

# PROGRAMACIÓN CON PYTHON

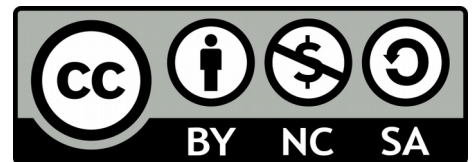
(CEFIRE CTEM)



## Interfaces gráficas con Tkinter Ejercicio obligatorio

Esta obra está sujeta a la licencia Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visitad

<http://creativecommons.org/licenses/by-nc-sa/4.0/>.



Autora: María Paz Segura Valero (mpazprofe@gmail.com)

## CONTENIDO

|                                   |   |
|-----------------------------------|---|
| 1. Introducción.....              | 2 |
| 2. Enunciado.....                 | 2 |
| 2.1. Interfaz gráfica.....        | 3 |
| 2.2. Lógica de la aplicación..... | 4 |

## 1. Introducción

En este documento puedes encontrar el **ejercicio obligatorio** de esta unidad. Es imprescindible entregarlo en tiempo y forma para superar esta parte del curso.

Tendrás la oportunidad de realizar la entrega de varias versiones del ejercicio hasta que consigas superarlo y la profesora te indicará en cada corrección las mejoras necesarias.

En cualquier momento puedes lanzar preguntas al **foro** del curso o realizar una **entrega parcial** del ejercicio acompañada de una **lista de dudas** para que la profesora pueda orientarte en su resolución.

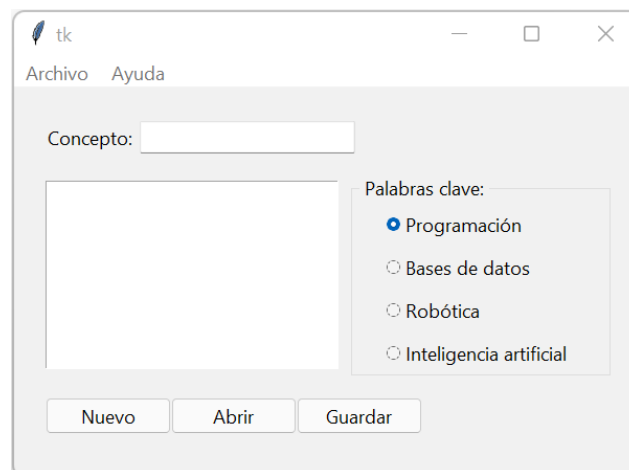
## 2. Enunciado

Vamos a crear una aplicación que nos permite gestionar un **Glosario** de términos informáticos.

Cada concepto estará almacenado en un fichero de texto distinto que contendrá una línea con el siguiente formato: `concepto|descripción|materia`.

El programa nos permitirá cargar los datos de un fichero, modificar su información y guardarlo.

El aspecto de la aplicación será el siguiente:



## 2.1. Interfaz gráfica

Las características de los widgets se recogen en la siguiente tabla:

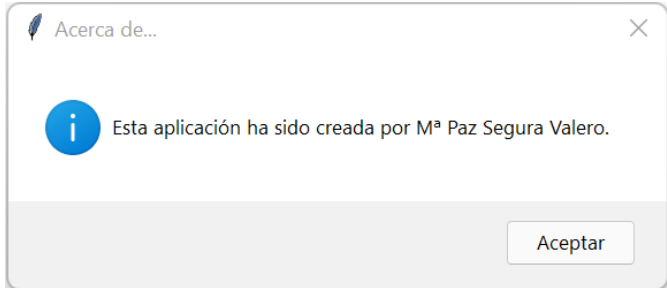
| Widget                 | Características   |
|------------------------|---|
| Raíz                   | <b>Ancho</b> de 600 píxeles, <b>altura</b> de 370 píxeles y posicionado ( <b>X, Y</b> ) donde más te guste.<br>Deberá tener 20 píxeles de <b>margen</b> entre el borde y su contenido.  |
| Marcos de distribución | Crear tres <i>Frame</i> para poder distribuir el resto de widgets. Cada marco tendrá estas características: <ul style="list-style-type: none"> <li>• <b>Margen</b> de 10 píxeles entre su borde y su contenido.</li> <li>• Rellenar (<b>fill</b>) todo el eje X.</li> <li>• Alineación (<b>side</b>) top.</li> </ul>  |
| Concepto               | <b>Etiqueta</b> que muestre el texto <i>Concepto</i> : alineado a la izquierda.<br>Campo de <b>entrada</b> corta que tenga asociada una variable de control de tipo <code>StringVar()</code> .<br>Se mostrará en el primer marco.   |
| Descripción            | Campo de <b>entrada multilínea</b> con un ancho de 23 caracteres y altura de 8.<br>Se mostrará en el tercer marco alineado a la izquierda.  |
| Palabras clave         | <b>Marco</b> de tipo <i>LabelFrame</i> que muestre el texto "Palabras clave:". Se mostrará en el tercer marco alineado a la derecha.<br><br>Dos <i>widgets</i> de tipo <i>Radiobutton</i> que mostrarán las opciones: <ul style="list-style-type: none"> <li>• "Programación" con valor 1 (valor por defecto)</li> <li>• "Bases de datos" con valor 2.</li> <li>• "Robótica" con valor 3.</li> <li>• "Inteligencia artificial" con valor 4.</li> </ul> Deberás asociarles una variable de control de tipo <code>IntVar()</code> .<br>Se mostrarán en el <i>LabelFrame</i> anterior. |
| Botones inferiores     | Crear un <b>botón</b> que mostrará el texto "Nuevo" y ejecutará la función <code>nuevo()</code> .<br>Crear un <b>botón</b> que mostrará el texto "Abrir" y ejecutará la función <code>abrir()</code> .<br>Crear un <b>botón</b> que mostrará el texto "Guardar" y ejecutará la función <code>guardar()</code> .<br><br>Se mostrarán en el cuarto marco alineados a la izquierda.  |

|      |  |
|------|--|
| Menú | <p>Crear el <b>menú principal</b> de la aplicación con las siguientes opciones:</p> <ul style="list-style-type: none"> <li>• Submenú <i>Archivo</i>: opciones <i>Nuevo</i>, <i>Abrir</i> y <i>Guardar</i>, un separador, opción <i>Salir</i>. <ul style="list-style-type: none"> <li>◦ Las opciones <i>Nuevo</i>, <i>Abrir</i> y <i>Guardar</i> ejecutarán las mismas funciones que los botones homónimos.</li> <li>◦ La opción <i>Salir</i> provocará el fin de la aplicación.</li> </ul> </li> <li>• Submenú <i>Ayuda</i>: opción <i>Acerca de...</i> <ul style="list-style-type: none"> <li>◦ Esta opción llamará a la función <code>acerca_de()</code>.</li> </ul> </li> </ul> |
|------|--|

## 2.2. Lógica de la aplicación

Ahora que disponemos de la interfaz gráfica, podemos empezar a rellenar con código. En la siguiente tabla se describen las funciones.

| Widget               | Características   |
|----------------------|---|
| <code>reset()</code> | Esta función se encargará de reinicializar todos los widgets de la interfaz y será llamada desde otras funciones.   |
| <code>nuevo()</code> | <p>Mostrará un diálogo <code>AskQuestion</code> con la siguiente pregunta: "Se borrarán los datos del formulario. ¿Estás seguro/a?"</p> <p>Si la respuesta es afirmativa entonces se llamará a la función <code>reset()</code>.</p>   |
| <code>abrir()</code> | <p>Mostrará un diálogo <code>AskQuestion</code> con la siguiente pregunta: "Se borrarán los datos del formulario. ¿Estás seguro/a?"</p> <p>Si la respuesta es afirmativa entonces mostrará un diálogo de tipo <code>AskOpenFileName</code> y permitirá seleccionar ficheros de tipo <code>.TXT</code>.</p> <p>Podemos utilizar como directorio inicial el directorio actual de la aplicación.</p> <p>Si se ha seleccionado un fichero entonces se leerá su contenido para rellenar con él los widgets de la interfaz.</p> <p>El fichero contendrá una única línea con el siguiente formato:</p> <pre>concepto descripción materia</pre> <p><u>Ejemplo:</u></p> <pre>compilador Programa que traduce... 1</pre> <p>La línea tiene tres campos separados por el símbolo <code> </code>.</p> |

|             |   |
|-------------|---|
|             | <p>Existe una función de cadenas de texto llamada <b>split()</b> que nos permite obtener una lista con todos los elementos de una cadena teniendo en cuenta un carácter de separación (blanco por defecto).</p> <p>Su <u>sintaxis</u> es la siguiente: <b>mi_cadena.split(carácter_separación)</b></p> <p><u>Ejemplo:</u></p> <pre>&gt;&gt;&gt; linea = "Polaroid polaroid.png Este dibujo vectorial... 1" &gt;&gt;&gt; linea.split(' ') ['Polaroid', 'polaroid.png', 'Este dibujo vectorial...', '1']</pre> <p>Puede que la línea del fichero acabe con el carácter de fin de línea '\n' y deberás tenerlo en cuenta.</p> <p>Si no se ha podido abrir el fichero, se mostrará un diálogo de error.</p> |
| guardar()   | <p>Mostrará un diálogo AskSaveAsFilename para preguntar el fichero que queramos actualizar con la información que haya en la interfaz. La extensión por defecto será .txt.</p> <p>Escribiremos una línea con el siguiente formato:</p> <pre>título nombre_imagen descripción licencia</pre> <p><u>Ejemplo:</u></p> <pre>Polaroid polaroid.png Este dibujo vectorial... 1</pre> <p>Recuerda que la línea tiene cuatro campos separados por el símbolo  .</p> <p>Si no se ha podido guardar en el fichero, se mostrará un diálogo de error.</p> <p>Si se ha podido guardar, se mostrará un diálogo de información.</p>  |
| acerca_de() | <p>Mostrar un diálogo de información con tus nombres y apellidos.</p> <p><u>Ejemplo:</u></p>    |

Puedes crear el programa desde cero o basarte en el esquema que tienes en el fichero **ud5\_ejercicio\_obligatorio (ESQUEMA).py** del aula virtual. En este caso, deberás cambiar las instrucciones *pass* que encuentres en el código.

Sube al aula virtual un fichero con el nombre **ud5\_ejercicio\_obligatorio.py**.