

Tema 14

Interface Gràfica tkinter

Programació
amb Python



```
from tkinter import *
from tkinter import ttk

root = Tk()

button = ttk.Button()
button.pack()

def callback():
    print("button was pushed")
```

A hand holding a pencil is pointing towards a green computer monitor. On the monitor, there is a window with a button labeled "button". The code in the foreground corresponds to the code running on the screen.

Tkinter is a module in Python 2x while **tkinter** is a module in Python 3x

1. Componentes gráficos: Widgets

1.1. Contenedor base o raíz : TK()

1.2. Marco contenedor : Frame()

1.3. Etiquetas : Labels ()

1.4. Gestión de la Geometría

1.4.1. Posición fija: place ()

1.4.2. Zona flexible: pack()

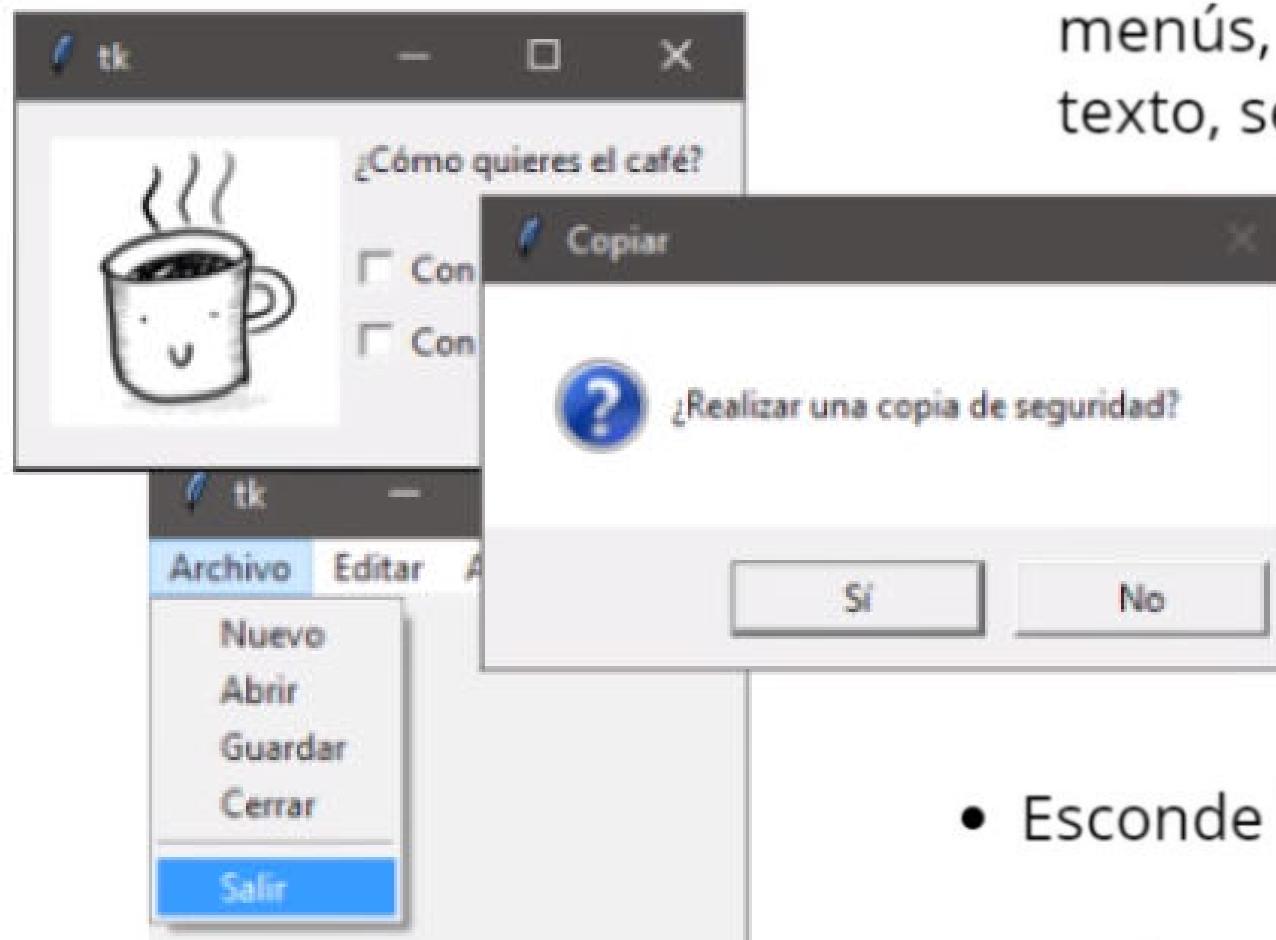
1.4.3. Formularios, rejilla : grid()

1.5. Textos largos : Text()

1.6. Botones y comandos de acción

1.7. Creando labels dinámicamente

Una interfaz gráfica es un medio visual a través del que los usuarios pueden interactuar y realizar tareas



- Se utilizan componentes gráficos como ventanas, menús, botones, campos de texto, seleccionables...

- Esconde lógica en segundo plano

- Python es un lenguaje multiplataforma
- Uno de sus retos más importantes fue ofrecer un sistema común para manejar la I.G.
- Los desarrolladores decidieron integrar una librería ya existente llamada Tk en lugar de crear la suya propia desde cero

The diagram illustrates the transition from Python 2.x to Python 3.x regarding the import statement for Tkinter. On the left, under 'Python 2.x', the code 'from Tkinter import *' is shown. On the right, under 'Python 3.x', the code 'from tkinter import *' is shown. A large computer monitor in the center displays a window with a red square and a cursor, with a red circle highlighting the word 'TK'. To the right of the monitor, the Python logo is enclosed in a green circle. Below the monitor, the text 'Interfaz gráfica con Tkinter' is displayed.

Python 2.x

```
from Tkinter import *
```

Python 3.x

```
from tkinter import *
```

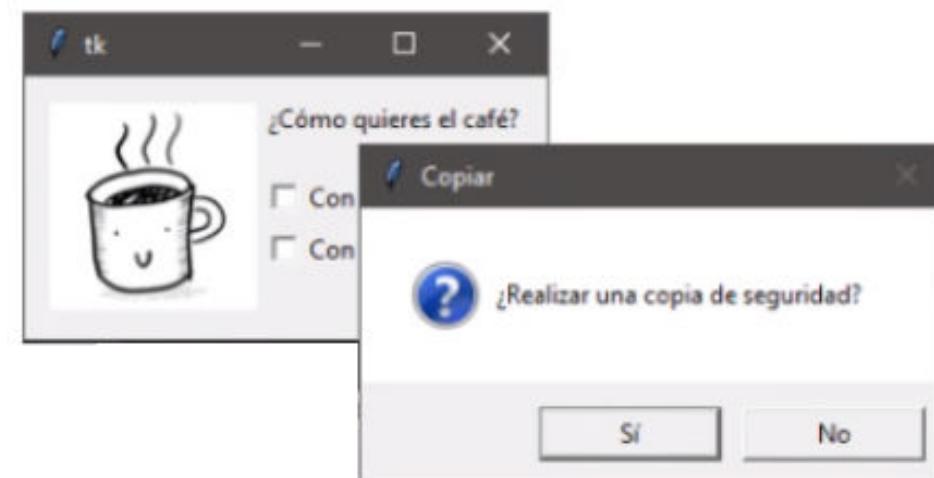
TK

Interfaz gráfica con Tkinter

1. Componentes gráficos: Widgets

El módulo Tkinter cuenta con una serie de **componentes gráficos** llamados Widgets, gracias a los cuales podemos diseñar nuestras interfaces.

Los widgets **deben seguir una jerarquía** a la hora de añadirse a la interfaz. Por ejemplo, un Marco (frame) forma parte del objeto raíz Tk. Y a su vez, un botón (button) puede formar parte de un contenedor como la raíz o un marco.





Los que veremos en esta introducción a Tkinter son:

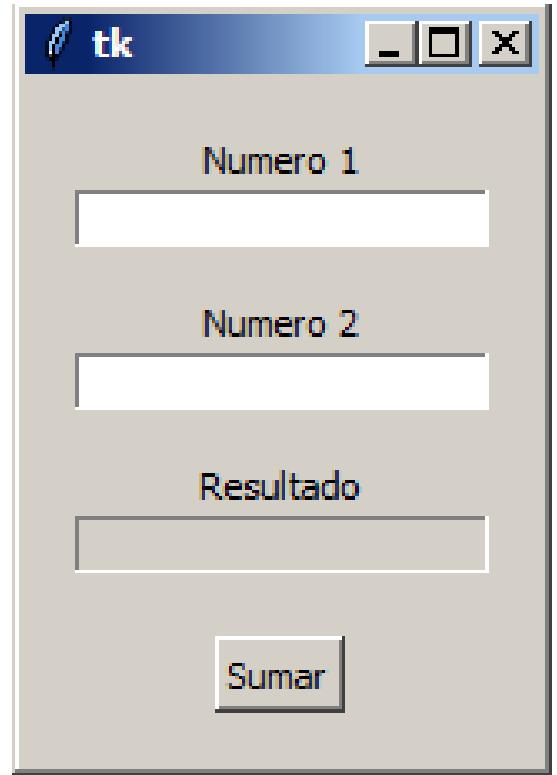
Tk: Contenedor base o raíz de todos los widgets que forman la interfaz. No tiene tamaño propio sino que se adapta a los widgets que contiene.

Frame: Marco contenedor de otros widgets. Puede tener tamaño propio y posicionarse en distintos lugares de otro contenedor (ya sea la raíz u otro marco).

Label: Etiqueta dónde podemos mostrar algún texto estático.

Entry: Campo de texto sencillo para escribir texto corto.
Nombres, apellidos, números..

Text: Campo de texto multilínea para escribir texto largo.
Descripciones, comentarios...



Button: Botón con un texto sobre el cual el usuario puede hacer clic.

Sigue los siguientes enlaces para estudiar en detalle cada uno de ellos

<https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-tk-raiz>

<https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-frame-marco/>

<https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-label-etiqueta-de-texto/>

<https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-entry-texto-corto/>

<https://docs.hektorprofe.net/python/interfaces-graficas-con-tkinter/widget-button-boton/>

Si has comprendido el contenido ...
salta a la diapositiva 28
y realiza la práctica P01.

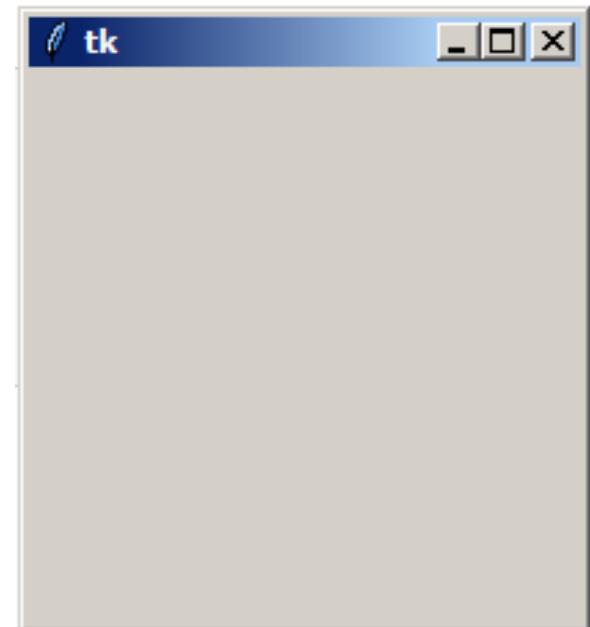
The image shows a window titled "Welcome to TutorialsPoint". Inside the window, there are four text input fields arranged vertically. The first field is labeled "First Name", the second "Last Name", the third "Email Id", and the fourth "Contact Number". Below these fields is a single "Submit" button. The window has a standard title bar with minimize, maximize, and close buttons.

1.1. Contenedor base o raiz : TK()

Crea una ventana

```
[*]: from tkinter import *
root = Tk()

#----Situar al final del programa
root.mainloop()
```



Le añade título, y crea una zona interior (frame)

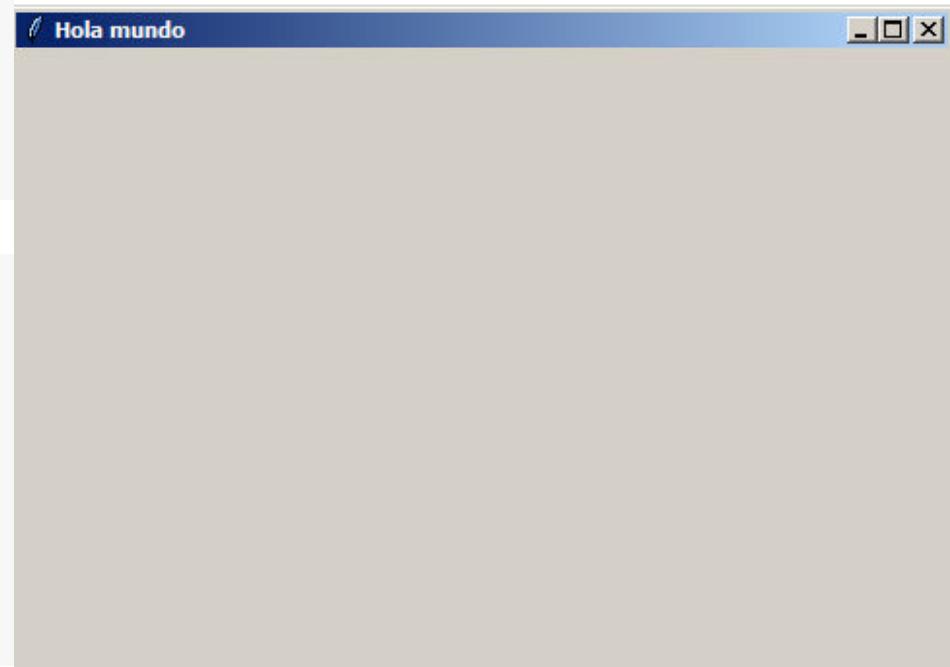
```
from tkinter import *
root = Tk()

root.title("Hola mundo")
root.resizable(1,1)

frame = Frame(root, width=480, height=320)
frame.pack()

frame.config(cursor="pirate")

#----Situar al final del programa
root.mainloop()
```



1.2. Marco contenedor : Frame()

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

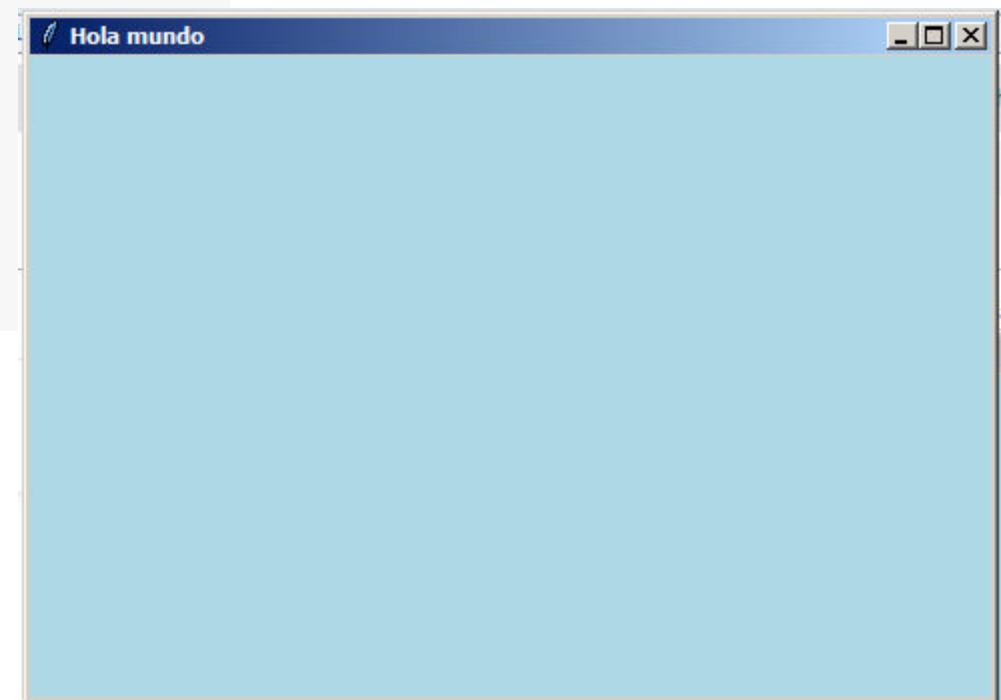
root.title("Hola mundo")

root.resizable(1,1)

frame = Frame(root, width=480, height=320)
frame.pack()

frame.config(cursor="pirate")
frame.config(bg="lightblue")

#----Situar al final del programa
root.mainloop()
```



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

root.title("holo")
root.resizable(1,0)

frame = Frame(root, width=480, height=320)
frame.pack()

frame.config(cursor="pirate")
frame.config(bg="#77340a")

root.config(cursor="arrow")
root.config(bg="blue")
root.config.bd=15

-----Situar al final del programa
root.mainloop()
```

Código para corta y pega



Con el cambio de color podemos distinguir la zona de la ventana y la zona del frame (azul claro)

1.3. Etiquetas : Labels ()

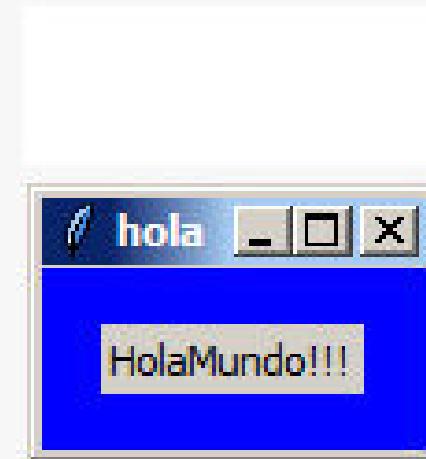
```
from tkinter import *
root = Tk()

root.title("hola")
root.resizable(1,0)

frame = Frame(root, width=480, height=320)
frame.pack()

frame.config(cursor="pirate")
frame.config(bg="lightblue")

label=Label(frame,text="HolaMundo!!!")
label.pack()
```



```
root.config(cursor="arrow")
root.config(bg="blue")
root.config.bd=15

-----Situar al final del programa
root.mainloop()
```

1.4. Gestión de la Geometría

Todos los widgets de Tkinter tienen acceso a métodos específicos de administración de geometría, que tienen el propósito de organizar widgets en todo el área del widget principal. Tkinter expone las siguientes clases de administrador de geometría: paquete, cuadricula y lugar.

- El método *place ()* : este administrador de geometría organiza los widgets colocándolos en una posición específica en el widget principal.
para mapas y juegos
- El método *pack ()* : este administrador de geometría organiza los widgets en zonas antes de colocarlos en el widget principal.
es flexible tipo web
- El método *grid ()* : este administrador de geometría organiza los widgets en una estructura similar a una tabla en el widget principal.
ideal para formularios

Estudiemos brevemente los métodos de gestión de la geometría

1.4.1. Posición fija: place ()

The diagram illustrates the placement of a button using the `place()` method. A coordinate system shows the origin at 0,0. A button labeled "Hola, mundo!" is positioned at (60, 40) with a width of 100px and a height of 30px. The button's top-left corner is at (60, 40). A red bracket indicates a vertical distance of 40px from the top edge to the button. A blue bracket indicates a horizontal distance of 100px from the left edge to the right edge of the button. The button has a 3px border.

`button.place(x=60, y=40, width=100, height=30)`

eje x

eje y

Break point & resolutions

- 320 (Phone – portrait)
- 480 (Phone – landscape + portrait)
- 600 (Small tablet – portrait)
- 768 (Large Tablet – portrait)
- 800 (Phone + Small tablet – landscape)
- 1024 (Large Tablet – landscape)
- 1280 (Desktop)
- 1440 (Wide screen desktop)

Width: 1440px

Width: 1280px

Width: 1024px

Width: 1024px

Width: 480px

Width: 320px

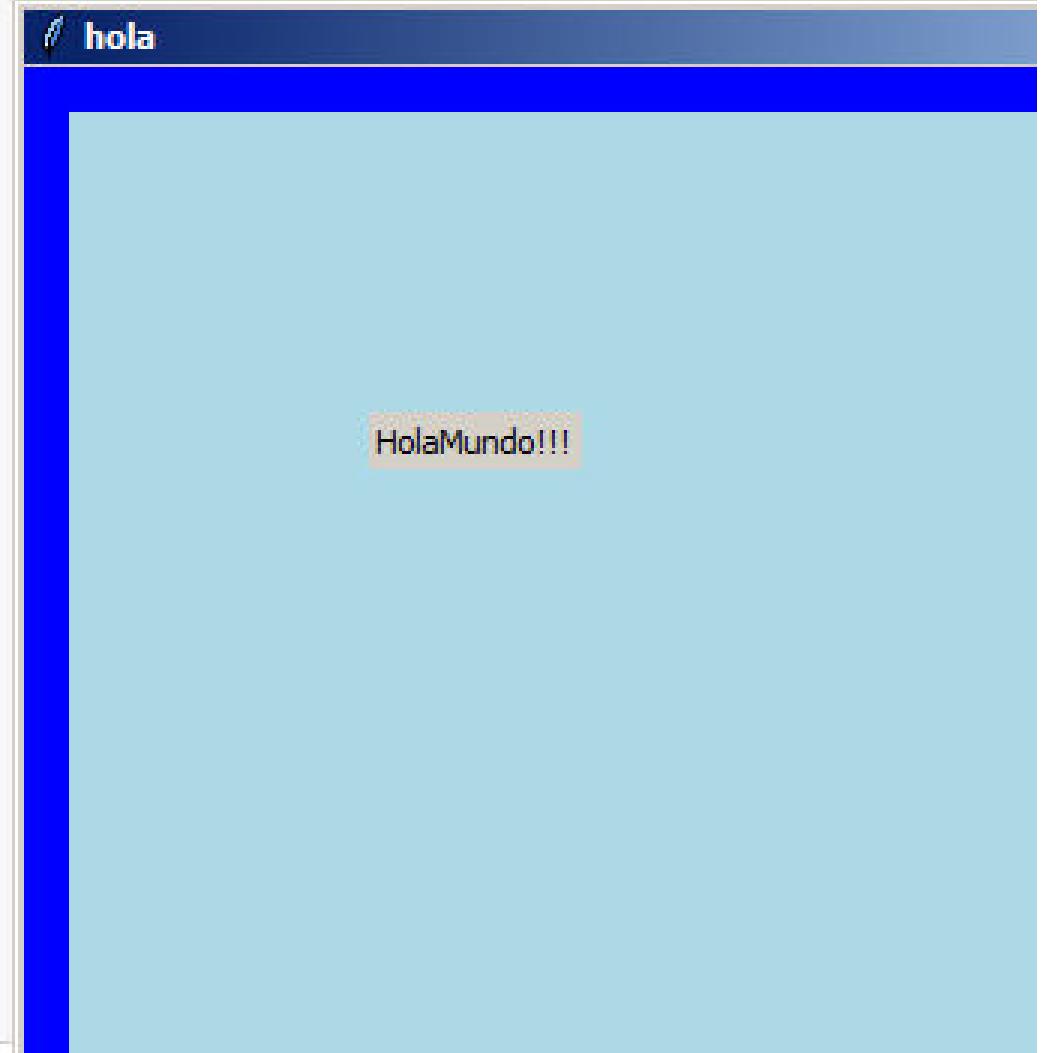
Breakpoints: horizontal widths we'll need to accommodate in our responsive design.

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

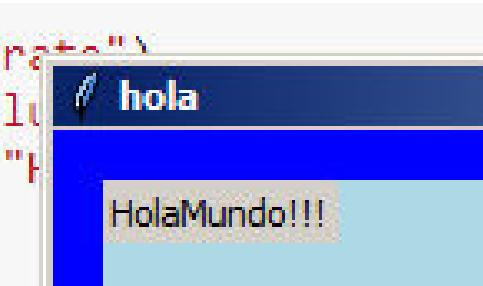
root.title("hola")
root.resizable(1,0)

frame = Frame(root, width=480, height=320)
frame.pack()

frame.config(cursor="pirate")
frame.config(bg="lightblue")
label=Label(frame,text="HolaMundo!!!")
label.place (x=100, y=100) 
```



```
frame.config(cursor="pirate")
frame.config(bg="lightblue")
label=Label(frame,text="HolaMundo!!!")
label.place (x=0, y=0)
```



```

from tkinter import *
root = Tk()

root.title("hola")
root.resizable(1,0)    ← eliminar

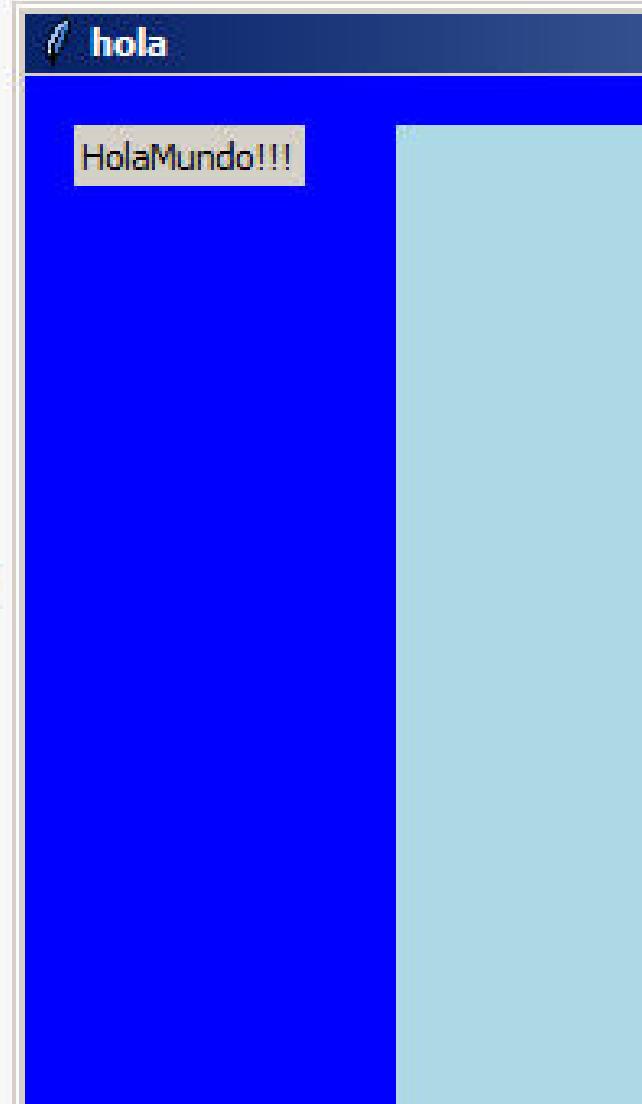
frame = Frame(root, width=480, height=320)
frame.pack()

frame.config(cursor="pirate")
frame.config(bg="lightblue")
label=Label(root, text="HolaMundo!!!!") ←
label.place (x=0, y=0)

root.config(cursor="arrow")
root.config(bg="blue")
root.config.bd=15

-----Situar al final del programa
root.mainloop()

```

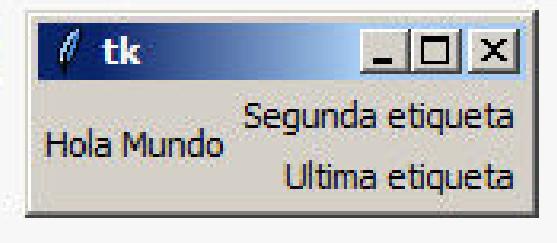


Prueba con distintas posiciones dentro y fuera del frame (marco) y redimensionando la ventana, hasta que hayas comprendido bien los conceptos y las medidas de las ventanas creadas.

1.4.2. Zona flexible: pack()

Side (Cara)

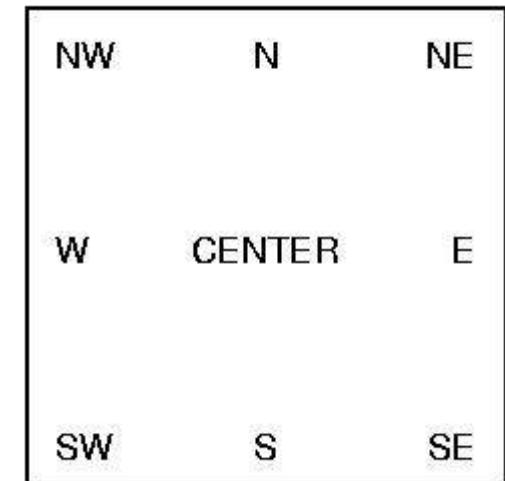
```
Label(root, text="Hola Mundo").pack(side="left")
Label(root, text="Segunda etiqueta").pack()
Label(root, text="Ultima etiqueta").pack(side="right")
#----Situar al final del programa
root.mainloop()
```



Anchor (Anclas)

Los anclajes se utilizan para definir dónde se coloca el texto en relación con un punto de referencia.

"HolaMundo!!! •_•"



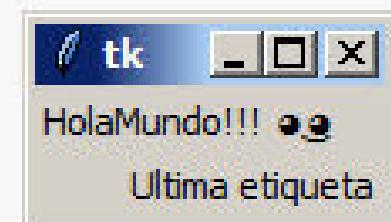
```
Label(root, text="HolaMundo!!! •_•").pack(anchor="nw")
```

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

Label(root,text="HolaMundo!!! •_•").pack(anchor="nw")
label=Label(root,text="Segunda etiqueta")

Label(root,text="Ultima etiqueta").pack(anchor="se")

-----Situar al final del programa
root.mainloop()
```

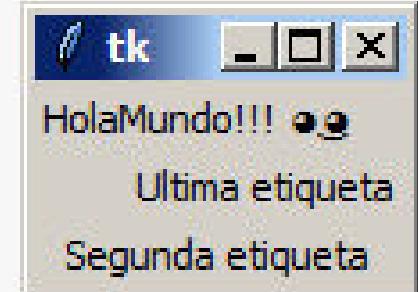


```
from tkinter import *
root = Tk()

Label(root,text="HolaMundo!!! •_•").pack(anchor="nw")
label=Label(root,text="Segunda etiqueta")

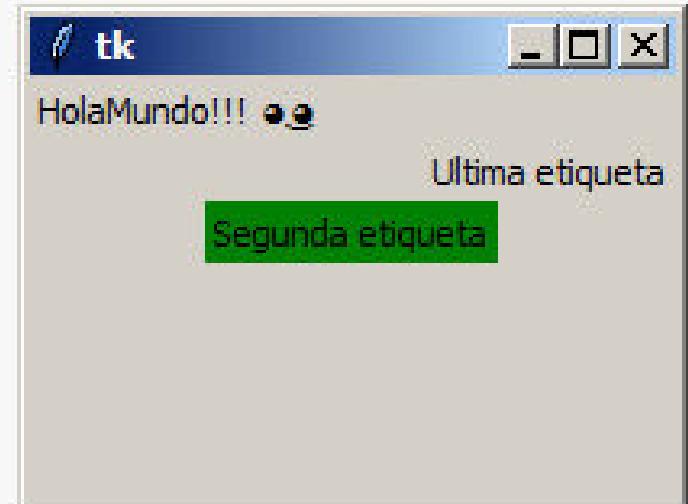
Label(root,text="Ultima etiqueta").pack(anchor="se")
label.pack(anchor="center")

-----Situar al final del programa
root.mainloop()
```



```
from tkinter import *  
  
root = Tk()  
  
Label(root, text="HolaMundo!!! •_•").pack(anchor="nw")  
label=Label(root, text="Segunda etiqueta")  
Label(root, text="Ultima etiqueta").pack(anchor="se")  
  
label.pack(anchor="center")  
label.config(bg="green")
```

-----Situar al final del programa
root.mainloop()



Código para corta y pega

```
#  
# Tkinter is a module in Python 2x while tkinter is a module in Python 3x.  
#  
from tkinter import *  
root = Tk()  
  
Label(root,text="HolaMundo!!! ☺_☺").pack(anchor="nw")  
label=Label(root,text="Segunda etiqueta")  
  
Label(root,text="Ultima etiqueta").pack(anchor="se")  
label.pack(anchor="center")  
  
#----Situar al final del programa  
root.mainloop()
```

```
from tkinter import *

root = Tk()

Label(root, text="HolaMundo!!! •_•").pack(anchor="nw")
label=Label(root, text="Segunda etiqueta")
Label(root, text="Ultima etiqueta").pack(anchor="se")

label.pack(anchor="center")
label.config(bg="green", fg="blue", font=("Verdana", 24))

#----Situar al final del programa
root.mainloop()
```



```
from tkinter import *

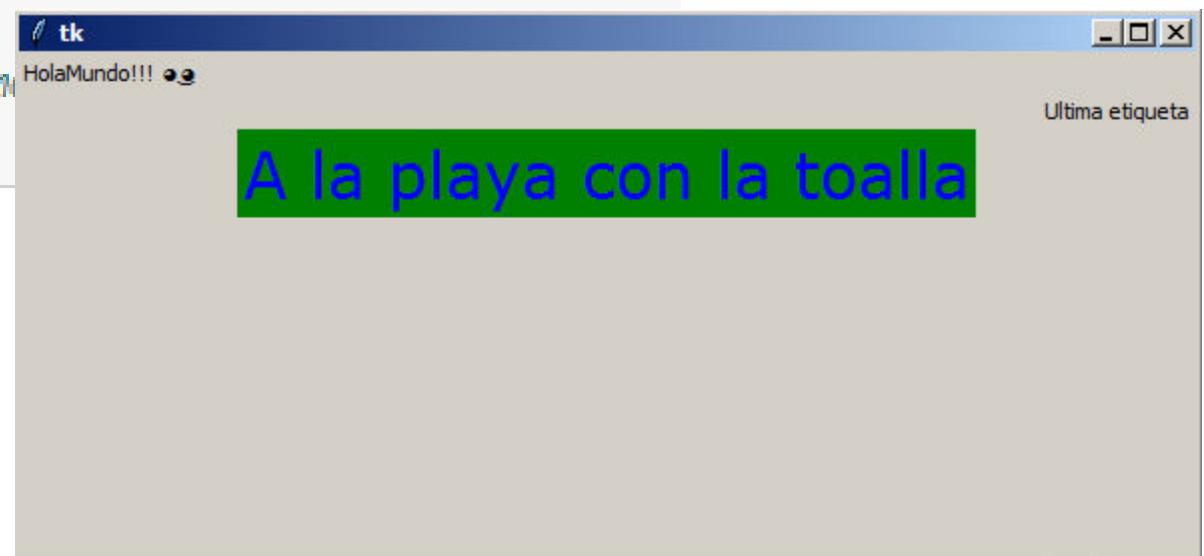
root = Tk()

Label(root, text="HolaMundo!!! ☺_☺").pack(anchor="nw")
label=Label(root, text="Segunda etiqueta")
Label(root, text="Ultima etiqueta").pack(anchor="se")

label.pack(anchor="center")
label.config(bg="green", fg="blue", font=("Verdana", 24))

texto=StringVar()
texto.set ("A la playa con la toalla")
label.config(textvariable=texto)
```

#----Situar al final del programa
root.mainloop()



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

```
Label(text="Nombre").pack(side="left")
Entry(root).pack(side="right")
```

```
# Situar al final del programa
root.mainloop()
```



Cajas de Texto : Entry()

Debajo del nombre añadimos lo necesario para pedir los apellidos

```
#----- Apellidos
Label(text="Apellidos").pack(side="left")
Entry(root).pack(side="right")
```



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

#----- Campo Nombre
frame1 = Frame(root)
frame1.pack()
Label(frame1, text="Nombre").pack(side="left")
Entry(frame1).pack(side="right")

#----- Apellidos
frame2 = Frame(root)
frame2.pack()
Label(frame2, text="Apellidos").pack(side="left")
Entry(frame2).pack(side="right")

#---Situar al final del programa
root.mainloop()
```



Con este sistema las cajas quedan desalineadas

1.4.3. Formularios, rejilla : grid()

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

Label(text="Nombre").grid(row=0, column=0)
Entry(root).grid(row=0, column=1)

Label(text="Apellidos").grid(row=1, column=0)
Entry(root).grid(row=1, column=1)

root.mainloop()
```

Copia-pegar

En lugar de usar pack,
organizamos con grid



- sticky west = alinea a la izquierda los textos
- padx, pady = crea un margen de 5 pixels

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

```
Label(text="Nombre completo").grid(row=0, column=0, sticky="w", padx=5, pady=5)
Entry(root).grid(row=0, column=1, padx=5, pady=5)
```

```
Label(text="Apellidos").grid(row=1, column=0, sticky="w", padx=5, pady=5)
Entry(root).grid(row=1, column=1, padx=5, pady=5)
```

```
root.mainloop()
```

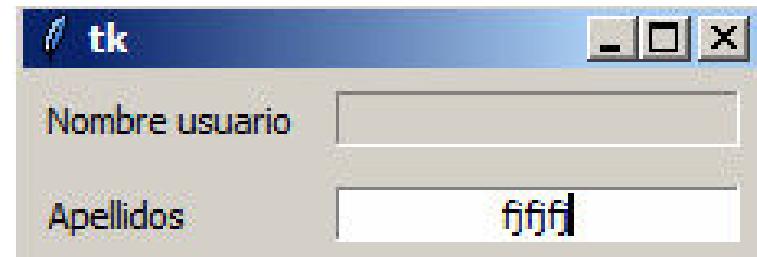
Si queremos configurar más parámetros, tendremos que asignar la función Entry a un objeto y pasar a llamar a los métodos por separado en lugar de ir encadenando instrucciones.

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

Label(text="Nombre usuario").grid(row=0, column=0, sticky="w", padx=5, pady=5)
e1 = Entry(root)
e1.grid(row=0, column=1, padx=5, pady=5)
e1.config(justify="right", state="disabled")

Label(text="Apellidos").grid(row=1, column=0, sticky="w", padx=5, pady=5)
e2 = Entry(root)
e2.grid(row=1, column=1, padx=5, pady=5)
e2.config(justify="center")

root.mainloop()
```



Usamos el método config para indicar el alineamiento del texto en la caja y si la caja está habilitada o desactivada.

Practica P01-Formulario Contacto

Modifica el programa que se incluye en la siguiente diapositiva, para que incluya los siguientes cambios.



Debe pedir : Nombre, Apellidos 1, Apellido 2, correo electrónico, teléfono, distrito postal

En lugar del botón Submit, debe incluir un botón de Aceptar y Otro de cancelar.

```
# -*- coding: utf-8 -*-
"""
Created on Thu Sep 10 19:00:21 2020

@author: Usuario
"""

from tkinter import *
from tkinter import ttk

window = Tk()
window.title("Welcome to TutorialsPoint")
window.geometry('400x200')
window.minsize(width=400, height=200)

frame = Frame(window)
Label(frame ,text = "First Name").grid(row = 0,column = 0)
Label(frame ,text = "Last Name").grid(row = 1,column = 0)
Label(frame ,text = "Email Id").grid(row = 2,column = 0)
Label(frame ,text = "Contact Number").grid(row = 3,column = 0)

Entry(frame).grid(row = 0,column = 1)
Entry(frame).grid(row = 1,column = 1)
Entry(frame).grid(row = 2,column = 1)
Entry(frame).grid(row = 3,column = 1)

btn = ttk.Button(frame ,text="Submit").grid(row=4,column=1)
frame.pack(anchor=NW, expand=1)

window.mainloop()
```

1.5. Textos largos : Text()

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

texto = Text(root)
texto.pack()

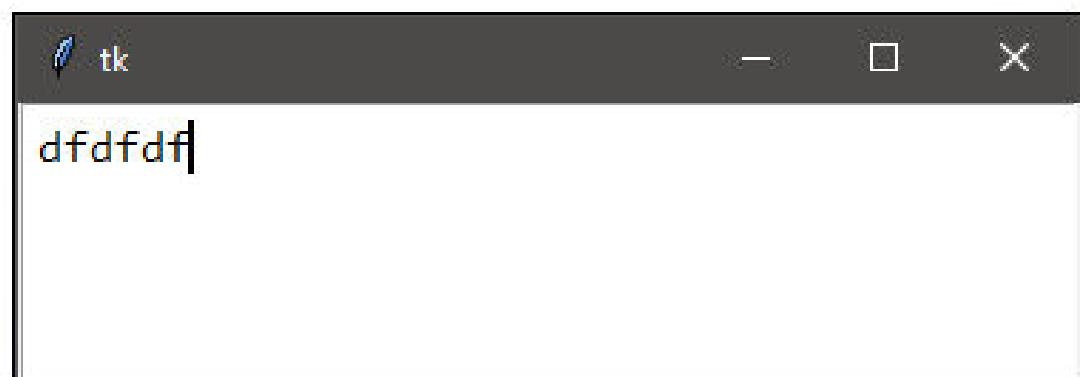
root.mainloop()
```

podríamos establecer una altura y anchura en caracteres, no píxeles:

```
texto.config(width=30, height=10)
```

También acepta propiedades visuales para los colores o la fuente:

```
texto.config(font=("Consolas", 12), selectbackground="red", padx=5, pady=5)
```



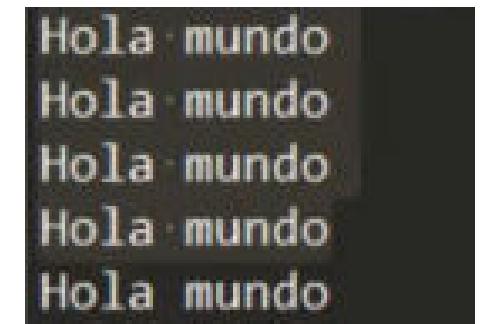
1.6. Botones y comandos de acción

```
# Podemos crearlos y empacarlos en una línea  
Button(root, text="Clícame").pack()
```



Al definir un botón podemos asociar la acción o “comando” a realizar en el momento que sea pulsado.

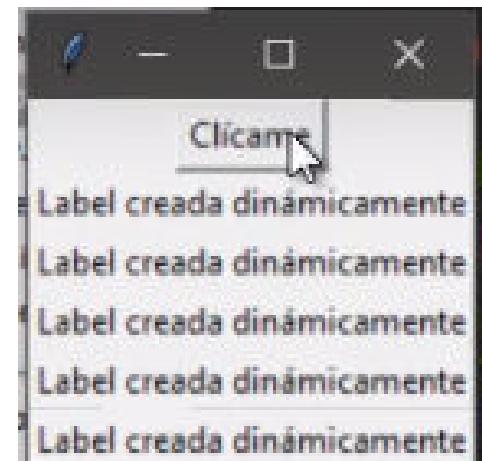
```
from tkinter import *  
  
# Definimos una función a ejecutar al clic el botón  
def hola():  
    print("Hola mundo!")  
  
root = Tk()  
  
# Enlazamos la función a la acción del botón  
Button(root, text="Clícame", command=hola).pack()  
  
root.mainloop()
```



1.7.Creando labels dinámicamente

```
def crear_label():
    Label(root, text="Label creada dinámicamente").pack()
```

```
Button(root, text="Clicame", command=crear_label).pack()
```



1.8. Como acceder a las variables del formulario

Cuando a una clase Entry() le asociamos una variable para poder leer o cambiar su valor debemos usar :
get() y set().

```
from tkinter import *

def validar():
    if num1.get() =="" :
        num1.set("rellene el campo")
#----- Proceso principal
root = Tk()
num1= StringVar()

Entry(root, justify="center", textvariable=num1).pack()
btnRes = Button(root, text="Sumar", command=validar).pack()
root.mainloop()
```

Practica P02

Ejercicio 1. Crea un formulario en que pidas dos números. Queremos saber si el primero es múltiplo o divisor del segundo.

Crea los botones :

[Múltiplo ?] que muestre un mensaje con Si o No dependiendo si el primer número es múltiplo del segundo.

[Divisor ?] que muestre un mensaje con Si o No dependiendo si el primer número es divisor del segundo.

The screenshot shows a window titled "tk" with a blue header bar. It contains two input fields: "Primer número : 78.9" and "Segundo número : 788.9". Below these are two buttons: "Multi" and "Divisor". At the bottom left is a label "Resultado" and at the bottom right is a field containing the word "No".

Ejercicio 2. ventas.py

Queremos generar datos de ventas por tienda, año y mes.

Pide la tienda, el año y el volumen máximo de Euros.

Crea el botón [Generar], al ser pulsado generará 12 registros csv con las ventas de la tienda para ese año, y cada mes. El volumen de ventas se obtendrá tomando un número aleatorio entre 0 y Máxim.

The window has three input fields: 'Tenda' with value 'BCN', 'Any' with value '2019', and 'Màxim' with value '700'. Below these is a text area labeled 'dados' containing the following CSV data:

```
BCN;2019;1;217;
BCN;2019;2;399;
BCN;2019;3;688;
BCN;2019;4;508;
BCN;2019;5;179;
BCN;2019;6;74;
BCN;2019;7;571;
BCN;2019;8;625;
BCN;2019;9;374;
BCN;2019;10;327;
BCN;2019;11;392;
BCN;2019;12;320;
```

A 'Generar' button is at the bottom.

The window has three input fields: 'Tenda' with value 'BCN', 'Any' with value '2019', and 'Màxim' with value '700'. Below these is a text area labeled 'dados' containing the following CSV data:

```
BCN;2019;1;217;
BCN;2019;2;399;
BCN;2019;3;688;
BCN;2019;4;508;
BCN;2019;5;179;
BCN;2019;6;74;
BCN;2019;7;571;
BCN;2019;8;625;
BCN;2019;9;374;
BCN;2019;10;327;
BCN;2019;11;392;
BCN;2019;12;320;
```

A 'Generar' button is at the bottom.

Graba los datos en un fichero csv y muestra el gráfico de barras del año generado, puedes usar las funciones de la página siguiente o crear funciones nuevas.

Ten en cuenta que la gráfica sale en la consola.

```
from io import open
import pandas as pd
import matplotlib.pyplot as plt

#----- Funciones
def graba_fichero(datos) :
    try :
        fichero = open('ventas.dat', 'w')
        fichero.write('tenda;anyy;mes;vendes;\n'+datos)
        fichero.close()
    except:
        r.set("Problemas al generar el archivo")

def vergrafo() :
    try :
        df = pd.read_csv("ventas.dat", sep=";")
        plt.plot(df['mes'], df['vendes'])
        plt.show()
    except:
        r.set("Problemas al generar el gráfico")
```

Practica P03-Enviando Formulario Contacto

Usando la práctica P01, modifica el programa para que:

- El botón Cancelar : deje las casillas en blanco
- El botón Aceptar :
 - deje las casillas en blanco,
 - muestre una etiqueta dinámica en la ventana indicando que se ha enviado.
 - Haga un print : “send”, cognom1, cognom2, correu

Nom	marta
Cognom 1	Lopez
Cognom 2	Sanchis
Correu	marta@gmail.com
Telèfon	625735523
Codi Postal	08003

Cancelar **Enviar**

Practica P04-Formulario Centres

Crea un programa que se llame centres.py con un formulario Tkinter que pida los siguientes datos

- codi centre (de tipo carácter máximo 12 posiciones)
- nom centre (de tipo carácter máximo 80 posiciones)
- adres 1 (tipo carácter, máximo 80)
- adres 2 (tipo carácter, máximo 80)
- codi postal (de tipo carácter máximo 5 posiciones)
- correu
- telèfon

Que es tkinter.ttk ?

tk	ttk
<i>Button</i>	Button
Canvas	Checkbutton
<i>Checkbutton</i>	Combobox
<i>Entry</i>	Entry
Frame	Frame
<i>Label</i>	Label
LabelFrame	LabeledScale
Listbox	Labelframe
Menu	Menubutton
Menubutton	Notebook
Message	OptionMenu
OptionsMenu	Panedwindow
PanedWindow	Progressbar
<i>Radiobutton</i>	Radiobutton
Scale	Scale
Scrollbar	Scrollbar
Spinbox	Separator
Text	Sizegrip
	Treeview

```
from tkinter import *
from tkinter import ttk
root = Tk()
ttk.Button(root, text="Hello World").grid()
root.mainloop()
```