

**EVIDENCIA 1**

Nombre de la Unidad de Aprendizaje: estructura de datos y su procesamiento

Nombre del proyecto: SISTEMA DE ALUMNOS

Programa educativo: Licenciado en Tecnologías de información

Semestre: 3 Grupo: 31

Nombre del maestro: FRANCISCO SALAZAR

Nombre del estudiante:

RANGEL GASPAR FRANCISCO JAVIER - 1805458

San Nicolás de los Garza, ciudad universitaria a (4/11/2020)

**INTRODUCCION Y PRESENTACION DEL PROYECTO**

***Español***

Este sencillo proyecto es un menú en el que se usara una clase con las diferentes funcionalidades que se requieren para llevar acabo las manipulaciones de datos ya sea agregar un nuevo registro, mostrar la lista de los mismos, modificarlos y por ultimo salir del sistema.

Se hará uso de listas, estructuras “for”, “while” e “if”.

***Inglés***

This simple project is a menu in which a class will be used with the different functionalities that are required to carry out the data manipulations, whether to add a new record, show the list of them, modify them and finally exit the system.

It will make use of lists, structures "for", "while" and "if".

**OBJETIVOS Y REQUERIMIENTOS**

OBJETIVO

El objetivo de este proyecto es darle la facilidad a los docentes que impartan clases a un pequeño grupo de alumnos al momento de dar calificaciones, ya que se sacara el promedio de cada alumno y se guardaran los datos (Nombre completo, Matricula, Promedio final)en un archivo de tipo TXT.

El promedio es el dato que se podrá modificar para corregir posibles errores de parte del docente o que el alumno le haga saber que hay algo mal y pida revisión.

REQUERIMIENTOS FUNCIONALES POR CUMPLIR

Se necesitarán registrar alumnos antes de tratar de hacer alguna modificación o verificar el listado de registros.

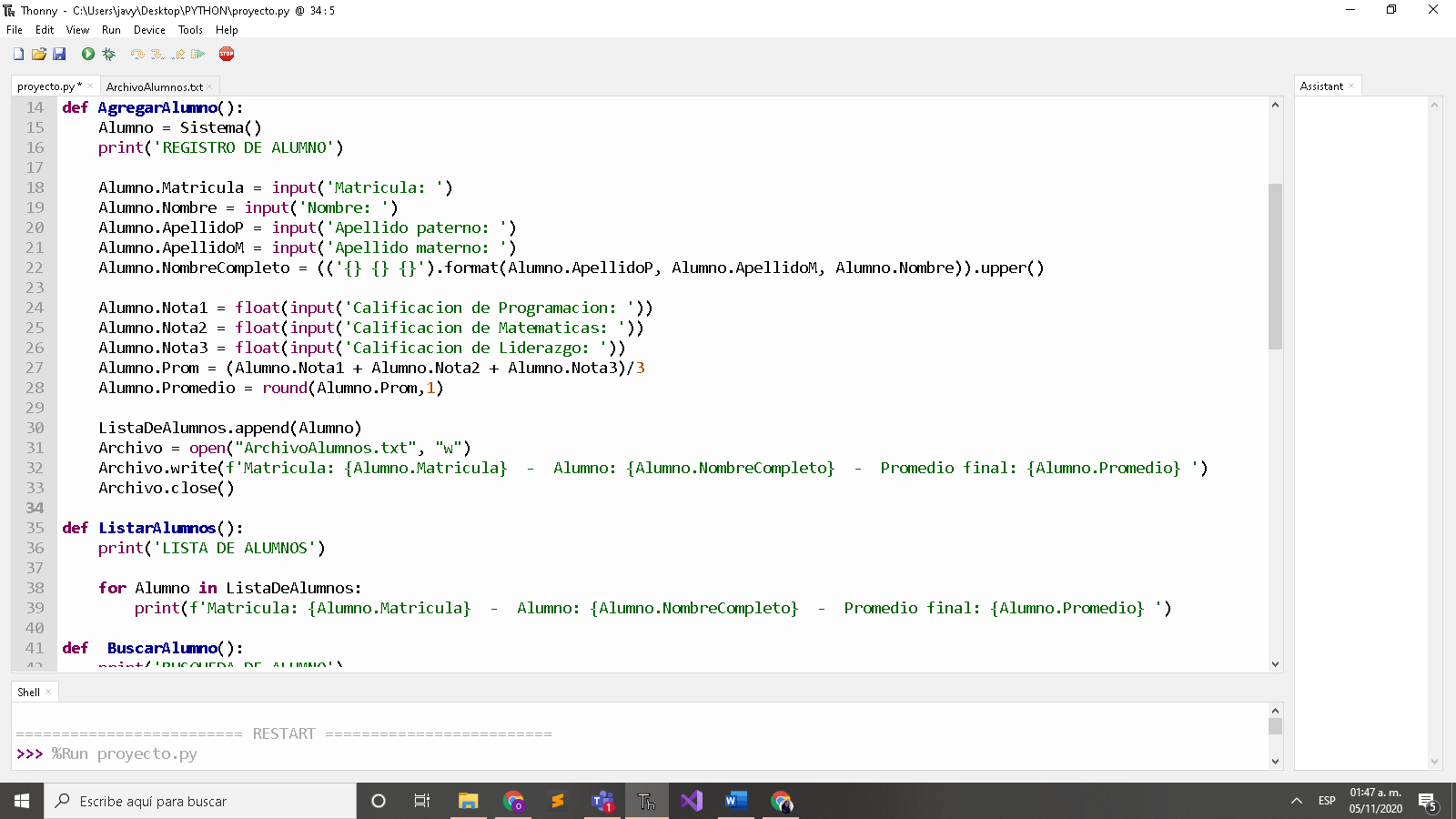
El archivo txt se creará una vez compilado el código.

**LIBRERIAS**

USO DE LIBRERIAS DE PYTHON PARA ACCEDER A ARCHIVOS

***MOVER ARCHIVOS***

* En la siguiente imagen podremos ver como después de consultar los datos y calificaciones de 3 materias se concatenará en orden de matricula, nombre completo y promedio final para escribirlo en un archivo llamado “ArchivoAlumnos.txt”, ese archivo nos servirá como sustituto de una base de datos



* Se tendrá la opción 5 en el menú que se encargará de eliminar los registros ya hechos, a la lista solo se le dará un valor de lista vacía y se borrara el archivo txt del filesystempara eso se necesitó importar el modulo os

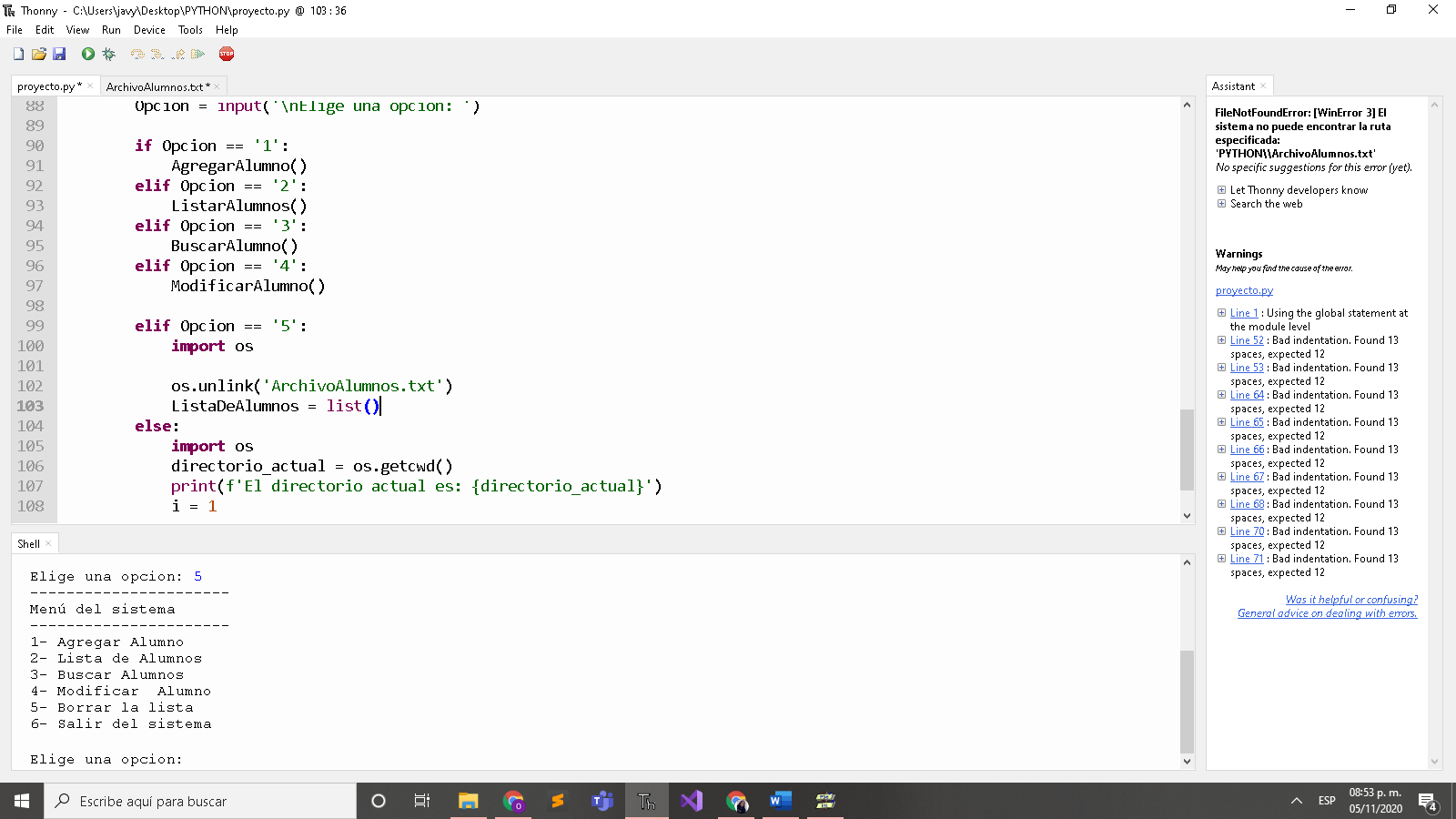
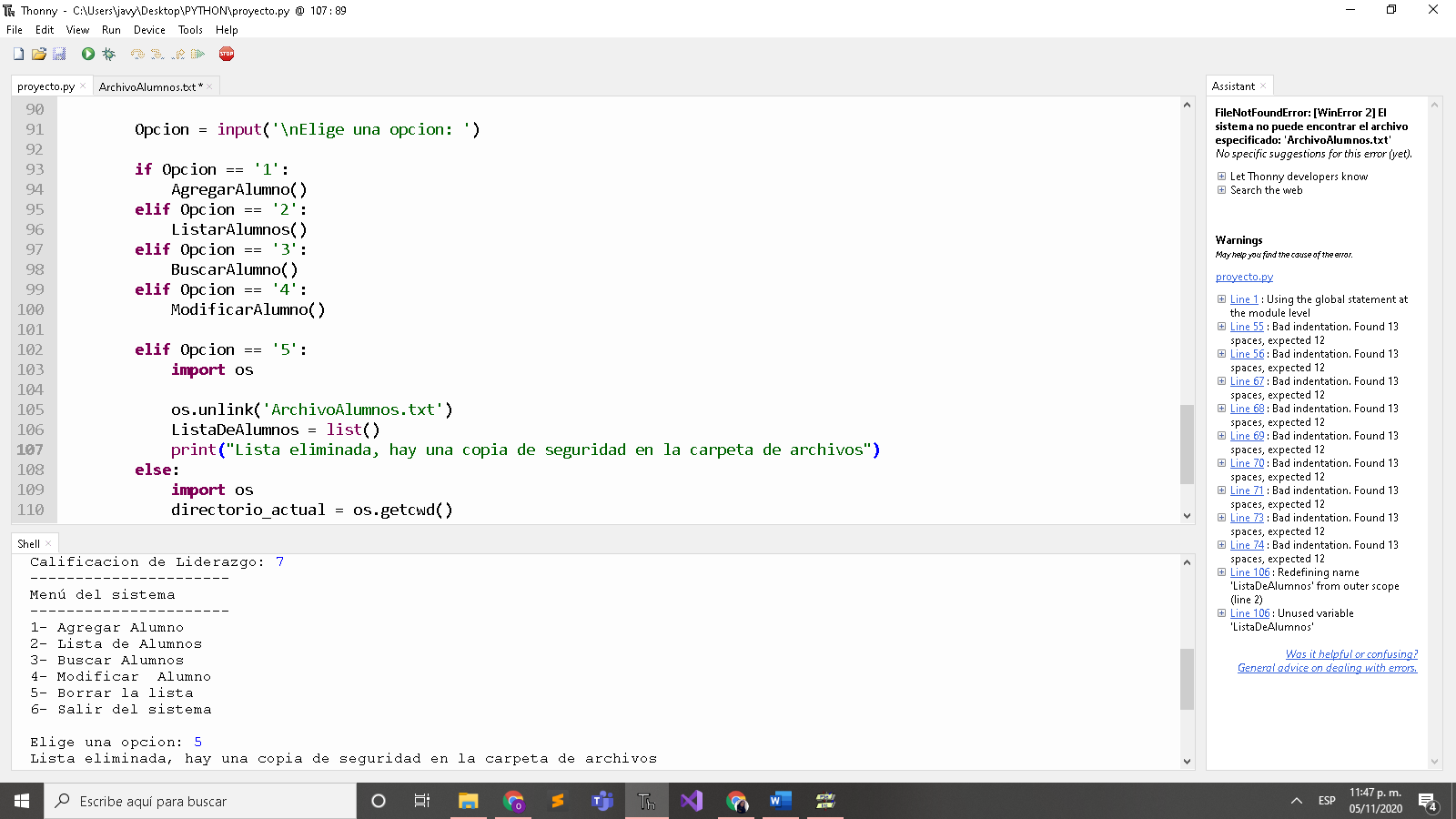


TABLA DE DATOS

|  |  |  |  |
| --- | --- | --- | --- |
| ID | TIPO | | DESCRIPCION |
| Archivo | CURSOR | | Nos sirve para llevar a cabo las instrucciones que se llevaran a acabo una vez abierto el “ArchivoALumnos.txt” |
| Opción | String | | Es la opción que será ingresada por el usuario para hacerle saber al sistema que es lo que quiere realizar |
| ListaDeAlumnos | List | Es la lista en la cual se almacenaran los registros, la lista nos permitirá acceder a varias funcionalidades útiles en este proyecto | |

PSEUDOCODIGO 1

1. Inicio
2. Se abre el archivo (ArchivoAlumnos.txt) y se le indica que será para escribir con una “w”, a la instrucción se le asignara el identificador con el nombre “Archivo”

El cual nos servirá de cursor

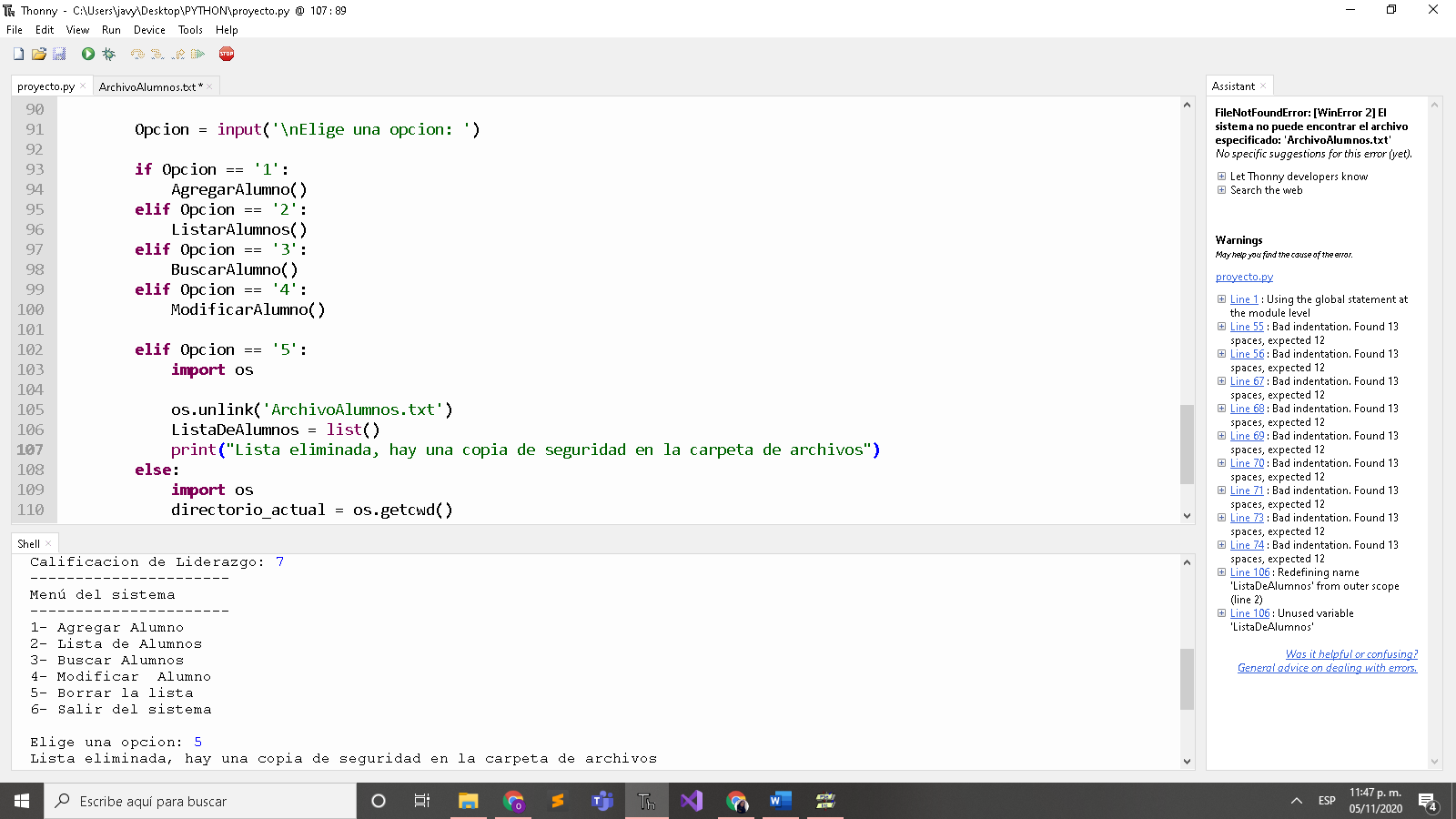
1. Con el cursor archivo daremos la instrucción write() para que nos escriba todos los datos anteriormente solicitados al usuario, el destino a donde se mandará la cadena será el “ArchivoAlumnos.txt”
2. Se le dará la instrucción close() para cerrar el archivo y evitar el consumo de recursos.
3. Fin

PSEUDOCODIGO 2

1. Inicio
2. Se importa el modulo os
3. Se usara la función unlink para remover el archivo
4. Se le asignara a la lista el valor que tenia desde un inicio, perdiendo así los registros.
5. Se envía un mensaje al usuario para hacerle saber que la lista se elimino con éxito y que contara con una copia de seguridad
6. Fin

MANUAL DE OPERACIÓN DEL USUARIO

Es muy sencillo, al empezar solo tendrás que seleccionar un numero del 1 al 6 dependiendo lo que quieras hacer, en este caso seleccionaremos el 5 que es para borrar la lista de alumnos y automáticamente se le dará un valor nulo a la lista y el archivo se eliminará.



***COPIAR ARCHIVOS***

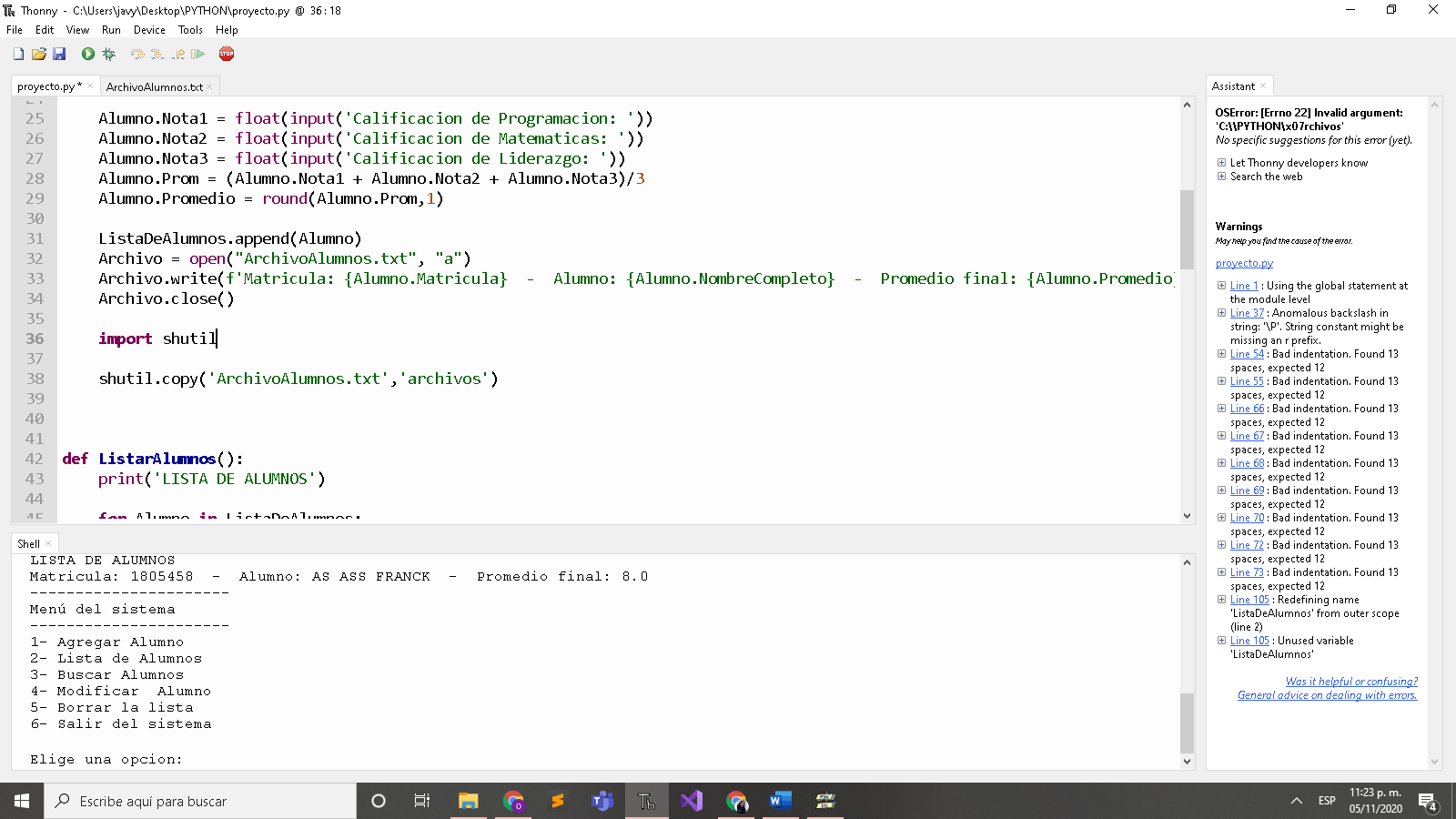
* Se hará una copia de seguridad de el archivo txt cuyo destino será una carpeta llamada archivos. Esto para tener a la mano un historial de cambios en caso de que se necesite

TABLA DE DATOS

|  |  |  |
| --- | --- | --- |
| ID | TIPO | DESCRIPCION |
| archivos | carpeta | Es un directorio donde se almacenaran los archivos de tipo texto |

PSEUDOCODIGO

1. Inicio
2. Se importa sutil
3. Se usara el método copy para copiar el archivo deseado
4. Se agregan los dos parámetros el primero es el archivo txt y el segundo la carpeta de destino
5. Fin

**ESTRUCTURAS DE DATOS NATIVAS**

**APLICACIÓN DE LISTAS COMO ESTRUCTURAS DE DATOS NATIVAS**

Los datos almacenados en las listas nos ayudaran a sacar el mejor promedio

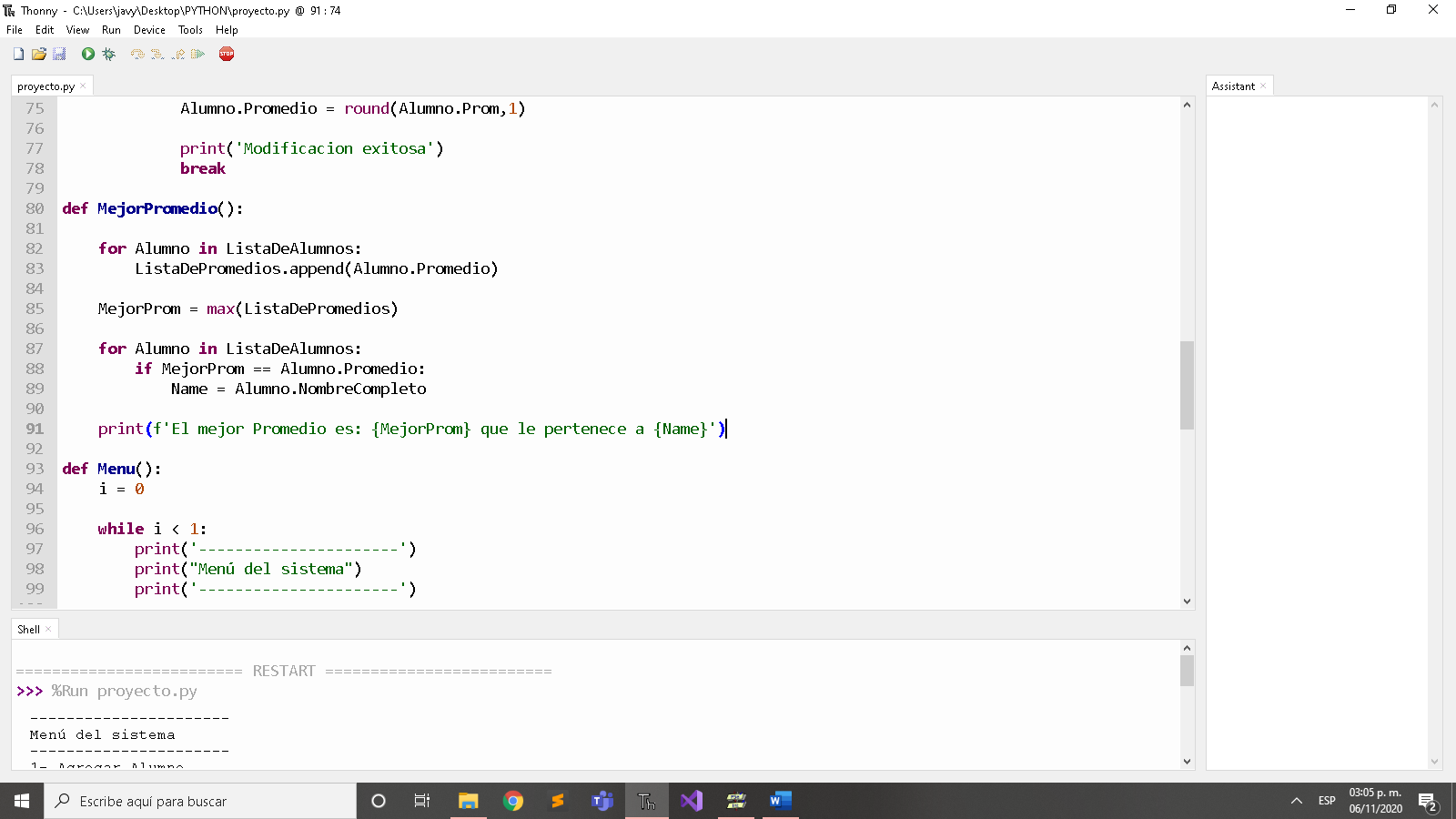


TABLA DE DATOS

|  |  |  |
| --- | --- | --- |
| ID | TIPO | DESCRIPCION |
| ListaDeAlumnos | list | Lista donde se guardaran los registros de alumnos |
| ListaDePromedios | list | Lista donde se almacenarán los promedios registrados |
| MejorProm | float | Aquí se pondrá el valor máximo entre los promedios registrados en la lista de promedios |
| Name | String | Aquí se recoge el nombre completo del alumno con el mejor promedio |

PSEUDOCODIGO

* Inicio
* Por cada alumno en la ListaDeAlumnos

-Agregar a la ListaDePromedios el (Alumno.Promedio)

- MejorProm = valormaximo(ListaDePromedios)

- Por cada alumno en la ListaDeAlumnos hacer

-Si MejorProm == Alumno.Promedio entonces

- Name = Alumno.NombreCompleto

- Escribir(‘El mejor promedio es MejorProm y le pertenece a Name’)

**APLICACIÓN DE TUPLAS COMO ESTRUCTURAS DE DATOS NATIVAS**

Use la lista para almacenarla en una variable que tendrá sus mismos datos pero ahora será una tupla y eso nos servirá para hacer las comparaciones.

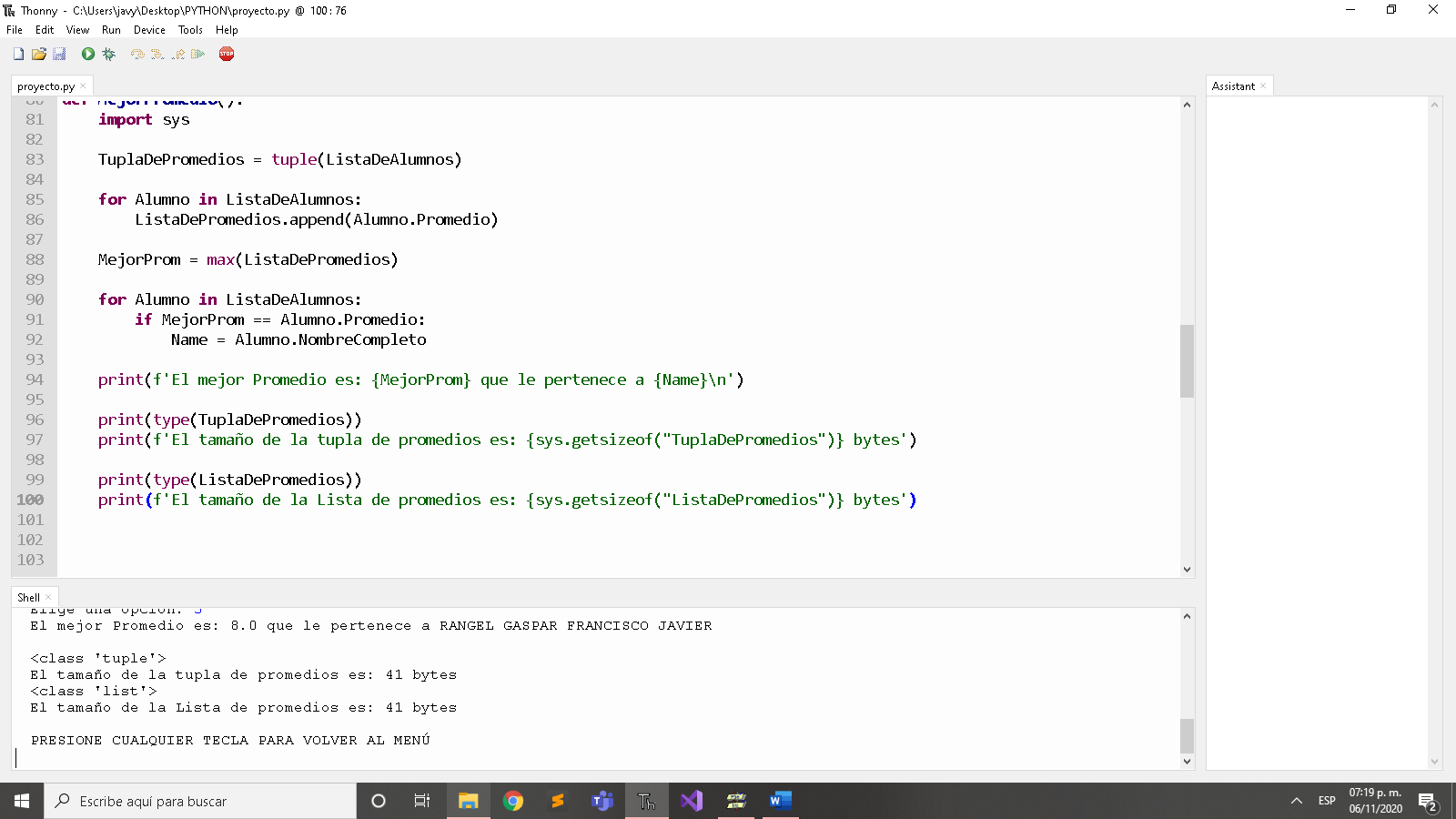
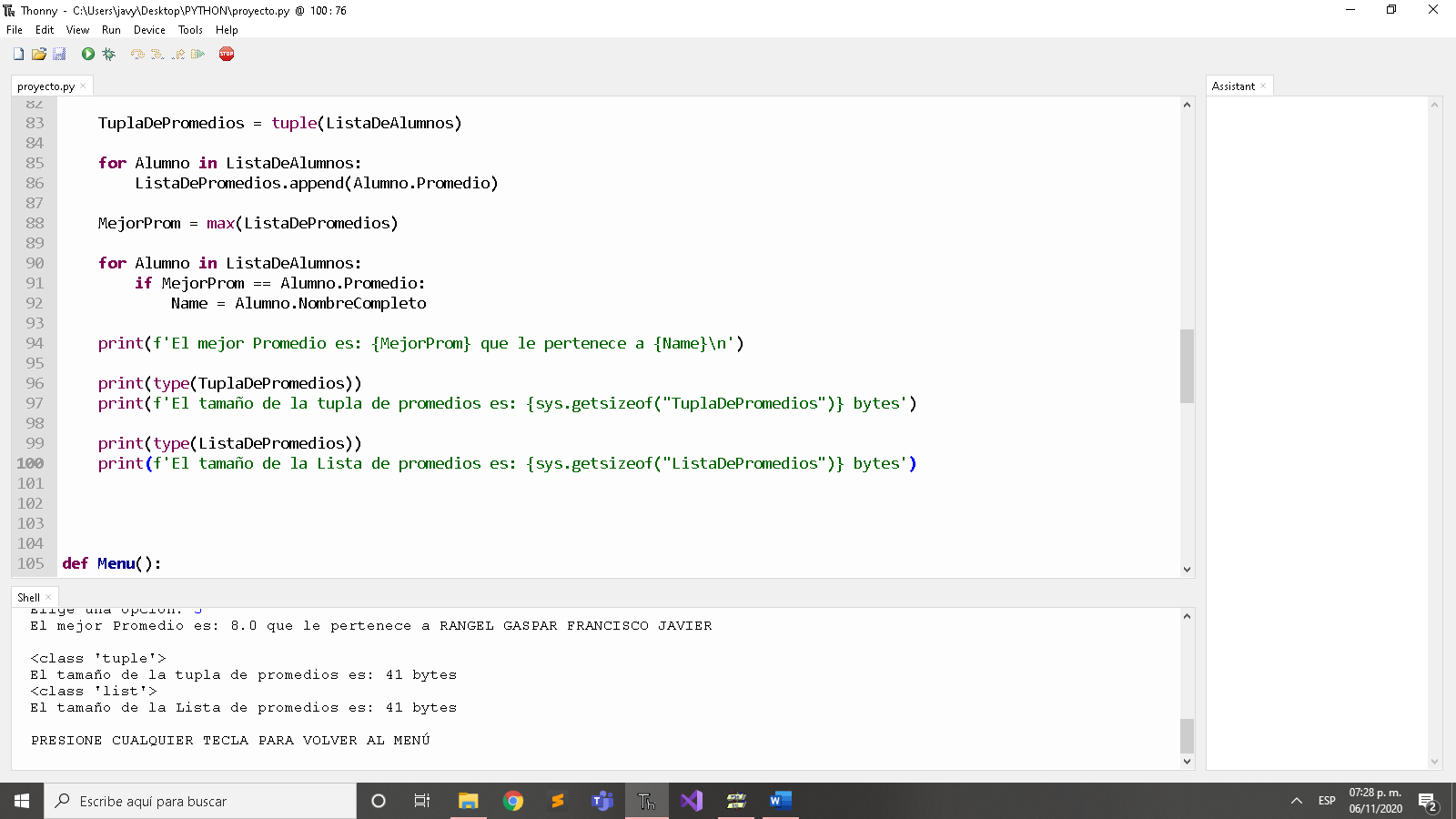
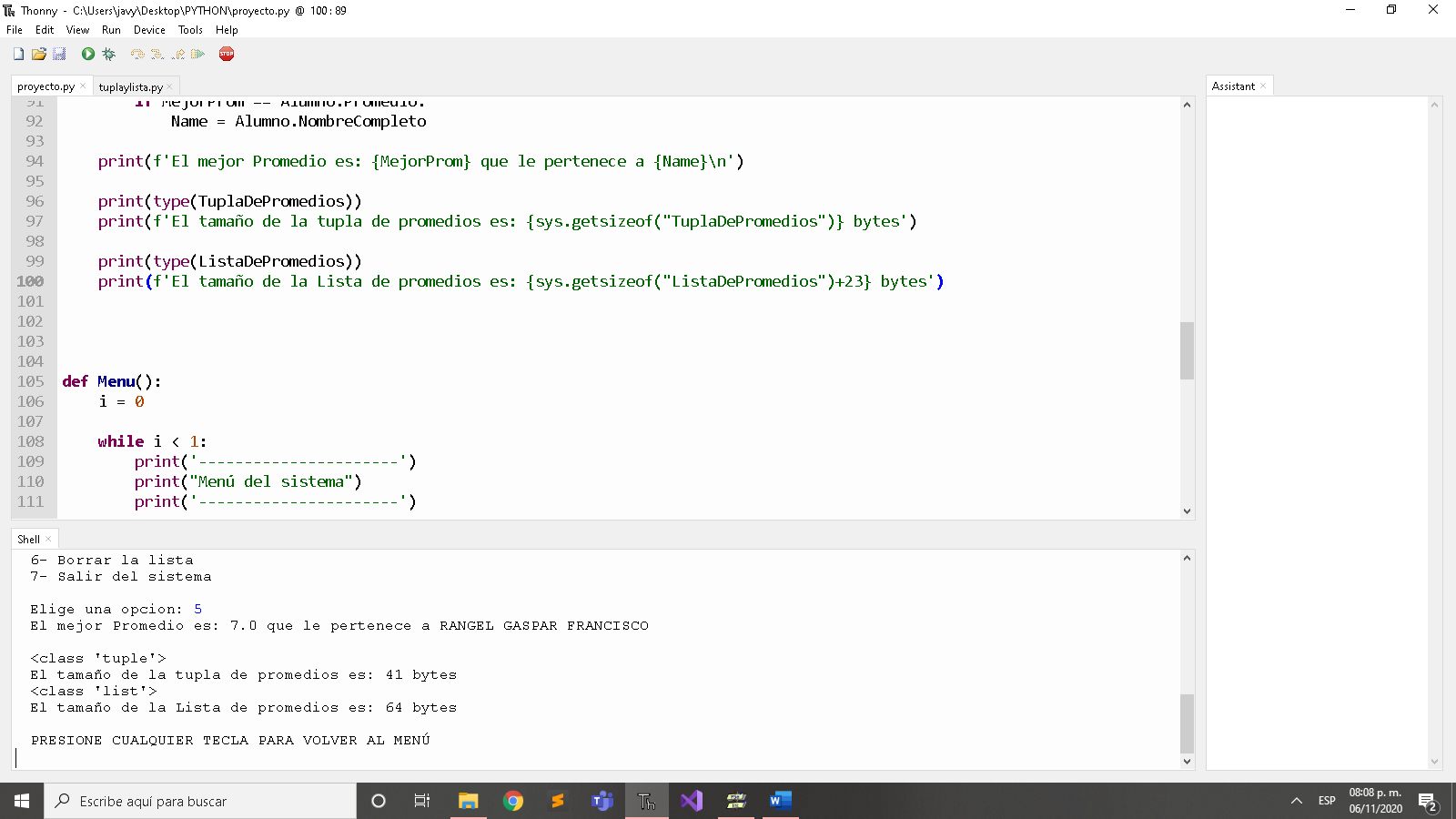


TABLA DE DATOS

|  |  |  |
| --- | --- | --- |
| ID | TIPO | DESCRIPCION |
| TuplaDePromedios | tuple | Tupla con los mismos datos que la lista |

CONSUMO DE RECURSOS LISTA VS TUPLA





El resultado va a variar dependiendo de los registros hechos en el programa , en este caso siempre será conveniente usar la lista ya que se ocupara modificar datos en todo momento

**ARGUMENTACION DE REQUERIMIENTOS FUNCIONALES**

Tenemos el uso de librerías para modificar el filesystem, movimos un archivo con todos los registros hacia una carpeta para proporcionarle al usuario una copia de seguridad en caso de que la necesite y se lo hacemos saber.

Use listas para tener acceso a los datos en todo momento y para poder moverme por cada unos de los datos dentro de ella con la estructura for.

Como se dijo con anterioridad este sencillo proyecto le será de uso a cualquiera que ocupe una pequeña ayuda a la hora de calificar a un pequeño grupo de alumnos. Hasta ahora a cumplido decentemente pero se tiene la conciencia de que no existe software perfecto, por lo que se le dará mantenimiento con frecuencia.

**APORTACIONES EN EL EQUIPO**

Lamento informar que tuve que hacer este proyecto solo, ya que nunca recibí respuestas solidas de parte de mi equipo y se tardaban en contestar, no me iba a quedar sin evidencia por ese motivo y decidí hacerlo solo. Espero su comprensión.

**CONCLUSION**

Este proyecto sirvió para darme cuenta que no ando tan bien como creía, ya que algunas cosas que creía sencillas se me complicaron un poco. También tengo que mejorar mucho en la limpieza de mi código para tener mis ideas mas ordenadas.

En lo que abarca a la evidencia también me ayudo a conocer lo que es github, el uso de repositorios y la buena utilidad que puede tener en el aprendizaje y trabajo de un desarrollador.