**main():**

databaseconfiguration se configuran todos los diferentes parámetros de la base de datos

(database) “emailDB” nombre de la carpeta donde se van a guardar los emails y

las configuraciones

utils.load\_database carga la base de datos usando la información guardada en

(utils)

show\_menu opciones del menú

read\_int\_option input message y lee la opción del usuario

- 1

list\_emails lista de emails en la base de datos

- 2

show\_email muestra el contenido del email

choose\_email el usuario elige el email deseado

- 3

create\_email pide al usuario rellenar los datos del email y lo crea en

la base de datos. Crea su archivo de texto

- 4

delete\_email elimina el email elegido

choose\_email el usuario elige el email deseado

- 5

show\_folders muestra los emails contenidos en la carpeta elegida

choose\_folder muestra las carpetas de la base de datos y el usuario

elige la deseada

- 6

create\_folder pide al usuario el nombre de la carpeta que se desea

crear y la crea en la base de datos

- 7

delete\_folder elimina la carpeta elegida

choose\_folder el usuario elige la carpeta deseada

- 8

add\_email\_to\_folder añade el email en la carpeta

choose\_email el usuario elige el email deseado

choose\_folder el usuario elige la carpeta deseada

- 9

remove\_email\_from\_folder elimina el email elegido de la carpeta elegida

show\_folder el usuario elige el email de la carpeta deseada

- 10

search busca el texto introducido por le usuario en la base de datos.

Devuelve los emails que contienen dicho texto

- 0 salida del programa

utils.write\_database guarda toda la información relacionada con los cambios realizados

(utils)

**database.py: base de datos, configuración y diferentes operaciones**

DatabaseConfiguration se configuran todos los diferentes parámetros de la base de datos

\_init\_ Inicializa la configuración de la base de datos con el directorio de la

base de datos (donde se encuentra el archivo de configuración).

También permite configurar un directorio de correo electrónico

diferente y una extensión de correo electrónico.

get\_config\_path Devuelve la ruta donde se encuentra el archivo de configuración de la

base de datos.

get\_email\_path habiendo recibido un id de correo devuelve su ubicación

Database Administra las diferentes operaciones de la base de datos

\_init\_ Inicializa la clase de base de datos con una semilla dada y la instancia

de configuración de base de datos que contiene la ruta donde se

almacenan los correos electrónicos.

self.emails = LinkedList()

add\_email Agrega el correo electrónico dado a la base de datos y a la carpeta

especificada. Si no se proporciona una carpeta, el correo electrónico

será añadido a la carpeta de la bandeja de salida por defecto.

remove\_email Elimina el correo dado de la carpeta dada. Si no se proporciona

la carpeta, el correo electrónico se elimina por completo de la base

de datos.

get\_email Busca el correo electrónico dado en la base de datos y lo devuelve.

get\_email\_ids Devuelve las identificaciones de correo electrónico de una carpeta

determinada.

create\_folder Añade la carpeta a la base de datos

remove\_folder Elimina la carpeta de la base de datos

search Busca el texto en el título y el contenido del correo y devuelve el correo

get\_folder\_names Devuelve una lista con las carpetas de la base de datos

**email.py: contiene clase de los correos**

Email Contiene toda la información del correo

\_init\_ Inicializa el correo

\_str\_ Se la llama cuando se quiere mostrar el contenido del correo

**folder.py: contiene clase de las carpetas**

Folder Contiene el nombre de la clase y la lista de correos

\_init\_ Inicializa la carpeta y le pone nombre

**utils.py: contiene cosas útiles del programa**

load\_email Carga el objeto de correo de un archivo de texto.

write\_email Escribe el texto del correo

delete\_email Elimina el correo

load\_database carga la base de datos usando la información guardada en Dbaseconfig

write\_database Escribe el archivo de texto de una base de datos dada

**linked\_list.py: linkedlist (cada elemento es un objeto separado): Los elementos de la**

**lista enlazada no se almacenan en una ubicación contigua están**

**vinculados mediante punteros.**

Node Son los objetos separados que conformaran la linkedlist

\_init\_ Inicializa un nodo que tendrá algún dato

Linkedlist Implementa una linked list

\_init\_ Inicializa la lista enlazada

append Añade un elemento al final de la lista

insert Inserta un elemento en la posición indicada

remove Elimina de la lista la primera aparición del elemento

pop Elimina el elemento de la posición indicada y lo devuelve

clear Elimina todos los elementos de la lista

index Devuelve el primer índice del valor

\_\_len\_\_ Devuelve la longitud de la lista

\_\_str\_\_ Devuelve una representación de la lista