

# Clases para ventanas de interacción

Juan Álvarez Rubio

En esta sección se presenta un módulo con clases que se utilizan para agregar a los programas interfaces de usuario intuitivas y amigables. Utilizando clases predefinidas los programas se comunican con los usuarios (personas) a través de ventanas que se muestran en la pantalla del computador y que tienen elementos de interacción como textos y botones.

## Ventana con un texto

En primer lugar, y para presentar los conceptos y elementos, se presenta un programa que abre una ventana en la pantalla con el texto “Hola” como contenido:



El programa correspondiente es el siguiente:

```
from tkinter import *
ventana = Tk()
saludo = Label(ventana, text="Hola")
saludo.pack()
ventana.mainloop()
```

La instrucción `from tkinter import *` deja disponibles todas las clases para desarrollar interfaces con el usuario que están disponibles en el módulo predefinido `tkinter` (*toolkit interface*). En particular, este programa utiliza las clases `Tk` y `Label`.

La instrucción `ventana=Tk()` crea un objeto de la clase `Tk` y guarda la referencia con el nombre `ventana`. El constructor de la clase `Tk` crea un objeto en la memoria que representa una ventana y devuelve la referencia a ese objeto.

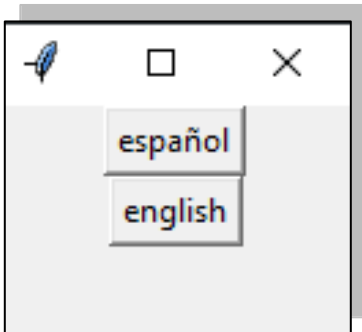
La instrucción `saludo=Label(ventana, text="Hola")` crea un objeto de la clase `Label` y su referencia se guarda con el nombre `saludo`. El constructor de la clase `Label` recibe dos parámetros: la referencia a la ventana y el texto con el string que se pone en la ventana. Un programa puede definir más de una ventana por lo que se necesita indicar en qué ventana se pondrá el texto.

La instrucción `saludo.pack()` pone o acomoda el objeto de la clase `Label` en el primer lugar de la ventana. En este caso la ventana tiene una sola componente pero igual es necesario utilizar el método `pack` de la clase `Label`.

La instrucción `ventana.mainloop()` muestra la ventana en la pantalla con las componentes (*widgets*) que fueron agregadas con los métodos `pack`. La ventana se muestra con la barra horizontal en la parte superior con las opciones de minimizar y de cerrar. La ventana permanecerá en la pantalla hasta que se cierre dando un *click* sobre la opción X.

## Ventana con botones

Una ventana, además de componentes con texto, puede contener botones, es decir componentes que se seleccionan y *cliquean* con el mouse. Por ejemplo, la siguiente ventana permite saludar eligiendo el idioma, de manera que al dar un *click* en el botón identificado con “español” entonces aparece el texto “hola”. En cambio, si se selecciona el botón “english” se muestra el texto “hello”.



El programa es el siguiente:

```
from tkinter import *
#metodos que se ejecutan al clicar botones
def saludar(): saludo.config(text="hola")
def greet(): saludo.config(text="hello")
#crear ventana
v = Tk()
#definir componentes
b1 = Button(v, text="español", command=saludar)
b2 = Button(v, text="english", command=greet)
saludo = Label(v)
#diagramar ventana
b1.pack()      #botón b1 arriba
b2.pack()      #botón b2 debajo de anterior
saludo.pack()  #label saludo abajo
#mostrar ventana (hasta que se cierre)
v.mainloop()
```

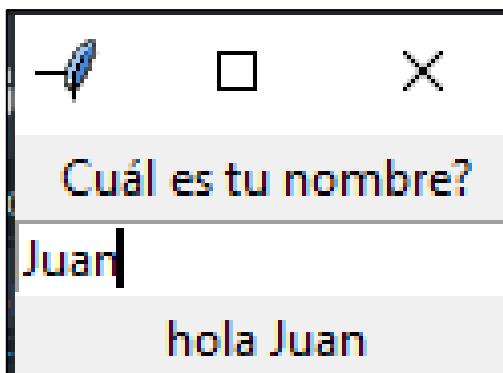
La ventana tiene tres componentes (*widgets*): dos objetos de la clase `Button` y un objeto de la clase `Label`. El orden en que aparecen en la ventana está determinado por el orden de las instrucciones `componente.pack()`.

El constructor de la clase Button necesita tres parámetros: la ventana que lo contiene, el texto que lo identifica, y el método que se invoca cuando se dé un *click* en el botón. En este programa los métodos asociados a los botones pondrán un texto en la componente saludo utilizando el método config de la clase Label.

La ventana se muestra en la pantalla y permanece abierta hasta que se cierre con el botón X de la barra de título. Cada vez que se seleccione un botón se muestra el saludo en el idioma elegido y el programa queda esperando un nuevo *click* en uno de los botones. Por lo tanto, el programa “saluda” todas las veces que se dé un *click* a un botón, sin necesidad de repetir instrucciones.

## Ventana con una componente para ingresar datos

Una ventana puede contener también una componente para ingresar un valor. Por ejemplo, un programa que solicite el nombre de una persona para saludarlo muestra la siguiente ventana:



Para ingresar un valor se necesita una componente de la clase Entry. Después de definir un objeto de la clase Entry se necesita invocar el método bind que requiere dos parámetros: el carácter que se utiliza para indicar el ingreso y el método que se invoca (que requiere un parámetro pero que en este caso no se utiliza).

```
from tkinter import *
#metodo que se ejecuta al ingresar nombre
def saludar(x):
    nombre=respuesta.get()
    saludo.config(text="hola "+nombre)
#definir ventana y componentes
v = Tk()
pregunta = Label(v,text="Cuál es tu nombre?")
respuesta = Entry(v)
respuesta.bind("<Return>",saludar)
saludo=Label(v)
#diagramar y mostrar ventana
pregunta.pack()
respuesta.pack()
```

```
saludo.pack()  
v.mainloop()
```

## Ventana con botones e ingreso de datos

Una ventana puede tener componentes de distintas clases. Por ejemplo, una ventana que implemente un saludo bilingüe tiene las siguientes 5 componentes:



Los métodos que se invocan al seleccionar el idioma tienen que escribir la pregunta en el idioma correspondiente e informar el método que se invocará al ingresar el nombre.

```
from tkinter import *  
  
#metodos para saludar en español  
def saludar(): #para botón  
    pregunta.config(text="Cuál es tu nombre?")  
    respuesta.bind("<Return>",saludarEnEspanol)  
def saludarEnEspanol(x): #para Entry  
    saludo.config(text="hola "+respuesta.get())  
  
#metodos para saludar en inglés  
def greet():  
    pregunta.config(text="What is your name?")  
    respuesta.bind("<Return>",greetings)  
def greetings(x):  
    saludo.config(text="hello "+respuesta.get())  
  
#definir ventana y componentes  
v=Tk()  
b1=Button(v,text="español",command=saludar)  
b2=Button(v,text="english",command=greet)  
pregunta = Label(v)
```

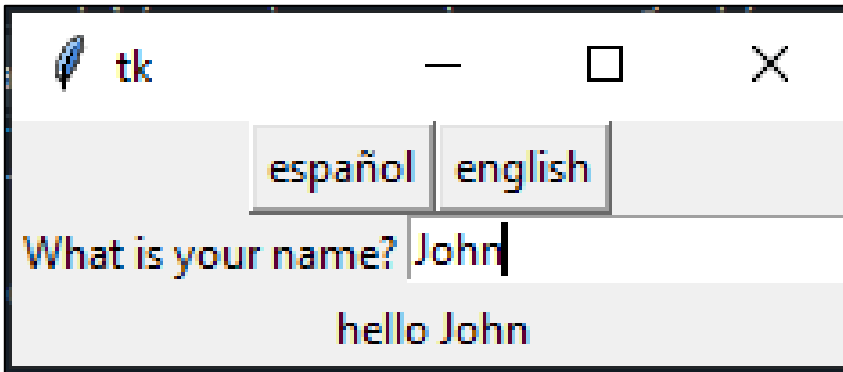
```
respuesta = Entry(v)
saludo=Label(v)
```

#### **#diagramar y mostrar ventana**

```
b1.pack()
b2.pack()
pregunta.pack()
respuesta.pack()
saludo.pack()
v.mainloop()
```

## Ventana con marcos para agrupar componentes

Cuando las ventanas tienen muchas componentes, conviene agruparlas de manera de facilitar la comunicación con el usuario. Por ejemplo, en el programa bilingüe para saludar se pueden agrupar los botones y la pregunta y respuesta del nombre:



Para agrupar componentes se utiliza un marco, es decir, un objeto de la clase Frame. Las componentes que se agrupan deben indicar que pertenecen a un marco. Por otra parte, al diagramar un marco se puede indicar que las componentes se acomoden de izquierda a derecha.

#### **#definir ventana y componentes**

```
v = Tk()
saludo=Label(v)
```

#### **#definir y diagramar marco para botones**

```
marco1=Frame(v)
b1=Button(marco1,text="español",command=saludar)
b2=Button(marco1,text="english",command=greet)
b1.pack(side=LEFT) #a la izquierda
b2.pack()          #a continuación
```

#### **#definir y diagramar marco para pregunta y respuesta**

```
marco2=Frame(v)
pregunta = Label(marco2)
respuesta = Entry(marco2)
```

```
pregunta.pack(side=LEFT)
respuesta.pack()
```

#### **#diagramar y mostrar ventana**

```
marco1.pack()
marco2.pack()
saludo.pack()
v.mainloop()
```

El programa debe incluir los mismos métodos del programa anterior para los botones y el objeto de la clase Entry.