

Soporte a la Gestión de Datos con Programación Visual

INTRODUCCIÓN A TKINTER

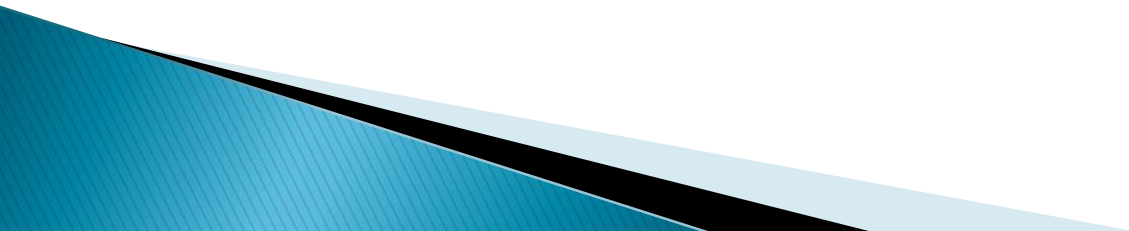
Tkinter

Es una interfaz para las librerías Tcl/Tk


Sería: Tk – inter

Portable a otras plataformas

Viene preinstalado con Python



Elementos de la Interfaz de usuario

- Ventana y lazo principal
 - Controles
 - Geometría. Disposición de los controles en la pantalla.
 - Manejo de eventos y llamadas a funciones para hacer un programa funcional
 - Opciones de estilo
- 

Generalidades

Import tkinter as tk (ó from tkinter import *)

3 puntos centrales:

•Qué se coloca en la pantalla? **Controles**

•Dónde se colocan? **Manejo de la Geometría**

•Cómo se comportan? **Eventos y callbacks**

Ventana principal

```
from tkinter import *  
root = Tk()  
root.mainloop()
```

- .Con la orden `root = Tk()` se crea un objeto ventana principal
- .Mainloop es el lazo infinito que permite la interacción del usuario. Captura eventos generados por el usuario, a los que responde el programa.
- . Cuando el usuario presiona el botón de cerrar, se sale del lazo infinito

Ventana principal

```
from tkinter import *  
root = Tk()  
root.title('Mi primer ventana')  
root.protocol("WM_DELETE_WINDOW", pide_cerrar)  
root.mainloop()
```

```
# Función que se invoca al pulsar en la X de  
# cierre  
def pide_cerrar():  
    root.destroy()    # Cierra la ventana creada  
    root.quit()       # Finaliza este programa
```

Controles

```
from tkinter import *  
root = Tk()  
label = Label(root, text="Soy una etiqueta")  
button = Button(root, text="Soy un botón")  
label.pack()  
button.pack()  
root.mainloop()
```

Sintaxis genérica para agregar controles:

`mi_ = ClaseDel (objeto contenedor, opciones)`

`Button()` crea un objeto de la clase botón

`pack()` es un método de geometría. Admite parámetro: centrado, der, izq
También están los métodos `grid()` y `place()` para manejar geometría.

No es imprescindible conservar referencias a los objetos visuales, a menos que se los vaya a usar después en el programa:

Por ej: `Label(root, "etiqueta").pack()`

Crea la etiqueta y la agrega a root

CONTROLES USUALES

Toplevel

Label

Button

Canvas

Checkbutton

Entry

Frame

LabelFrame

Listbox

Message

Menu

Menubutton

OptionMenu

PanedWindow

Radiobutton

Scale Scrollbar

Spinbox

Text

Bitmap Class

Image Class

PROPIEDADES DE LOS CONTROLES

Para conocer las propiedades disponibles en cada control se puede emplear el método `config()`:

```
form1 = Tk()
bot1 = Button(form1)
print( bot1.config() )      # Devuelve t-upla con la información:
```

Indice	Significado	Ejemplo
0	nombre	'font '
1	idem	'font'
2	clase	'Font'
3	Valor por defecto	
4	Valor actual	'TkDefaultFont'

Se le puede pasar el valor de una propiedad a conocer:

```
print(bot1.config('bg'))    #--- devuelve config de background
```

Controles básicos

```
from tkinter import *  
from tkinter import ttk, font
```

Label `Label(objRaiz, text = "", font = objFuente)`

Entry `Entry((objRaiz, txtvariable= nombreVar, width=nn, show="*")`

Button `Button(objRaiz, text = "...", command = nombreFunc)`

Separator `Separator(objRaiz, orient=HORIZONTAL)`

Frame `Frame(objRaiz, borderwidth=2, relief="raised", padding=(10,10))`

Para que un control reciba el foco: `control. focus_set()`

`objFuente = font.Font(weight='bold')`



Ventana3 - Botón

Ver ventana3_muestratexto.py

```
self.tinfo = Text(self.raiz, width=40, height=10)

self.binfo = ttk.Button(self.raiz, text='Info',
                        command=self.verinfo)

self.bsalir = ttk.Button(self.raiz, text='Salir',
                        command=self.raiz.destroy)

def verinfo(self):
    ...
    self.tinfo.insert("1.0", texto_info)
```

Entry

Es un objeto para que el usuario ingrese datos

```
from tkinter import *
```

```
raiz = Tk()
etiq = Label( raiz, text= "Edad:" )
nvar = StringVar()          # --- método de tkinter para generar objeto str
oentrada = Entry( raiz, textvariable = nvar )      # vincula con el objeto nvar
```

Para asignar un valor y obtener el valor del objeto nvar, hay que usar:
nvar.set() y nvar.get()

Variables de control. Opciones:

entero	= IntVar()	# Declara variable de tipo entera
flotante	= DoubleVar()	# Declara variable de tipo flotante
cadena	= StringVar()	# Declara variable de tipo cadena
booleano	= BooleanVar()	# Declara variable de tipo booleana

Ejercicio:

Hacer ventana que pida un nro y se muestre el mismo.

Marco (Frame)

Es un objeto contenedor.

Está contenido dentro de una ventana, y contiene a su vez otros controles

```
from tkinter import *  
from tkinter import ttk, font
```

```
raiz    =Tk()  
marco =Frame( raiz, relief = FLAT)  
etiq    = Label( marco, text = "hola" )    # va dentro del marco!
```

Propiedad relief= FLAT (llano), RAISED (elevado), SUNKEN (hundido), GROOVE (hendidura) y RIDGE (borde elevado).

Gestores de geometría

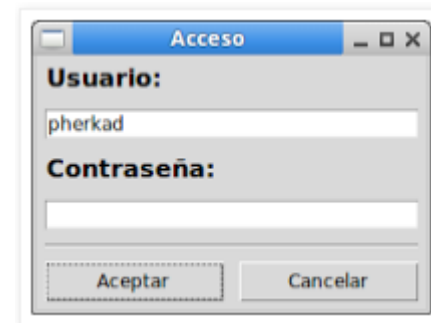
`pack()`

`grid()`

`place()`

A su vez, se pueden usar marcos, paneles (frames), que definen sectores dentro de una ventana.

Ejemplo ventana4_pack.py



Gestor de geometría pack()

Se crean primero los objetos controles y luego se los acomoda en la ventana ordenadamente con el comando pack()

Opciones:

side = TOP LEFT RIGHT BOTTOM

fill = X, Y, BOTH

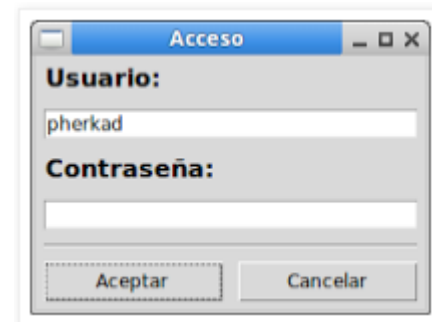
expand = TRUE, FALSE

padx, pady indican la cant de pixeles que separa

ipadx, ipady separación en pixels, interna al formulario

anchor "n", "ne", "e", "se", "s", "sw", "w", "nw", "center"

Ejemplo ventana4_pack.py



Ventana – OOP

```
from tkinter import *  
from tkinter import ttk
```

En este ejemplo se crea una clase Python para definir la interfaz de usuario de la aplicación.

```
class Aplicacion():  
    def __init__(self):  
        raiz = Tk()  
        raiz.geometry('300x200')  
        raiz.configure(bg = 'beige')  
        raiz.title('Aplicación')  
        ttk.Button(raiz, text='Salir',  
                    command=raiz.destroy).pack(side  
                    =BOTTOM)  
  
        raiz.mainloop()
```

---función main() para probar la clase

```
def main():  
    mi_app = Aplicacion()  
    return 0
```

si el módulo es ejecutado independiente

```
if __name__ == '__main__':  
    main()
```


Frame

```
import tkinter as tk

class CreaFrame(tk.Frame):
    def __init__(self, padre=None, titulo = ""):    #--- recibe un form contenedor 'padre'
        super().__init__(padre)
        self.padre = padre
        self.pack(side='top')
        self.crear_controles(titulo)

    def crear_controles(self, titulo):
        self.lblTitulo = tk.Label(self, text=titulo)
        self.lblTitulo.pack()
        self.btn_saludo = tk.Button(self)    #--- el padre es un frame
        self.btn_saludo["text"] = "Hola todos\n(Haz Click!)"
        self.btn_saludo["command"] = self.saluda
        self.btn_saludo.pack(side="top")

        self.btn_salir = tk.Button(self, text="QUIT", fg="red",
                                   command=self.padre.destroy)    #--- se cierra el form padre
        self.btn_salir.pack(side="bottom")

    def saluda(self):
        print("Estoy aquí ...!")

if __name__ == '__main__':
    form1 = tk.Tk()
    app = CreaFrame(padre=form1, titulo='frame 1')
    #app2 = CreaFrame(padre=form1, titulo='frame 2')    # se podrían crear varios
    form1.mainloop()
```

Gestor de geometría grid()

Ej.: Ventana no dimensionable

```
raiz = Tk()
```

```
raiz.resizable(0,0)
```

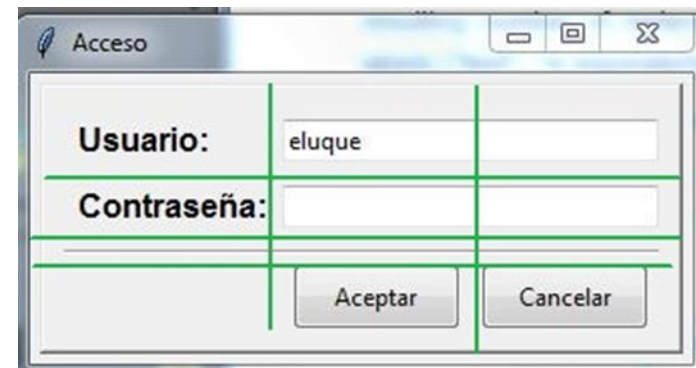
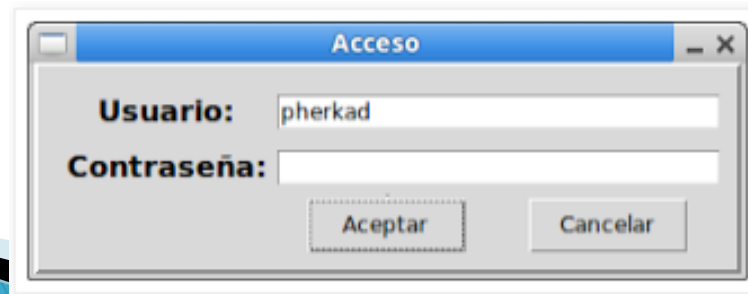
```
vp = Frame(raiz)
```

```
vp.grid(column=0, row=0, padx=(50,50), pady=(10,10))
```

```
etiqueta = Label(vp, text="Valor")
```

```
etiqueta.grid(column=2, row=2)
```

Ver Ejemplo ventana7



Introducir texto - Grid

```
import sys
from Tkinter import *

def hacer_click():
    try:
        _valor = int(entrada_texto.get())
        _valor = _valor * 5
        etiqueta.config(text=_valor)
    except ValueError:
        etiqueta.config(text="Introduce
un numero!")

raiz = Tk()
raiz.title("Mi segunda App Gráfica")
raiz.columnconfigure(0,weight=1)
raiz.rowconfigure(0,weight=1)
```

```
vp = Frame(raiz)
vp.grid(column=0, row=0,
        padx=(50,50), pady=(10,10))

vp.columnconfigure(0, weight=1)
vp.rowconfigure(0, weight=1)

etiqueta = Label(vp, text="Valor")
etiqueta.grid(column=2, row=2)

boton = Button(vp, text="OK!",
               command=hacer_click)

boton.grid(column=1, row=1)

valor = ""
entrada_texto = Entry(vp, width=10,
                      textvariable=valor)

entrada_texto.grid(column=2, row=1)

raiz.mainloop()
```

Gestor de geometría grid()

Ej.: Ventana dimensionable

```
raiz = Tk()
```

```
....
```

```
marco.grid(column=0, row=0, padx=5, pady=5, sticky= tk.N + tk.S+ tk.E+  
tk.W)
```

```
etiql.grid(column=0, row=0, sticky= tk.N+tk.S+tk.E+tk.W)
```

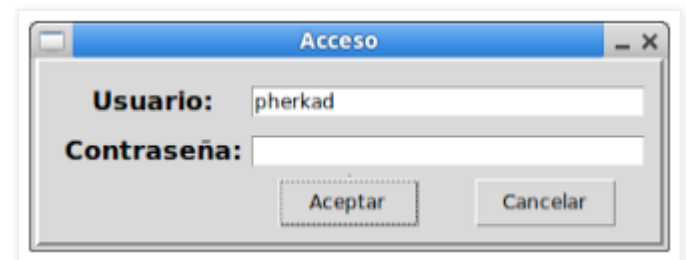
```
.....
```

```
raiz.columnconfigure(0, weight=1)      # formulario modificable vertical  
raiz.rowconfigure(0, weight=1)         # formulario modificable horizontal  
marco.columnconfigure(0, weight=1)    # marco modificable ...
```

Se usan la propiedad sticky en los controles visuales, y en los contenedores se asigna weight:

= 0 → no redimensionable = 2 → crece el doble que con = 1

Ejemplo ventana6_gridDimensionable.py

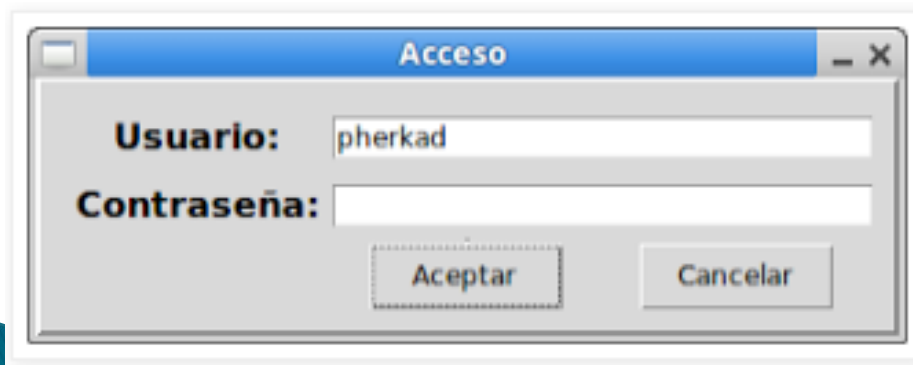


Gestor de geometría grid()

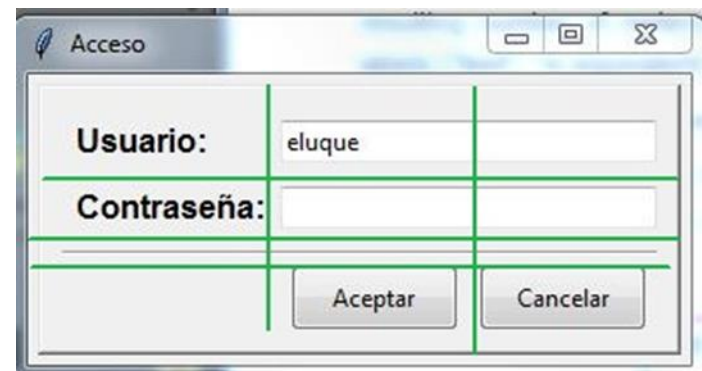
Otras propiedades usuales:

Rowspan y colspan Para que un objeto ocupe más de una celda (vertical u horizontal)

En el ejemplo se usó colspan. Indica la cantidad de columnas que abarca



A screenshot of a Java Swing window titled "Acceso". It contains two text input fields: "Usuario:" with the text "pherkad" and "Contraseña:". Below the fields are two buttons: "Aceptar" and "Cancelar".



A screenshot of a Java Swing window titled "Acceso" demonstrating a grid-based layout. The window is divided into a 3x2 grid by green lines. The "Usuario:" label and "eluque" text are in the top-left cell. The "Contraseña:" label and an empty text field are in the middle-left cell. The "Aceptar" and "Cancelar" buttons are in the bottom-right cell, demonstrating the use of colspan=2.

Gestión de eventos

```
raiz = Tk()
```

```
nvar = StringVar()
```

```
text1 = Entry( raiz, textvariable = nvar )
```

```
text1.bind('<button-1>', borrar_mensa) # --- al hacer click con el botón  
izq en el textbox, se disparará el método borrar_mensa
```

Otros ejemplos de acciones que se pueden capturar:

<double-button-1>, <buttonrelease-1>, <enter>, <leave>,

<focusin>, <focusout>, <return>, <shift-up>, <key-f10>,

<key-space>, <key-print>, <keypress-h>, etc.

Ejemplo ventana7_place.py

