



Practica 1

Manual

Técnico

2022

Lenguajes Formales y de Programación

Damaris Julizza
Muralles Véliz
202100953



Detalles de desarrollo

Este software se desarrollo en el lenguaje python, utilizando como interfaz gráfica la librería de Tkinter.

Descripción general

El programa es un software que permite presentar de una forma más simple la información relacionada con el pensum de estudios.

Cuenta con distintas opciones para desplegar la información de un pensum de estudios, facilitando la forma visual de cada uno de ellos para saber así qué cursos puede llevar. Todo el desarrollo de la interfaz como se a mencionado con anterioridad se realizo mediante Tkinter.

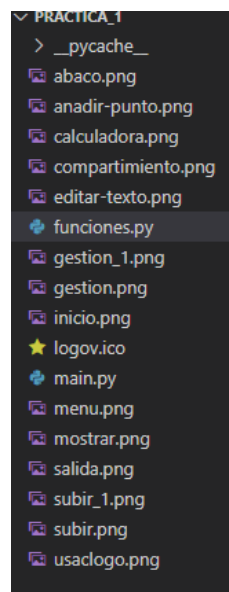
Paradigma utilizado

El paradigma utilizado en esta practica fue el paradigma declarativo ya que se nos describió que se debía realizar en el programa declarando propiedades y reglas que deben cumplirse y no se especifica como se realizaría concretamente.

Lógica del programa

Recursos Utilizados:

En el folder “Practica_1” encontraremos todos los recursos y archivos utilizados, como lo son las imágenes usadas y los archivos de las clases utilizadas, estos ultimas son “funciones” y “main”.



Funciones:

```
class funcionesgestion():
```

Este archivo contiene la clase “funcionesgestion” el cual este compuesto por varios métodos que realizan cierta acción sobre la lista que se encarga de almacenar los datos que se ingresen en el software, entre estas funciones tenemos:

➤ Método `__init__`:

Este se inicializa los atributos del objeto creado, en este caso se coloco dentro el constructor de la lista utilizada para almacenar y manejar toda la información que se estará agregando al sistema denominada “listalineas”.

➤ Función leer:

En este atributo se espera recibir un atributo el cual será la dirección de donde se encuentra el archivo que se leerá, luego realiza las validaciones necesarias antes de abrir el archivo y almacenar la cadena de texto encontrada en este en otra variable, al hacer esto el archivo se cierra y se procede a llamar al método de enlistar.

En caso de que el atributo recibido no cumpla las validaciones se procede a generar un mensaje para advertir al usuario al respecto. Para realizar todas las validaciones de este método se utilizaron las librerías de messagebox de tkinter, la librería os y pathlib, así como también una condición if.

➤ Función enlistar:

El método recibe la cadena de texto almacenada en el método leer y haciendo uso de la del método Split() para separar la cadena respecto a los saltos de línea y las comas, para implementar esto de manera eficiente se hizo unos de ciclos for para ir introduciendo los datos separados en la lista principal con el método append() propio de las listas.

Cabe mencionar que se hizo la validación de que el dato ingresado no puede estar dos veces en la lista general, por lo que antes de utilizar el método append(), se utilizo otro ciclo for y condiciones if que buscara en todo lo contenido en la lista encontrando así, si hay datos repetidos, en caso de encontrarse un dato similar se sobrescribirán los datos respectivos a este.

➤ Función mostrar:

Esta función hará una búsqueda entre todos los datos de la lista general para encontrar una coincidencia, para ello esta función también espera que se le pase un valor al ser llamado.

Si encuentra coincidencia retornara información en tipo de lista que corresponde a los datos correspondientes a la coincidencia, en caso de no ser encontrada una coincidencia generara un mensaje para informar al usuario, para todo esto se utilizaron ciclos for y condiciones if.

➤ **Función Agregar:**

Recibe dos valores como atributo, uno de ellos de tipo lista y mediante esta información primero realiza una búsqueda en la lista general mediante el uso de un ciclo for para encontrar información repetida, luego agregara a la lista general los datos contenidos en la variable de tipo lista según las validaciones correspondientes realizadas con la condicional if.

➤ **Función editar:**

Esta función al igual que el anterior espera recibir dos valores de los cuales uno es de tipo lista y posteriormente mediante ciclos y condiciones hace validaciones para encontrar en la lista un dato que coincida con el que le fue proporcionado, si lo encuentra sobre escribe la información en esta posición de la lista.

➤ **Función eliminar:**

Al igual que las demás funciones este espera un valor y mediante ciclos for y condiciones if valida que haya coincidencia de este dato con los datos de lista, ubicada la posición en la que se encuentra una coincidencia procede a eliminar este dato mediante el método pop() propio de las listas.

➤ **Funciones aprobados, cursados, pendientes, hastasemestre y semestre:**

Estas funciones tienen una misma finalidad, la cual es sumar los datos correspondientes a los créditos agregados en la lista, para esto hacen uso de ciclos for y condiciones if y luego retornan el valor de la suma para su uso.

Las primeras tres funciones hacen una búsqueda respecto a condiciones determinadas, mientras que las últimas dos lo harán respecto al valor que se le sea pasado a su atributo.

main:

```
class Ventana(object):
```

Aquí se encuentra la clase ventana, la cual contiene las funciones necesarias para la lógica del programa y todo lo relacionado con la interfaz, por lo cual se hizo utilización de muchas librerías de Tkinter y también se importó el archivo funciones que explicamos con anterioridad.

➤ **Método `__init__`:**

Como este método sirve para inicializar lo que utilizaremos, aquí se creó la ventana y se hizo la configuración del tamaño para esta utilizando la librería Tk de Tkinter, también se crearon los frames que se usaron como divisiones y se asignaron las variables que se estarían utilizando en los cuadros de entrada de texto, también se llamó a la función de “elementos” en la cual se encuentra todo lo relacionado con la interfaz.

➤ **Funciones `pantalla_**`:**

Este grupo de funciones se encarga de hacer un llamado de las páginas o vistas que tendremos, así como las configuraciones necesarias para que los elementos llamados puedan estar posicionados de una forma adecuada haciendo uso de los métodos `columnconfigure()` y `rowconfigure()`.

➤ **Función `salir`:**

Esta función hace uso del método `destroy()` y permite cerrar la ventana principal del programa.

➤ **Función `menu_lateral`:**

Esta función es la encargada de las configuraciones para la animación del menú desplegable, para ello hace uso de los métodos `config()`, `update()`, `grid_forget()` y `grid_propagate()`, utiliza ciclos `for` u condiciones `if`, también hace un llamado a la función `pantalla_inicio()`.

➤ **Función `limpiar`:**

En esta función, mediante el uso del método `set()`, se asigna a las variables creadas en el `__init__` un espacio vacío lo que equivale a limpiar los datos que se les hayan asignado en el transcurso de la ejecución del programa.

➤ **Funciones `leerarchivo`, `listarcursos`, `mostrarcurso`, `editarcurso`, `eliminarcurso`, `creditoshastaNsemestre`, `creditosensemestre` y `detallecreditos`:**

Estas funciones son las que se encargan de la lógica principal del programa, estos son llamados al presionar botones en la interfaz y después de realizar ciertas validaciones mediante uso de condiciones `if` y `try-except`, llama a los métodos que creamos en nuestra

clase “funcionesgestion” que se encuentra en el archivo “funciones”, además de que la mayoría también espera el valor retornado de dichos métodos para su uso posterior.

➤ **Función elementos:**

Esta función como mencionamos posee toda la programación referente a la interfaz, primero encontramos la definición de las imágenes que utilizaremos mediante la librería PhotoImage de Tkinter.

Posterior a esto se usaron buttons y labels para colocarlos en la ventana principal creando así el menú desplegable.

Para este programa solo se utilizó una ventana principal y el resto se hizo mediante la utilización de la librería ttk.Notebook de Tkinter se hicieron las configuraciones para definir su estilo y luego se crearon las “paginas” incorporando frames dentro de estas.

Una vez definidas las paginas a utilizar se comenzó a diseñar la interfaz incorporando labels, buttons, entrys y spinbox definiendo la información que se mostrará en cada cosa y las configuraciones respecto a color de letra, la posición que ocuparían y el command que se utilizaron solo en los botones para llamar a las funciones creadas para la lógica del programa.

También se hizo una treeview para generar la tabla en la que se muestra toda la información cargada al sistema, para esta tabla también se hicieron configuraciones de color, posición, ancho, alto, etc.