

# CHECKPOINT

SECCIÓN D06  
L-MI. 1:00-3:00PM



COMPUTACIÓN  
TOLERANTE A  
FALLAS  
DR. MICHEL  
EMMANUEL LOPEZ  
FRANCO

## CHECKPOINT

UNIVERSIDAD DE GUADALAJARA,  
CUCEI  
MURILLO CORTES, JEANETTE



creado. Para la búsqueda del archivo, se comprueba si hay notas existentes mediante un try-except, donde en el try, abre el archivo de Notas con 'rb' *read binary*, que como su nombre lo dice, lee el archivo en binario. Mientras que exista el archivo, retorna los objetos existentes en el archivo binario, para cargarlas como objetos mediante la función 'load()' de pickle y mostrarlo.

```
filename=input("\nEscribe el
nombre del archivo que quieres
abrir: ")
try:
    fileread=open(filename,'rb')
    while True:
        try:
            note=pickle.load(fileread)
            print("\nTitulo:",filename)
            print("Contenido:",note)
        except:
            break
    fileread.close()
```

```
print("\n\t\t EDITAR NOTA")
print(os.listdir())
filename=input("\nEscribe el nombre del archivo que
quieres editar: ")
fileopc=int(input("Editar: 1.Titulo 2.Contenido. Escoge
1 o 2: "))
if(fileopc == 1):
    try:
        newtitle=input("Escribe el nuevo nombre del
archivo: ")
        path_actual = Path(filename)
        path_objetivo = Path(newtitle)
        Path.rename(path_actual,path_objetivo)
    except:
        print("Ocurrió un error: no existe ese archivo.\n")
        os.system("pause")
elif(fileopc == 2):
    try:
        os.remove(filename)
        fileread = open(filename, 'wr')
        while True:
            try:
                print("\nTitulo: ", filename)
                print("Contenido: ")
                newcontent=input("\nEscribe el contenido de
la nota: ")
                pickle.dump(newcontent, fileread)
                os.system("pause")
                time.sleep(3)
                fileread.close()
            except:
                break
        except:
            print("Ocurrió un error: no existe ese archivo.\n")
            os.system("pause")
```

Para editar una nota, se listan los archivos que se encuentran en la carpeta de la práctica mediante la función 'listdir()', para de esta manera buscar el archivo a editar, y seleccionar si se desea editar el título o el contenido de la nota.

Si el archivo existe y se quiere editar el título, se escribe el nuevo nombre y se guarda en una variable, la cuál se utiliza para renombrar el archivo mediante 'Path.rename()'. En caso de querer editar el contenido, si el archivo existe, se imprime el título de la nota y se da la opción de guardar el nuevo contenido mediante el uso de 'wr' *write binary*, el cuál permite escribir en el archivo. Además, también se solicita el contenido de la nota para serializarlo como objeto en el archivo, utilizando la función 'dump()' de pickle para grabar en el archivo, para por último, cerrar el archivo.

## Añadir Nota

```
1. Añadir nueva nota    2. Mostrar nota    3. Editar nota    4. Salir

Escoge una opción: 1

                        AÑADIR NOTA
Editando...
Titulo de la nota: Nuevo
Escribe el contenido de la nota: Computación Tolerante a Fallas
Guardando...
█
```

## Mostrar Nota

```
Escoge una opción: 2

                        MOSTRAR NOTA
Notas actuales:
['checkpoint.py', 'Clase', 'Ejemplo', 'FileList', 'Hola!', 'NotesPickle', 'Nuevo']

Escribe el nombre del archivo que quieres abrir: Nuevo

                        N O T A

Titulo: Nuevo
Contenido: Computación Tolerante a Fallas

Presione una tecla para continuar . . . █
```

## Editar el Titulo de una Nota

```
Escoge una opción: 3

                        EDITAR NOTA
Notas actuales:
['checkpoint.py', 'Clase', 'Ejemplo', 'FileList', 'Hola!', 'NotesPickle', 'Nuevo']

Escribe el nombre del archivo que quieres editar: Nuevo
Editar: 1. Titulo    2. Contenido. Escoge 1 o 2: 1
Escribe el nuevo nombre del archivo: Tolerancia a Fallas █

Notas actuales:
['checkpoint.py', 'Clase', 'Ejemplo', 'FileList', 'Hola!', 'NotesPickle', 'Tolerancia a Fallas']
```

## Editar el Contenido de una Nota

```
Escoge una opción: 3

                        EDITAR NOTA
Notas actuales:
['checkpoint.py', 'Clase', 'Ejemplo', 'FileList', 'Hola!', 'NotesPickle', 'Tolerancia a Fallas']

Escribe el nombre del archivo que quieres editar: Tolerancia a Fallas
Editar: 1. Titulo    2. Contenido. Escoge 1 o 2: 2

                        N O T A

Titulo: Tolerancia a Fallas
Contenido:

Escribe el contenido de la nota: Computación Tolerante a Fallas, Sección D06, impartida por Michel Emmanuel López Franco █

Presione una tecla para continuar . . .
Guardando...
█

Escribe el nombre del archivo que quieres abrir: Tolerancia a Fallas

                        N O T A

Titulo: Tolerancia a Fallas
Contenido: Computación Tolerante a Fallas, Sección D06, impartida por Michel Emmanuel López Franco

Presione una tecla para continuar . . . █
```

## CONCLUSIÓN

El desarrollo de esta práctica nos permite desarrollar sistemas más robustos para nuestras aplicaciones gracias a la implementación de guardar el estado en el que el usuario esta interactuando, para que por si la aplicación se cierra, se pueda conservar información. Como es importante conocer este tipo de técnicas, nos garantiza tener mejor visión con respecto a lo que podría ocurrir durante el uso de una aplicación, y cómo solucionarlo.



## BIBLIOGRAFÍA

- *"Programming and Problem Solving through Python Language O Level / A Level"*. Microsoft Word - ALevel\_1\_Python\_19May.docx (nielit.gov.in)  
[https://www.nielit.gov.in/gorakhpur/sites/default/files/Gorakhpur/ALevel\\_1\\_Python\\_19May\\_SS.pdf](https://www.nielit.gov.in/gorakhpur/sites/default/files/Gorakhpur/ALevel_1_Python_19May_SS.pdf)
- *"pickle – Python object serialization"* – Python 3.10.4 documentation  
<https://docs.python.org/3/library/pickle.html>