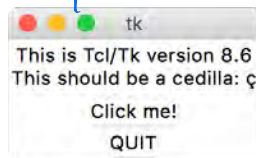
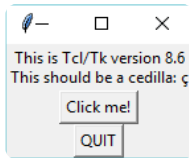


## UNIDAD 2 - tkinter, EL PAQUETE VISUAL OFICIAL DE PYTHON

tkinter es un paquete, un módulo que contiene otros módulos (o submódulos), con todo lo que necesitamos para desarrollar aplicaciones visuales usando el lenguaje de programación Tcl y el conjunto de herramientas para interfaces gráficas de usuario TK, con las que podemos crear aplicaciones nativas ejecutables, sin cambios en el código, sobre MS-Windows, macOS, Linux y otros sistemas operativos.

Si ejecutamos `python -m tkinter` en la terminal se abrirá una aplicación visual mostrando una interfaz simple de TK con información sobre la versión que tenemos instalada. En macOS muestra un



menú muy completo a través del que podemos estudiar diferentes ejemplos de aplicaciones visuales desarrollados con tkinter.

Si escribimos `tkinter` en la ayuda interactiva de Python obtendremos una documentación muy completa sobre el paquete, con una descripción concisa que incluye un ejemplo básico, y la lista de módulos y clases que contiene con una detallada descripción de cada uno de ellos y de sus métodos. Su enorme extensión nos debe hacer reflexionar sobre las enormes posibilidades que nos ofrece este paquete, que no debe asustarnos porque está bien documentado. Es conveniente tener abierto la ayuda interactiva de Python siempre que programemos en este lenguaje.

### MÓDULOS

El paquete tkinter contiene 12 módulos:

`constants` define constantes simbólicas.

`dialog` define una clase interna.

`simpledialog` define diálogos para solicitar datos al usuario.

`commanddialog` es la clase base de los diálogos nativos.

`messagebox` mensajes nativos del sistema operativo.

`colorchooser` selector de color nativo del sistema operativo.

`filedialog` selector de archivos " " " " "

`dnd` gestiona el Drag-and-Drop (Arrastrar-y-Soltar)

`font` accede a los fuentes tipográficos instalados en el sistema.

`scrolledtext` define un entrada de texto multilineal con barra de desplazamiento vertical.

`tix` extiende la interfaz gráfica, pero está obsoleto.

`ttk` define widgets basados en temas.

Su uso es relativamente sencillo, y se explicará cuando sea necesario utilizar sus clases o funciones en los ejemplos que completan estos apuntes.



## CLASES

El módulo `tkinter` define 14 clases base, de las que derivan los widgets y clases necesarias para que los componentes visuales de nuestras aplicaciones tengan el aspecto y posición que queramos y una completa interacción con los usuarios.

`Wm` conecta las ventanas de alto nivel de nuestra aplicación con el gestor de ventanas del sistema operativo.

`CallWrapper`, `Event` y `EventType` gestionan la interacción del usuario con la aplicación.

`Grid`, `Pack` y `Place` gestionan el aspecto visual de la aplicación.

`XView` e `YView` gestionan los vistas cuando algo no puede mostrarse por problemas de espacio.

`Image` gestiona las imágenes que muestra la aplicación.

`Variable` es la base de cuatro clases con las que se vinculan valores enteros, de como `PlotData`, booleanos o cadenas de caracteres a los widgets de la aplicación.

`Misc` define una miscelánea de funciones que ayudan a gestionar todos los elementos de la aplicación.

`BaseWidget` y `Widget` son la base de todos los widgets.

No utilizaremos ninguno de estas clases directamente.

Para gestionar imágenes usaremos `BitmapImage` y `PhotoImage`.  
Las variables vinculadas a los widgets los gestionaremos a través de `IntVar`, `DoubleVar`, `StringVar` y `BooleanVar`.

Usaremos `TK` y `Toplevel` para crear ventanas de alto nivel y `Button`, `Canvas`, `Checkbutton`, `Entry`, `Frame`, `Label`, `LabelFrame`, `Listbox`, `Menu`, `Menubutton`, `OptionMenu`, `Message`, `PanedWindow`, `Radiobutton`, `Scale`, `Scrollbar`, `Spinbox` y `Text` para completar la interfaz gráfica de nuestras aplicaciones visuales.

Desde la terminal de tu sistema operativo, abre la consola interactiva de Python. Abre la ayuda interactiva de Python y revisa rápidamente la documentación de los módulos y clases de `tkinter`.

```
$ python  
>>> help()  
help >
```

```
help > tkinter  
---  
(Pulsa Q para salir)
```

```
help > tkinter.constants
```

```
---  
help > tkinter.colorchooser  
---
```

## APLICACIONES VISUALES DESARROLLADAS CON tkinter

Las aplicaciones visuales interactúan con el usuario a través de ventanas de alto nivel en las que el desarrollador ha colocado widgets que capturan los eventos de ratón o teclado lanzados por el usuario.

La aplicación en sí es una ventana de alto nivel del tipo `TK`, a lo que habitualmente llamaremos ventana principal (`v_principal`, `root` o `main` es el nombre que solemos dar en el código a este objeto, que en estos apuntes reduciré a `v` por simplificar).

Para crear una aplicación visual seguiremos siempre estos pasos:

- 1.- Importamos los módulos y clases necesarios.
- 2.- Definimos las funciones con las que la aplicación responde a los eventos de ratón y teclado del usuario.
- 3.- Creamos la ventana principal, un objeto del tipo `TK` al que damos un nombre y configuramos.
- 4.- Creamos los componentes de la aplicación, configurando en su constructor algunos de sus aspectos visuales (color, fuente, cursor...) y, si procede, su comportamiento ante el evento que lanzará el usuario cuando interactúe con el componente del modo esperado (haciendo clic con el botón izquierdo del ratón sobre un botón...).
- 5.- Vinculamos a cada componente visual otros eventos del usuario que queramos capturar y gestionar (hacer doble clic sobre un botón...).
- 6.- Colocamos los componentes visuales en la ventana, configurando su comportamiento visual.
- 7.- Ejecutamos la aplicación llamando al método `mainloop` de la ventana principal.

Si la aplicación utiliza más ventanas de alto nivel, las crearemos a partir de la clase `TopLevel` haciéndolas dependientes de la ventana principal a través de su opción `master`, usaremos los pasos 1 a 6 para dar el aspecto y funcionamiento deseado y los mostramos u ocultamos al usuario según proceda.

En estos apuntes empezaré con un análisis técnico de las clases y módulos que usaremos necesitados, basado en lo posible en la ayuda interactiva de Python. Una vez expuesta la teoría expandiré algunos ejemplos y, finalmente, el lector podrá aprender más viendo algunos ejercicios.

