

# **Universidad Fidélitas**

Proyecto Programación Básica

Alumnos: SANTIAGO SUAREZ MORERA  
KEVIN PORTILLA MORALES  
AARON STEVEN AGUILERA CORDERO  
DENNIS ERICK SANCHEZ OLIVAS

## **Ingeniería en Sistemas de Computación**

Programación Básica (Introducción a la Programación)

Profesor: Sebastián Chaves Bermúdez

8 Marzo 2023

## **Introducción**

Se presenta una situación donde un hotel conocido como “*Hotel Verde Ujarras S.A*” necesita que algunos desarrolladores estructuren una aplicación que les permite llevar un récord de los tours que ofrece dicho hotel para facilitar el manejo de información y poder brindársela a los clientes. La problemática de este trabajo es crear dicha aplicación con todos los objetivos enseñados en clase como funciones, arreglos, etc.

En este proyecto se desea investigar y desarrollar una aplicación en el lenguaje de programación conocido como Python. Esta aplicación resolvería la problemática presentada en el proyecto de estudio brindado en la clase. Esto nos ayuda a entender el enfoque escogido y nuestra solución a la problemática, por medio de varios métodos aplicados y estudiados durante clase.

## **Objetivo General**

- Identificar la importancia del uso de Python a la hora de crear nuestro aplicativo de consola.

## **Objetivos específicos**

- Determinar el significado y ejemplos específicos del uso de Python en la ingeniería en sistemas.
- Exponer las ventajas y desventajas que presenta el uso de Python en los diferentes campos de la ingeniería en sistemas.
- Analizar las aplicaciones de Python en el campo laboral actual de un ingeniero en sistemas.

## **Justificación**

Este proyecto es de gran importancia para nuestro grupo, ya que, se cree que la problemática presentada asiste al aprendizaje de la temática dada en clase, e impulsa a mejorar la resolución de problemas. Además, ayuda a desarrollar las habilidades requeridas en cualquier ámbito para poder enfrentar y resolver cualquier incertidumbre en la vida cotidiana.

## **Pseudocódigo**

Después de haber realizado el programa para el caso anterior, se les presentará una explicación sobre cada parte del caso.

Se inicia el código con varios import

```
1  import os
2  import time
3  import sys
4  import shutil
5  from datetime import date
```

*Fuente: Elaboración Propia*

El import "os" hace posible utilizar funcionalidades dependientes al sistema operativo, el cual se utilizó en el programa para poder limpiar la pantalla después de utilizar otra función del programa; import "time" nos permite cronometrar la duración de un intervalo de tiempo, el cual se utiliza para darle el tiempo al usuario de leer y luego limpiar la pantalla; import "sys" provee acceso a algunas variables usadas o mantenidas por el intérprete y a funciones que interactúan fuertemente con el intérprete, se utiliza para poder terminar el programa; import "shutil" Mueve de forma recursiva un archivo o directorio a otra ubicación y retorna el destino; import "datetime" proporciona clases para manipular fechas y hora.

Se declara la función \*Opción Menú\*:

```
6  def OpcionMenu():
7      while True:
8          try:
9              opc = int(input("<----Bienvenido al TaskMaganer del Hotel Verde Ujarras---->"
10                             "\n----Cual opcion desea utilizar---- ?")
11              "\n----> 1) Visualizar Tours "
12              "\n----> 2) Opcion numero 2 "
13              "\n----> 3) Opcion numero 3 "
14              "\n----> 4) Opcion numero 4 "
15              "\n----> 5) Opcion numero 5 "
16              "\n----> 6) Salir "
17              "\n:")
18              break
19          except ValueError:
20              print("No es un numero, intentelo de nuevo")
21              time.sleep(2)
22              borrarPantalla()
23      return opc
```

*Fuente: Elaboración Propia*

Se declara un ciclo While con la condición true (Hasta que devuelva el valor de true el ciclo va a terminar). Dentro del try se declara la variable opc (Se guarda la opción del usuario) si el usuario ingresa un valor incorrecto que no sea un número, tenemos el except "ValueError" en este hace un print con la cadena de texto "No es número, reintente" utilizamos la función \*time.sleep\* para que el usuario pueda leer el texto antes de que se ejecute la función borrarPantalla() y por último si el valor que ingreso el usuario es correcto devolvemos la variable opc.

Se declara la función borrarPantalla:

```
31  #Funcion para borrar pantalla segun sistema operativo
32  def borrarPantalla():
33      if os.name == "posix":
34          os.system ("clear")
35      elif os.name == "ce" or os.name == "nt" or os.name == "dos":
36          os.system ("cls")
```

*Fuente: Elaboración Propia*

En esta iniciamos con un `"if"`, el cual está preguntando cuál es el nombre del sistema operativo según este va a utilizar el código respectivo para limpiar la pantalla del intérprete.

Se declara la función finalizar:

```
38  #Funcion para preguntar si desea utilizar otra funcion
39  def finalizar(var):
40      if var == None:
41          var = str(input("Desea utilizar otra funcion ? \n Y/N \n:"))
42
43      if var == "Y" :
44          borrarPantalla()
45          Main()
46
47      elif var == "N":
48          print("Salio del sistema")
49          print("Saliendo en 1...2..")
50          time.sleep(2)
51          borrarPantalla()
52          sys.exit()
53      else:
54          print("Opcion no valida")
55          time.sleep(2)
56          borrarPantalla()
57          finalizar(None)
```

*Fuente: Elaboración Propia*

Después de cada opción otorgada, se les presentará con la función `*finalizar*`, donde le permite al usuario volver a recibir el menú de opciones o terminar el proceso. La variable `*var*` se iguala al parámetro `None` para que se le pueda preguntar al usuario que desea hacer. `var = str(input("..."))` le presenta la opción de regresar a los menús, o cerrar el programa. En la primera condicional, si este ingresa "Y", el usuario regresa a las opciones, y en el segundo condicional, si este ingresa "N", el programa termina. Al finalizar, si este

ingresa otro dato no correspondiente a las opciones brindadas, se ejecuta else y se le presenta con el print ("opción no válida").

Se declara la función Opcion1:

```
233 #Ver Tours (Aaron)
234 def Opcion1():
235     if ValidarCarpetaVacía("Datos") == True:
236         LeerTodosLosArchivos("Datos")
237         var = str(input("Desea utilizar otra funcion ? \n Y/N \n:"))
238         finalizar(var)
```

*Fuente: Elaboración Propia*

Se declara un \*if\* con la condición si el return de la invocación "ValidarCarpetaVacía" es igual a True va a invocar la función "LeerTodosLosArchivos" y por último invocamos a la función finalizar pasándole por parámetros el \*var\*.

Se declara la función FacturacionTours():

```
8 def FacturacionTours():
9     print("Facturando Tours..")
10    LeerTodosLosArchivos("DatosAsignados")
11    TourSeleccionado=input("Seleccione el nombre del tour que desea facturar: ")
12    if os.path.exists("DatosFacturacion/Factura" + TourSeleccionado + ".txt") :
13        print("El tour ya fue facturado")
14        time.sleep(2)
15        borrarPantalla()
16        Main()
17    else:
18        url = "DatosAsignados/" + TourSeleccionado + ".txt"
19        archivo = open(url, "r")
20        lista = archivo.readlines()
21        archivo.close()
22        urlNuevo = "DatosFacturacion/Factura" + TourSeleccionado + ".txt"
23        archivo = open(urlNuevo, "w")
24        archivo.write(lista[0])
25        archivo.write(lista[1])
26        archivo.write(lista[2])
27        archivo.write(lista[3])
28        archivo.write(lista[4])
29        archivo.write(lista[5])
30        archivo.write(lista[6])
31        archivo.write(lista[7])
32        archivo.write(lista[8])
33        archivo.write(lista[9])
34        archivo.write(lista[10])
35        archivo.write(lista[11])
```

*Fuente: Elaboración Propia*

Se hace un "print" diciéndole al usuario que se están facturando los tours , invocamos la función "LeerTodosLosArchivos" y le pasamos por parámetros "DatosAsignados" para que esta pueda leer todos los archivos que están en esa carpeta ,

declaramos “ToursSeleccionado” y utilizamos la función “input” posteriormente entra en un \*if\* con la condición de “os.path.exists” y con el url del archivo nuevo que estamos creando , si este existe le imprime al usuario que ya está facturado y vuelve al menú, si no entra al “else” donde vamos a declarar el url del tour que fue seleccionado anteriormente este lo lee y además crea una variable donde está guardando el archivo por líneas luego lo cierra , Crea el url del nuevo archivo y lo abre para poder escribir dentro de él , le escribimos al txt nuevo las 11 primeras líneas del anterior txt.

```
costosdelTour = int(lista[4])
nochesdeseadas = int(input("Ingrese las noches que desea: "))
costototal= costosdelTour * nochesdeseadas
IMPUESTO = 16
impuesto = costototal * IMPUESTO / 100
costototal = costototal + impuesto
print("El costo total es de: " + str(costototal))
cuantoPaga = int(input("Ingrese cuanto paga el cliente: "))
if cuantoPaga >= costototal:
    FechaFactura = date.today()
    cambio = cuantoPaga - costototal
    print("-----")
    print(f"Se facturo el tour con exito a {ToursSeleccionado}")
    print(f"El cambio es de {cambio}")
    print(f"Fecha de la factura: {FechaFactura}")
    archivo.write("-----" + "\n")
    archivo.write("Informacion de la factura" + "\n")
    archivo.write("Fecha de la factura: " + str(FechaFactura) + "\n")
    archivo.write("Costo total por noche: " + str(costototal) + "\n")
    archivo.write("Impuesto: " + str(impuesto) + "\n")
    archivo.write("Pago del cliente: " + str(cuantoPaga) + "\n")
    archivo.write("Cambio: " + str(cambio) + "\n")
    archivo.close()
    Continuar = input("Desea continuar ? (Precione cualquier tecla): ")
    if Continuar == "Si":
        borrarPantalla()
        Main()
    else:
        borrarPantalla()
        Main()
else:
    print("El cliente no pago lo suficiente")
    time.sleep(2)
    borrarPantalla()
    Main()
```

*Fuente: Elaboración Propia*

Declaramos la variable “costosdeltour” donde le estamos asignando la línea que contiene el valor del tour , preguntamos las noches que desea y luego hacemos el cálculo en la variable “costototal” multiplicando el “costodeltour” individual con las noches deseadas declaramos una variable constante la cual es el impuesto luego “costototal” lo multiplicamos por el impuesto y lo dividimos por 100 para saber el costo final sumamos el “costototal” más el impuesto , imprimimos el “costototal”, con la variable “cuantoPaga” hacemos la condición donde si cuantopaga es mayor o igual al costo total pasa el if si no imprime que no pagó lo suficiente , si este si lo pago lo que hace es mostrar los prints de la información de la factura y luego los escribe en el archivo y lo devuelve al menu principal.

Se declara la función AsignarTours():

```

72 #Funcion para asignar los tours deseados (Santiago / Aaron)
73 def AsignarTours():
74     print("Asignando tours...")
75     LeerTodosLosArchivos("Datos")
76     TourSeleccionado=input("Escriba el nombre del tour que desea asignar: ")
77     if os.path.exists("Datos/" + TourSeleccionado + ".txt") :
78         url = "Datos/" + TourSeleccionado + ".txt"
79         archivo = open(url, "r")
80         lista = archivo.readlines()
81         archivo.close()
82         nombreUsuario = input("Ingrese el nombre del Cliente: ")
83         apellidoUsuario = input("Ingrese el apellido del Cliente: ")
84         telefonoUsuario = input("Ingrese el telefono del Cliente: ")
85         CedulaUsuario = input("Ingrese la cedula del Cliente: ")
86         urlNuevo = "DatosAsignados/" + nombreUsuario + ".txt"
87         archivo = open(urlNuevo, "w")
88         archivo.write(lista[0])
89         archivo.write(lista[1])
90         archivo.write(lista[2])
91         archivo.write(lista[3])
92         archivo.write(lista[4])
93         archivo.write(lista[5])
94         archivo.write(lista[6])
95         archivo.write("-----" + "\n")
96         archivo.write("Informacion del usuario" + "\n")
97         archivo.write("Nombre del usuario: " + nombreUsuario + "\n")
98         archivo.write("Apellido del usuario: " + apellidoUsuario + "\n")
99         archivo.write("Telefono del usuario: " + telefonoUsuario + "\n")
100        archivo.write("Cedula del usuario: " + CedulaUsuario + "\n")
101        archivo.close()
102        print("Tour asignado con exito")
103        time.sleep(2)
104        borrarPantalla()
105        Main()
106    else:
107        print("El tour no existe")
108        time.sleep(2)
109        borrarPantalla()
110        Main()

```

*Fuente: Elaboración Propia*

Se declara un print con el texto asignando tours que le avisa en que apartado está el usuario, se llama la función “LeerTodosLosArchivos” con parámetros en datos, luego un input que le señala al usuario que acción debe de realizar con el texto Escriba el nombre del tour que desea asignar. Un if que lo que realiza es que verifica si el archivo txt existe con parámetros que le indican dónde buscar ese archivo. Se crea una url como dirección de donde se encuentra los archivos, abrimos el archivo con un open, pero solo lo leemos con la letra r y con el readlines leemos el archivo, pero lo guarda por líneas, al final cerramos el archivo.

Declaramos variables nuevas que funcionaran como información del usuario para después guardarlas. Se crea una nueva url para los datos asignados. Abrimos el archivo con letra w sé lo que hace es sobrescribir los datos, pero para no ser eliminados guardamos las líneas del archivo antes de agregar la nueva información. Agregamos los datos nuevos con los antiguos.

Al final se hace un print diciendo si el tour se asignó correctamente, si no un print diciendo que no existe el tour.

Declaramos la funcion ValidarCarpetaVacía():

```

113  def ValidarCarpetaVacía(carpeta):
114      if os.listdir(carpeta) == []:
115          if carpeta == "Datos":
116              print("Necesita agregar tours primero")
117              time.sleep(2)
118              borrarPantalla()
119              Main()
120          if carpeta == "DatosFacturacion":
121              print("No hay facturas disponibles")
122              time.sleep(2)
123              borrarPantalla()
124              Main()
125          if carpeta == "DatosAsignados":
126              print("No hay tours asignados")
127              time.sleep(2)
128              borrarPantalla()
129              Main()
130      else:
131          return True

```

*Fuente: Elaboración Propia*

La función recibe por parámetros la carpeta que quiere verificar si está vacía, entra por un if donde la condición es "os.listdir" el nombre de la carpeta y lo iguala a los paréntesis cuadrados, esto es para ver si está vacía si esta busca el nombre de la carpeta y mande el print de lo que tiene que hacer antes, si existe hace el return True.

Declaramos la función ValidarExisteTours():

```

134  def ValidarExisteTour(nombreTour):
135      if os.path.exists("Datos/" + nombreTour + ".txt") :
136          print("El tour ya existe , reintente")
137          time.sleep(2)
138          borrarPantalla()
139          AgregarTour()
140

```

*Fuente: Elaboración Propia*

En esta función recibe por parámetros el nombre del tour que vamos a revisar si existe o no, entra a un if donde la condición es "os.path.exists" y el url del tour si este existe lo mande al menú y a la función de "AgregarTour".



Declaramos la funcion ValidarExisteID():

```
141 #Funcion para validar si existe el ID que se desea agregar (Aaron)
142 def ValidarExisteElID(id):
143     for file in os.listdir("Datos"):
144         if file.endswith(".txt"):
145             archivo = open(os.path.join("Datos", file), "r")
146             for line in archivo:
147                 if id in line:
148                     print("El ID ya existe")
149                     time.sleep(2)
150                     borrarPantalla()
151                     AgregarTour()
152             archivo.close()
```

*Fuente: Elaboración Propia*

La función recibe por parámetros el ID. El for lo que hace es leer todos los archivos que están en la carpeta datos, Una condición que verifica si el archivo que se está leyendo termina en txt. Se abre el archivo verificando todos los archivos que están en datos y además que tenga la extensión txt, se lee las líneas del archivo y comprueba si el ID ya existe, si es así sale un print que indica si ya existe el ID. Al final se cierra el archivo.

Declaramos la función AgregarTour():

```

155 def AgregarTour():
156     print("Agregando un nuevo tour...")
157     id = str(input("Ingrese el numero del tour: "))
158     ValidarExisteElID(id)
159     nombreTour = str(input("Ingrese el nombre del tour: "))
160     ValidarExisteTour(nombreTour)
161     url = "Datos/" + nombreTour + ".txt"
162     archivo = open(url, "w")
163     archivo.write("-----\n")
164     archivo.write(f"ID: {id}" + "\n")
165     descripcion = str(input("Ingrese la descripcion del tour: "))
166     if descripcion == "":
167         descripcion = "N/A"
168     archivo.write("Descripcion: " + descripcion + "\n")
169     CostoPorNoche = int(input("Ingrese el costo por noche: "))
170     if CostoPorNoche == "":
171         CostoPorNoche = "N/A"
172     archivo.write("Costo por noche: \n")
173     archivo.write(f"{CostoPorNoche} \n")
174     NombreGuia = str(input("Ingrese el nombre del guia: "))
175     if NombreGuia == "":
176         NombreGuia = "N/A"
177     archivo.write("Nombre del guia: " + NombreGuia + "\n")
178     Duracion = str(input("Ingrese la duracion del tour: "))
179     if Duracion == "":
180         Duracion = "N/A"
181     archivo.write("Duracion: " + Duracion + "\n")
182     Transporte = str(input("Ingrese (si/no) si el tours tiene tran
183     if Transporte == "":
184         Transporte = "N/A"
185     archivo.write("Transporte: " + Transporte + "\n")
186     archivo.close()
187     print("Tour creado con exito")
188     time.sleep(2)
189     borrarPantalla()
190     Main()

```

*Fuente: Elaboración Propia*

Se imprime el mensaje agregando un nuevo tour y se declara la variable id y el "nombreTour" estas dos variables se les pregunta al usuario y se validan con las funciones que invocamos "ValidarExisteELID" y "ValidarExisteTours" luego de eso creamos la url del archivo con el nombre del tour que fue escrito por el usuario y lo rellenamos con información del tour como la descripción , costoponoche , nombreguia , duracion, transporte y mandamos un mensaje de éxito.

Declaramos la función LeerTodosLosArchivos():

```
192 #Funcion para Leer todos Los archivos (Santiago)
193 def LeerTodosLosArchivos(carpeta):
194     for file in os.listdir(carpeta):
195         if file.endswith(".txt"):
196             nombreTour = file[:-4]
197             print("Nombre del Tour --> " , nombreTour)
198             archivo = open(os.path.join(carpeta, file), "r")
199             print(archivo.read())
200             archivo.close()
```

*Fuente: Elaboración Propia*

La función recibe por parámetros la carpeta. El for lo que hace es leer todos los archivos que están en la carpeta y una condición que verifica si el archivo que se está leyendo termina en txt. Creamos la variable nombre tour donde la igualamos al file la cual es el nombre del tour, pero quitándole las últimas 4 letras para que aparezca sin él .txt y lo imprimimos. Abrimos el archivo con la letra r, se imprime y se cierra el archivo.

Se declara la función LeerTodosLosTitulos():

```
203 v def LeerTodosLosTitulos():
204 v     for file in os.listdir("Datos"):
205 v         if file.endswith(".txt"):
206             nombreTour = file[:-4]
207             print("Nombre del Tour --> " , nombreTour)
```

*Fuente: Elaboración Propia*

Se hace un ciclo for donde recorre los archivos de la carpeta datos y entra a un if donde la condición es que tiene que tener él .txt creamos la variable nombre tour donde la igualamos al file la cual es el nombre del tour pero quitándole las últimas 4 letras para que aparezca sin él .txt y lo imprimimos.

Se declara la función EliminarTours():

```
246  def EliminarTours(nombreTour):  
247      source = "Datos/" + nombreTour + ".txt"  
248      destination = "DatosEliminados/" + nombreTour + ".txt"  
249      shutil.copy(source, destination)  
250      print("Archivo eliminado exitosamente.")  
251      os.remove("Datos/" + nombreTour + ".txt")
```

*Fuente: Elaboración Propia*

Se para por parámetros el nombre del tour , en este creamos la variable source el cual contiene la ruta donde está el archivo y creamos el destination donde va a ir el archivo, en este caso a la carpeta de datos eliminados utilizamos la función shutil.copy con los parámetros source y destination este va a copiar el archivo que seleccionamos a otra carpeta y finalmente hacemos el os.remove para eliminar el archivo de la carpeta de datos.

Declaramos la función ModificarTour():

```

209 #Funcion para modificar los tours (Kevin)
210 def ModificarTour(nombreTour):
211     url = "Datos/" + nombreTour + ".txt"
212     archivo = open(url, "r")
213     lista = archivo.readlines()
214     archivo.close()
215     archivo = open(url, "w")
216     archivo.write(lista[0])
217     archivo.write(lista[1])
218     descripcion = str(input("Ingrese la descripcion del tour: "))
219     if descripcion == "":
220         descripcion = "N/A"
221     archivo.write("Descripcion: " + descripcion + "\n")
222     CostoPorNoche = str(input("Ingrese el costo por noche: "))
223     if CostoPorNoche == "":
224         CostoPorNoche = "N/A"
225     archivo.write("Costo por noche: \n")
226     archivo.write(f"{CostoPorNoche} \n")
227     NombreGuia = str(input("Ingrese el nombre del guia: "))
228     if NombreGuia == "":
229         NombreGuia = "N/A"
230     archivo.write("Nombre del guia: " + NombreGuia + "\n")
231     Duracion = str(input("Ingrese la duracion del tour: "))
232     if Duracion == "":
233         Duracion = "N/A"
234     archivo.write("Duracion: " + Duracion + "\n")
235     Transporte = str(input("Ingrese el transporte del tour: "))
236     if Transporte == "":
237         Transporte = "N/A"
238     archivo.write("Transporte: " + Transporte + "\n")
239     archivo.close()
240     print("Tour modificado con exito")
241     time.sleep(2)
242     borrarPantalla()
243     Main()

```

*Fuente: Elaboración Propia*

La función recibe parámetros del “nombreTour”. Una url que contiene la información de donde se va a realizar la búsqueda. Se abre el archivo con la letra “r”, se leen las líneas del archivo y se cierra. Se vuelve a abrir el archivo, pero con la letra “w” y conservamos las dos primeras líneas para después rellenar con información nueva del tour como la descripción, costo por noche, nombre guía, duración, transporte y se imprime un mensaje de éxito.

Declaramos la función Opcion2():

```
311 #Agregar Tours (Aaron)
312 def Opcion2():
313     AgregarTour()
```

*Fuente: Elaboración Propia*

Se invoca la función “agregarTour”.

Declaramos la función Opción3():

```
315 #Asignar Tours (Santiago)
316 def Opcion3():
317     if ValidarCarpetaVacía("Datos") == True:
318         AsignarTours()
```

*Fuente: Elaboración Propia*

Se declara un if con la condición si el return de la invocación “ValidarCarpetaVacía” es igual a True, va a invocar la función “asignarTours”.

Declaramos la función Opcion4():

```
320 #Modificar Tours (Kevin)
321 def Opcion4():
322     if ValidarCarpetaVacía("Datos") == True:
323         print("Lista de Tours: ")
324         LeerTodosLosTitulos()
325         print("-----")
326         TourSeleccionado=str(input("Cual tour desea modificar:\n"))
327         if os.path.exists("Datos/" + TourSeleccionado + ".txt") :
328             ModificarTour(TourSeleccionado)
329         else :
330             print("El tour no existe")
331             time.sleep(2)
332             borrarPantalla()
333             Main()
```

*Fuente: Elaboración Propia*

Se declara un if con la condición si el return de la invocación “ValidarCarpetaVacía” es igual a True sale un print con Lista de Tours y va a invocar la función “LeerTodosLosTitulos”, le pide al usuario indicar cuál tour desea modificar, válida si existe e

invoca la función “modificarTour” con parámetros “TourSeleccionado”. Si no retorna un print con el tour no existe.

Declaramos la función Opcion5():

```
335 #Eliminar Tours Logico (Kevin)
336 def Opcion5():
337     if ValidarCarpetaVacía("Datos") == True:
338         print("Lista de Tours: ")
339         LeerTodosLosTitulos()
340         print("-----")
341         TourSeleccionado=str(input("Escriba el tour que desea eliminar \n"
342                                   "Si desea regresar al menu principal digite Y \n:"))
343         if TourSeleccionado == "Y":
344             borrarPantalla()
345             Main()
346         if os.path.exists("Datos/" + TourSeleccionado + ".txt") :
347             confirmacion = input("Esta seguro que desea eliminar el tour " + TourSeleccionado + " ? \n Y/N \n:")
348             if confirmacion == "Y":
349                 print("Seleccionado correctamente")
350                 EliminarTours(TourSeleccionado)
351                 finalizar(None)
352             elif confirmacion == "N":
353                 print("Reintentando en 1...2..")
354                 time.sleep(2)
355                 borrarPantalla()
356                 Main()
357         else:
358             print("No existe, reintentente")
359             print("Reintentando en 1...2..")
360             time.sleep(2)
361             borrarPantalla()
362             Opcion5()
```

*Fuente: Elaboración Propia*

Se declara un if con la condición si el return de la invocación “ValidarCarpetaVacía” es igual a True sale un print con Lista de Tours y va a invocar la función “LeerTodosLosTitulos”, le indica al usuario si desea regresar al menú. Si no le pide al usuario indicar cuál tour desea eliminar, válida si existe y le vuelve a decir al usuario si está seguro de eliminar, avisa que fue seleccionado con éxito e invoca la función “eliminarTour” con parámetros “TourSeleccionado”. Y se llama la función finalizar. Si el usuario no está seguro de eliminar, retorna un print con reintentando y regresa al menú. Si al final el tour que quiere eliminar no existe, regresa al usuario al menú y pide que reintentente.

Declaramos la función SubMenu6():

```
364 #SubMenu Facturacion (Aaron)
365 def SubMenu6():
366     borrarPantalla()
367     try:
368         opc = int(input("<----Bienvenido al SubMenu Facturacion del Hotel Verde Ujarras---->"
369             "\n----Cual opcion desea utilizar---- ?"
370             "\n----> 1) Visualizar Facturas"
371             "\n----> 2) Facturar un Tour"
372             "\n----> 3) Atras"
373             "\n:"))
374         if opc == 1:
375             Opcion7()
376         elif opc == 2:
377             if ValidarCarpetaVacía("DatosAsignados") == True:
378                 FacturacionTours()
379         elif opc == 3 :
380             borrarPantalla()
381             Main()
382         else:
383             print("Opcion no valida , reintente")
384             time.sleep(2)
385             borrarPantalla()
386             Main()
387     except ValueError:
388         print("No es un numero, intentelo de nuevo")
389         time.sleep(2)
390         borrarPantalla()
391         Main()
```

*Fuente: Elaboración Propia*

Hacemos un try para que el usuario ingrese la opción correcta sin que el programa se caiga, declaramos la variable opc donde si esta es 1 ingresa a opcion7 , si es dos ingresa a “facturaciónTours” con la condición del if se cumple y si es 3 regresa al menú , si esta es una opción que no está en el menú imprime que no es válido

Se declara la funcion Opcion7():

```
393 #Funcion Visualizar Facturas (Aaron)
394 def Opcion7():
395     if ValidarCarpetaVacía("DatosFacturacion") == True:
396         LeerTodosLosArchivos("DatosFacturacion")
397         var = str(input("Desea utilizar otra funcion ? \n Y/N \n:"))
398         finalizar(var)
```

*Fuente: Elaboración Propia*



En esta tiene un if donde la condición es que la carpeta de datos facturación no tiene que estar vacía si esta no está , lee los archivos que están en él y pregunta si desea hacer otra funcion.

Se declara la funcion menú():

```
400 #Menu principal (Aaron)
401 def menu(opc):
402     while True:
403         if opc == 1:
404             Opcion1()
405         elif opc == 2:
406             Opcion2()
407         elif opc == 3 :
408             Opcion3()
409         elif opc == 4 :
410             Opcion4()
411         elif opc == 5 :
412             Opcion5()
413         elif opc == 6 :
414             SubMenu6()
415         elif opc == 7:
416             var = "N"
417             finalizar(var)
418         else:
419             print("Opcion no valida , reintente")
420             time.sleep(2)
421             borrarPantalla()
422             Main()
```

*Fuente: Elaboración Propia*

En esta se hace un while con la condición true , en esta funcion se pasa por parámetros la variable opc según el número entra a la opción correspondiente si no tira opcion no válida reintente.

Declaramos la funcion Main():

```
424 #Main xd (Aaron)
425 def Main():
426     opc = OpcionMenu()
427     menu(opc)
428
429
430 Main()
```

*Fuente: Elaboración Propia*

En esta creamos la variable opc e invocamos la funcion “OpcionMenu” para tener un valor en la variable invocamos el menú y le pasamos la variable opc por último para que funcione invocamos Main.

## Conclusión

Para concluir, utilizando Python como nuestro método de resolución de problemas, este no solo nos va a poder guiar para darnos aquel resultado que buscamos mientras se desarrolla el programa, también, nos da las herramientas para entender casos y darnos razonamiento para poder alcanzar esa meta que buscamos mediante el programa.

El caso del hotel asignado, nos permitió observar las diferentes dificultades de planteamiento y ejecución de dicho programa, pero con ayuda de Python como lenguaje de programación y diferentes fuentes, se nos otorgaron las herramientas para poder entender el proceso de formación y de desarrollo del programa, y esto resulta provechoso para no solo para el acto de programación, si no, para la resolución de problemas en la vida cotidiana.

## Referencias

Brugués, A. (2021, December 1). Sentencias Condicionales en Python. Programa En Python.

<https://www.programaenpython.com/fundamentos/sentencias-condicionales-en-python/>

Fecha y hora. Python. Bartolomé Sintés Marco. [www.mclibre.org](http://www.mclibre.org). (n.d.).

<https://www.mclibre.org/consultar/python/lecciones/python-biblioteca-time.html>

García, M. (2017, July 4). 07. Python: ciclos (ejemplos).

<https://codingornot.com/07-python-ciclos-ejemplos>

if . . . elif . . . else . . . Python. Bartolomé Sintés Marco. [www.mclibre.org](http://www.mclibre.org). (n.d.).

<https://www.mclibre.org/consultar/python/lecciones/python-if-else.html>

os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.10.

(n.d.-a). <https://docs.python.org/es/3.10/library/os.html>

os — Interfaces misceláneas del sistema operativo — documentación de Python - 3.10.10.

(n.d.-b). <https://docs.python.org/es/3.10/library/os.html>

sys — Parámetros y funciones específicas del sistema. (n.d.). Python Documentation.

<https://docs.python.org/es/3/library/sys.html>