

4.5. LA CLASE Scrollbar



Con esta clase podemos añadir y vincular barras de desplazamiento a los widgets que definen las opciones `xscrollcommand` e `yscrollcommand`, widgets que pueden tener contenido de mayor tamaño que el usado en pantalla.

Para agrupar visualmente las barras de desplazamiento y el widget vinculado es recomendable crear un `Frame` que los agrupe de todos ellos. Si solo creamos una barra de desplazamiento podemos colocarla en el `Frame` con el método `pack()`. Si añadimos ambas barras es preferible usar el método `grid()`, colocando el widget en la cuadrícula $(0,0)$, la barra vertical en $(0,1)$ y la horizontal en $(1,0)$.

Los widgets `Entry` y `Spinbox` derivan de la clase `XView`, que permite mover horizontalmente su contenido cuando es más ancho que el espacio del que dispone el widget. Si queremos añadirle una barra de desplazamiento horizontal, `sbh`, tenemos que vincular la opción `xscrollcommand` del widget a la propiedad `sbh.set`, y la opción `command` de `sbh` al método `yview()` del widget.

```
from tkinter import TK, Entry, StringVar, Scrollbar
v = TK()
vs = StringVar(v, 'En un lugar de la Mancha...')
e = Entry(v, textvariable=vs)
e.pack(fill='x')
s = Scrollbar(v, orient='horizontal', command=e.xview)
s.pack(fill='x')
e['xscrollcommand'] = s.set
v.mainloop()
```

Dos widgets `Canvas`, `Listbox` y `Text` derivan de las clases `XView` e `YView` para mover su contenido horizontal y verticalmente cuando sea más ancho o alto que el widget. Hay que crear un objeto `Scrollbar` para cada barra de desplazamiento que queramos añadirle, y vincular cada una con la opción y el método correspondiente del widget.

```
from tkinter import TK, Scrollbar, Listbox
v = TK()
sb = Scrollbar(v)
sb.pack(side='right', fill='y')
lb = Listbox(v, yscrollcommand=sb.set, bd=0)
for i in range(1000):
    lb.insert('end', str(i))
lb.pack(side='left', fill='both')
sb.config(command=lb.yview)
v.mainloop()
```