

f = open("Personitas.txt", "x")

f.write("""

        Amigos:{

                Iris Yulit Jasso Cortes {cualidades: divertida, amable, mi mejor amiga}

                Joshua Abdiel Barragan Guardado {cualidades: honesta, un amor, directa, en general eres mi mejor amigo}

                Rodrigo Cruz Bartolo {cualidades: divertida, chistosa, y super adoro hablar contigo}

            }

        Crushes:{

                Lorenzo Grebe {cualidades: eres una persona amable, educado, alto, bonito, ojos bonitos, soliamos estudiar juntos, sonrisa hermosa, coqueto, te sientas adelante en el salon}

                Secreto {cualidades: listo, buena onda, lees mucho, eres divertido, usas frases chistosas, me gusta hablar contigo, desyunas muy rapido y me tienes que esperar mientras yo desayuno}

        }

        """)

f.close()

f = open("Personitas.txt", "r")

personitasCartita = f.read()

f.close()

print(personitasCartita)

# We create the frinds letter

if os.path.exists("Platilla\_Amigos.txt"):

    os.remove("Platilla\_Amigos.txt")

f = open("Platilla\_Amigos.txt", "x")

f.write("""

        Feliz San Valentin nombre,

        Esta carta la escribo con todo mi cariño,

        esperando te encuentres bien, a continuación

        te voy a decir algunas de las razones por las

        que te quiero demasiado y por qué eres importante

        para mi:

        Eres una persona cualidades.

        Pero no solo son ese tipo de cosas las que te hacen especial

        para mi, lo que realmente aprecio de nuestra amistad, es que es

        verdadera, que puedo contar contigo en las buenas y en las malas,

        que no vas a temer decirme la verdad, pero tambien estar

        ahi cuando lo necesite.

        Te quiero muchooooo.

        Dany.

        """)

f.close()

f = open("Platilla\_Amigos.txt", "r")

cartita\_amiguitos = f.read()

print(cartita\_amiguitos)

# We create our crush letter

if os.path.exists("cartitaCrush.txt"):

    os.remove("cartitaCrush.txt")

f = open("cartitaCrush.txt", "w")

f.write("""

        Feliz San Valentin, nombre

        Probablemente te preguntes por qué te estoy escribiendo una carta,

        lo mismo me pregunto yo, sin embargo, una tarea me ha llevado a esto,

        si bien creo que en San Valentin puedes festejar las amistades,

        también creo que es para sentirse enamorado y disfrutar de este sentimiento.

        Y he aqui mi carta, probablemente a estas alturas ya sepas que siento

        algo por ti. Tendré que ser honesta y admitir que no es un simple crush,

        al menos no del todo.

        Eres una persona a la que he conocido poco, pero en ese breve tiempo,

        me hiciste pasarmela increíble, la verdad te me haces un niño espectacular

        y de alguna manera lograste meterte en mi mente, debido a ciertas cosas que

        he notado de ti, como que eres: cualidades.

        En fin puede que algunas sean un poco torpes y superficiales, pero en realidad

        son validas para mi, son de esas pequeñas razones que me hacen sentir que quiero

        conocerte más.

        Entonces dejando esta carta a tu alcance, te pregunto si estas interesado en

        conocer alguna de tus posibles futuras razones para encontrar tus razones que te

        hagan sentir de la misma manera que yo por ti.

        Así que feliz 14 de Febrero, espero podamos vernos pronto.

        Dany.

        """)

f.close()

f = open("cartitaCrush.txt", "r")

cartitaCrush = f.read()

f.close()

# Expresión regular ajustada para extraer las categorías (Amigos, Crushes) y su contenido

categorias = re.findall(r"(\w+):\s\*{([^{}]+(?:\{[^}]\*\}[^{}]\*)\*)}", personitasCartita)

# Habilitamos la variable para el diccionario

amigos = {}

crushes = {}

# Mostramos los bloques encontrados

for categoria, contenido in categorias:

    #print(f"Categoría: {categoria}")

    #print(f"Contenido:\n{contenido}\n")

    #Capturar el nombre

    nombres = re.findall(r"([\w\s]+?)\s\*\{cualidades:", contenido)

    # Limpiar los saltos de línea y espacios adicionales de cada nombre

    nombres\_limpios = [nombre.strip() for nombre in nombres]

    #print(nombres\_limpios)

    #Capturar cualidades

    cualidades = re.findall(r"\{cualidades:\s\*([^}]+)\}", contenido)

    #print(cualidades)

    # Creamos un diccionario

    if categoria == "Amigos":

        for i, nombre in enumerate(nombres\_limpios):

            amigos[nombre] = cualidades[i]

    else:

        for i, nombre in enumerate(nombres\_limpios):

            crushes[nombre] = cualidades[i]

# print(amigos)

# print(crushes)

if not os.path.exists("Amigos"):

    os.makedirs("Amigos")

if not os.path.exists("Crushes"):

    os.makedirs("Crushes")

# Desglosamos el glosario de Amigos

for nombre, cualidades in amigos.items():

    arch\_personalizado = os.path.join("Amigos", f"{nombre}.txt")

    if os.path.exists(arch\_personalizado):

        os.remove(arch\_personalizado)

    with open(arch\_personalizado, "w") as f:

        carta = cartita\_amiguitos.replace("Crushes", nombre).replace("cualidades", cualidades)

        f.write(carta)

        f.close()

# Desglosamos el glosario de Crushes

for nombre, cualidades in crushes.items():

    arch\_personalizado = os.path.join("Crushes", f"{nombre}.txt")

    if os.path.exists(arch\_personalizado):

        os.remove(arch\_personalizado)

    with open(arch\_personalizado, "w") as f:

        carta = cartitaCrush.replace("nombre", nombre).replace("cualidades", cualidades)

        f.write(carta)

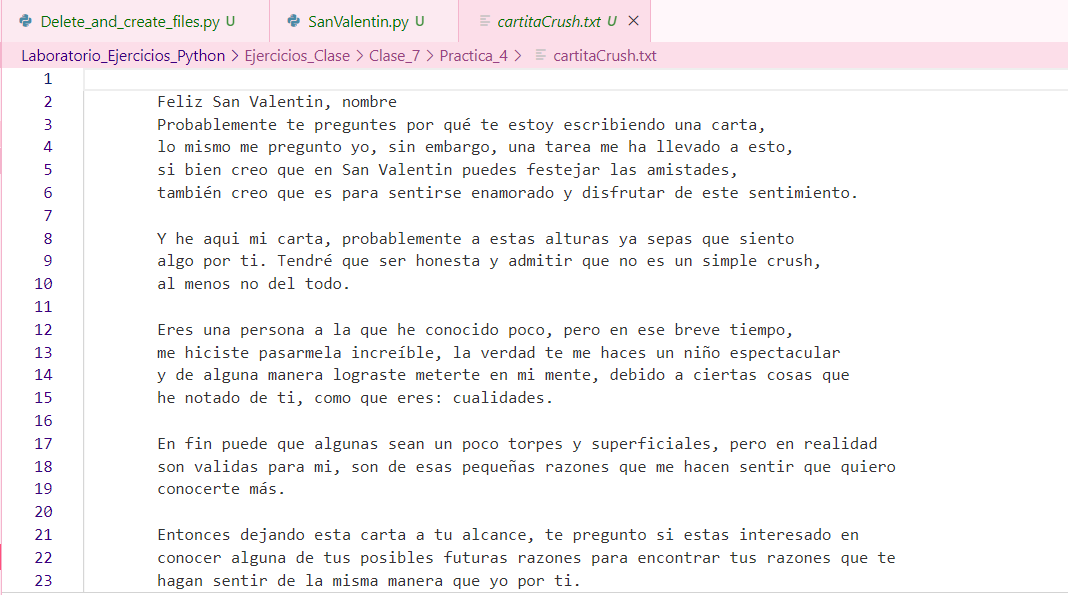
        f.close()

Archivo de las Personitas a las que se va a enviar carta:

A screenshot of a computer

Description automatically generated

Archivo Cartita Crush:



Archivo Cartita Amigos:

A screenshot of a computer

Description automatically generated

Cada carta se guarda en el folder que le corresponde:

A screenshot of a computer

Description automatically generated

Resultado carta amigos

A screenshot of a computer

Description automatically generated

Resultado carta Crushes

A screenshot of a computer

Description automatically generated

Conclusión:

El manejo de cadenas de texto ha sido súper útil para poder personalizar las cartas en esta práctica. Al trabajar con funciones como replace(), pude reemplazar de manera sencilla los nombres y cualidades de las personas dentro de las cartas. Además, la combinación de cadenas con la lectura y escritura de archivos me permitió automatizar el proceso y crear cartas personalizadas para cada uno de mis amigos y crushes. También, el uso de expresiones regulares y diccionarios fue clave para organizar y extraer la información de manera eficiente. Todo esto me ayudó a entender cómo las cadenas y los diccionarios pueden hacer que tareas repetitivas sean mucho más rápidas y fáciles de manejar, y de paso, aprendí cómo adaptar estos conceptos a situaciones reales como la creación de cartas personalizadas.

Fuentes:

*W3Schools.com*. (s. f.). http://www.w3schools.com/python/matplotlib\_intro.asp

*W3Schools Python Exercise*. (s. f.). http://www.w3schools.com/python/exercise.asp?x=xrcise\_file\_handling1

*W3Schools.com*. (s. f.-b). <http://www.w3schools.com/python/python_comments.asp>

*W3Schools Python Exercise*. (s. f.-b). http://www.w3schools.com/python/exercise.asp?x=xrcise\_operators1