



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías

Serialización

Alumno: Bryan De Anda Reyes

Código de alumno: 216195537

Materia: Computación Tolerante a Fallas (D06)

Horario: lunes y miércoles (11:00 – 13:00)

Carrera: INCO

Fecha de entrega: 21 de febrero de 2022

Este código está realizado en **python** para hacer manejo de la serialización con ayuda de la librería “pickle” para no perder la información que se va ingresando, guardando la información de manera que, cuando se desea recuperar esos datos podamos tenerlos.

Para este ejemplo creé una clase de tipo Pelicula, la cual servirá para ir pidiendo la información de cada película nueva que se desea ingresar, para la clase se definió el constructor con los siguientes atributos; nombre, género, director y protagonista de la película. También utilice un método especial para convertir los objetos en una cadena de texto.

```
import pickle

class Peliculas:
    def __init__(self, nombre, genero, director, protagonista):
        self.nombre = nombre
        self.genero = genero
        self.director = director
        self.protagonista = protagonista
        print("\nSe a agregado una película nueva con el nombre de: ", self.nombre, "\n")

    def __str__(self):
        return "{} {} {} {}".format(self.nombre, self.genero, self.director, self.protagonista)
```

Como se pretende que mientras se van ingresando más películas se vayan agregando a un archivo externo para no perder esa información, lo que hice fue crear una lista para almacenar los objetos que se van a guardar en el archivo y de la misma manera, poder obtener esa información de una manera más sencilla. Para esto cree una clase llamada “ListaPeliculas”, dentro de la clase inicialice una lista llamada “películas” la cual servirá para lo antes mencionado.

```
class ListaPeliculas:

    peliculas = []
```

Dentro de esta clase creé un método llamado “AgregarPeliculas” que recibe un parámetro de tipo Peliculas, agregué lo que recibe la función a la lista (peliculas), después se manda a llamar la función GuardarPeliculasExterno que se explica su

funcionamiento más adelante. También tenemos la función “MostrarPelículas”, esta va recorriendo e imprimiendo en pantalla la lista que guarda las películas.

```
def AgregarPelículas(self, p):  
    self.peliculas.append(p)  
    self.GuardarPelículasExterno()  
  
def MostrarPelículas(self):  
    for p in self.peliculas:  
        print(p)
```

Para leer la información en un fichero externo creé un constructor justo después de la lista de películas dentro de la clase “ListaPelículas”, cada vez que instanciamos la clase el constructor se encargara de abrir el archivo y pasar la información a la lista con ayuda de la función “pickle.load()”.

```
def __init__(self):  
    listaDePelículas = open("ficheroExterno", "ab+")  
    listaDePelículas.seek(0)  
  
    try:  
        self.peliculas=pickle.load(listaDePelículas)  
    except:  
        print("El fichero esta vacio")  
    finally:  
        listaDePelículas.close()  
        del(listaDePelículas)
```

Después tenemos la función de “GuardarPelículasExterno” que guarda la información en el archivo externo con ayuda de la función “pickle.dump()” que recibe como parámetros lo que se va a guardar y donde se va a guardar.

```
def GuardarPelículasExterno(self):  
    listaDePelículas = open("FicheroExterno", "wb")  
    pickle.dump(self.peliculas, listaDePelículas)  
    del(listaDePelículas)
```

La función para imprimir en pantalla las películas, recorre la lista mostrando cada una de ellas.

```
def MostrarInfoFicheroExterno(self):  
    print("\nLa información de las películas almacenadas es: \n")  
    for p in self.peliculas:  
        print("Nombre: ", p.nombre)  
        print("Genero: ", p.genero)  
        print("Director: ", p.director)  
        print("Protagonista: ", p.protagonista, "\n")
```

Para buscar una película, está compara los nombres de cada película con el ingresado por el usuario.

```
def BuscarPeliculaEnElFichero(self, nom):  
    for p in self.peliculas:  
        if p.nombre == nom:  
            print("\nPelicula encontrada: ")  
            print("Nombre: ", p.nombre)  
            print("Genero: ", p.genero)  
            print("Director: ", p.director)  
            print("Protagonista: ", p.protagonista, "\n")  
            break
```

Al final tenemos el menú de opciones que se repite hasta que el usuario ingrese un "0".

```
miLista = ListaPeliculas()  
while True:  
    try:  
        print("----Menú de almacen de peliculas----")  
        print("1) Ingresar una nueva pelicula")  
        print("2) Mostrar todas las peliculas")  
        print("3) Buscar una pelicula por su nombre")  
        print("0) Salir")  
        op = int(input("Ingrese una opcion: "))  
        if op == 1:  
            nomb = input("Ingresa el nombre de la pelicula: ")  
            gen = input("Ingresa el genero de la pelicula: ")  
            dire = input("Ingresa el director de la pelicula: ")  
            prota = input("Ingresa el protagonista de la pelicula: ")  
            peli = Peliculas(nomb, gen, dire, prota)  
            miLista.AgregarPeliculas(peli)  
        elif op == 2:  
            miLista.MostrarInfoFicheroExterno()  
        elif op == 3:  
            nomb = input("Ingresa el nombre de la pelicula a buscar: ")  
            miLista.BuscarPeliculaEnElFichero(nomb)  
        elif op == 0:  
            break  
    except:  
        print("\nLa opción no se encuentra en el menu\n")
```

A continuación, se muestra la ejecución del programa:

1. Ingresar una película:

```
----Menú de almacen de peliculas----
1) Ingresar una nueva pelicula
2) Mostrar todas las peliculas
3) Buscar una pelicula por su nombre
0) Salir

Ingrese una opcion: 1

Ingresa el nombre de la pelicula: Fragmentado

Ingresa el genero de la pelicula: Suspenso

Ingresa el director de la pelicula: M. Night Shyamalan

Ingresa el protagonista de la pelicula: Kevin Wendell Crumb

Se a agregado una pelicula nueva con el nombre de: Fragmentado
```

2. Mostrar todas las películas:

```
----Menú de almacen de peliculas----
1) Ingresar una nueva pelicula
2) Mostrar todas las peliculas
3) Buscar una pelicula por su nombre
0) Salir

Ingrese una opcion: 2

La información de las peliculas almacenadas es:

Nombre: Fragmentado
Genero: Suspenso
Director: M. Night Shyamalan
Protagonista: Kevin Wendell Crumb

Nombre: Iron Man 3
Genero: Accion
Director: Shane Black
Protagonista: Iron Man
```

3. Cerrar el programa y volver a mostrar las películas

```
----Menú de almacen de peliculas----
1) Ingresar una nueva pelicula
2) Mostrar todas las peliculas
3) Buscar una pelicula por su nombre
0) Salir

Ingrese una opcion: 0

In [2]:
```

```

In [2]: runfile('C:/Users/Bryan De Anda/OneDrive/
Programa2/Serializacion.py', wdir='C:/Users/Bryan
tolerante a fallas/Programa2')
---Menú de almacen de peliculas---
1) Ingresar una nueva pelicula
2) Mostrar todas las peliculas
3) Buscar una pelicula por su nombre
0) Salir

Ingrese una opcion: 2

La información de las peliculas almacenadas es:

Nombre: Fragmentado
Genero: Suspenso
Director: M. Night Shyamalan
Protagonista: Kevin Wendell Crumb

Nombre: Iron Man 3
Genero: Accion
Director: Shane Black
Protagonista: Iron Man

---Menú de almacen de peliculas---
1) Ingresar una nueva pelicula
2) Mostrar todas las peliculas
3) Buscar una pelicula por su nombre
0) Salir

```

Con esto podemos comprobar que la información fue respaldada correctamente.

4. Buscar una película

```

---Menú de almacen de peliculas---
1) Ingresar una nueva pelicula
2) Mostrar todas las peliculas
3) Buscar una pelicula por su nombre
0) Salir

Ingrese una opcion: 3

Ingresa el nombre de la pelicula a buscar: Fragmentado

Pelicula encontrada:
Nombre: Fragmentado
Genero: Suspenso
Director: M. Night Shyamalan
Protagonista: Kevin Wendell Crumb

```

Link al código en el repositorio: <https://github.com/BryanDeAnda/Serializacion.git>

Conclusión

Finalizando con esta practica y con su respectivo reporte, me gustaría decir que no había considerado que hacer en los programas cuando se va la luz o por accidente terminas la ejecución sin antes haber respaldado de la información y en muchas ocasiones puede llegar a ser información que llevo mucho tiempo capturarla, por esta razón cuando se crean programas en los cuales se va a ingresar información importante, es necesario constantemente respaldarla.