

## Unidad de Competencia 2

Nombre: Steve Gomez Huamani

- La entrega fuera de hora representa 13 puntos menos
- Se debe visualizar el proceso a travez de commit en git
- Compartir el link de github
- Entregar el documento al ula virtual en pdf

1. Preliminar:

Commit inicial:"UC 2 – Inicio"

```
USER@DESKTOP-C258LR2 MINGW64 /d/CICLO 6/LP3/Semana6 (main)
$ git add -A

USER@DESKTOP-C258LR2 MINGW64 /d/CICLO 6/LP3/Semana6 (main)
$ git commit -m "UC 2 – Inicio de GOMEZ HUAMANI STEVE"
[main 3bc7ad6] UC 2 – Inicio de GOMEZ HUAMANI STEVE
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Semana07/UC2/Unidad de Competencia 2.docx
create mode 100644 Semana07/UC2/~Sidad de Competencia 2.docx

USER@DESKTOP-C258LR2 MINGW64 /d/CICLO 6/LP3/Semana6 (main)
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 12.51 kib | 400.00 kib/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/cerdanflor/2022-1_LP3-S1_Python.git
  38569a..3bc7ad6 main -> main
```

2. Crear un módulo llamado: **gestion\_archivos.py**, que considere los métodos:
1. crear\_archivo, que reciba como parámetro el nombre del archivo y el contenido.
  2. eliminar\_archivo, que considere como parámetro el nombre del archivo.
  3. agregar\_contenido\_archivo, que considere como parámetro el nombre del archivo y el contenido.
  4. leer\_archivo, que considere como parámetro el nombre del archivo.

```

import os
# Para crear archivo, recibe como parámetro el nombre del archivo
# y el contenido del archivo
def crear_archivo(nombre, contenido):
    archivo = open(nombre, "wt")
    archivo.write(contenido)
    archivo.close()

# Para eliminar recibe como parámetro el nombre del archivo a eliminar
def eliminar_archivo(nombre):
    os.remove(nombre)

# Para agregar contenido a un archivo plano, debe existir un archivo
# Se envía como parámetro el nombre y el contenido del archivo
def agregar_contenido_archivo(nombre, contenido):
    archivo = open(nombre, "at")
    archivo.write("\n" + contenido)
    archivo.close()

# Para leer un archivo, ya debe existir un archivo plano
# Ejemplo: txt, py, java, ...
# Recibe como parámetro el nombre del archivo a leer
# y devuelve el contenido del archivo
def leer_archivo(nombre):
    archivo = open(nombre, "rt", encoding='utf8')
    contenido = archivo.read()
    return contenido

```

```

USER@DESKTOP-C258LR2 MINGW64 /d/CICLO 6/LP3/Semana6
$ git add -A

USER@DESKTOP-C258LR2 MINGW64 /d/CICLO 6/LP3/Semana6
$ git commit -m "UC 2 - Gestión de Archivos de GOMEZ HUAMANI STEVE"
[main fdab97e] UC 2 - Gestión de Archivos de GOMEZ HUAMANI STEVE
2 files changed, 34 insertions(+)
create mode 100644 Semana07/UC2/Gestion_Archivos.py

USER@DESKTOP-C258LR2 MINGW64 /d/CICLO 6/LP3/Semana6
$ git push
Enumerating objects: 10, done.
counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 205.50 KiB | 12.84 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/cerdanFlor/2022-1_LP3-S1_Python.git
3bc7ad6..fdab97e main -> main

```

En otro archivo llamado: **aplicacion.py**, diseñar:

- Un menú de opciones:

Menú
1. Crear archivo
2. Eliminar archivo
3. Agregar contenido
4. Mostrar contenido de archivo
5. Salir

- Considere los métodos en **aplicacion.py**
  - menu()
  - crear()
  - eliminar()
  - agregar()
  - listar()
  - salir()

## 7. error()

```
import Gestion_Archivos

def menu():
    print("*** MENU PRINCIPAL ***")
    print("1. Crear Archivo")
    print("2. Eliminar Archivo")
    print("3. Agregar Contenido")
    print("4. Mostrar Contenido de Archivo")
    print("5. Salir")

def crear():
    print("*** CREAR ARCHIVO ***")
    archivo = input("Crear archivo: ")
    contenido = input("Contenido del archivo: ")
    Gestion_Archivos.crear_archivo(archivo, contenido)

def eliminar():
    print("*** ELIMINAR ARCHIVO ***")
    archivo = input("Quiero eliminar el archivo: ")
    Gestion_Archivos.eliminar_archivo(archivo)

def agregar():
    print("*** AGREGAR CONTENIDO A UN ARCHIVO ***")
    archivo = input("Quiero agregar contenido al archivo: ")
    contenido = input("El contenido adicional del archivo será: ")
    Gestion_Archivos.agregar_contenido_archivo(archivo, contenido)

def listar():
    print("*** MOSTRAR CONTENIDO DE UN ARCHIVO ***")
    archivo = input("Quiero mostrar el contenido del archivo: ")
    print(Gestion_Archivos.leer_archivo(archivo))
```

Mostrar contenido de un archivo

```

9
10 def menu():
11     print("*** MENU PRINCIPAL ***")
12     print("1. Crear Archivo")
13     print("2. Eliminar Archivo")
14     print("3. Agregar Contenido")
15     print("4. Mostrar Contenido de Archivo")
16     print("5. Salir")
17
18 def crear():
19     print("*** CREAR ARCHIVO ***")
20     archivo = input("Crear archivo: ")
21     contenido = input("Contenido del archivo: ")
22     Gestion_Archivos.crear_archivo(archivo, contenido)
23
24 def eliminar():
25     print("*** ELIMINAR ARCHIVO ***")
26     archivo = input("Quiero eliminar el archivo: ")
27     Gestion_Archivos.eliminar_archivo(archivo)
28
29 def agregar():
30     print("*** AGREGAR CONTENIDO A UN ARCHIVO ***")
31     archivo = input("Quiero agregar contenido al archivo: ")
32     contenido = input("El contenido adicional del archivo será: ")
33     Gestion_Archivos.agregar_contenido_archivo(archivo, contenido)
34
35 def listar():
36     print("*** MOSTRAR CONTENIDO DE UN ARCHIVO***")
37

```

```

19     print("*** CREAR ARCHIVO ***")
20     archivo = input("Crear archivo: ")
21     contenido = input("Contenido del archivo: ")
22     Gestion_Archivos.crear_archivo(archivo, contenido)
23
24 def eliminar():
25     print("*** ELIMINAR ARCHIVO ***")
26     archivo = input("Quiero eliminar el archivo: ")
27     Gestion_Archivos.eliminar_archivo(archivo)
28
29 def agregar():
30     print("*** AGREGAR CONTENIDO A UN ARCHIVO ***")
31     archivo = input("Quiero agregar contenido al archivo: ")
32     contenido = input("El contenido adicional del archivo será: ")
33     Gestion_Archivos.agregar_contenido_archivo(archivo, contenido)
34
35 def listar():
36     print("*** MOSTRAR CONTENIDO DE UN ARCHIVO***")
37     archivo = input("Quiero mostrar el contenido del archivo: ")
38     print(Gestion_Archivos.leer_archivo(archivo))
39
40 def salir():
41     print("Gracias... Vuelva pronto")
42
43 def error():
44     print("Opción incorrecta")
45
46 # La lógica para el menu de opciones

```

[badkitten-bug/UC2 \(github.com\)](https://github.com/badkitten-bug/UC2)