

▼ Caso de Estudio IV: Diccionario

Para este caso de estudios debe suponer que ya tiene una **clase Diccionario**, la cual corresponde a una secuencia ordenada de **palabras** (identificadores que no se repiten) con **significados** (información asociada).

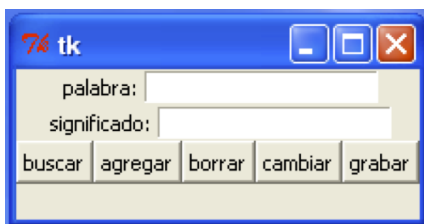
Esta clase cuenta con las siguientes operaciones:

Operación	Significado
D=Diccionario("x.txt")	Crea diccionario con palabras y significados en líneas de archivo
D.buscar(x)	devolver significado de palabra x None si no existe
D.agregar(x,y)	agregar palabra x con significado y True si ok
D.borrar(x)	borrar palabra x True si ok
D.cambiar(x,y)	cambiar significado de palabra x por y True si ok
D.grabar("x.txt")	Graba diccionario en archivo

Un ejemplo de uso de esta clase podría ser para darle una mejor implementación a la aplicación AI2E (*Abreviación de Internet a Español*) que vimos en una clase anterior. O podría ser para simplemente implementar un simple diccionario "Español-Inglés" por ejemplo.

Problema 1

Escriba un programa que maneje un diccionario a través de la siguiente ventana



¿Qué funcionalidad queremos?

La ventana tiene 3 sectores:

1. Campo palabra. Al lado de "palabra:" hay un campo de texto.
2. Campo significado. Al lado de "significado:" hay un campo de texto.
3. Etiqueta de mensaje: abajo de los botones hay una etiqueta que mostrará el resultado de la operación.

Al iniciar la aplicación (ventana), se debe leer el contenido del diccionario desde un archivo de texto "diccionario.txt". Este archivo contiene líneas de la forma "palabra.significado", por ejemplo:

```
hola.hi
chao.good bye
```

estas líneas deben cargarse en el diccionario antes que el/la usuario haga nada.

Luego, la funcionalidad deseada es la siguiente:

- El/la usuario debe poder ingresar una palabra y
 - Apretar el botón `buscar`: muestra el significado en el campo de texto significado. En la etiqueta mensaje muestra "encontrado" o "no encontrado", dependiendo del resultado.
 - Apretar el botón `borrar` borra la palabra del diccionario. No muestra nada en el campo de texto significado. En la etiqueta mensaje muestra la palabra "borrado" si la palabra existe y fue borrada, o "no existe" si la palabra no existía.
- El/la usuario debe poder ingresar una palabra y un significado y
 - Apretar el botón `agregar` agrega la palabra y el significado asociado al diccionario. Si la palabra ya existe despliega "ya existe" en la etiqueta mensaje. Si no existe pero agregarla al diccionario no funciona (por ej. el método `agregar()` de la clase `diccionario` responde `False`) entonces despliega "no hay espacio".
 - Apretar el botón `cambiar` cambia el significado de una palabra. El campo de texto palabra debe tener la palabra a cambiar (la cual debe existir en el diccionario), y el campo de texto significado debe tener el significado asociado. Si el cambio funciona bien, debe mostrar "cambiado" en la etiqueta mensaje. Si no, debe mostrar "no existe".
- Finalmente, el/la usuario debe poder apretar el botón `grabar` para almacenar el contenido del diccionario actual en uso en un archivo de texto de nombre "diccionario.txt", siguiendo el formato de las líneas del archivo indicado más arriba.

▼ Solución:

Partiremos armando *visualmente* la ventana. Creamos la ventana con sus campos y sus botones usando Tkinter. (Todavía no haremos las funciones asociadas a las acciones de los botones.)

```
# programa principal

v=Tk()
f1=Frame(v)
f2=Frame(v)
f3=Frame(v)
```

```

Label(f1,text="palabra:").pack(side=LEFT)
palabra=Entry(f1)
palabra.pack()

Label(f2,text="significado:").pack(side=LEFT)
significado=Entry(f2)
significado.pack()

Button(f3,text="buscar",command=buscar).pack(side=LEFT)
Button(f3,text="agregar",command=agregar).pack(side=LEFT)
Button(f3,text="borrar",command=borrar).pack(side=LEFT)
Button(f3,text="cambiar",command=cambiar).pack(side=LEFT)
Button(f3,text="grabar",command=grabar).pack()

respuesta=Label(v)

f1.pack(); f2.pack(); f3.pack(); respuesta.pack()
v.mainloop()

```

Ahora agregaremos las funciones de las acciones y el resto. Todo lo siguiente va *antes, más arriba* del código que crea la ventana mostrado justo antes.

```

from tkinter import *
from Diccionario import *

D=Diccionario("diccionario.txt")

#metodo para boton buscar
def buscar():
    s=D.buscar(palabra.get())
    if s==None:
        responder("no existe")
    else:
        significado.delete(0,END) #limpiar
        significado.insert(0,s)
        responder("encontrado")

#metodo para boton agregar
def agregar():
    p=palabra.get()
    if D.agregar(p,significado.get()):
        responder("agregado")
    elif D.buscar(p) != None:
        responder("ya existe")
    else:
        ..

```

```
    responder("no hay espacio")
```

```
#metodo para boton borrar
```

```
def borrar():  
    if D.borrar(palabra.get()):  
        responder("borrado")  
    else:  
        responder("no existe")
```

```
#metodo para boton cambiar
```

```
def cambiar():  
    if D.cambiar(palabra.get(),significado.get()):  
        responder("cambiado")  
    else:  
        responder("no existe")
```

```
#metodo para boton grabar
```

```
def grabar():  
    D.grabar("diccionario.txt")  
    responder("grabado")
```

```
#método para mostrar mensaje de respuesta
```

```
def responder(x):  
    respuesta.config(text=x)
```

Problema 2: Crear la clase Diccionario

La aplicación anterior funciona si existe la clase Diccionario con los métodos indicados. La vamos a implementar ahora.

Implementación de la clase Diccionario:

Hay dos posibles soluciones:

- [utilizando diccionarios de Python](#)
- [utilizando listas de Python](#)

