



# Programación



# Fácil

---

## El método grid() - Tkinter desde cero - Capítulo 3

El método grid() - Tkinter de cero - Capítulo 3



En el capítulo anterior ya vimos cómo añadir widgets, en concreto el widget de marco, `Frame()`. En este vamos a ver cómo añadir otro widget nuevo, `Label()` y verás las diferencias entre los métodos `pack()` y `grid()`.

# Label() de Tkinter

Empecemos escribiendo una etiqueta `Label()`. Esta etiqueta nos sirve para añadir una etiqueta de texto en la ventana o en un marco.

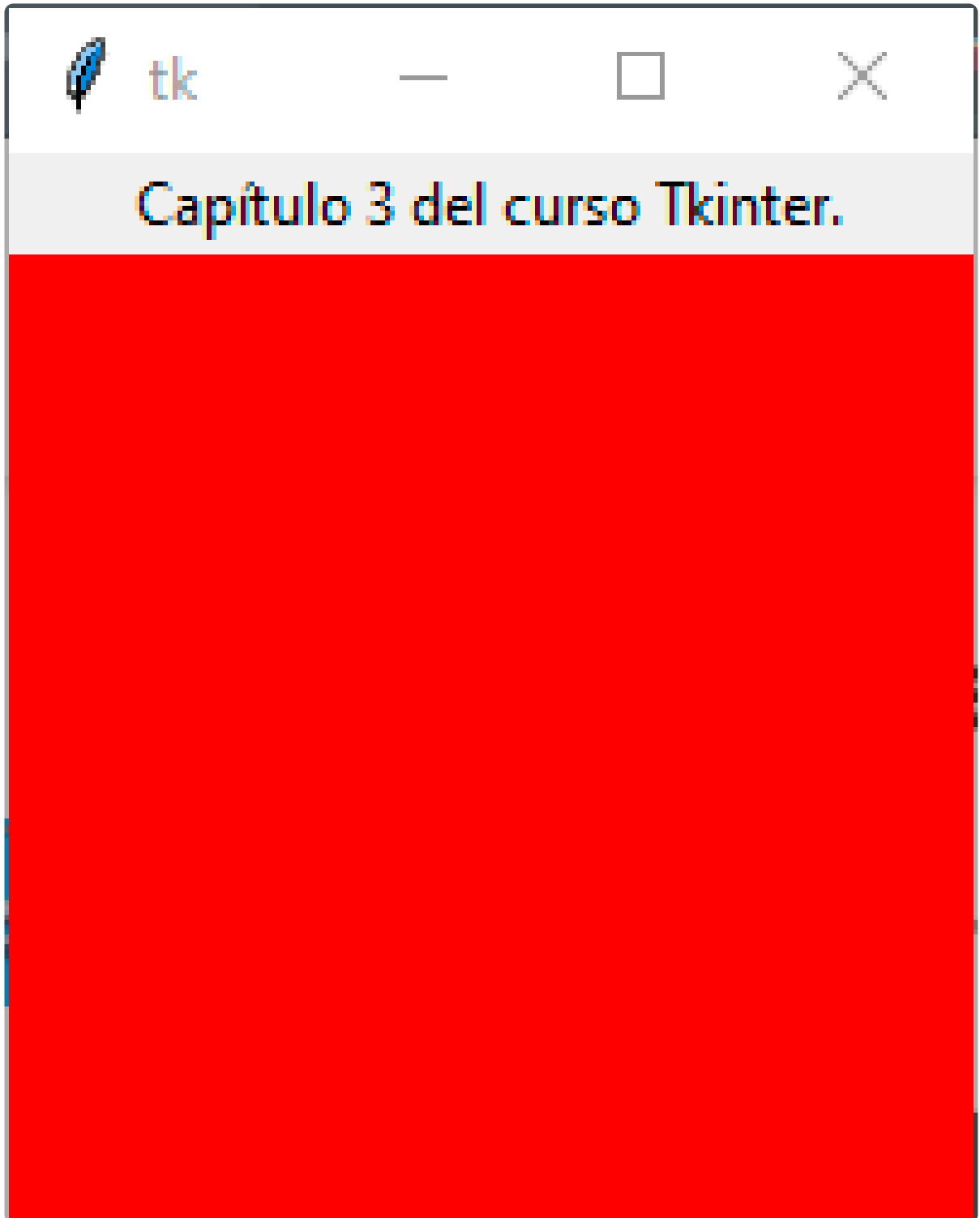
## Código Python

```
1 from tkinter import *  
2 root = Tk()  
3  
4 # Dibuja un marco en la ventana  
5 marco_principal = Frame()  
6  
7 #Etiqueta (Muestra un texto)  
8 texto = Label(root, text="Capítulo 3 del curso Tkinter.")  
9 texto.pack()  
10  
11 marco_principal.config(width=200, height=200, bg="red")  
12 marco_principal.pack()  
13  
14 root.mainloop()
```

## El método pack() de Tkinter

El método `pack()` se ejecuta según el flujo de ejecución de Python, el `pack()` que lea primero se muestra arriba en la ventana.

## Resultado



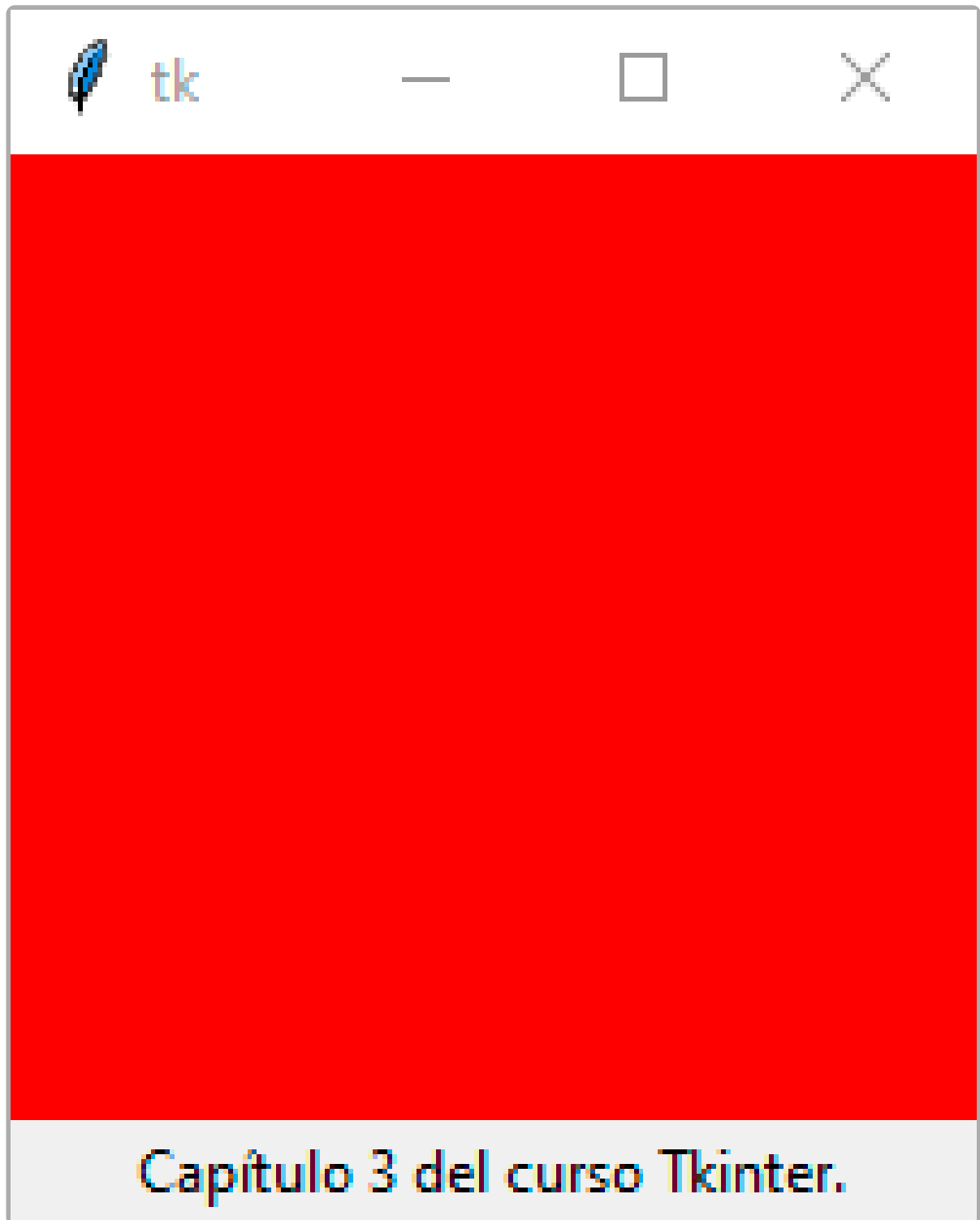
Si cambiamos este orden, el texto aparecerá debajo y el `Frame()` rojo arriba (te he marcado la parte del código que he cambiado de posición):

## Código Python

```
1 from tkinter import *  
2 root = Tk()
```

```
3 | LF
4 | # Dibuja un marco en la ventana LF
5 | marco_principal = Frame() LF
6 | LF
7 | #Etiqueta (Muestra un texto) LF
8 | texto = Label(root, text="Capítulo 3 del curso Tkinter.") LF
9 | LF
10 | marco_principal.config(width=200, height=200, bg="red") LF
11 | marco_principal.pack() LF
12 | LF
13 | texto.pack() LF
14 | LF
15 | root.mainloop()
```

🖥️ Resultado 🖥️



# El método grid() de Tkinter

Entramos en algo muy interesante de Tkinter, el método `grid()` el cuál es capaz de mostrar los elementos de un programa en forma de tabla bidimensional, es decir, podemos colocar las

cosas por columnas y filas y olvidarnos del orden en el flujo de ejecución.

Bien, ¿pero cómo especificamos estas columnas y filas?  
Con números de posición, así de fácil.

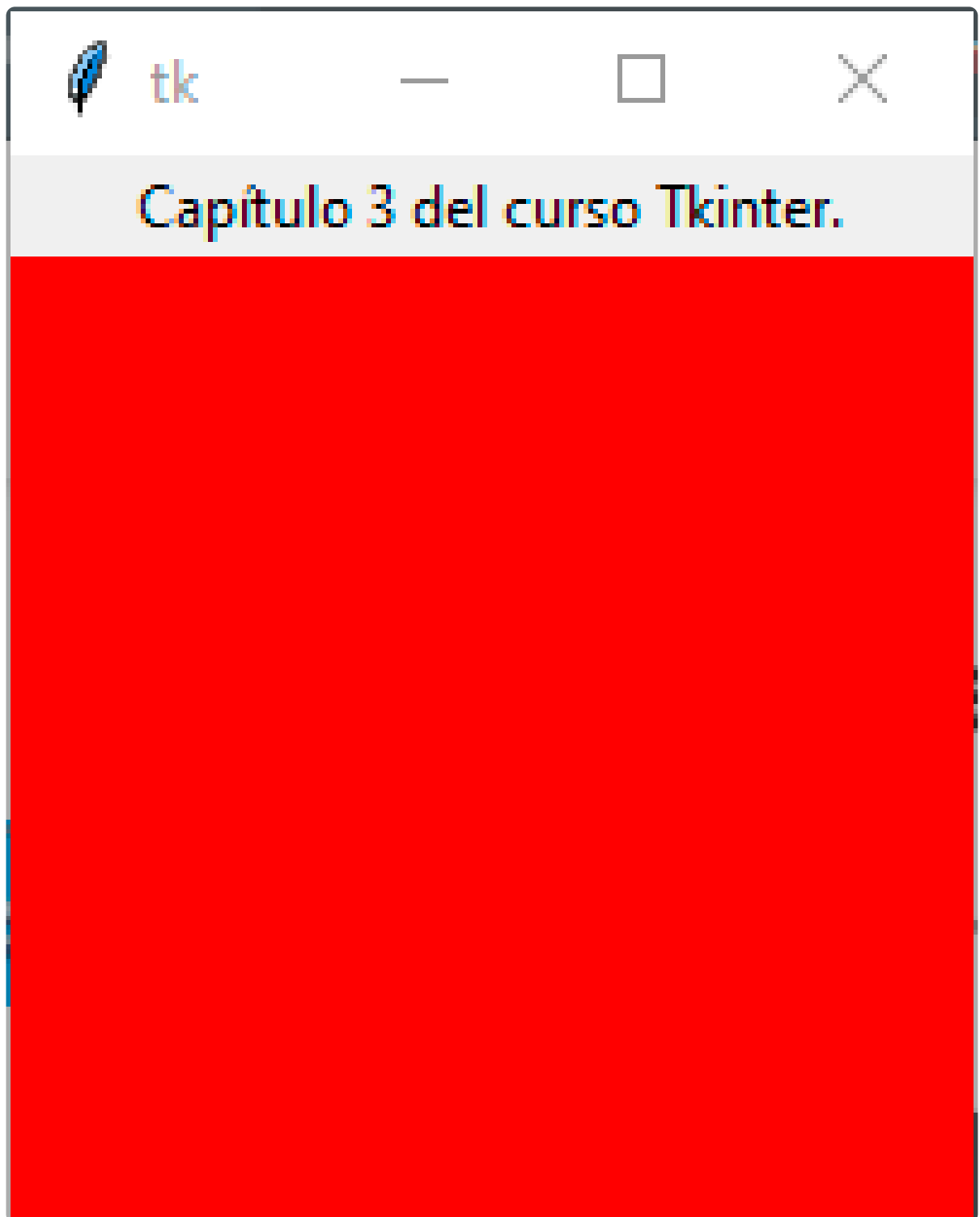
Vamos a sustituir los `pack()` por `grid()`, que en este caso, `grid()` también permite mostrar los widgets y además de eso, posicionarlos de una forma muy intuitiva.

Veamos algún ejemplo de colocación:

### Código Python

```
1 marco_principal.grid(row=1,column=0) LF
2 LF
3 texto.grid(row=0,column=0)
```

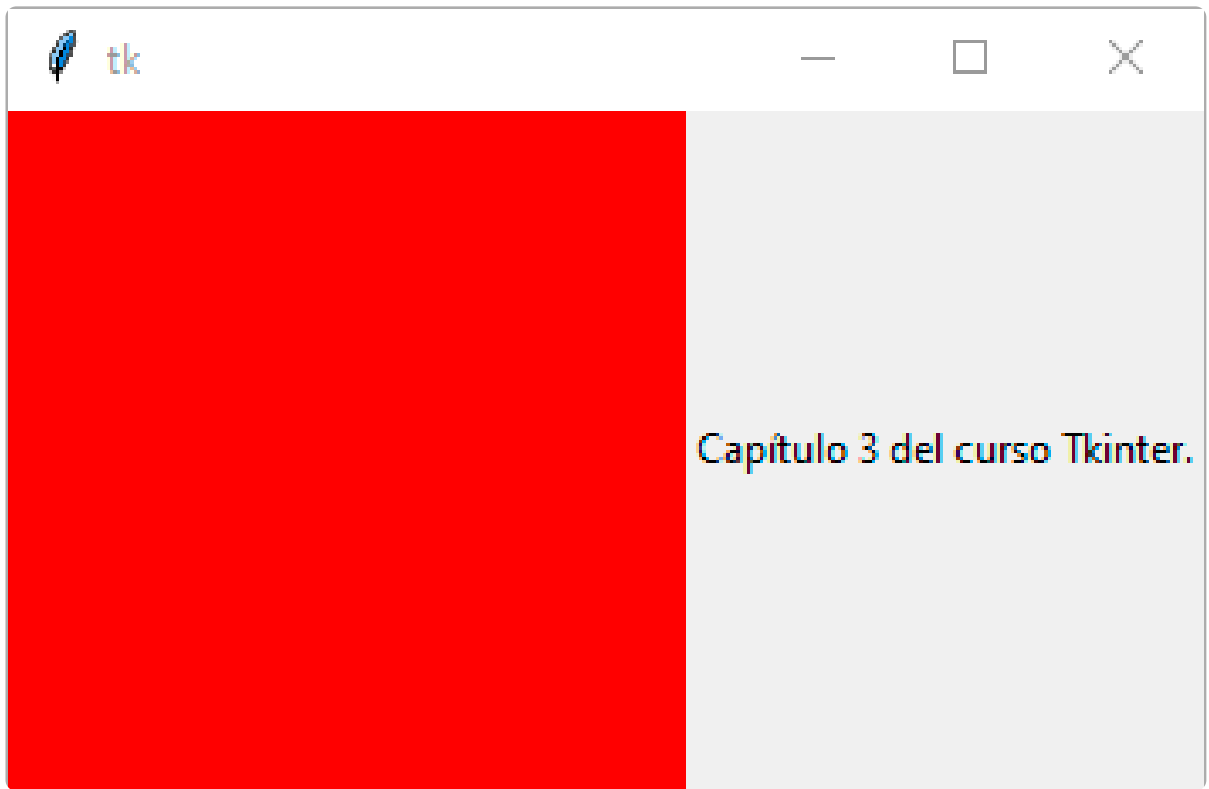
### Resultado



### Código Python

```
1 marco_principal.grid(row=0,column=0) LF
2 LF
3 texto.grid(row=0,column=1)
```

### Resultado



En el vídeo tienes algún ejemplo más y por supuesto, vamos a seguir utilizando `grid()` y `pack()` a lo largo del curso.

---

## Suscríbete a mi canal de YouTube para apoyarme

Si te ha gustado este curso y crees que el trabajo merece la pena, te agradeceré eternamente que te suscribas a mi canal de YouTube para apoyarme y que pueda seguir haciendo cursos gratuitos.

Además, si te encanta la programación, tienes un montón más de cursos gratuitos para ver.

No solo eso, podrás participar enviándome comentarios con tus sugerencias para temas específicos o cursos completos o incluso



las dudas que tengas y las intentaré ir resolviendo en los cursos que estén todavía abiertos.



Programación Fácil

YouTube

12 K

---

# Comentarios

Si te quedan dudas sobre el temario, sobre Python, o cualquier otra cosa relacionada o simplemente quieres agradecer, aquí tienes tu sitio para dejar tu granito de arena. Gracias por tus comentarios y por darle vida a este sitio web.

Nickname

Email (optional)

Reply...

**Comment**

[Comments powered by Cusdis](#)