



UNIVERSIDAD DE GUADALAJARA  
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS  
DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

# **Generar un programa que sea capaz de restaurar el estado de ejecución**

Oscar Evanilson Gutiérrez Pérez

Carrera: Ingeniería en Computación

Código de estudiante: 219748308

18 de febrero de 2022

En este programa, utilice el lenguaje de Python para poder codificar la restauración de un estado en ejecución. El ejemplo que codifiqué fue un simple formulario con tres campos: nombre de alumno, numero de lista y calificación. Estos datos se almacenan en un fichero llamado lista.txt.

La herramienta que utilice fue el manejo de archivos dentro de este lenguaje para ir guardando datos en un fichero. A continuación, su explicación:

```
1  import os
2
3  try:
4      respaldo = open('respaldo.txt', 'r')
5      datos = respaldo.readlines()
6      lista = open('lista.txt', 'a')
7      if len(datos) == 1:
8          print("Datos restaurados\n")
9          print("Nombre:", datos[0])
10         numero_lista = int(input("Ingresa el numero de lista: "))
11         calificacion = int(input("Ingresa la calificación: "))
12         lista.write(datos[0])
13         lista.write('\n')
14         lista.write(str(numero_lista))
15         lista.write('\n')
16         lista.write(str(calificacion))
17         lista.write('\n')
18         lista.write('\n')
19         lista.close()
20     elif len(datos) == 2:
21         print("Datos restaurados\n")
22         print("Nombre:", datos[0])
23         print("Numero de lista:", datos[1])
24         calificacion = int(input("Ingresa la calificación: "))
25         lista.write(datos[0])
26         lista.write('\n')
27         lista.write(str(datos[1]))
28         lista.write('\n')
29         lista.write(str(calificacion))
30         lista.write('\n')
31         lista.write('\n')
32         lista.close()
33     elif len(datos) == 3:
34         print("Datos restaurados\n")
35
36     os.system('cls')
37     print("Alumno registrado...\n")
38     respaldo.close()
39     os.remove('respaldo.txt')
```

Lo primero que se ven en el programa es la librería os con la que puedo eliminar ficheros de respaldo, que más adelante explicaré.

Esta primer parte del código es lo que el programa debe de hacer en caso de encontrar un archivo de respaldo, estos archivos se crean automáticamente en caso de que se cierre la aplicación o de que no se complete la ejecución del mismo.

Si se encuentra un archivo de respaldo se va a leer, donde va a contener los datos que no se guardaron en la lista y va a mostrarlos para continuar con el formulario

que fue interrumpido. Lo que se hace es leer los datos en el archivo de respaldo, mostrarlos y pedir los datos que hicieron falta y guardar los datos en la lista. Después de eso elimina el archivo de respaldo ya que no se necesita más.

```
pickie.py x lista.txt x
44 opcion = 0
45 while opcion != 3:
46     print("Selecciona la opción deseada:\n1.-Agregar alumno\n2.-Ver lista de alumnos\n3.-Salir")
47     opcion = int(input())
```

En la siguiente parte del código encontramos el menú que se muestra en consola, hecho con un simple while que muestra las opciones y solicita la que el usuario desea realizar, en este menú tenemos tres opciones.

```
48 if opcion == 1:
49     os.system("cls")
50     crear_respaldo = open('respaldo.txt', 'a')
51     nombre = input("Ingresa el nombre del alumno: ")
52     crear_respaldo.write(nombre)
53     crear_respaldo.write('\n')
54     crear_respaldo.close()
55     crear_respaldo = open('respaldo.txt', 'a')
56     numero_lista = int(input("Ingresa el numero de lista: "))
57     crear_respaldo.write(str(numero_lista))
58     crear_respaldo.write('\n')
59     crear_respaldo.close()
60     crear_respaldo = open('respaldo.txt', 'a')
61     calificacion = int(input("Ingresa la calificación: "))
62     crear_respaldo.write(str(calificacion))
63     crear_respaldo.write('\n')
64     crear_respaldo.write('\n')
65     crear_respaldo.close()
66     lista = open("lista.txt", "a")
67     lista.write(nombre)
68     lista.write('\n')
69     lista.write(str(numero_lista))
70     lista.write('\n')
71     lista.write(str(calificacion))
72     lista.write('\n')
73     lista.write('\n')
74     lista.close()
75     os.system("cls")
76     os.remove('respaldo.txt')
77     print("Alumno registrado...\n")
```

La primera opción es agregar un nuevo alumno, en esta opción se crea un archivo de respaldo ya que aquí es donde se guarda la información en caso de una interrupción de la ejecución del programa. Después se van solicitando los datos del alumno, uno a uno, y se van guardando dentro de este archivo de respaldo por si llega a cerrarse esta ejecución no se pierdan los datos que se han ingresado. Si se completan todos los datos y la ejecución continua, se guarda el alumno en la lista y

el archivo de respaldo es eliminado. En caso de que se cierre la aplicación o se interrumpa la ejecución de esta la información que se estaba ingresando va a quedar guardada en un archivo de respaldo y al momento de volver a ejecutar la aplicación, se va a detectar el respaldo y se realizará la restauración de el estado de ejecución para seguir completando el formulario.

```
78 elif opcion == 2:
79     os.system("cls")
80     lista = open("lista.txt", "r")
81     lineas = lista.readlines()
82     if len(lineas) > 0:
83         for linea in lineas:
84             if linea != "\n":
85                 print(linea)
86     lista.close()
87 else:
88     os.system("cls")
89     print("No hay alumnos registrados...\n")
90 elif opcion == 3:
91     os.system("cls")
92     input("Programa Finalizado...")
93 else:
94     os.system("cls")
95     print("Opcion Incorrecta, vuelva a intentar\n")
```

La segunda opción de realizar en el menú es ver a los alumnos registrados, donde se recorren las líneas del archivo de la lista de alumnos y se imprimen en pantalla.

La tercera opción es salir de menú, donde solamente se muestra un mensaje y se finaliza el programa al dar enter.

Captura de pantalla al ingresar correctamente un alumno:

```
C:\Windows\py.exe
Ingresa el nombre del alumno: Juan
Ingresa el numero de lista: 24
Ingresa la calificación: 100
```

```
C:\Windows\py.exe
Alumno registrado...

Selecciona la opción deseada:
1.-Agregar alumno
2.-Ver lista de alumnos
3.-Salir
```

Captura de pantalla al recuperar los datos:

```
C:\Windows\py.exe
Ingresa el nombre del alumno: Aldo
Ingresa el numero de lista: 1
Ingresa la calificación: _
```

```
C:\Windows\py.exe
Datos restaurados

Nombre: Aldo

Numero de lista: 1

Ingresa la calificación: _
```

```
C:\Windows\py.exe
Alumno registrado...

Selecciona la opción deseada:
1.-Agregar alumno
2.-Ver lista de alumnos
3.-Salir
```

Captura de pantalla de la opción para ver alumnos registrados:

```
C:\Windows\py.exe
Oscar
80
100
Arantza
10
95
Ale
2
100
Juan
24
100
Aldo
1
100
```

Link de repositorio en GitHub para revisar el código:

<https://github.com/oscarevanilson/Computacion-Tolerante-a-Fallas>

## **Conclusiones**

Para la realización de esta actividad tuve que investigar acerca de la restauración de un estado de ejecución, y la verdad es que no pude encontrar nada de información acerca de este tema, por lo que tuve que acercarme a mi profesor para preguntarle acerca de lo que el programa debía de hacer, gracias a esto tuve la idea de hacer un formulario.

Al momento de estar programando pensé inicialmente en utilizar la herramienta pickle pero cada vez que utilizaba alguna función de esta librería tenía errores, por lo que decidí hacerlo más sencillo con el manejo de ficheros, que realmente hace una función similar a esta herramienta.

Al final de esta tarea me he puesto a reflexionar y ahora creo que no es nada complicado realizar respaldos de información o restaurar un estado de ejecución y nunca lo había tomado en cuenta en otros programas o tareas durante mi carrera en la universidad, y de verdad me parece muy importante y me llama la atención lo útil que puede llegar a ser en cosas o programas de mayor dificultad o importancia para un futuro.

Gracias a esta tarea estoy más consciente de lo que se debe de hacer para guardar la información y evitar la pérdida de datos por cualquier error en un programa y me ha quedado más claro lo que hemos visto en clase esta semana.