



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

COMPUTACIÓN TOLERANTE A FALLAS – Do6 – 2024A

MICHEL EMANUEL LOPEZ FRANCO

Navarro Velazquez Andres – 218127792

Fecha: 05/02/2024



En este programa se utiliza la función de pickle, esta puede crear un archivo el cual restaura los datos ya introducidos, aquí se puede ver como se utiliza, con varias estructuras, la primera se declaran 2 variables, después otra para guardar el estado y la última para cargar el estado, ya por último en el programa se les da valores a las variables y se guardan en el archivo, después se les cambia de valor a esas variables pero como ya guardamos los datos anteriores los restauramos como se asignaron en un principio.

```
import pickle

class EstadoEjecucion:
    def __init__(self):
        self.variable1 = None
        self.variable2 = None

def guardar_estado(estado, archivo):
    with open(archivo, 'wb') as f:
        pickle.dump(estado, f)
    print(f'Estado guardado en {archivo}')

def cargar_estado(archivo):
    with open(archivo, 'rb') as f:
        estado = pickle.load(f)
    print(f'Estado cargado desde {archivo}')
    return estado

if __name__ == "__main__":

    estado_actual = EstadoEjecucion()
    estado_actual.variable1 = 'Hola'
    estado_actual.variable2 = 50

    guardar_estado(estado_actual, 'estado.pkl')

    estado_actual.variable1 = 'Nuevo valor'
    estado_actual.variable2 = 100

    estado_recuperado = cargar_estado('estado.pkl')

    print('Estado recuperado:')
    print(f'Variable 1: {estado_recuperado.variable1}')
    print(f'Variable 2: {estado_recuperado.variable2}')
```

```
import pickle

class EstadoEjecucion:
    def __init__(self):
        self.variable1 = None
        self.variable2 = None

def guardar_estado(estados, archivo):
    with open(archivo, 'wb') as f:
        pickle.dump(estados, f)
    print(f'Estado guardado en {archivo}')

def cargar_estado(archivo):
    with open(archivo, 'rb') as f:
        estado = pickle.load(f)
    print(f'Estado cargado desde {archivo}')
    return estado

if __name__ == "__main__":

    estado_actual = EstadoEjecucion()
    estado_actual.variable1 = 'Hola'
    estado_actual.variable2 = 50

    guardar_estado(estado_actual, 'estado.pkl')



    estado_actual.variable1 = 'Nuevo valor'
    estado_actual.variable2 = 100

    estado_recuperado = cargar_estado('estado.pkl')

    print('Estado recuperado:')
    print(f'Variable 1: {estado_recuperado.variable1}')
    print(f'Variable 2: {estado_recuperado.variable2}')
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Boolean ignoreIfNoAction, Object arg)
    en Microsoft.PowerShell.PSConsoleReadLine.InputLoop()
    en Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace, EngineIntrinsics engineIntrinsics)---
PS C:\Users\Andre\OneDrive\Escritorio\Escuela\computacion tolerante a fallas\programas tolerante a falla> & C:
te a fallas/programas tolerante a fallas/Restaurar.py"
Estado guardado en estado.pkl
Estado cargado desde estado.pkl
Estado recuperado:
Variable 1: Hola
Variable 2: 50
PS C:\Users\Andre\OneDrive\Escritorio\Escuela\computacion tolerante a fallas\programas tolerante a fallas>
```

Nombre	Fecha de modificación	Tipo	Tamaño
 estado.pkl	05/02/2024 02:04 p. m.	Archivo PKL	1 KB
 Restaurar.py	05/02/2024 01:57 p. m.	Python File	1 KB

Conclusión:

Esta actividad de realizar un programa que implemente una restauración de los datos me ayudo a comprender como funciona el método pickle el cual me ayudo principalmente a cumplir el objetivo de la actividad, yo desconocía esta función aun que con algo de investigación pude darme cuenta que no es del todo seguro.