# Practica 6 - Base de datos (Python)

# Jackh Emmanuel Narváez guerra

# Juan Alejandro Gualteros Fonseca

# Universidad de Antioquia

# Facultad de ingeniería

# Medellín – Antioquia

# 2020

# Practica 6 – base de datos (Python)

# Estudiantes:

# Jackh Emmanuel Narváez guerra

# Juan Alejandro Gualteros Fonseca

# Docentes:

# Aníbal Guerra

# Sebastián Isaza

# Universidad de Antioquia

# Facultad de ingeniería

# Medellín – Antioquia

# 2020

# Objetivos

Objetivo Principal

Elaborar un programa modular en Python, que a base de tipos de datos simples, como las cadenas de caracteres, las tuplas, las listas, los diccionarios y los archivos, se pueda diseñar una base de datos, que almacene la información de la estaciones ambientales de la región, y que además de poder almacenar un gran volumen de información, este se pueda mostrar de forma clara y tangible, debe haber tres tipos de usuarios con diferentes permisos y funciones para que la base de datos funcione, y además de eso la información debe ser clara y fácil de acceder.

Objetivos Específicos

1. Aprender a hacer programas que utilicen los diferentes objetos de datos estructurados que ofrecen Python: cadenas, tuplas, listas y diccionarios.
2. Conocer y utilizar los métodos (funciones) asociadas a los datos estructurados de Python.
3. Consolidad el desarrollo de programas en Python siguiendo las directrices de la programación modular, utilizando funciones.
4. Practicar la interacción, de los programas con archivos de texto para leer y escribir datos.

# Introducción

1. Programa asociado con el monitoreo del clima, de una manera muy resumida esa es la función principal de este programa modular, ya que es un programa que recolecta datos del clima, dado que la situación ambiental del planeta ha cambiado y los niveles de contaminación ha aumentado, se hace necesario el estudio de datos específicos, y de la variación de estos mismos datos durante el tiempo ya que esta información es muy valiosa, tiene que estar bien almacenada en bases de datos que sean fiables y que guarden los datos de forma correctamente organizada lo que haga más fáciles acceder a ellos bajo condiciones específicas, además de que estas bases de datos tiene que ser públicas para el uso del común de la población y que también puedan ser modificadas por personas especificas con ciertos comandos de acceso que les permitan corregir o arreglar información.

El diseño del programa tiene cierta complejidad, ya que se usarán tipos de datos específicos con que nos permiten organizar la información en orden, la modificación de estos tipos de datos con base a sus funciones básicas, se usaran funciones en Python para el diseño modular y la organización en archivos de almacenamiento.

# Practica 6

1. La práctica se basa en el diseño de un programa modular en Python, que pueda almacenar los datos de estaciones climáticas en todo el valle de aburra, para conocer la situación con la polución del aire, y los niveles de contaminación del ambiente, situación que en ella actualidad con un cambio climático comprobado y una gran cantidad de situaciones ambientales dispersas se vuelve indispensable el almacenamiento correcto de la información, algo que le permita ser de nivel correcto, tiene que ser información correcta, la fidelidad y la confianza en la base de datos debe ser fundamental algo que nos permite diseñar la información de diseño correcta, diseño que se puede correlacionar, algo que se puede administrar de manera correcta y de forma fiable ante el ambiente, debe ser una base de datos organizada y de correcto funcionamiento que nos permita acceder a la información de manera sencilla y que sea agradable con el usuario.
2. Requisitos

Usuarios: Registrados y visitantes -> Base de Datos type (Basic)

| Documento, Contraseña

Hay dos registrados: Administrador Operador

Administrador: Puede crear, editar y eliminar usuarios y estaciones del sistema.

Operador: Puede ingresar al sistema la información de estaciones de monitoreo, y también puede visualizarlas.

Base de datos

Una base de datos que almacenara:

* Usuarios registrados
* Estaciones existentes
* Variables monitoreadas
* Los municipios y el registro de datos

Todo en archivos de texto, leída por el programa y que carga la memoria

Menú inicial

* Usuarios registrados → autenticar
* Usuario visitante
* Salir del sistema → Guarda los datos

\*Menú correspondiente

Menú usuario operador

* **Elegir municipio:** Selecciona el municipio en el que esta ubicado ó volver al menú inicial
* **Elegir estación:** ó volver a elegir municipio
* **Acción:**

**|**

**-** Listar Medida

**-** Ingresar medidas → El sistema toma automáticamente la fecha y hora actual para cada medida.

* **Ingresar valores:** De cada una de las variables del sistema

**-** Se valida que el valor ingresado este dentro del rango precisado

**except ← Try ← error ← type**

**-** Si el valor no esta disponible falta o fallo del sensor se ingresa **“ND”** y se guarda la cantidad **-999**

Menú usuario administrador

* **Volver al menú inicial**
* **Gestionar estaciones**

**|**

* + **Crear estación**

**|**

**-** Ingresar estación (código numérico incrementa)

**-**  Seleccionar el municipio de la lista

* + **Editar estación**

**|**

* + - Elegir la estación de la lista

**-** Mostrar el nombre actualde la estación e ingresar el nuevo. El código de la estación no se modifica.

* + - Seguir editando
    - Salir a menú administrador
  + **Eliminar estación:**

**|**

**-** Elegir la estación a eliminar con todo y sus valores (NO se puede eliminar una estación sin antes eliminar los valores debe lanzar una advertencia)

* **Gestionar Usuario**

**|**

* + **Crear usuario**

**|**

* + - Ingresar documento, que debe ser diferente a todos los anteriores registrados (Solamente puede tener números)
    - Ingresar nombre completo (Solo se pueden contener letras y espacios )
    - Ingresar contraseña, que debe tener como mínimo 4 caracteres
    - Confirmar contraseña
    - Seleccionar el rol de usuario

* + **Editar usuario**

**|**

* + - Elegir el usuario de la lista
      * Cambiar nombre
      * Cambiar contraseña → Validar código de seguridad
      * Cambiar e Rol
  + **Eliminar usuario**
    - Muestra lista de Usuarios
    - Pide Documento que es la clave del diccionario para eliminar

**Menú Usuario visitante**

* **Visualizar estadísticas**

**|**

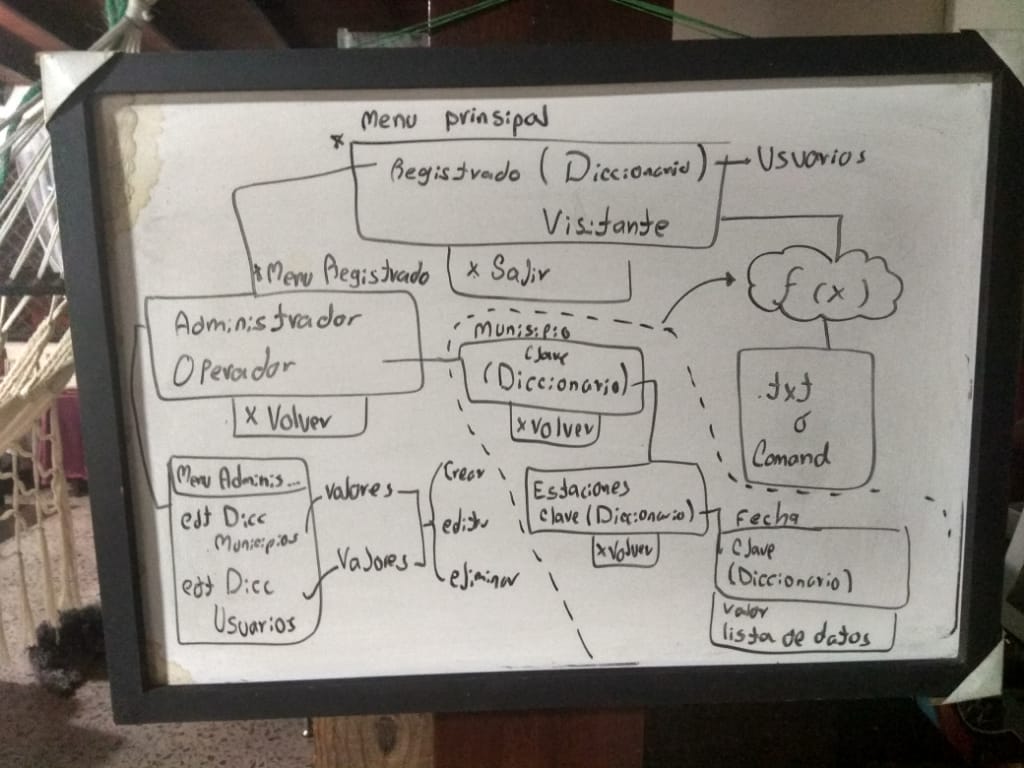
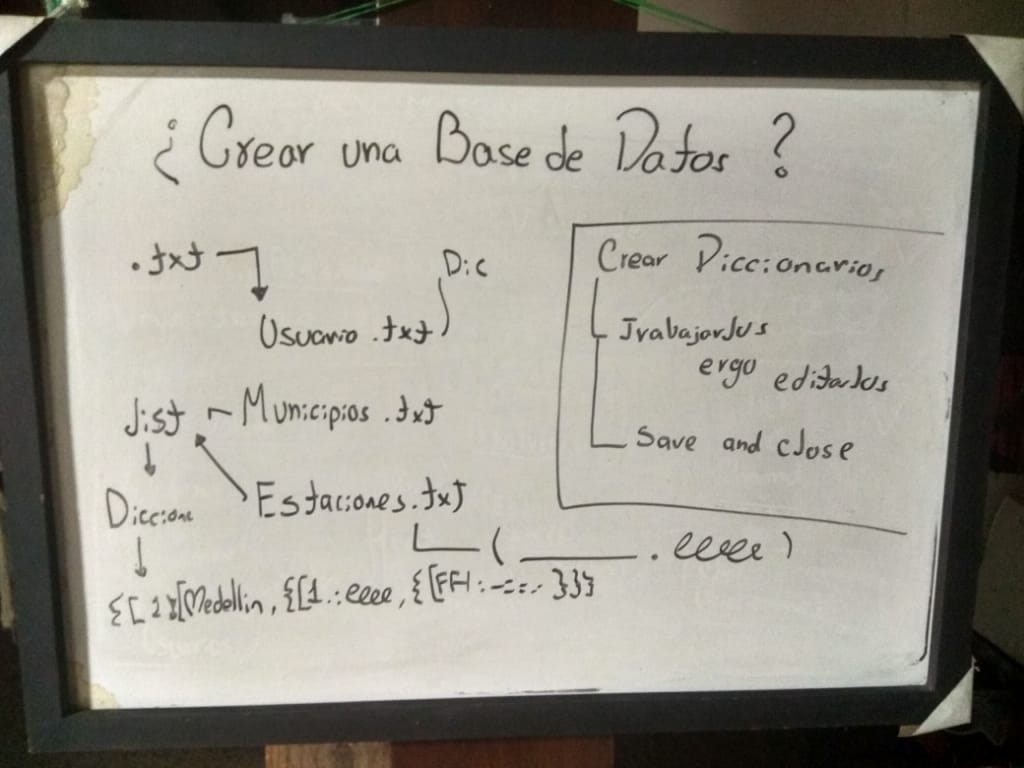
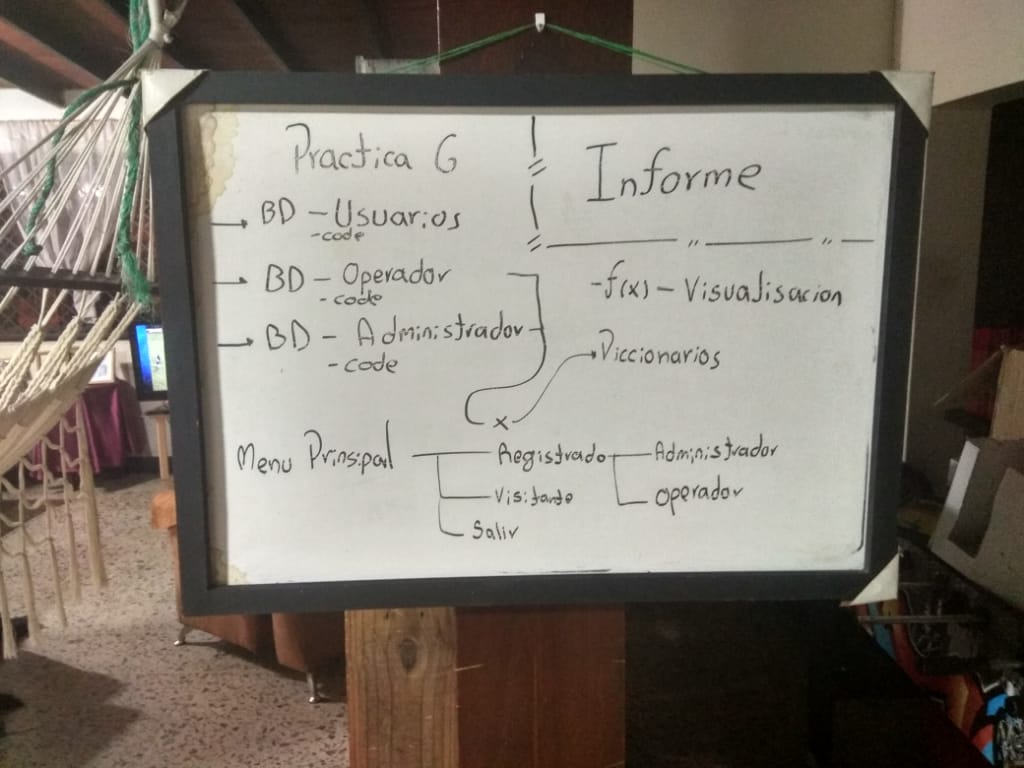
* + Elegir el periodo de tiempo a evaluar

|

* + - Últimos 7 días
    - Últimos 30 días
    - Elegir fechas manualmente
  + Elige las variables a analizar del listado, pueden ser una o varias
  + Elegir los municipios para el análisis pueden ser elegir uno o varios

Opcional

Agregar a las estadísticas un ranking de las 5 horas del día con mayor concentración de cada una de las variables seleccionadas.

Validaciones – Usuarios

* Fecha ( yyyy / mm / dd )
* **Nombre** ( Solo puede tener letras y espacios )
* **Documento** ( Debe tener entre 10 a 7 caracteres )
* **Contraseña** ( Debe tener mínimo 4 caracteres )
* **Valor de la medida**

Debe ser un numero y estar dentro del rango definido para la variable respectivamente. La única excepción es la disponible ( -999 ó “ND” según se elija )

* + PM10 [ 0.0 : 100.0 , μg/m³ ] → Polvo, hollín en el aire
  + PM2.5 [ 0.0 : 200.0 , μg/m³ ] → polución del ambiente
  + Temperatura [ -20.0 : 50.0 , °C]
  + Húmeda [ 0.0 : 100 , %]

Análisis Físico

# Funciones\_DIC

Este es primer script de python diseñado, ya que tiene un propósito fundamental que es la creación de un diccionario con base a la información, dad dentro de la carpeta Base, dentro del documento del trabajo por lo que hace de mucha importancia conocer la bases de la información analizada ya que con base a los datos se maneja la información persistente, información que es dada con la intención de diseñar in diccionario con toda la información dada ya que esta tiene una ubicación y se tiene que respetar la fidelidad de la misma, información que se conserva y cuya intención es no modificarla según los lugares y momentos específicos.

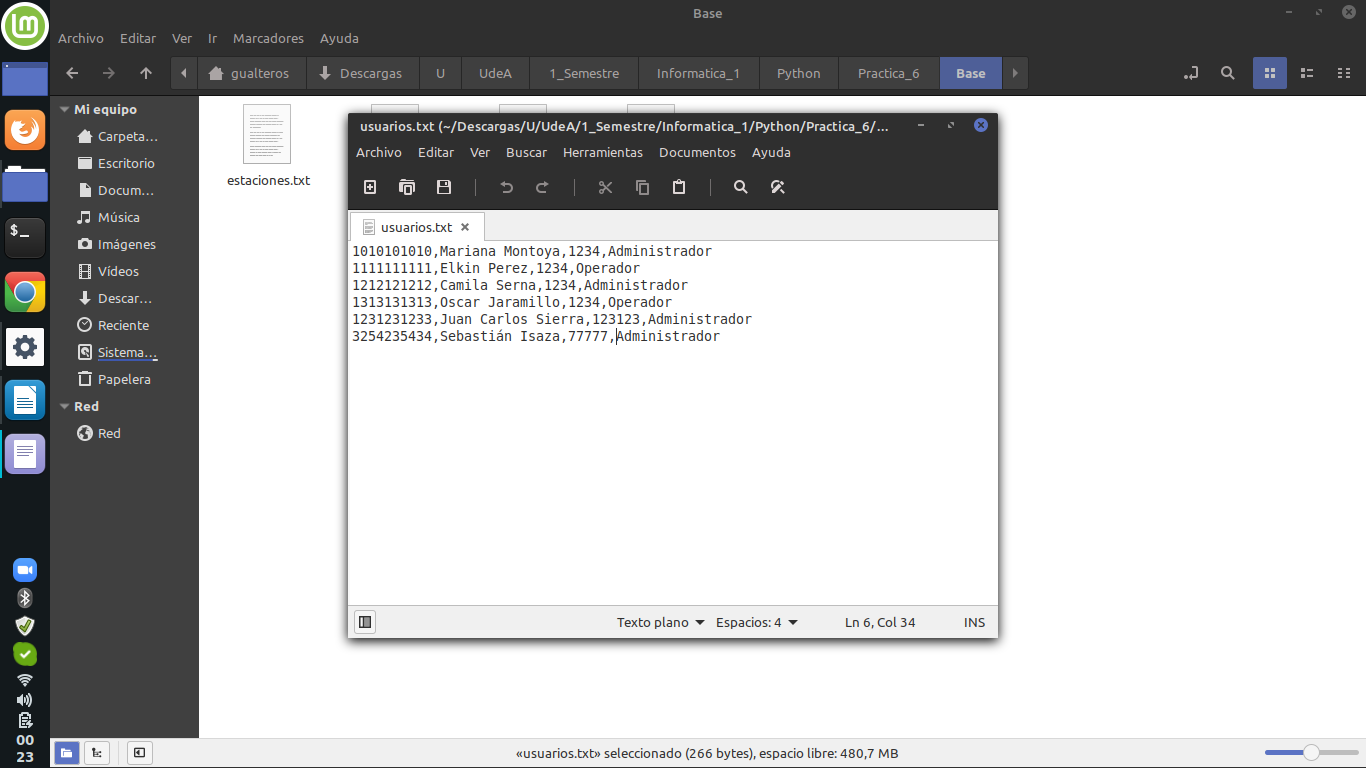
Dentro del script lo primero que se hace es buscar la información de donde se encuentra el archivo porque por antonomasia no podemos saber en que ubicación esta el archivo de trabajo por lo que buscarnos con el modulo .os la ubicación en la que nos encontramos, también importamos el modulo .io lo que nos permite administrar leer y conocer la información dada anteriormente.

Ya conocido la ubicación actual en la que nos encontramos podemos comenzar a buscar con base a es información la carpeta Base, que debe estar en la misma carpeta donde el este el documento script, algo que nos dará mas facilidad a la hora de trabajar y que programa asume, como una carpeta que tiene que estar donde este el script de trabajo principal.

Y ahora damos una ruta dirección en la que sabremos la ubicación en la que nos encontramos es dentro de la carpeta Base, dentro de esta carpeta deben estar los archivos .txt, donde se almacenara la información que se trabaja luego con cada función en especifico según el documento que se necesite.

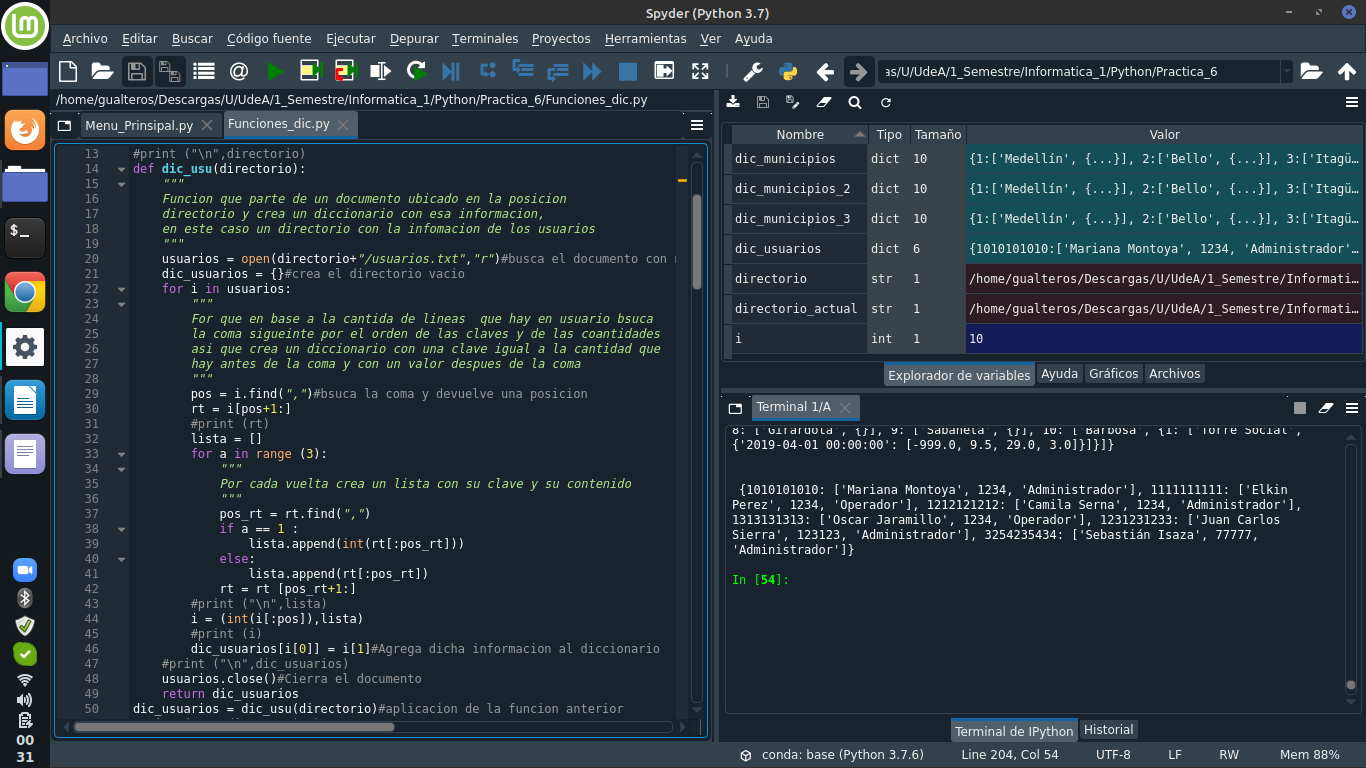
dic\_usu(directorio)

Esta función toma como base la ubicación persistente para buscar dentro de la carpeta el documento de usuario, documentado que contiene la información a analizar, la información de este documento tiene una forma de separación clara que nos permite lego por la función convertir esa linea de texto en un diccionario:



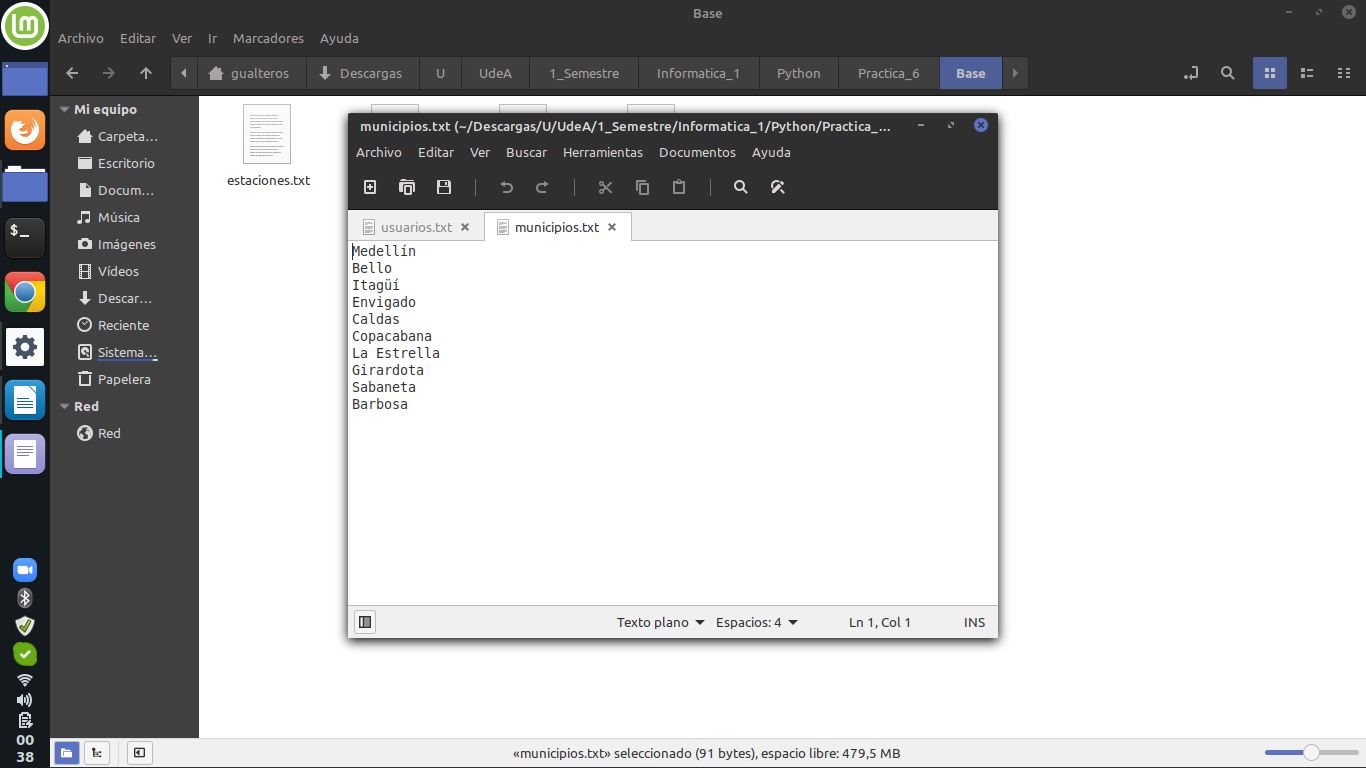
Se mantiene separado por comas según la necesidad del uso de la información por ejemplo la primer coma se usa para el clave del diccionario, y del resto de la información es para crear una lista que contenga los datos necesarios de uso para la creación de una cuenta de usuario y que dependiendo del ultimo valor se mostrara la función de trabajo según su rol si es administrador o operador.

Y hacia crea un diccionario de esta formar:



dic\_1(directorio)

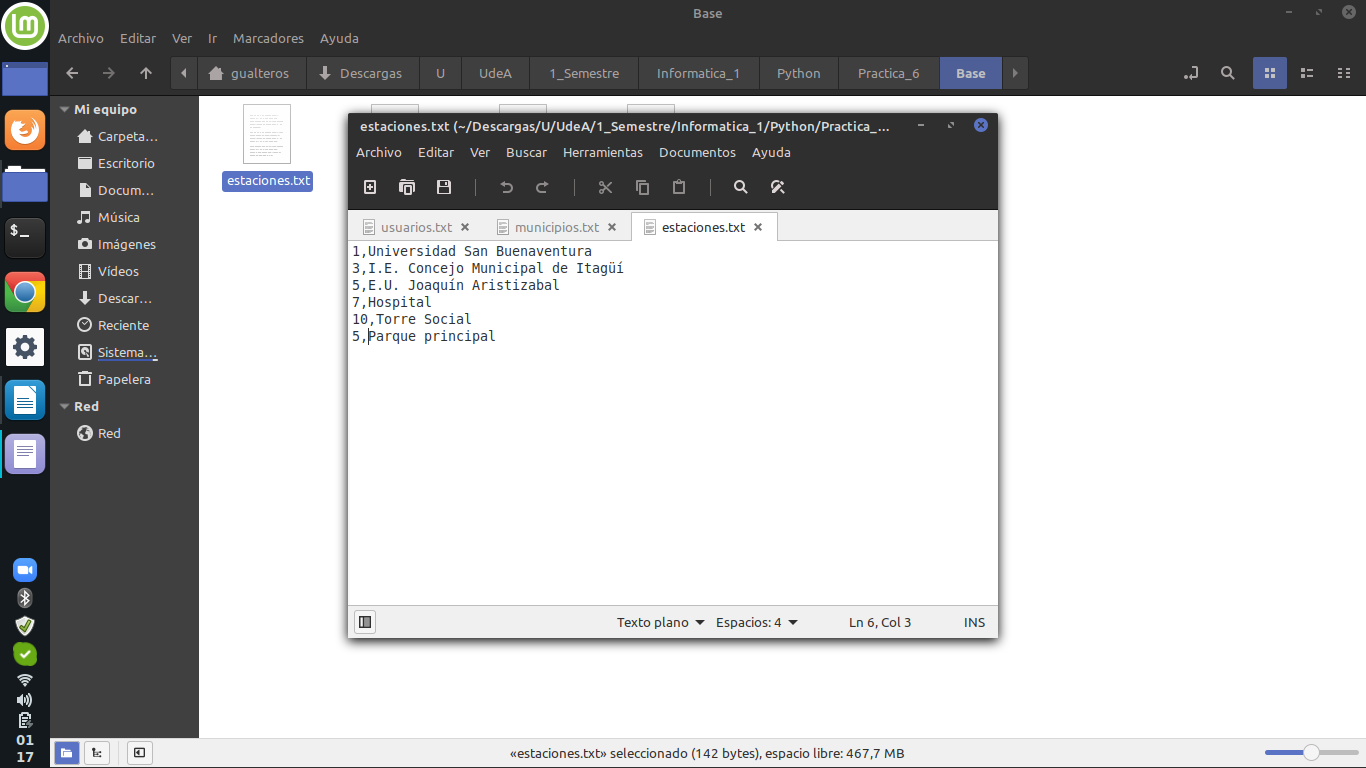
Toma como base para el uso de información el documento municipios.txt para analizar cuantos municipios hay y que por ende cuantos municipios podemos analizar para saber cuales podemos usar, y a cuales podemos acceder, ya que según la cantidad de datos aquí datos nos permiten trabajar , algo que nos da la información precisada:



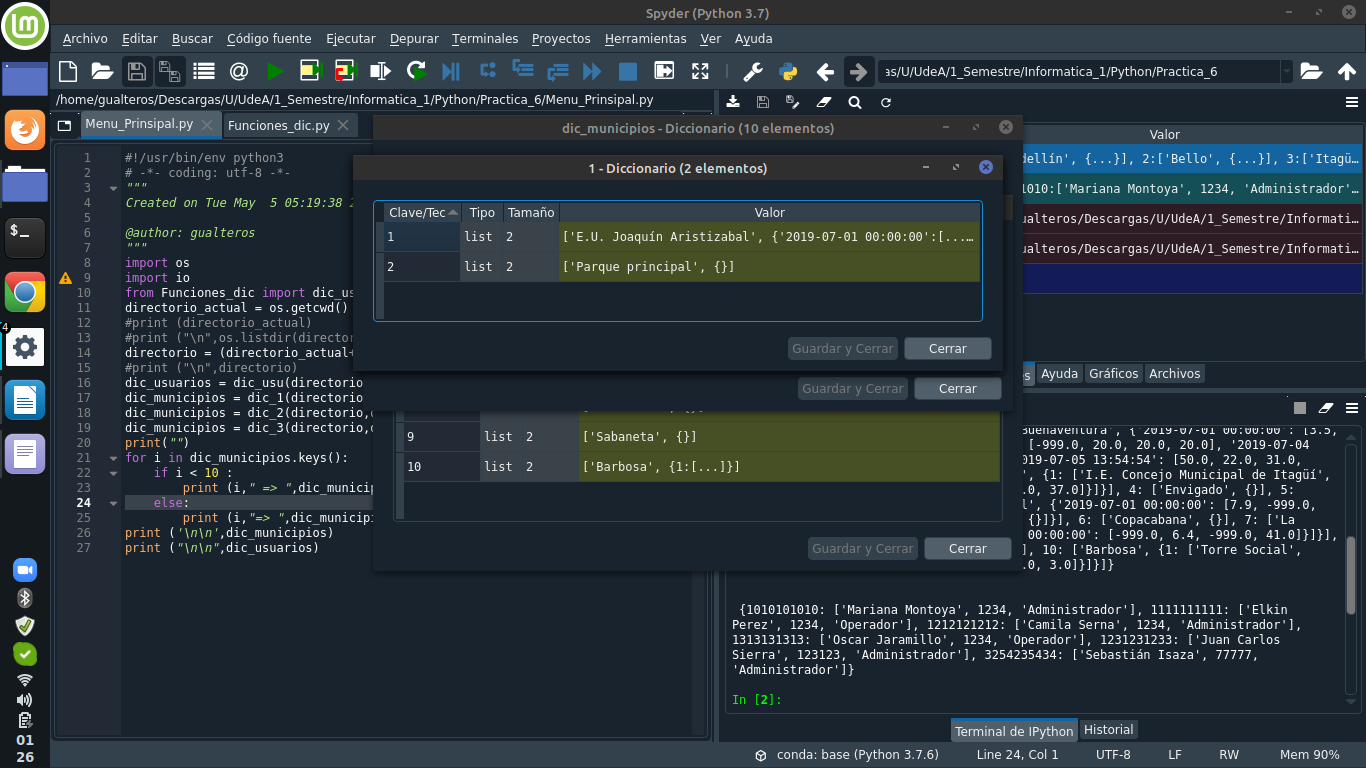
Se puede notar que solo se tiene la información en cada fila del lugar para así poder analizar los municipios que tengan no estaciones algo que nos permite crear una gran base da datos según cada parte de la información, según cada linea termina siendo el valor de un diccionario que cuya clave es el numero de la fila en la que este ubicado, por lo que se vuelve un documento inmodificable a menos que se agregue mas valores y que no se cambie el posicionamiento.

dic\_2(directorio,dic\_municipios)

Esta función recibe una dirección de búsqueda como las funciones anteriores, y trabaja con el documento estaciones, el propósito que tiene es el de crear un diccionario con ese documento, con base a cada posición predefinida ya que el diccionario estacione se separa por comas según el nombre de la estación y el numero del lugar donde se encontrá, ya que nos hace mas sencillo así poder editar a futuro este documento en caso de querer eliminar o modificar una estación, buscar una estación con otro nombre y que se establezca con este tipo de almacenamiento de información:

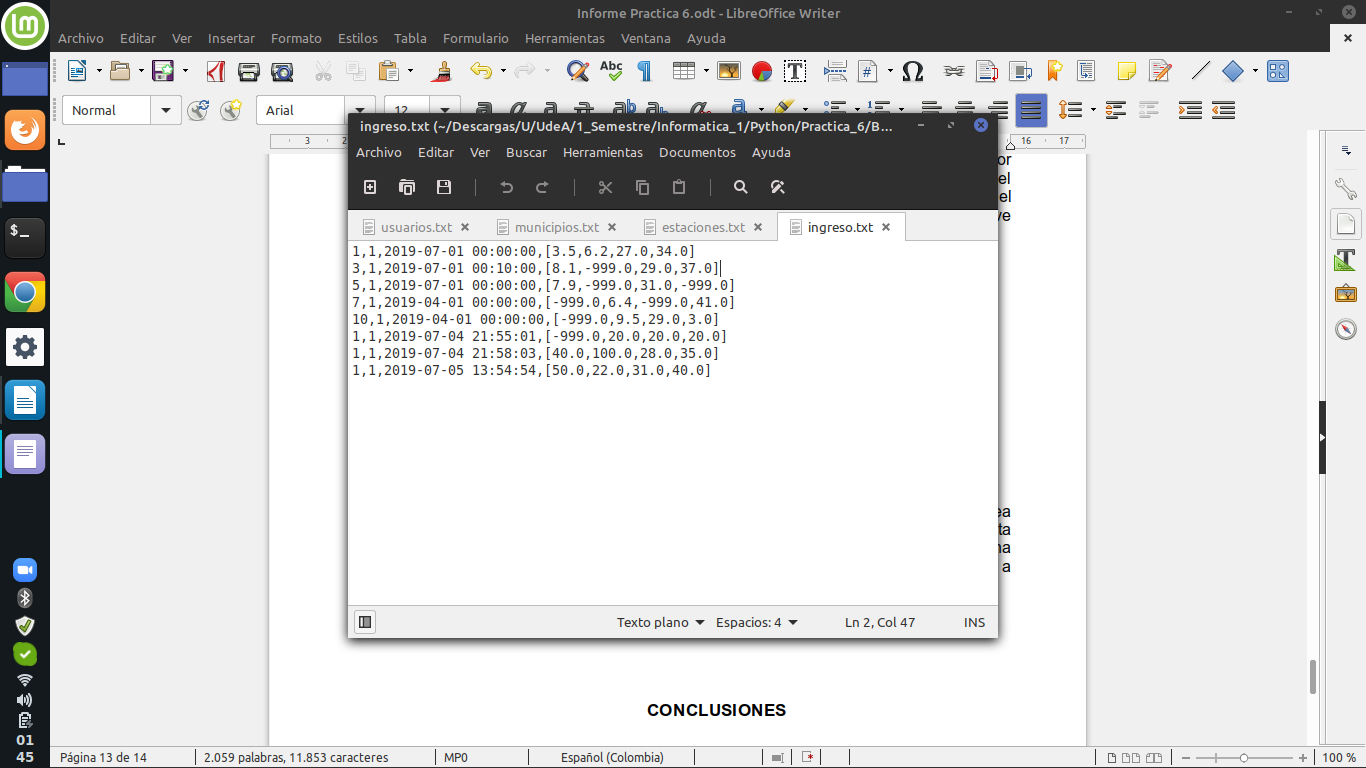


En cada linea ya que se analiza linea por linea de cada parte de la lista creada con esta información, se nota un numero, que representa el numero de la clave o ciudad a la que pertenece, lo que nos permite separarlas y colocar según la cantidad que halla de una ciudad para poder listarlos creando otro diccionario por ejemplo en el caso de caldas que se identifica dentro de esta imagen con el numero 5 como tiene dos estaciones se lista que se encuentre primero según el análisis linea por linea, ya que crea el diccionario con base a ese listado con clave 1 el E.U. Joaquín Aristizabal y con el numero 2 Parque principal:



dic\_3(directorio,dic\_municipios)

Esta función recibe la información del documento ingreso que tiene linea por linea la información dada por los aparatos de entrada dados anteriormente, y de esta forma crea un diccionario con la clave que es la fecha y hora tomada con una cadena de texto y según las comas previas nos dice a que municipio pertenece y a que estación se encontrá:



El formato como recibe la información es importante ya que el primer numero antes de la coma es el de la ubicación por municipio o el municipio como tal al que pertenece el dato y el segundo después de que la función elimine el primero es el dato de la estación de dicho municipio al que pertenece, formato de fidelidad para guardar la información, y como ultimo la fecha y hora como clave del diccionario al final que contiene toda la información necesaria para su análisis.

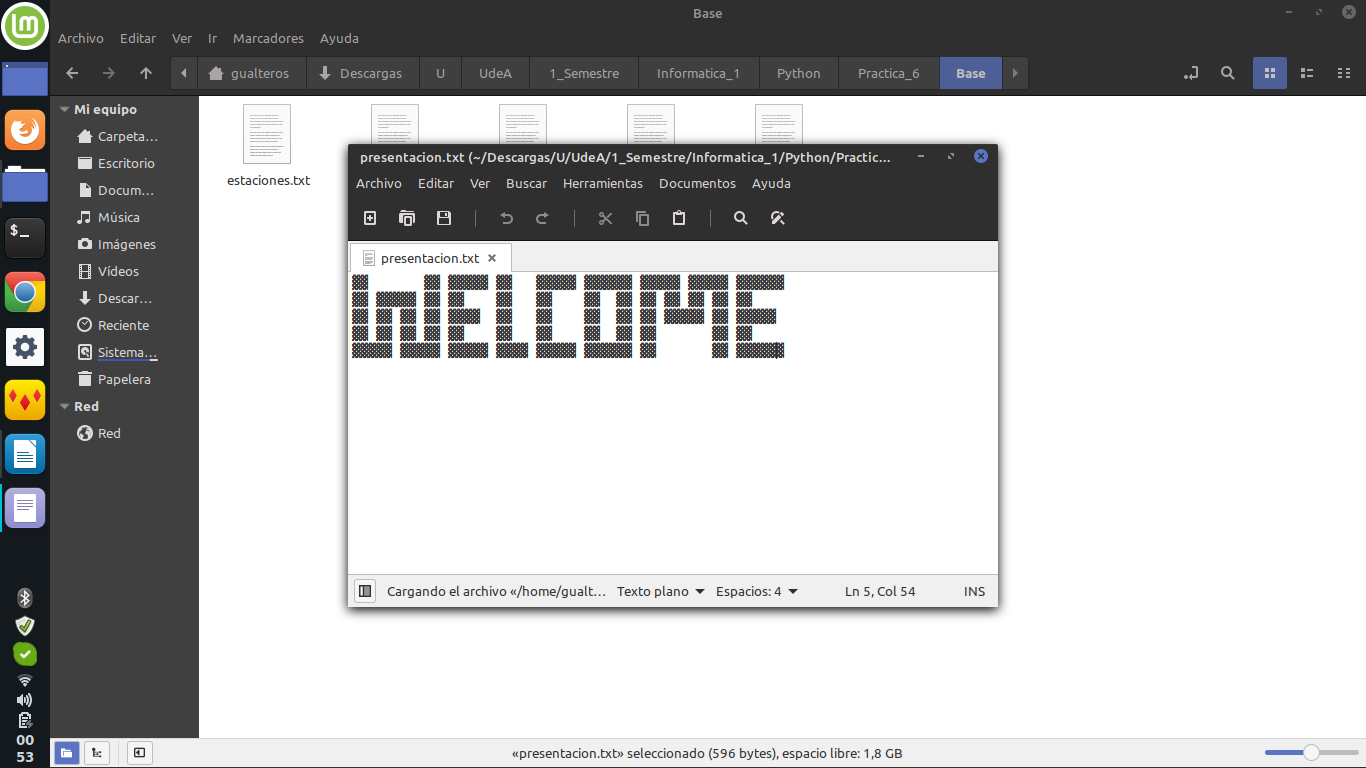
# Funciones\_ini

Son las funciones de inicio de operaciones, ya que después de lograr definir los diccionarios tenemos que saber cuando usarlo, algo que es muy importante, saber el momento correcto cada vez que lo necesitamos, por eso se realiza el script, que contiene la funciones que se trabajan en estos momentos y luego se exportan a la archivo de menú principal, se encurta dentro 3 funciones de uso, presentacion, Error, y la mas importante visua\_visi, ya que dentro de este se encuentran varias funciones que de paso utilizan Error para funcionar y buscar dentro el documento ingreso la información necesaria del trabajo.

Solo se manejan estas funciones de manera principal dentro del documento, ya que solo se hace necesario trabajar con el resto de funciones dentro de visua\_visi, algo que hace mas fácil portar dicha función como contenedor de una acción principal que es la busqueda de eh impreciso de los contenido dentro de ingreso según la necesidad especifica del usuario, lo que otorga un método de función para busqueda por medio de visitante algo que nos ser vira mas adelante para el menú de operador.

presentacion (directorio)

Esta función solo presenta la introducción cada vez que volvemos al menú principal, ademas de imprimir un comentario explicado las razones por las que se diseño este programa, así se logra que el usuario sepa cual es el propósito del mismo y tenga un contexto antes de comenzar a introducir valores de uso continuo. Imprime una bienvenida:

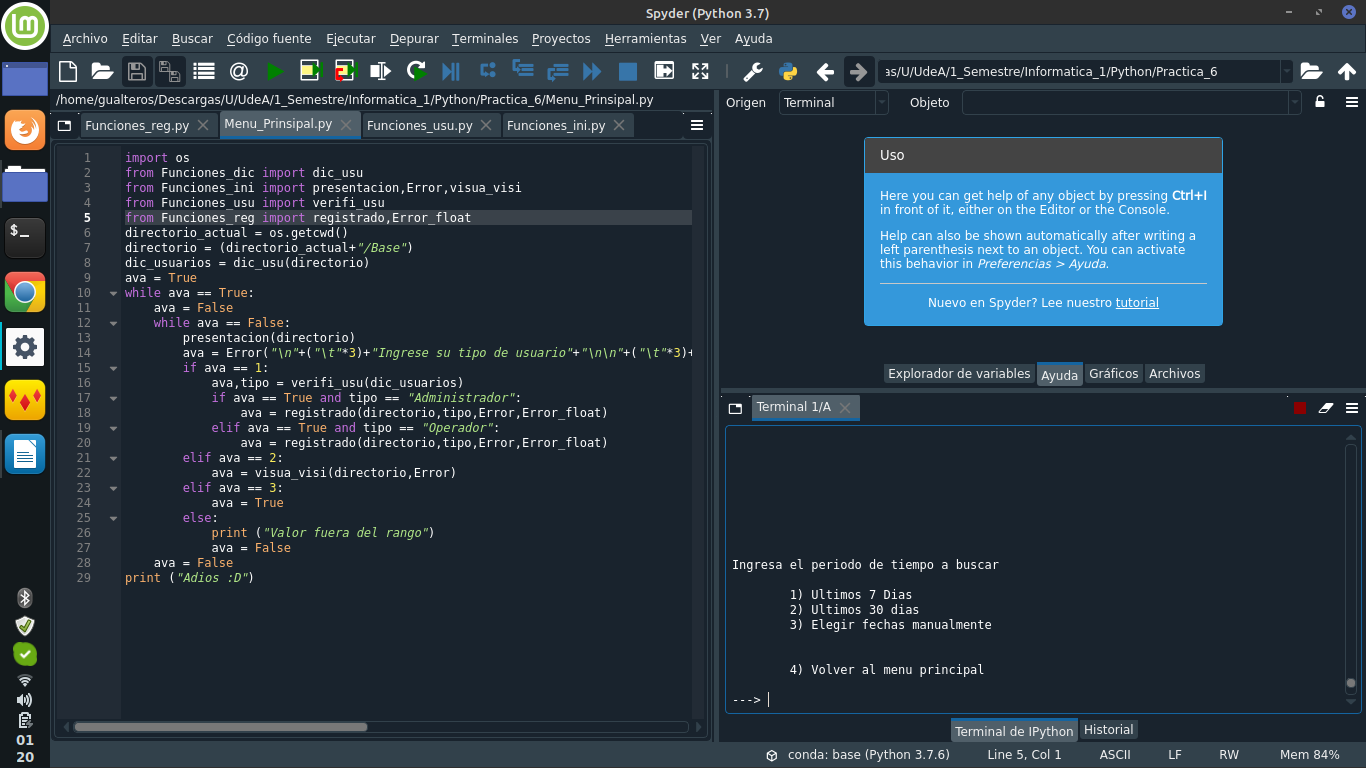


Error (tex)

Esta función tiene una importancia en especial, ya que muchas veces necesitamos estar seguros de que una variable sea de una cantidad y que en caso de que el usuario se equivocara y colocara en una petición de ingresos de datos o un input este ingresar aun str, algo que puede detener el programa algo que no queremos así que se valida pro medio de un Try, o intento de ejecución de un programa, y si este falla debería entregar un error, algo que podemos manejar y seguir operando de esta forma evitamos el fallo con un while que solo termina cuando el valor ingresado no es un str, después fuera de esta función podemos definir otro while para que siempre este dentro del rango queremos algo que evita y vuelve mas sencillo evitar los errores del usuario.

visua\_visi (directorio,Error)

La función como tal, al entregarle una ubicación de directorio done este los archivos de análisis de ingreso y una función de Error para medir los errores, que se ven necesarios de contar, busca dentro e este archivo midiendo las cantidades de fecha las ubicaciones precisadas, dependiendo de las fechas ingresadas por el usuario lo que quiera ya que muestras un menú de esta forma:



Ya que depende mucho de los valores que vaya a ingresar el usuario, para saber si quieres tener ciertas cantidades entre las mas comunes y fechas en especifico cada función anidada dentro de esta función tiene una razón de haberse realizado

ya que cumple con un propósito en especifico por lo que depende mucho de lo que ingrese el usuario si se desea continuar o saber la resultante del mismo.

* cambio\_fecha (d,day,month,year)

Como su nombre lo indica esta función realiza un cambio de fecha que es muy importante en caso de la opción 1 y 2 del menú anterior ya que dependiendo de lo que escoja le usuario esta función tiene que restar unos días a dicha cantidades y buscar desde esos días mas adelante hasta la fecha actual.

Por ejemplo si la diferencia de esto solo es un par de días hay que conocer si se produce un cambio de mes ya que nuestro calendario gregoriano se divide en meses algo que nos hace mas complejo el análisis, la teoría seria así:

Sabemos que el año tiene en su totalidad 12 meses y que en total hay 365 días en años que no son bisiestos y 366 días en años bisiestos, los años bisiestos nos agregan mas problemas al análisis ay que de entrada si hablamos de un cambio de mes hacia febrero tenemos que preguntar que mes es si un febrero de 28 o de 29 días, así mismo si hay un cambia de mes a cual me se cambia, sabemos que :

* + Enero → 31 días
  + Febrero → 28 o 29 días
  + Marzo → 31 días
  + Abril → 30 días
  + Mayo → 31 días
  + Junio → 30 días
  + Julio → 31 días
  + Agosto → 31 días
  + Septiembre → 30 días
  + Octubre → 31 días
  + Noviembre → 30 días
  + Diciembre → 31 días

Ya sabemos cuantos días tiene cada mes ahora, sabemos que a la cantidad actual de días hay que restarle un numero d que sera de 7 o de 30 días ya que en casos de rangos distintos es mas factible usar la opción 3 que nos permite medir rango algo mas útil, sabemos que si dicha resta resulta en un numero igual a 0 o menor a este osea un numero negativo este resultara significar un cambio de mes al mes anterior, por lo que a la cantidad de días que tenga el mes que cambio se le restara dicho numero negativo dando de por si la fecha medida por precisión, y obteniendo el numero del día anterior querido, la función solo esta diseñada para estos casos ya que si se restaran muchos días cabe la posibilidad de que haya dos cambios de meses algo que no se contemplan como posibilidad y generen un error al no saber exactamente en que se esta, y en tal caso es mas factible usar la opción 3.

Ahora aunque el mes ya se sepa, por lo menos en la mayoría de los casos sabemos a que numero le debemos restar la cantidad faltaste pero en el caso de febrero primero tenemos que analizar que el año que estemos analizando sea un año bisiesto, ya que en caso de serlo se le restaría a 29 y en caso de no serlo se le restaría a 28.

Un año bisiesto es aquel año divisible en 4 osea que no deja residuo al dividirse en cuatro que, con la excepción de los años que son múltiplos de 100 y que no son múltiplos de 400, por ejemplo el 2000 es bisiesto porque es divisible en 4, y aunque también es divisible en 100, este a su vez es divisible también en 400 por lo que es un numero bisiesto, diferente del 1900 que aunque es divisible en 4 también es divisible en 100 pero no es divisible en 400.

después preguntamos si hubo un cambio de mes en febrero, si lo hubo en que año fue y dependiendo si este es bisiesto o no restamos la cantidad correspondiente, por lo que se opera según esas mismas medias.

Aun así hay que hacerle un análisis mas si hay un cambio también de año y eso solo ocurre entre enero y diciembre, dicho de otro modo si el cambio de mes se produce en enero con fecha hacia Diciembre se le resta a la cantidad de 31 (días de diciembre) el numero a llegar a la fecha pasada y se le resta 1 al año de crear, así la función resultante nos entrega una fecha menos la cantidad de días que hallamos definido en d.

Opción 3

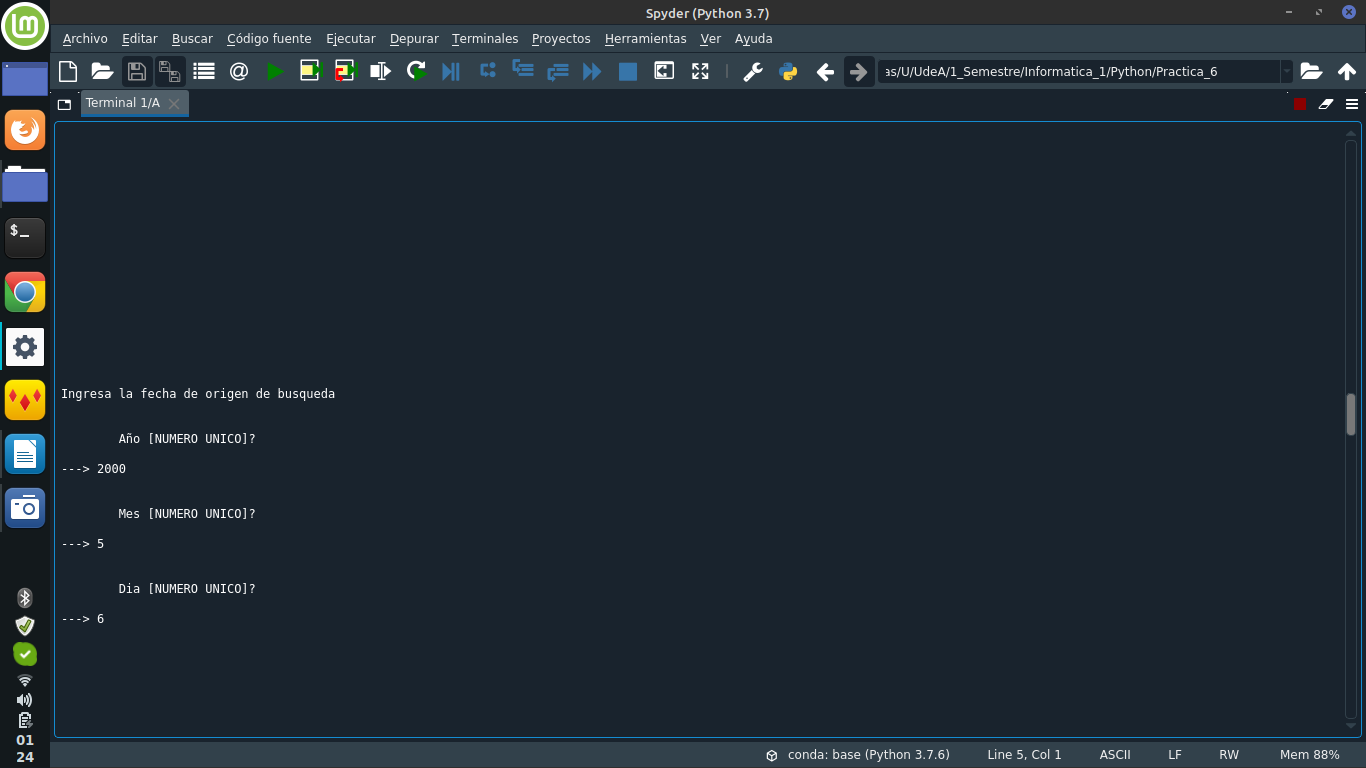
Si accedemos a la opción 3 nos solicita que ingresemos una fecha, de lo cual hay mas información en la función ingreso\_fecha, después de ingresarla, resultara en este menú:

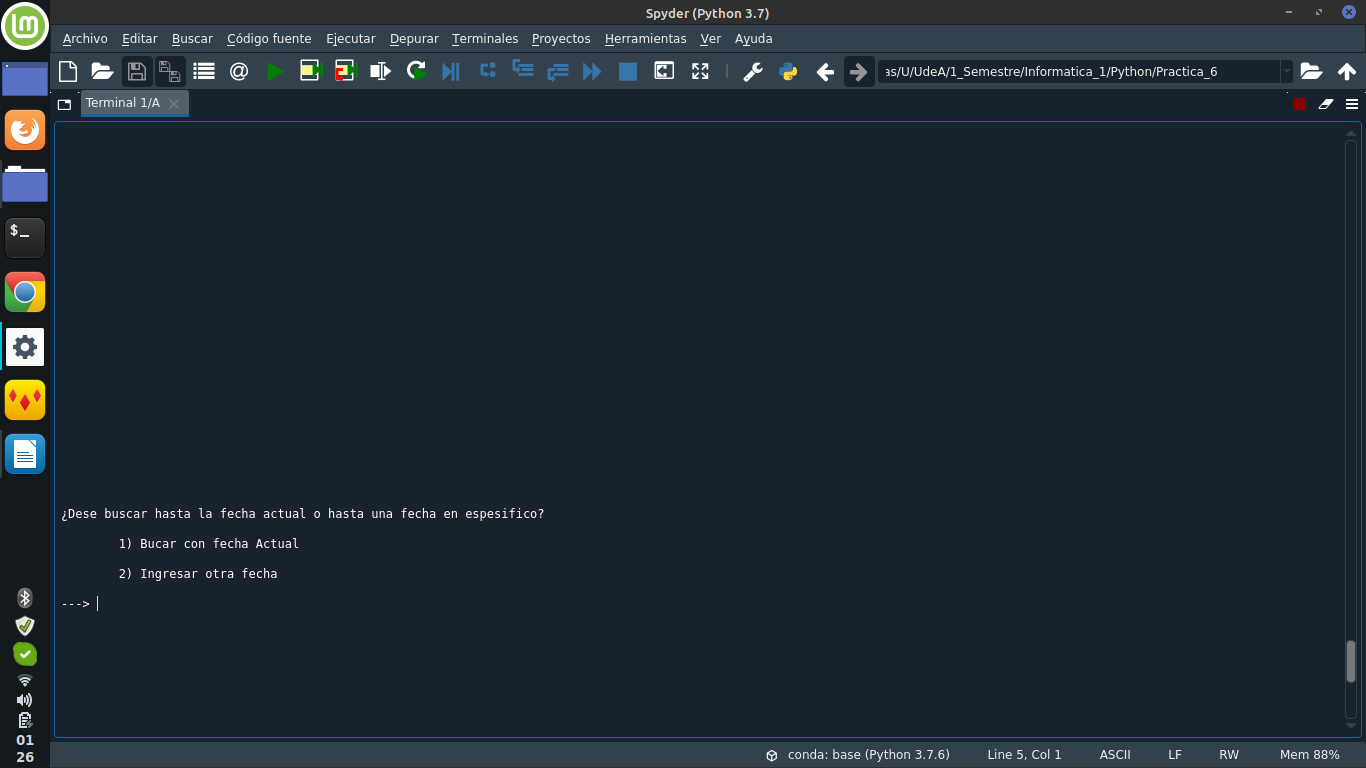
Que nos permite buscar desde la fecha ingresada hasta, la fecha que escojamos si es la actual buscara en todos los registros hasta la fecha entregada si no lo es y

escogemos la opción dos nos pedirá que ingresemos otra fecha y buscara en el rango de estas fechas, hay que recordar que cada menú que se analice de esta forma de ingreso numérico para cada opción trabaja con la función Error, mencionada anteriormente así que si ingresamos un valor no debido un un numero mayor a los mostrados, este nos pedirá que sigamos ingresando hasta dar con un numero lógico según el menú e trabajo.

* ingreso\_fecha (Error)

Cuando bajo ciertos momentos se nos pide que ingresemos un fecha, se deben pedir todo los datos asociados a la misma como lo es el año el mes y el día, eso es lo que hace esta función, lo que nos permite ingresar un valor de dato de fecha que inicia con el año el cual podemos colocar cualquier valor que corresponda a un numero lógico en el presente no podemos colocar fechas futuras ya que no se tiene registro de años futuros, por lo que pasa lo mismo en los meses, no podemos colocar mas de 12 meses donde utilicemos la función ya que en tal caso hay un repetidor que nos pude servir de comprobante, por lo que si no se coloca el valor esperado seguirá procesando con la función Error también en caso de ingresar un str, el menú de trabajo tiene a siguiente forma:



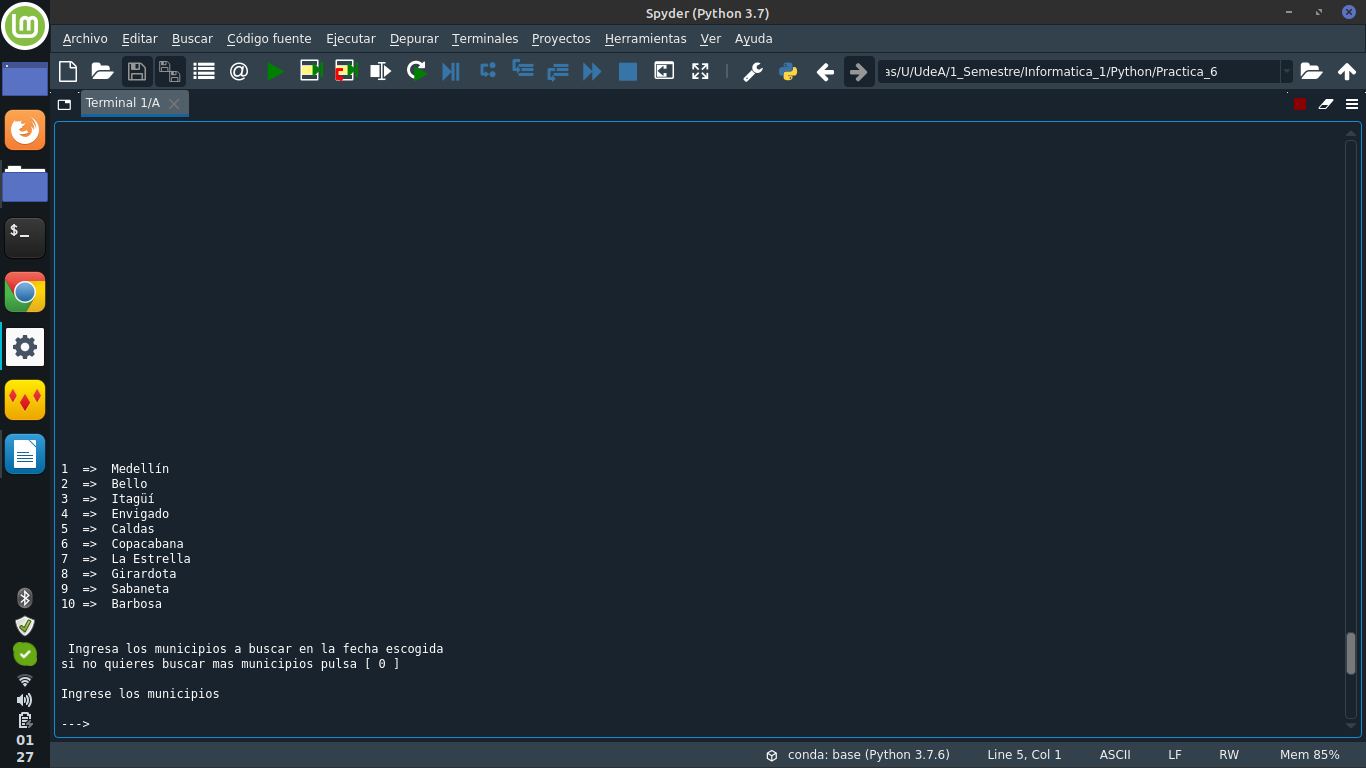


También sabemos que a la hora de procesar la información de días estos tiene que coincidir con el mes ingresado previamente evitando que por ejemplo se ingrese un 30 de febrero para así poder permitirnos un análisis correcto de las fechas, para que se de a entregar la forma de variable de vuelta sea el dato que se este pidiendo o en otras palabras una información factible de su uso en otras partes del programa.

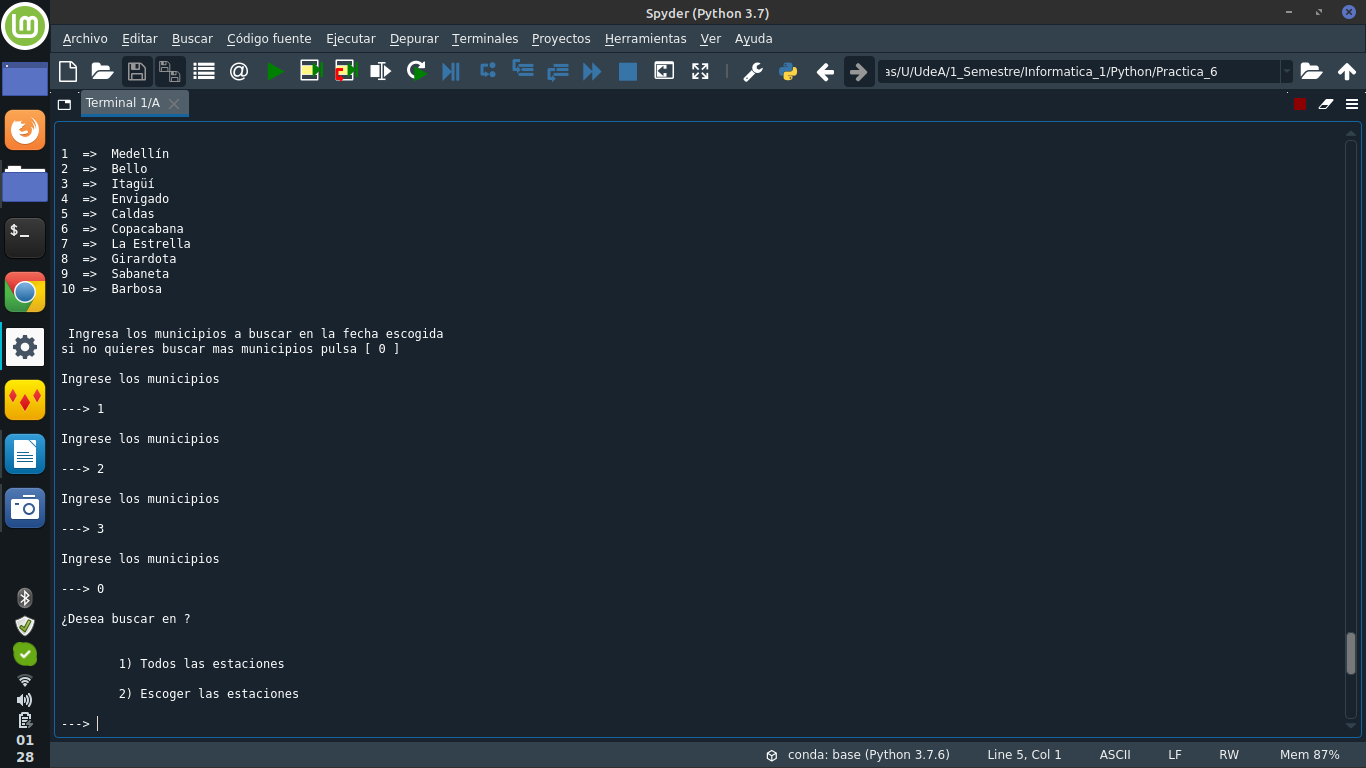
* muni\_est\_val (directorio,Error)

Cada función tiene una importancia especial, esta se creo para el uso en la busqueda mas específicamente en las funcione de busqueda, y busqueda\_2 ya que nos permite obtener una lista con los municipios a buscar, las estaciones de cada municipio que se desean buscar, y los valores que se desean imprimir de la misma forma.

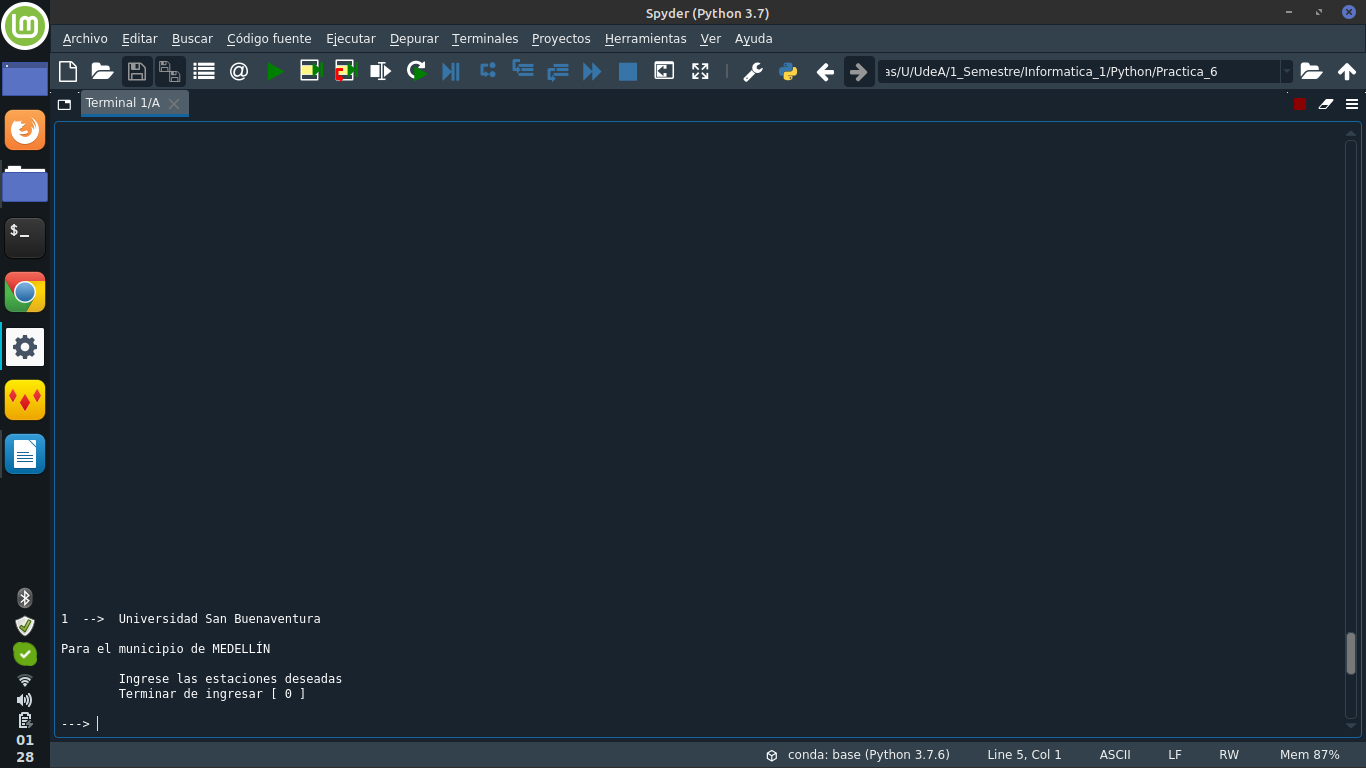
Lo primero que pregunta es si se desea buscar en todos los municipios o si se desea seleccionar los municipios a buscar, para eso abre un menú de pregunta con la función Error, algo que nos demuestra la importancia de esta función para el desarrollo avanzado en estos sistemas de creación continua y de producción de contenido especializado para el desarrollo del trabajo, si se selecciona todos los municipios, inmediatamente genera una lista con las claves del diccionario de municipios creada anteriormente con la función dic\_1 que se menciono antes, por lo que agrega cada municipio para hacer el análisis correspondiente, si se selecciono la opción de escoger municipios se imprime con este mismo diccionario la ubicación de la clave y su valor en forma de cadena de caracteres para que le usuario entienda la forma de uso, y así poder seleccionar las opciones de municipios, sin que hallan municipios que se salgan de este rango, ya que hay una cantidad de claves predefinidas y que terminan en 0 para dejar de contar, este es el menú mostrado:



Luego de ingresar las variables en dicho menú, pregunta las estaciones según las cuales se desea buscar para poder tener una busqueda mas personalizada de los lugares de información, muestra un menú de esta forma, un menú que también utiliza la función Error

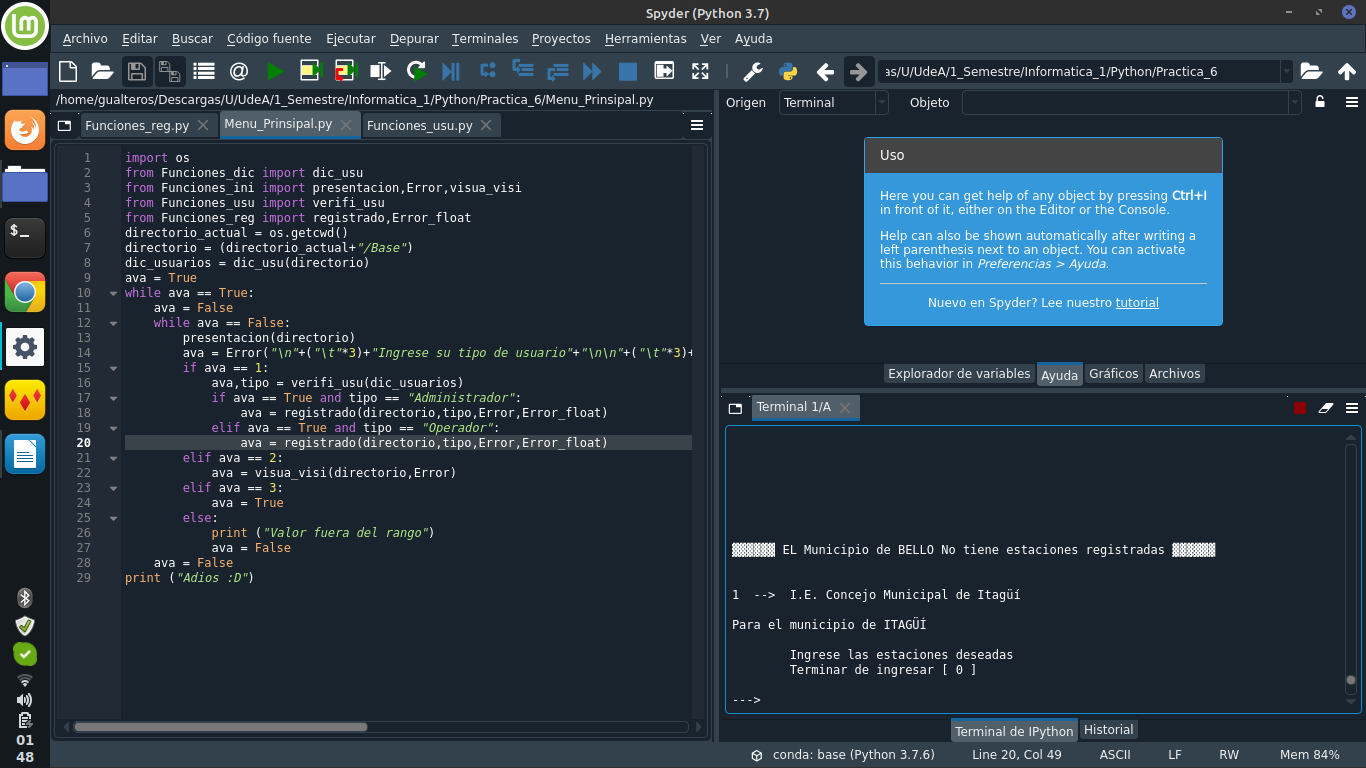


Ahora que sabemos si el usuario quieres trabajar con todas las estaciones de los municipios seleccionados o solo con las estaciones que el selecciono, para ellos se abre un menú personalizado dependiendo de los municipios seleccionados:

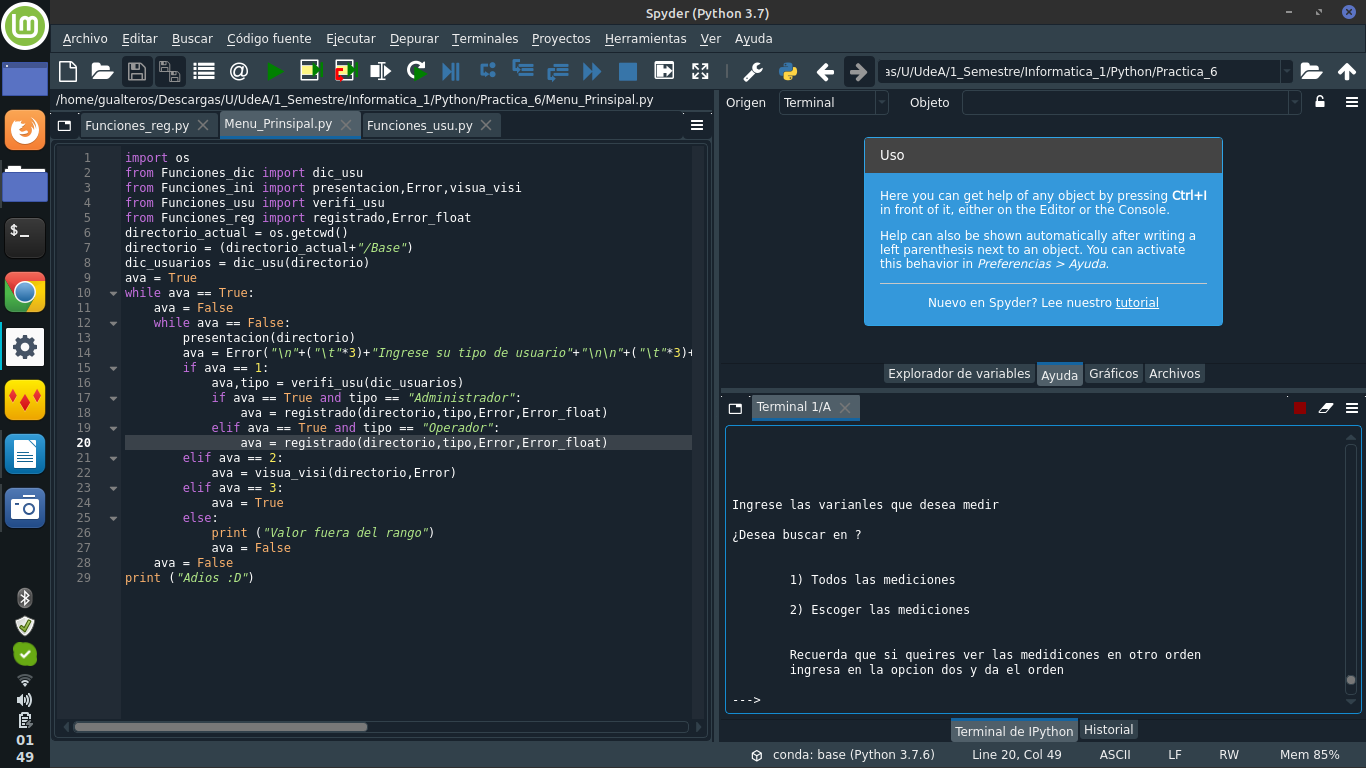


Se realiza un menú personalizado que primero imprime todas las estaciones del municipio analizado y luego imprime un comentario con el nombre del municipio en mayúscula y la opción de ingresar los valores según se quiera pero recordando que como trabaja con la función Error no se pueden valores invalidados como str, y tampoco se permite el uso de valores fuera del rango ya que es algo que analiza la función antes de agregarlo a la lista de lista de estaciones, ya que para cada municipio se le asigna una ubicación mismo que usaremos para también encontrar las estaciones a la hora de imprimir.

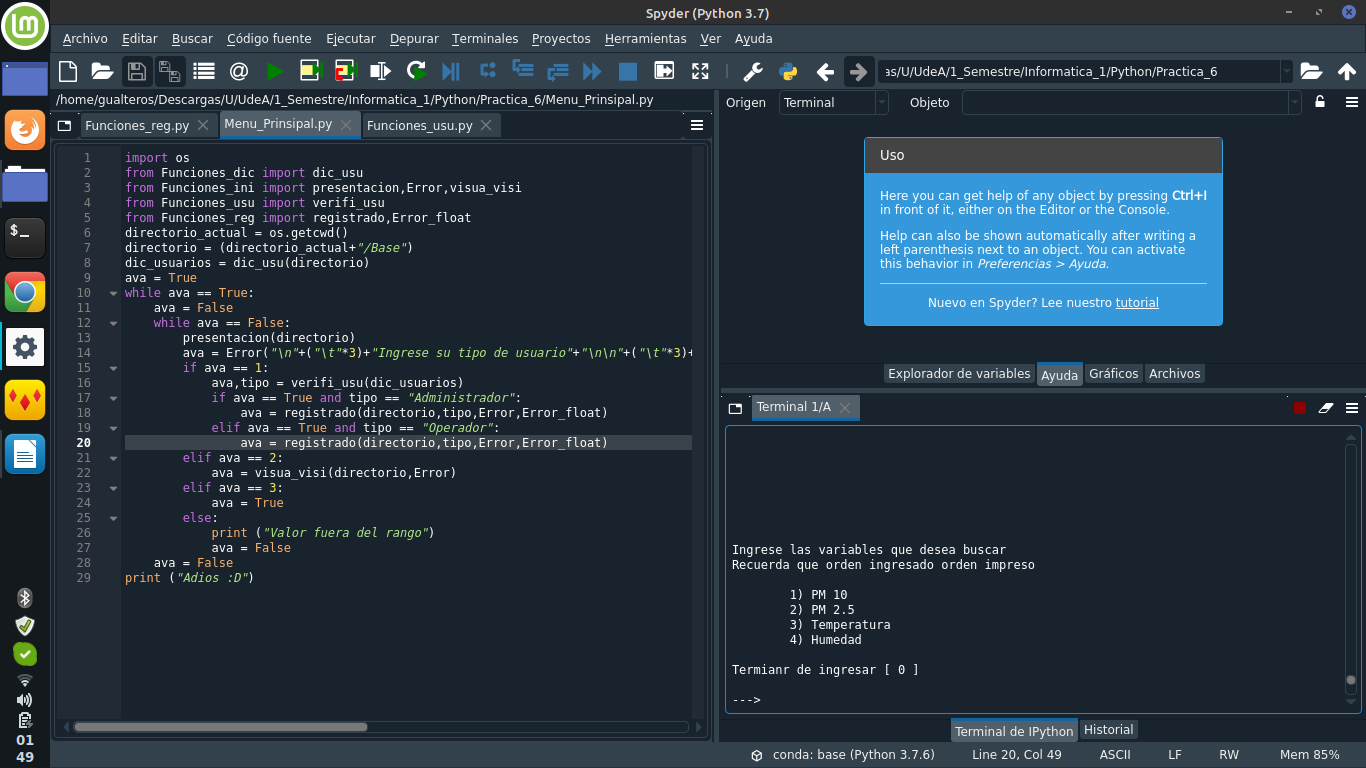
En caso de que un municipio seleccionado no tenga estaciones seleccionadas imprime el siguiente mensaje y pasa al siguiente municipio a a analizar para facilitar el ingreso de la información, con el mismo método anterior:



Luego de haber ingresado las variables de municipios y estaciones que queremos analizar, también queremos saber del usuario si este quiere trabajar con todas las variables o solo con las variables que el escoja, se muestra el siguiente menú:



Si el usuario desea recibir la información en otro orden para todas las variables puede seleccionar la opción 2 y dar el orden que se quiera ya que si se selecciona todas las mediciones ingresa todas las 4 posibilidades dadas anteriormente de para su impresión, ya con el cambio definido se sabe que si no son las 4 opciones en el orden estándar o son menos medidas variables se muestras el siguiente menú donde se nos permite ingresar según los valores :



Igual que los menús anteriores ya que trabaja con la función Error por lo que no permite ingresar un valor fuera del rango o un valor diferente a los contados ya que se estarán sumando dentro de las posibilidades entre 1 y 4, y al llenarse las lista con 4 valores dejara de contar por eso es que se pueden repetir la información, cambiar el orden o solo pedir un valor según la necesidad que tenga el visitante y termina ingresando 0, creando al final una listad de valores según posiciones que usaremos en las funciones de busqueda mas adelante.

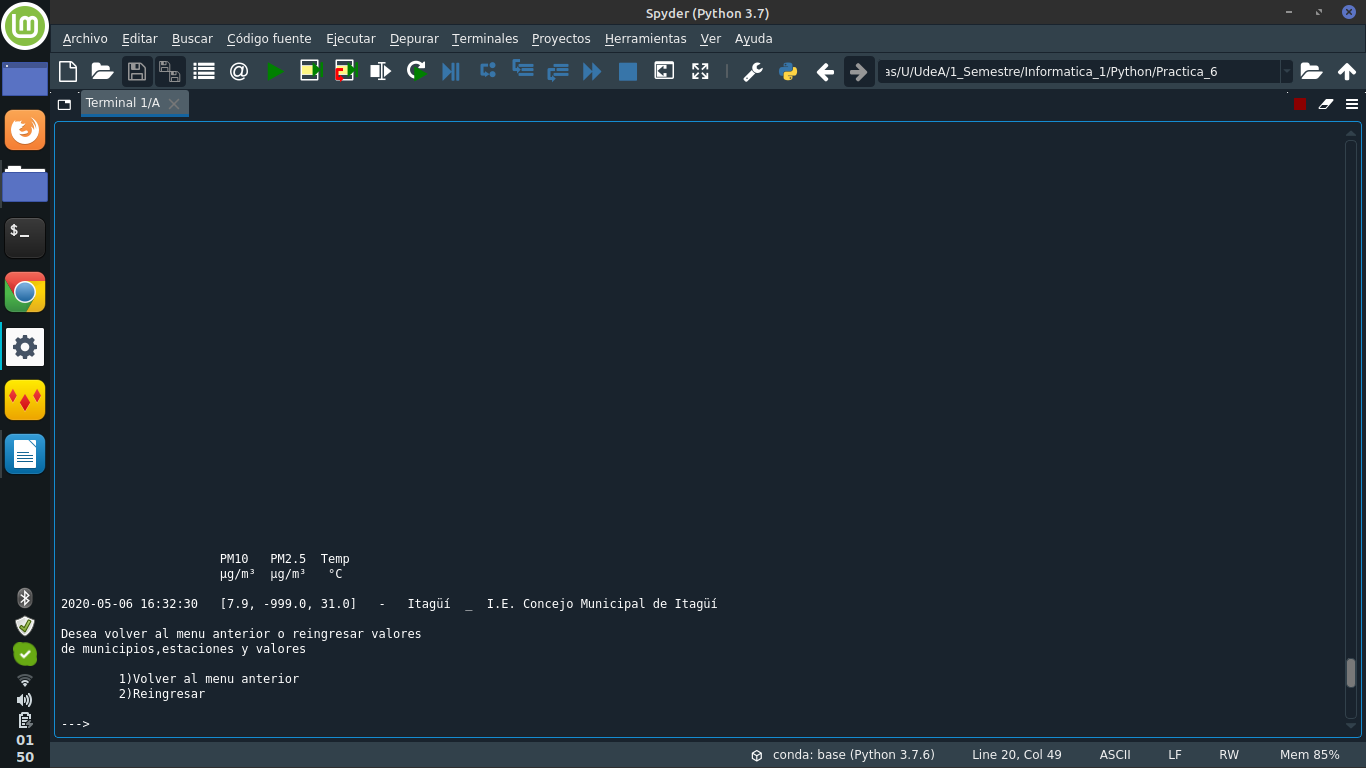
* busqueda (year,month,day,directorio,mun\_est\_val,Error)

A la hora de buscar algo se necesitan unos parámetros de busqueda y un lugar donde buscar, por lo que la función anterior muni\_est\_val es justa para obtener unos parámetros de busqueda, por lo que esta nos entrega tres listas, cada una con un valor diferente, algo que nos permite conocer por ejemplo en que municipios vamos a buscar, ahora se abre el documento ingreso.txt, que nos permite buscar linea por linea y comparar según los parámetros necesarios.

Una novedad es que dependiendo de los valores se imprime un comentario que da la razón o forma de medición de las cantidades variables que el usuario pide analizar, algo que a la hora de la impreciso hace mas preciso su trabajo, ya que depende de los valores que el usuario requiere, también el tipo de valores que le usuario necesita conocer para así poder conocer la importancia de los mismos datos, ya que cada valor de la variable es solicitable y se maneja dentro de los siguientes rangos de posibilidades de impresiones:

* + PM10 → Polvo, hollín en el aire
  + PM2.5 → Polución del ambiente
  + Temperatura
  + Húmeda

Estos datos son solicitados eh impresos según la necesidad eh imprime el siguiente comentario para luego imprimir las variables que se soliciten:



Ya sabiendo el comentario que se tiene que usar se comienza con dividir la la linea de texto del documento en partes primero los años y se compara con las variables de años dadas, si este año es mayor o igual al año analizado, continua, y se corta la cantidad del año en el str de linea para continuar con el análisis, luego se corta el pedazo de mes y se avanza pregunta si dicho mes es menor al mes analizado en caso de que estemos el mismo año, por lo que esos casos se descartan por estar en el pasado de la fecha requerida, ahora si resulta que nos encontramos en el mismo año mismo mes y el día resulta ser menor al día de la fecha de análisis, esos casos se descartan por estar en el pasado y solo le analiza los que estén en el futuro de la fecha requerida a buscar.

Luego de encontrarnos con las lineas dentro del texto que necesitamos saber, y que se encuentra dentro de los parámetros, se comienza con crear la lista de valores, ya que se sabe que dicha lista va en un orden predefinido por posición de datos creada, lo primero que hace es actuar con una linea de trabajo creando un diccionario con todos los datos d el información de la linea, ya que a cada ubicación dentro del diccionario con un ciclo for para la creación del diccionario en posición igual a la ya definida dentro de los valores de val ya obtenidos anteriormente, por lo que la lista que se creo y se imprime busca dentro el diccionario las posiciones que se pidieron y se agregan a la lista según lo necesitado, para luego imprimir una linea de texto concatenada con los valores pedidos y los municipios de busqueda que hacen mas sencillo su ubicación dentro de cada busqueda, por lo que hace el mismo procedimiento para cada linea.

* busqueda\_2 (year\_1,month\_1,day\_1,year\_2,month\_2,day\_2, directorio,mun\_est\_val,Error)

Las cosas no cambian mucho según la variable anterior mas que todo porque la variable anterior analiza desde la fecha previa entregada hasta el final, y este algoritmos analiza desde la fecha previa integrada hasta fecha posterior también integrada, ahora ya que sabemos que dicha función tiene que buscar las lineas dentro del código, lo primero que aplica es la función de muni\_est\_val ya que nos permite averiguar sobre que datos estamos analizado osea que municipios vamos a buscar que estaciones respecto a que municipios, y los valores analizados, por ejemplo igual a la función anterior, imprime según los valores que quiera el usuario

los comentario que informan que variables estamos viendo.

Cuando se hace la comparación con la fecha solo cambia el valor de year por year\_2 que se predefine, como valor de inicio de busqueda como year\_1, ya que la posibilidad del trabajo se modifica según la cantidad de resoluciones que este toma, y las impresiones se hacen directamente del documento ingreso.txt analizando cada linea y solo imprimiendo las lineas que estén dentro del rango.

# FUNCIONES\_USU

Este documento solo tiene una función que utiliza a la variable, de dic\_usuarios, para confirmar si valores posteriores ingresados por el usuario se encuentran dentro de la variable diccionario creada a base de la función dic\_usu del documento Funciones\_dic que nos permite usar esta variable para verificar si se encuentra o no dentro de la base de datos de usuario.

verifi\_usu

La función en si no es es compleja ya que usando el diccionario de usuario como base comparativa pregunta al usuario si quiere ingresar o volver al menú inicial, y en caso de querer ingresar, le solicita que ingrese unos valores de usuario y unos valores de contraseña, para luego verificar su este valor de usuario es la clave de alguna lista dentro del diccionario usuario, lo que nos permite saber si este usuario existe o no, y en caso de existir confirma la contraseña, que en caso de ser invalida vuelve desde el comienzo solicitando ingreso de usuario o continuación por valor incorrecto, y en caso de ser correcta, crea la validación booleana acerca de continuar o no, eh imprime el tipo dentro de la lista antes mencionada, dicho tipo se devuelve junto a conti para continuar con el uso de las funciones de administrador o operador, según el tipo de usuario.

# CONCLUSIONES

Al realizar el trabajo se comprende veraz mente la importancia de las las funciones según el tipo de dato, ay que permiten hacer muchas modificaciones a los tipos de usuario algo que es muy importante y beneficioso para los datos, ademas de darnos unas buenas bases para avanzar en la investigación intensiva de los datos que se analizan y la información que estos mismo conforman ya que es muy importante los datos que se analizan y que siempre varia según la cantidad de datos necesarios para el uso de dicha información de los datos escenarios de trabajo, algo que se hace fundamental a la hora de realizar correctas ediciones de trabajo, ya que al final de cuentas eso es lo importante en la realización de esta practica la conservación de los datos analizados, algo que se suele llamar conservación de información y que según el análisis del mismo nos permita administrar, la información proporcionada.

El uso de la información dada anteriormente fue correcta, ya nos sirvió de guía para la investigación personal que se realizo con la intención de mejorar el análisis de uso de datos, por lo que se puede proporcionar esta misma información de manera muy correcta así se puede usar los datos de trabajo de manera que estos ser conserven el orden y formato requerido por el usuario,

Se aprendió bastante y eso es de buen uso general que nos entrega una correcta base para poder continuar estudiando.

# BIBLIOGRAFÍA

1. Archivo de trabajo:
2. <https://www.dropbox.com/s/t0zqqfeunt0sql2/GuiaLab6.pdf?>