**1-Feedback**

**Antonio pardal López**

Empecé descomponiendo el programa en problemas más pequeños, como crear dos diccionarios (clients{}) y (client{}) donde introduciremos los datos de los usuarios.

clients = {}  
client = {}

El siguiente problema fue la introducción de datos del usuario y sus comprobaciones que será la opción uno asique cree una función (crear cliente) para ello.

def crear\_cliente():

Importamos os

Importamos sys

cree varias funciones para hacer las comprobaciones y las importamos utilizando el método de ruta absoluta para importar de la carpeta funciones, estas funciones que comprobaran los datos introducidos por el usuario.

import os  
import sys  
script\_dir = "../funciones" #variable donde le diremos el directorio que queremos  
sys.path.append(os.path.abspath(script\_dir)) #metodo ruta absoluta  
import validarDni, validarTlf, emailValido, validarfecha, clientehabitual

La primera función será la comprobación del DNI que es el archivo validarDni y la función dniValido.

Dentro de crear cliente introduje un bucle while True, donde le preguntamos al usuario su DNI:

while True:  
 dni = input("introduce tu dni: ")

Y comprobamos con un if y la llamada a la función dniValido el DNI, y para que este sea valido debe cumplir las condiciones de la función y saldrá del bucle con un break.

if validarDni.dniValido(dni): #dentro de validarDni llamamos a la funcion dniValido() para comprobar el dni  
 break

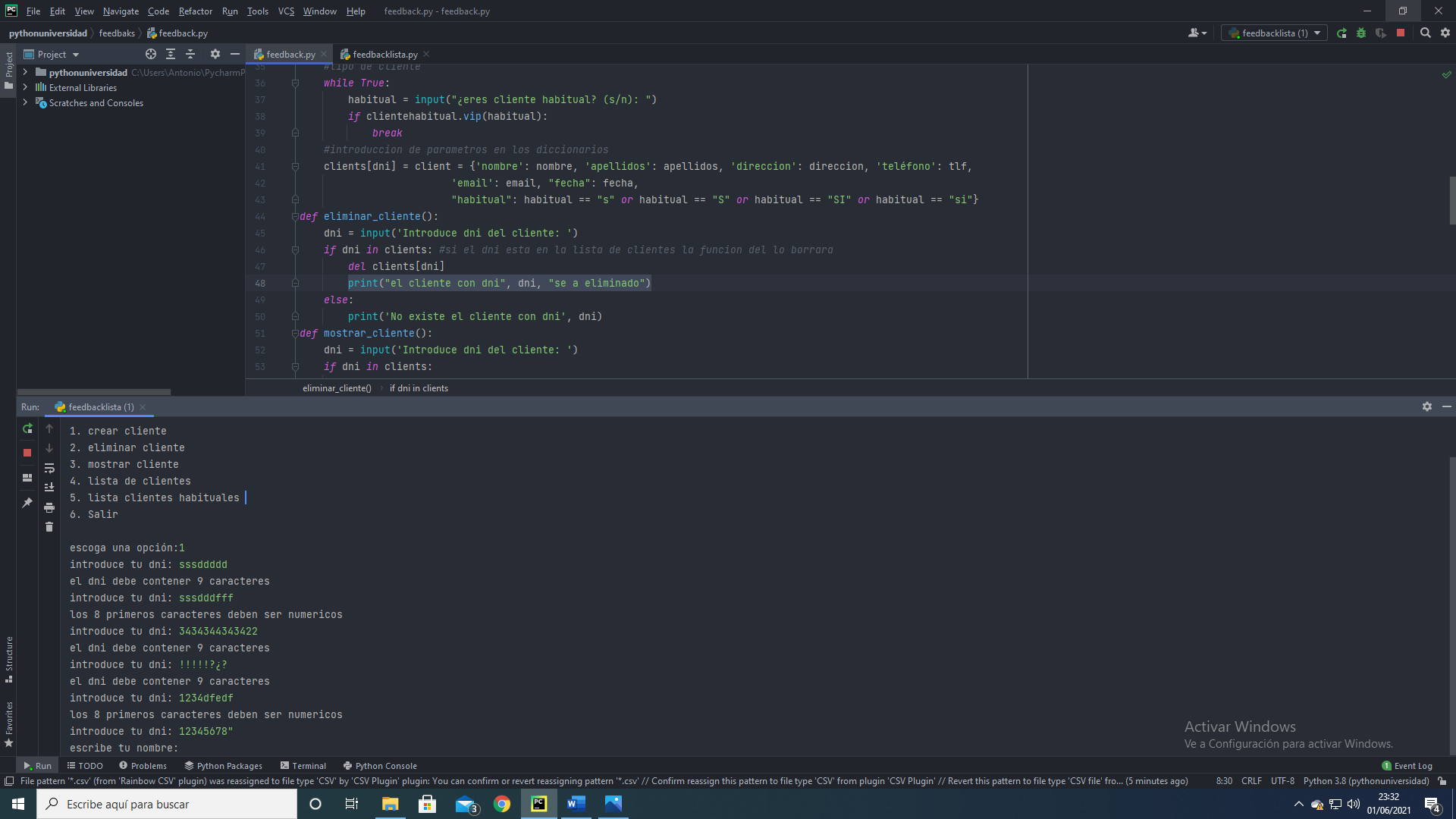
**La función dniValido:**

una variable (nif) para contar que los caracteres del 0 al 8 son numéricos, y esto lo hacemos con otra variable (n) y pasando el comando isidigit()

variable (i) pasándole el comando len() para contar los caracteres introducidos por el usuario, el cual comprobaremos con un if para saber si (i) es diferente de 9 que son los caracteres del DNI, un elif para comprobar si (n) es verdadero o falso que nos dará verdadero solo en el caso de que los ocho primeros caracteres sean numéricos Y si pasa estas comprobaciones retornara el valor True y validara el DNI.

def dniValido(dni): # definimos la funcion  
 nif = dni[:8] 5 # definimos una variable para contar dni de 0 hasta 8 que son los numeros que queremos contar para comprobar el dni  
 n = nif.isdigit() # creamos variable que te dice si todos los strings de 0 a 8 son numericos con el comando isdigit  
 i = len(dni) # variable para contanr el numero de caracteres  
 if i != 9: # si el numero de caracteres es diferente de 9  
 print("el dni debe contener 9 caracteres") # imprimira esto  
 return False  
 elif n is False: # si n es falso  
 print("dni incorrecto") # imprimira esto  
 return False  
 else:  
 return True

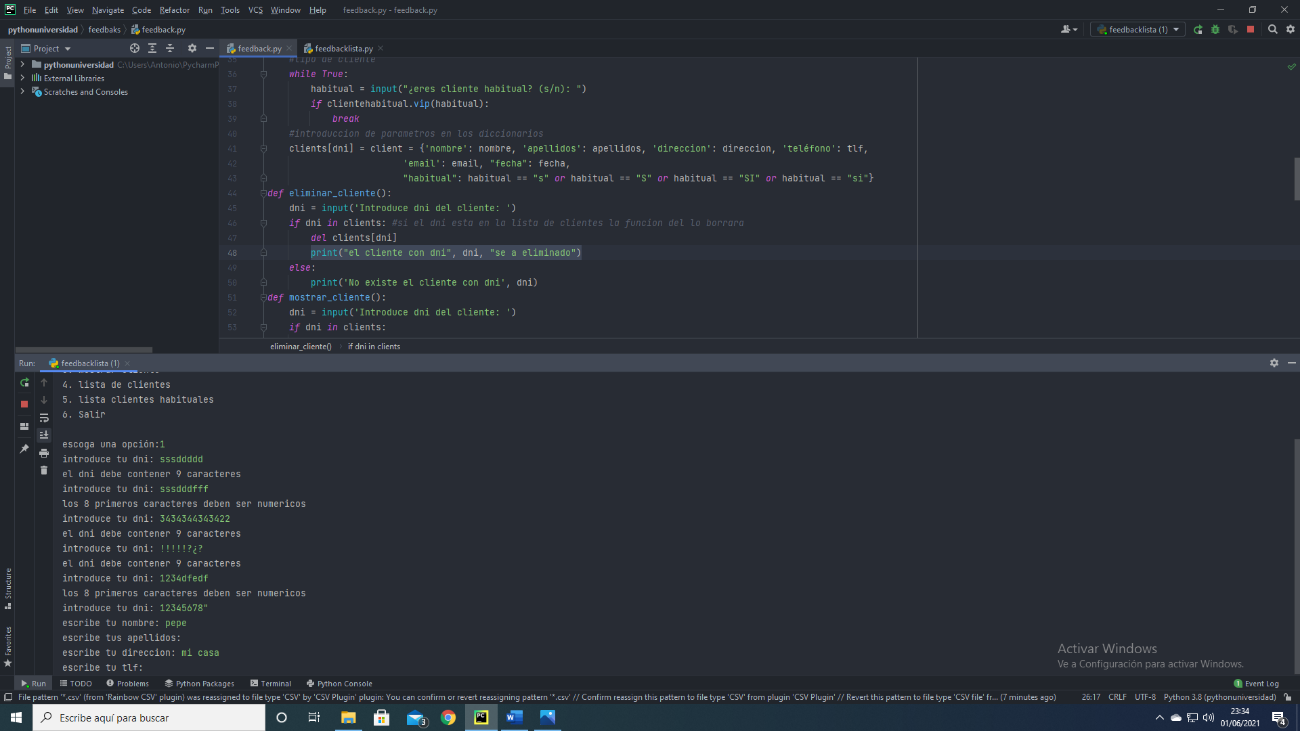
**COMPROBACIONES:**



Mientras introduzcamos los 8 primeros caracteres numéricos y cualquier otro carácter nos dejará pasar, en el caso de no llegar a 9 caracteres salta el mensaje el DNI debe contener 9 caracteres y en el caso de que alguno de los 8 primeros caracteres no sea numérico nos imprimirá el mensaje DNI incorrecto.

**#nombre, dirección, apellidos.**

Lo siguiente que introduje fueron el nombre, apellidos y dirección las cuales no tienen ninguna comprobación por lo cual nos dejara pasar al siguiente punto introduzcamos el dato que introduzcamos e incluso sino introducimos ninguno.



**#telefono**

El siguiente problema fue la comprobación del teléfono el cual debemos comprobar si tiene 9 caracteres y si son numéricos. Con un bucle while, la variable tlf para preguntar el teléfono al usuario, un if para pasarle del archivo validarTlf la funcion tlfValido y un break para salir de la iteración cuando la función llamada se cumpla.

while True:  
 tlf = input("escribe tu tlf: ")  
 if validarTlf.tlfValido(tlf):  
 break

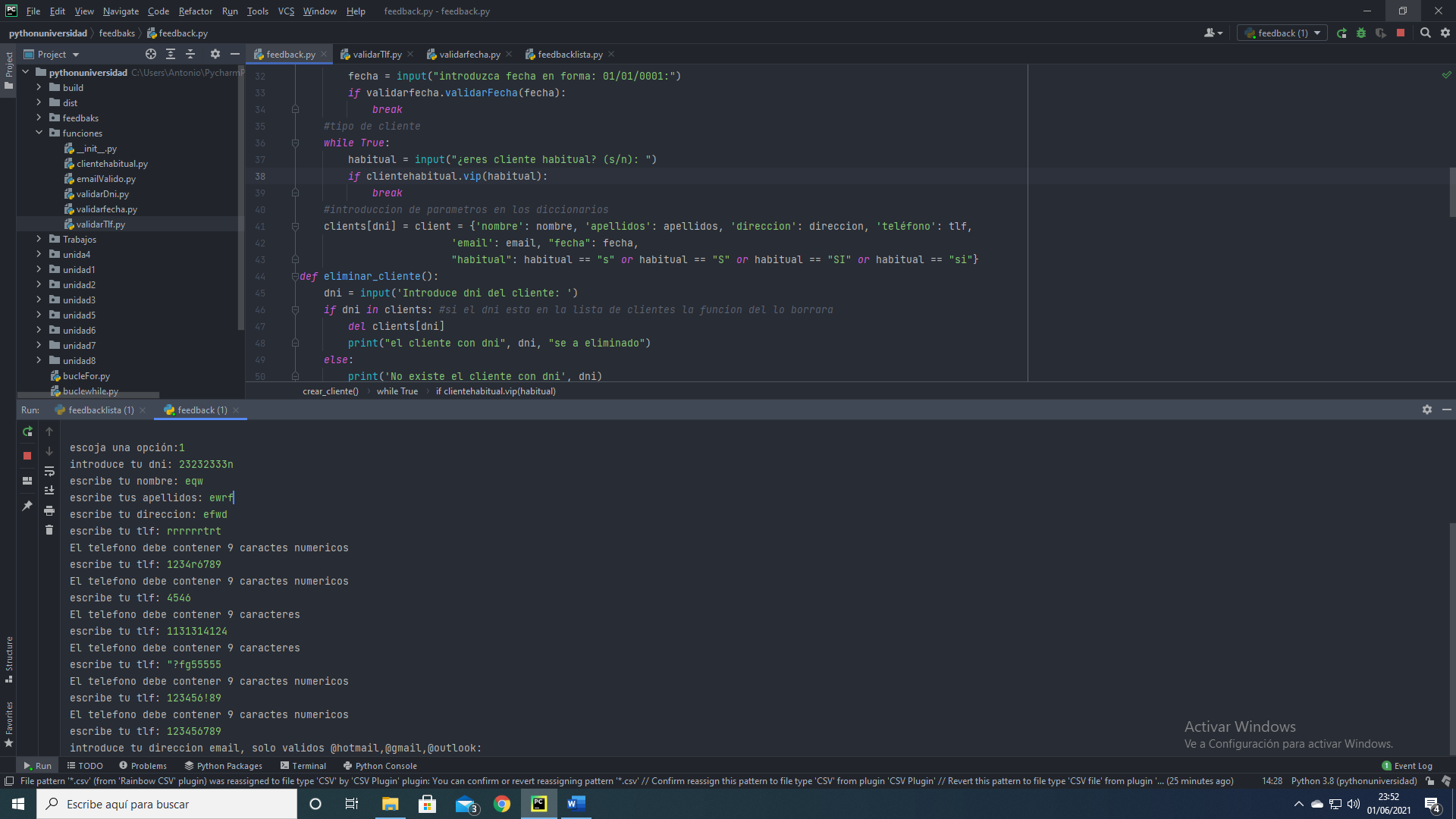
**La función tlfValido:**

La variable (t) para utilizar el comando isdigit() de nuevo. Con un if y el comando len() comprobamos si tlf es mayor o menor de 9 caracteres, con un elif si (t) es verdadero o falso y si pasamos estas dos comprobaciones retornaremos el valor True para validar el tlf.

def tlfValido(tlf):  
 t = tlf.isdigit() #creamos una variable con la que comprobaremos si los caracteres son numericos con el comando isdigit  
 if len(tlf) < 9 or len(  
 tlf) > 9: # utilizamos la funcion len para contar los caracteres de tlf  
 print("el telefono debe contener 9 caracteres")  
 return False  
 elif t is False:  
 print("el tlf debe contener 9 caracteres numericos")  
 return False  
 else:  
 return True

Hasta aquí la comprobación del teléfono la cual solo nos dejara pasar a la siguiente opción si introducimos 9 caracteres numéricos.

**COMPROBACIONES:**



**#email**

El siguiente problema fue la comprobación del email el cual comprobaremos que se trata de un email @hotmail, @gmail, @outlook.

Con un bucle y le pregunto al usuario el email, con un if le pasamos del archivo emailValido la función EmailValido y saldremos del bucle si el valor es true con un break.

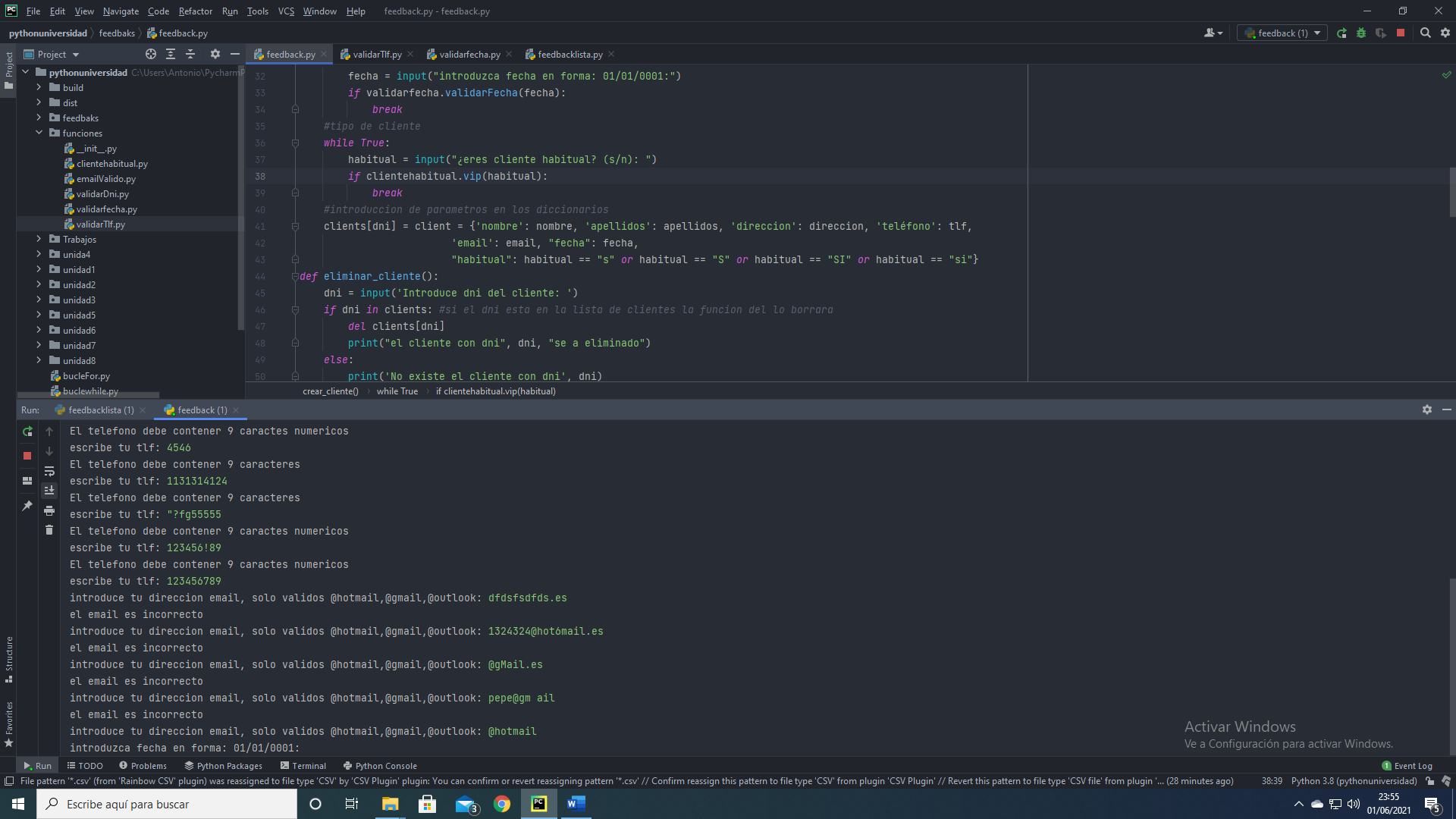
#email  
while True:  
 email = input("introduce tu direccion email, solo validos @hotmail,@gmail,@outlook: ")  
 if emailValido.EmailValido(email):  
 break

**La funcion EmailValido:**

Con un if donde pasándole a email el comando count comprueba si el email introducido por el usuario es de uno de los tres tipos que pedimos y en caso de que sea así retornar el valor true, en caso contrario entrara el else para decirle al usuario que el email es correcto y retornara false con lo que nos volverá a pedir el email.

def EmailValido(email):  
 if email.count("@hotmail") or email.count("@gmail") or email.count("@outlook") and email.count("."):  
 return True  
 else:  
 print("el email es incorrecto")  
 return False

**COMPROBACIONES:**



Hasta aquí la comprobación del email el cual nos dejara pasar a la siguiente opción siempre que se cumpla una de las tres condiciones que ponemos que son los tipos de email

Incluso nos dejara pasar si solo introducimos @hotmail como podemos ver en las comprobaciones.

**FECHA**

El siguiente problema fue la comprobación de la fecha.

con un bucle donde le pedimos al usuario que introduzca la fecha en un formato 01/01/0001 y con un if del archivo validarfecha llamaremos a la funcion ValidarFecha para comprobar esta y saldremos del bucle si la función nos devuelve un valor true

#fecha  
while True:  
 fecha = input("introduzca fecha en forma: 01/01/0001:")  
 if validarfecha.validarFecha(fecha):  
 break

**La funcion validarFecha:**

Con la variable (i) a la cual le pasaremos a fecha el comando len()

Creamos una variable (fechaSplit) para utilizar el método Split introduciendo / y dividiendo la cadena en una lista.

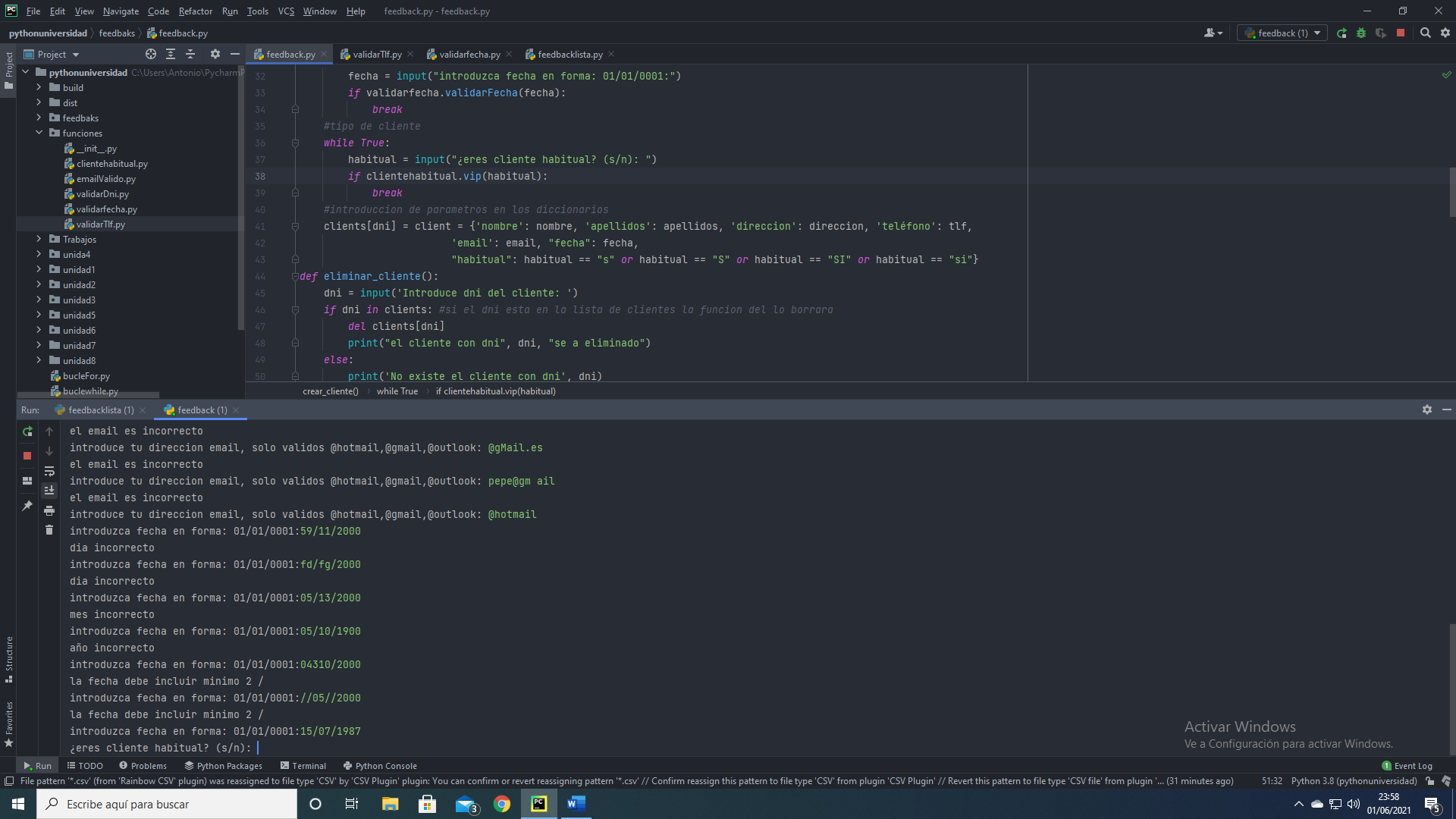
Con un if comprobaremos si (i) tiene 10 caracteres,

Con el siguiente if comprobamos que hay / o que la cadena se a dividido en 3 partes y que (i) es mayor o igual que 8 y menor o igual a 10.

En el else le introducimos varios if para comprobar los días, meses y años si son correctos, para ser correctos deben estar dentro de los valores que hemos introducido, en el caso de los días del 1 al 31, de los meses del 1 al 12 y el año entre 1920 y 2100 en cuyas comprobaciones todos los if se acompañan de un else por si no fuera correcta alguna de las comprobaciones y si las 3 condiciones se cumplen retornara el valor True.

def validarFecha(fecha):  
 i = len(fecha) # creamos una variable para contar los caracteres  
 fechaSplit = fecha.split("/") #metodo split introduce una / como separador y divide una cadena en una lista  
 if i != 10: # si los caracteres son diferentes de 10 imprimira la siguiente linea  
 print("la fecha debe contener 10 caracteres / incluidas")  
 if (not fecha.count("/")) or len(fechaSplit) != 3 and i >= 8 and i <= 10: #comprobamos que hay / o que la cadena se a dividido en 3 partes y que i es mayor o igual que 8 y menor o igual que 10  
 print("la fecha debe incluir minimo 2 /")  
 else:  
 if (fechaSplit[0].isdigit() and int(fechaSplit[0]) > 0 and int(fechaSplit[0]) < 32):#comprobamos que la primera parte en este caso los dias son caracteres numericos que el numero introducido es mayor que 2 y menor de 32  
 if (fechaSplit[1].isdigit() and int(fechaSplit[1]) > 0 and int(fechaSplit[1]) <= 12):#comprobamos que la segunda parte en este caso los meses son caracteres numericos, mayor que 0 y menor o igual 12  
 if (fechaSplit[2].isdigit() and int(fechaSplit[2]) < 2100 and int(fechaSplit[2]) >= 1920):#comprobamos que la tercera parte en este caso el año son caracteres numericos, menor que 2100 y mayor que 1920  
 return True  
 else:  
 print("año incorrecto")  
 return False  
 else:  
 print("mes incorrecto")  
 return False  
 else:  
 print("dia incorrecto")  
 return False

**COMPROBACIONES:**

Hasta aquí las comprobaciones de la fecha la cual solo nos dejara pasar si introducimos los datos dentro de los valores asignados.

**TIPO DE CLIENTE:**

Utilizamos un bucle para preguntarle al usuario el tipo de cliente que es,

Con un if llamamos del archivo clientehabitual a la función vip y utilizamos un break para que salga del bucle cuando la función retorne un valor true.

#tipo de cliente  
while True:  
 habitual = input("¿eres cliente habitual? (s/n): ")  
 if clientehabitual.vip(habitual):  
 break

**La función vip:**

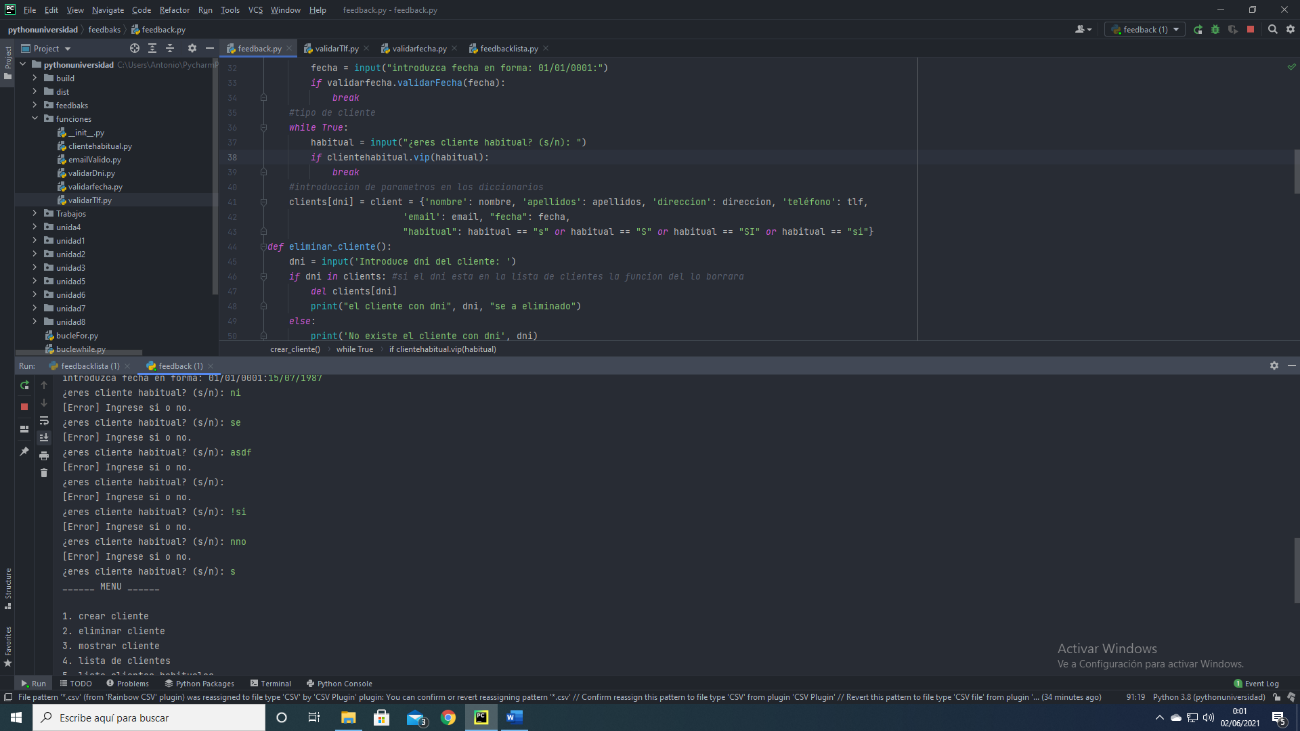
Con un if comprobaremos si habitual es igual a (si) y en el caso de que sea si el tipo de cliente será habitual y retornara un valor true

Con un elif comprobaremos si habitual es igual a (no) en cuyo caso el tipo de cliente será otro tipo y retornara valor true

Con un else si ninguna de las condiciones anteriores se cumple retornaremos mensaje de error y retornaremos false para que no salga del bucle.

def vip(habitual):  
 if habitual == "s" or habitual == "S" or habitual == "SI" or habitual == "si":  
 tipo = "Cliente habitual"  
 return True  
 elif habitual == "n" or habitual == "N" or habitual == "NO" or habitual == "no":  
 tipo = "otro tipo"  
 return True  
 else:  
 print("[Error] Ingrese si o no.")  
 return False

**COMPROBACIONES:**



**Introducción de parámetros en los diccionarios:**

Introducimos todos los datos solicitados anteriormente en clientes y dentro de clientes creamos cada cliente individual.

#introduccion de parametros en los diccionarios  
clients[dni] = client = {'nombre': nombre, 'apellidos': apellidos, 'direccion': direccion, 'teléfono': tlf,  
 'email': email, "fecha": fecha,  
 "habitual": habitual == "s" or habitual == "S" or habitual == "SI" or habitual == "si"}

Aquí terminamos con la función de la opción 1(crear cliente).

Vamos con la opción 2 la cual debe eliminar el cliente:

**Eliminar cliente:**

Con la función llamada eliminar cliente, la variable DNI para pedir DNI al usuario, comprobamos con un if si el DNI introducido está en clients la función del elimina el cliente, si el DNI no está en clients nos dirá "no existe el cliente con DNI ...

llamamos a esta función en la opción 2

def eliminar\_cliente():  
 dni = input('Introduce dni del cliente: ')  
 if dni in clients: #si el dni esta en la lista de clientes la funcion del lo borrara  
 del clients[dni]  
 print("el cliente con dni", dni, "se a eliminado")

**COMPROBACIONES:**

**escoja una opción:2**

**Introduce DNI del cliente: 234567890**

**No existe el cliente con DNI 234567890**

**escoja una opción:2**

**Introduce DNI del cliente: 53456f**

**No existe el cliente con DNI 53456f**

**escoja una opción:2**

**Introduce DNI del cliente: 12345678"**

**el cliente con DNI 13345678” se ha eliminado**

#opcion 3 mostrar cliente donde nos debe mostrar todos los datos del usuario con ese DNI

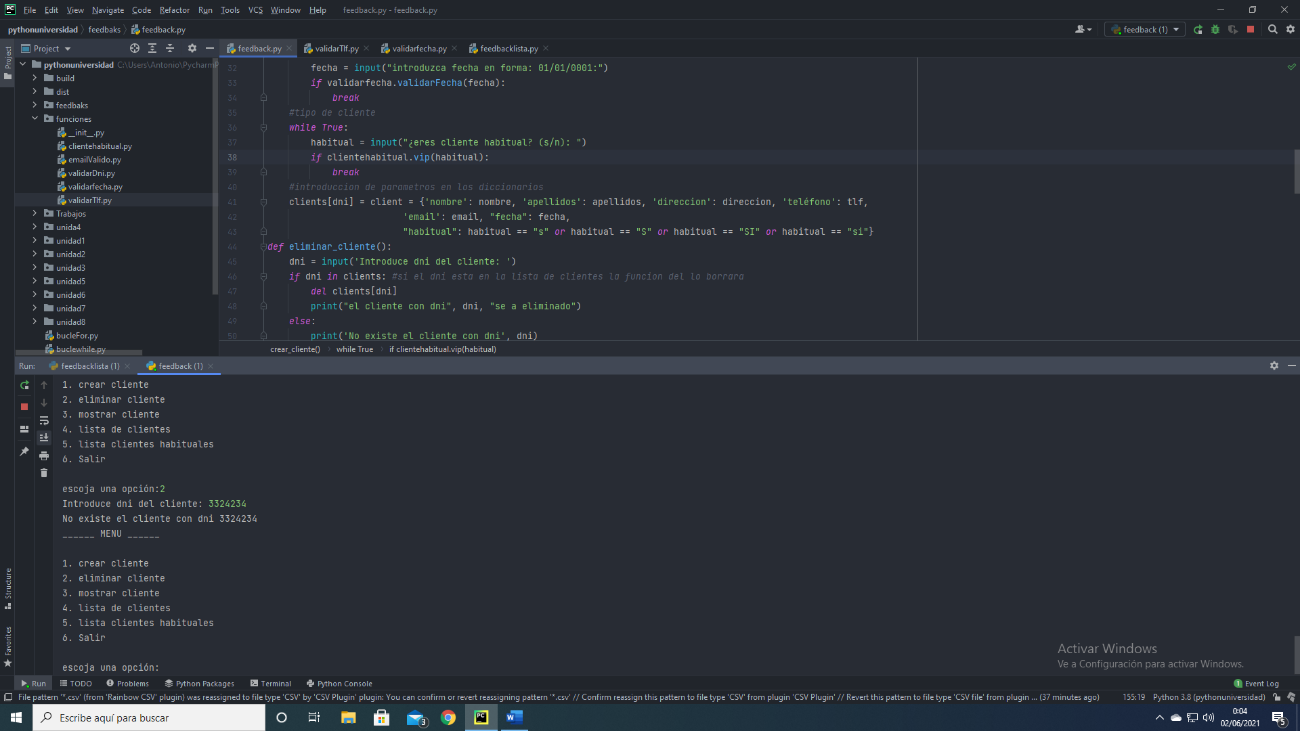
*def* mostrar\_cliente():  
 dni = input('Introduce dni del cliente: ')  
 *if* dni *in* clients:  
 print('dni:', dni)  
 *for* key, value *in* clients[dni].items(): *#para la clave valor en clientes donde la clave sera el dni y los valores los datos* print(key.title() + ':', value)

La función mostrar cliente: la variable DNI para pedirle al usuario el DNI, un if para comprobar si el DNI está en clients, si esta imprimirá el DNI del cliente y saltara el bucle for donde la clave será el DNI y el valor todos los datos del cliente con el método ítems()que devuelve una lista de claves valor.

imprimiremos el nombre del dato que pedimos y el que nos introdujeron con el método title()para convertir una cadena en formato titulo.

llamamos a esta función en la opción 3

**COMPROBACIONES:**

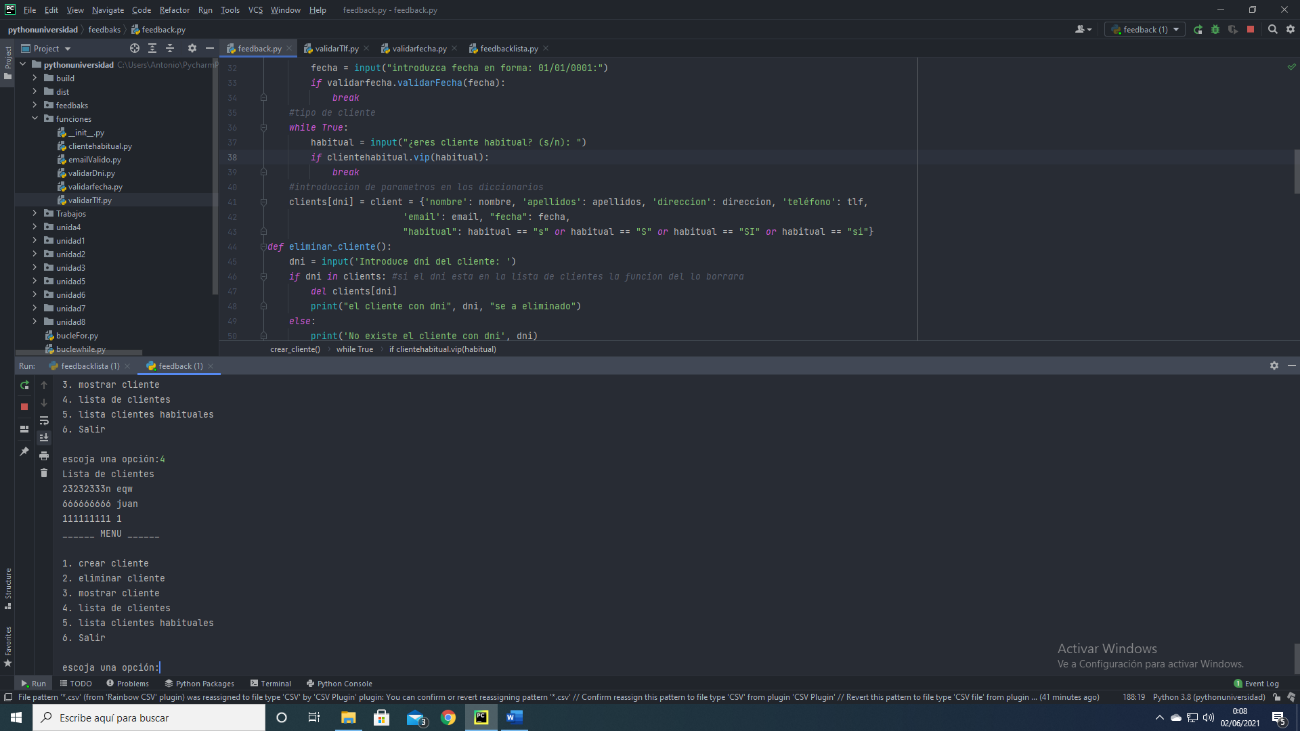


**#opcion 4:** nos tiene que devolver una lista con los nombres y dni de los clientes

*def* lista\_clientes():  
 print('Lista de clientes')  
 *for* key, value *in* clients.items(): *#para la clave valor en clients la funcion items devuelve una tupla con los pares clave-valor del diccionario.* print(key, value["nombre"])

La función lista clientes (), un bucle for donde le decimos que para la clave valor en clientes. Ítems() nos imprima la clave (DNI) y el valor (nombre) de los clientes. Llamamos a esta función en la opción 4

**COMPROBACIONES:**



**#opcion 5:** la cual nos debe devolver una lista de los clientes habituales

*def* lista\_habituales():  
 print('Lista de clientes habituales')  
 *for* key, value *in* clients.items():  
 *if* value['habitual']:  
 print(key, value['nombre'])

La función lista habituales y le introducimos un bucle for igual que en la opción 4 pero le añadimos un if para comprobar los clientes con valor true que serán los habituales y imprimirá sus DNI y nombres. llamamos a esta función en la opción 5

**COMRPOBACIONES:**

