

# DIPLOMATURA EN PYTHON

## Trabajo intermedio y final (Obligatorios)

*Profesor:*

**Juan Barreto**

*Alumno:*

**Nahuel Matías Fortuna**

**-2023 -**

**INDICE**

1) Introducción..... 1

2) Consignas..... 1

3) Mi Resolución..... 2

4) Aplicación Lograda..... 4

5) Mi Código..... 9

6) Conclusión.....12

### 1) Introducción:

Conforme transcurre la cursada se irá realizando un trabajo final para ir integrando los conceptos aprendidos en cada unidad. El trabajo pretende tener una continuidad desde la unidad 3 hasta la unidad 8.

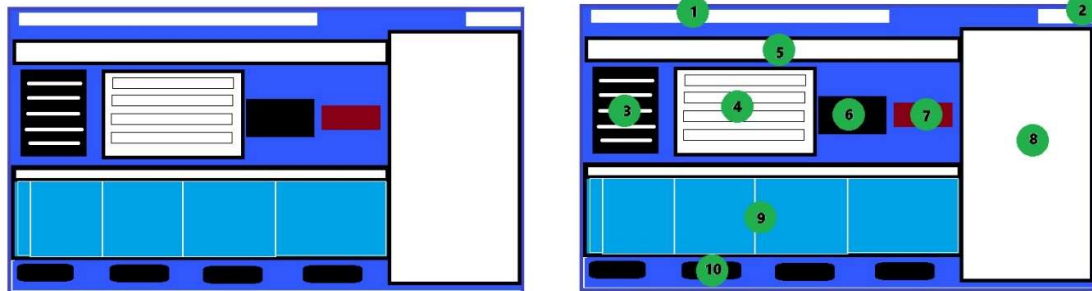
### 2) Consignas:

- **IMPORTANTE:** Puede ser realizado a elección del alumno mediante el uso de Tkinter o usando sólo Python.
- **Nivel de exigencia:** Alto
- **Objetivos:** Probar los conocimientos adquiridos y que el alumno comience a pasar de la teoría a la práctica.
- **Participantes:** El trabajo puede ser realizado de forma individual pero para afianzar el trabajo en equipo y la participación, se permite que los alumnos puedan conformar grupos de hasta 5 personas.
- Esta entrega es a los fines que chequear el trabajo de cada alumno y poder brindar ayuda en el caso de que se note que el alumno o los alumnos no están participando. No es considerada con nota.
- La presentación se realiza en el foro de la unidad 5.
- Consta de la entrega de una app a la cual se le irán agregando:
- **Unidad 1 a 5** – Toma de datos
- **Unidad 2 a 5** – Uso de funciones, condicionales y bucles.
- **Unidad 6** – Uso de base de datos (SQLite3 a elección)..  
Aquí se pide:
  - Creación de base de datos y tablas desde Python.
  - Realizar un alta de registro en la base de datos solicitando al usuario que ingrese los datos ya sea mediante consola si solo se utiliza Python o mediante campos del tipo Entry si se utiliza Tkinter.
- **Unidad 7** : Implementación de regex para validación de uno de los datos tomados en la unidad 6, realización de abmc (crud en ingles) (**alta**, **baja**, **modificación**, **consulta**)

### 3) Mi resolución:

La idea es que como yo soy administrativo contable, manejo mucho lo que es planilla de Excel y demás, pero a veces es engorroso, tener que entrar por consultas para solo buscar la clave fiscal y contraseña en cada Excel. Es por esto que, para facilidad de obtención de estos datos, creare una aplicación registro, integrando la ocupación del mismo, cuit/cuil y clave fiscal, teniendo un registro solo de esos datos y no la planilla completa con todos los datos que puede manejar de un cliente.

La idea de diseño será la siguiente:



Referencias:

- 1 - Encabezado de la aplicación dando referencia al estudio contable.
- 2 – Allí aparecen las opciones típicas de ventanas (minimizar, maximizar o cerrar).
- 3 – Serán los nombres de las variables a rellenar.
- 4- Campos a rellenar de Nombre, Apellido, Rubro, CUIT/CUIL y Clave fiscal.
- 5- Título de estudio contable y referencia de ingreso de datos.
- 6- Imagen ilustrativa de AFIP.
- 7- Boton que redirecciona a la página principal de AFIP.
- 8- Imagen ilustrativa a modo de favorecer la visualización.
- 9- Tabla de contenido con los datos de los clientes.
- 10- Botonera (ALTA, BAJA, MODIFICAR, ELIMINAR).

**4) Aplicación Lograda:**

La aplicación lograda es una aplicación realizada con Python con el apoyo de la base de datos Sqlite3 y ademas la utilización de Tkinter.

La idea de esta aplicación es el facilitar los datos de inicio de sesión a la cuenta correspondiente de cada cliente en AFIP, teniendo la posibilidad de ingresar o redirigirse desde la aplicación misma, llevando un control también de los clientes.

La aplicación permite la carga del cliente y poder observar los siguientes datos:

- Nombre.
- Apellido.
- Rubro o Oficio.
- CUIT/CUIL.
- Clave fiscal.

Pudiendo interactuar con los mismos, a continuación, veremos cómo se visualiza.

Aquí podemos ver la aplicación y su ventana terminada como se lo había diseñado previamente, buscando una armonía de colores y de fácil entendimiento para el usuario que utilice la misma.

ID	Nombre	Apellido	Rubro / Oficio	CUIT / CUIL	Clave Fiscal
7	Milagros	Torres	Ingeniera	27418754755	mllenaing90
4	Martina	Gutierrez	Oficinista	27452114596	martinaofi89
3	Hector	Gutierrez	Chapista	20233659651	puchi4567puchi
2	Julia	Ramirez	Nutricionista	27454522513	JULIA000
1	Ramon	Amor	Oficinista	20407042299	RAMON547/

Presenta el titular como habíamos dicho, donde allí podría ir el nombre del estudio contable, y las opciones correspondientes a minimizar, maximizar o cerrar la misma.

Podemos ver el titulo con su instrucción, siguiendo hacia la izquierda las variables que participan para el llenado de los campos vacíos que nombramos con anterioridad, y los campos para que el usuario pueda cargar un nuevo cliente.

Nombre	<input type="text"/>
Apellido	<input type="text"/>
Rubro/Oficio	<input type="text"/>
CUIT/CUIL	<input type="text"/>
Clave Fiscal	<input type="text"/>

Posee un método de seguridad para colocar la contraseña así se posee privacidad.

Clave Fiscal	<input type="password" value="*****"/>	<input type="checkbox"/> Mostrar contraseña
Clave Fiscal	<input type="text" value="rulemanes456"/>	<input checked="" type="checkbox"/> Mostrar contraseña

Luego yendo hacia la derecha podemos ver el logo de AFIP como se había pensado, con un botón que llame la atención de fácil acceso que redirige al usuario hacia la pagina principal de AFIP si lo desea.



Un poco mas hacia la derecha encontramos nuestra imagen ilustrativa que complementa nuestra aplicación, dando una mejor visualización y tamaño a la misma. Además es la imagen que une todo el contexto de la aplicación de manera que concatena con la carga de datos y la tabla con los clientes del estudio contable.

Se utiliza el azul de fondo y de contraste para la tabla y los datos ya llenados un celeste más claro cómo se puede apreciar.

ID	Nombre	Apellido	Rubro / Oficio	CUIT / CUIL	Clave Fiscal
4	Martina	Gutierrez	Oficinista	27452114596	martinaofi89
3	Hector	Gutierrez	Chapista	20233659651	puchi4567puchi
2	Julia	Ramirez	Nutricionista	27454522513	JULIA000
1	Ramon	Amor	Oficinista	20407042299	NahuelIA345



Por último encontramos la botonera personalizada que se decidió darles un color negro con letras blancas, formato también de las variables, donde allí podrá el usuario interactuar y dar las distintas órdenes.

Agregar Cliente

Consultar Cliente

Modificar Cliente

Eliminar Cliente

Max/Min

Las ordenes son:

- Agregar Cliente: Se debe completar todos los casilleros correspondientes y luego clic en dicho botón para cargarlos en la base de datos. En caso de error saldrá ventana emergente, al igual que si se cargó con éxito.
- Consulta Cliente: Selecciona el cliente en la planilla y al poner consultar lo tendrá presente en los campos a rellenar.
- Modificar Cliente: Seleccionado el cliente, podrá reemplazar o modificar algún cliente que se ha cargado erróneamente.
- Eliminar Cliente: Borra el cliente de la planilla correspondiente a la base de datos.
- Maximizar o minimizar: Lo que realiza es Maximizar o minimizar la ventana para que se lean o no los datos correspondientes a otros clientes, ocultándolos.

Podemos ver en las siguientes capturas nos permite poder consultar la base de datos si lo requerimos, y si queremos la minimizamos para que otro no vea esa información sensible.

- Ventana minimizada, contraseña con caracteres y podemos utilizar cualquier función de la botonera. Si marcamos mostrar contraseña nos mostrara la contraseña.

Registro De Clientes - Estudio Contable

### Plantilla de clientes - Ingrese sus datos

Nombre: Milagros  
 Apellido: Torres  
 Rubro/Oficio: Ingeniera  
 CUIT/CUIL: 27418754755  
 Clave Fiscal:  ☐ Mostrar contraseña

**Ingreso a AFIP**

Agregar Cliente Consultar Cliente Modificar Cliente Eliminar Cliente Max/Min

Registro De Clientes - Estudio Contable

### Plantilla de clientes - Ingrese sus datos

Nombre: Milagros  
 Apellido: Torres  
 Rubro/Oficio: Ingeniera  
 CUIT/CUIL: 27418754755  
 Clave Fiscal:  ☒ Mostrar contraseña

**Ingreso a AFIP**

Agregar Cliente Consultar Cliente Modificar Cliente Eliminar Cliente Max/Min

- Ventana maximizada logramos ver la base de datos para poder consultar las contraseñas y cuit/cuil. Si desmarcamos mostrar contraseña nos mostrara la contraseña.

Registro De Clientes - Estudio Contable

### Plantilla de clientes - Ingrese sus datos

Nombre: Milagros  
 Apellido: Torres  
 Rubro/Oficio: Ingeniera  
 CUIT/CUIL: 27418754755  
 Clave Fiscal:  ☐ Mostrar contraseña

**Ingreso a AFIP**

ID	Nombre	Apellido	Rubro / Oficio	CUIT / CUIL	Clave Fiscal
7	Milagros	Torres	Ingeniera	27418754755	mllenaing90
4	Martina	Gutierrez	Oficinista	27452114596	martinaofi89
3	Hector	Gutierrez	Chapista	20233659651	puchi4567puchi
2	Julia	Ramirez	Nutricionista	27454522513	JULIA000
1	Ramon	Amor	Oficinista	20407042299	RAMON547/

Agregar Cliente Consultar Cliente Modificar Cliente Eliminar Cliente Max/Min

Algunas de las ventanas emergentes configuradas podemos observar en los siguientes casos:

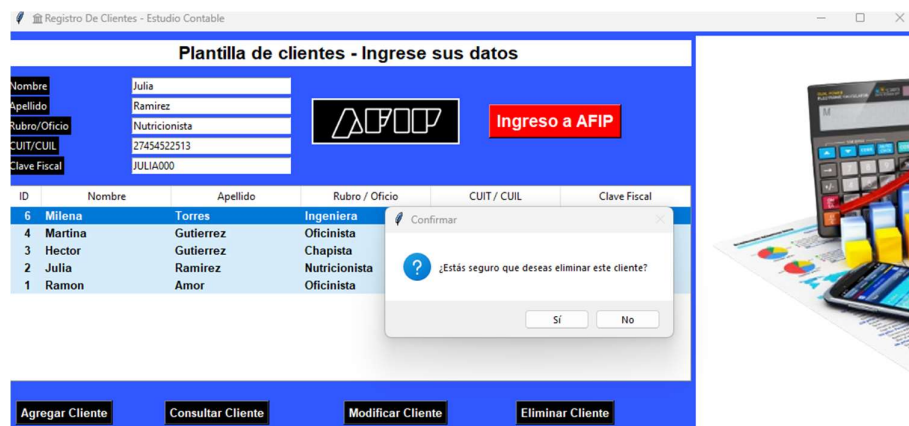
- No se cargo uno o ninguno de los datos, pide que los complete.



- Cuando cargamos un cliente nuevo, nos avisa que fue con éxito.



- Cuando queremos eliminar un cliente, nos pregunta si realmente queremos borrarlo o nos equivocamos.





- En caso de que la opción sea si y por lo tanto eliminamos un cliente entonces nos aparecerá el siguiente mensaje.



- Si modificamos a un cliente, entonces sale la ventada de que fue modificada con éxito.



### 5) Mi código:

A manera orientativa se la aplicación realizada, se trato de aplicar los conocimientos que se han dado a lo largo de la cursad en el curso de Python inicial.

Se baso un orden en el que va de la siguiente manera:

- MODULOS IMPORTADOS
- MODELO – FUNCIONES
- VISTAS
- TREEVIEW
- BOTONERA

Cada una de estas partes muy importantes para la realización de la misma, y que permitieron que pueda seguir un camino.

Explicando un poco de cada una, vemos lo siguiente:

Se utilizó los siguientes módulos importados, como se dijo con anterioridad se usó Tkinter y todas sus herramientas para visualizar, sqlite3 para la base de datos, el módulo re para las operaciones especiales de regex y por último para el botón de redireccionamiento a la página principal de AFIP el webbrowser.

```
#
#
#MODULOS IMPORTADOS
#
from tkinter import *
from tkinter.messagebox import *
from tkinter import ttk
import sqlite3
import re
import webbrowser
```

Dentro del modelo se coloca todo lo que se refiere a las funciones, que permiten la conexión, creación de tablas, control, interacción con el usuario (ALTA, BAJA, MODIFICAR, ELIMINAR), como también funciones complementarias para validar datos.

```
#
#
#MODELO PROPIO REGISTRO DE CLIENTES
#SOLO DE INGRESO A AFIP
#
#FUNCIONES
#
#Conexion y creacion de tabla clientes con control de conexion.
def conexion():
    con = sqlite3.connect('base_contable.db')
    return con

def crear_tabla(con):
    cursor = con.cursor()
    sql = "CREATE TABLE IF NOT EXISTS clientes(id INTEGER PRIMARY KEY AUTOINCREMENT, nombre varchar(50))"
    cursor.execute(sql)
    con.commit()

try:
    con = conexion()
    crear_tabla(con)
except sqlite3.Error as e:
    print("ERROR DE CONEXION:", e)
```

En vista se utilizó varios componentes, el principal fue el programa Tkinter, aunque también se complementa con imágenes, detalles de colores, tipos de letras y espacios.

```
#
#
#VISTA VISTA VISTA
#
root = Tk()
root.title("Registro De Clientes - Estudio Contable ")
root.configure(bg="#3157FF")

# Agregar imagenes
row_offset = 1
imagen = PhotoImage(file="cpn.png", width=250, height=450)
imagen_label = Label(root, image=imagen, bg="white")
imagen_label.grid(row=0, column=0, rowspan=8, padx=5, pady=5, sticky="e")

imagen_afip = PhotoImage(file="afip.png")
label_imagen_afip = Label(root, image=imagen_afip, bg="white")
label_imagen_afip.grid(row=row_offset, column=1, rowspan=5, padx=(0, 10), pady=(0, 10), sticky="w")

titulo = Label(root, text="Plantilla de clientes - Ingrese sus datos", font=("Arial", 16, "bold"), fg="black", bg="white")
titulo.grid(row=0, column=0, columnspan=6, pady=(10, 10), sticky="we")

# Variables de control
a_val = StringVar()
```

Aquí tenemos las TREEVIEWS que bueno termina definiendo un poco como se va a ver nuestra tabla con los datos de clientes y sus variables.

```
#
# TREEVIEWS
#
tree = ttk.Treeview(root)
tree = crear_marco_contenedor()
tree["columns"] = ("col1", "col2", "col3", "col4", "col5")
tree.column("#0", width=40, minwidth=40, anchor=W)
tree.column("col1", width=150, minwidth=150, anchor=W)
tree.column("col2", width=150, minwidth=150, anchor=W)
tree.column("col3", width=150, minwidth=150, anchor=W)
tree.column("col4", width=150, minwidth=150, anchor=W)
tree.column("col5", width=150, minwidth=150, anchor=W)

tree.heading("#0", text="ID")
tree.heading("col1", text="Nombre")
tree.heading("col2", text="Apellido")
tree.heading("col3", text="Rubro / Oficio")
tree.heading("col4", text="CUIT / CUIL")
tree.heading("col5", text="Clave Fiscal")
tree.grid(row=row_offset + 5, column=0, columnspan=6, pady=(10, 10))
#
```

Por último tenemos todos los Botones que permiten al usuario realizar ALTA/BAJA/MODIFICAR/ELIMINAR, editándolos, acompañando gráficamente con colores y letras legibles.

Terminando así el código con un actualizar para poder tener presente la tabla de clientes.

```
#
# BOTONERA
#
boton_alta = Button(root, text="Agregar Cliente", command=lambda:alta(a_val.get(), b_val.get(), c_val.get(), d_val.get(), e_val.get(), tree), bg="black",
boton_alta.grid(row=row_offset + 6, column=0, padx=(0, 10), pady=10)

boton_consulta = Button(root, text="Consultar Cliente", command=lambda:consultar(tree, a_val, b_val, c_val, d_val, e_val), bg="black", fg="white", font=('
boton_consulta.grid(row=row_offset + 6, column=1, padx=(0, 10), pady=10)

boton_modificar = Button(root, text="Modificar Cliente", command=lambda:modificar(tree, a_val, b_val, c_val, d_val, e_val), bg="black", fg="white", font=('
boton_modificar.grid(row=row_offset + 6, column=2, padx=(0, 10), pady=10)

boton_borrar = Button(root, text="Eliminar Cliente", command=lambda:borrar(tree), bg="black", fg="white", font=('Arial', 10, 'bold'))
boton_borrar.grid(row=row_offset + 6, column=3, padx=(0, 10), pady=10)

boton_afip = Button(root, text="Ingreso a AFIP", command=abrir_afip, bg="red", fg="white", font=('Arial', 14, 'bold'))
boton_afip.grid(row=row_offset, column=3, rowspan=5, padx=(0, 10), pady=(0, 10), sticky="W")

boton_maximizar_minimizar = Button(root, text="Max/Min", command=maximizar_minimizar_ventana, bg="black", fg="white", font=('Arial', 10, 'bold'))
boton_maximizar_minimizar.grid(row=row_offset + 6, column=4, padx=(0, 10), pady=10)

# Actualizar el Treeview con datos existentes
actualizar_treeview(tree)

root.mainloop()
```

## **6) Conclusión:**

Con base en la experiencia y el aprendizaje inicial en Python, se han adquirido conocimientos fundamentales que abarcan diversas áreas de programación y desarrollo de aplicaciones. A continuación, se presentan las conclusiones clave:

### **1. Dominio de Estructuras de Datos:**

- La comprensión y aplicación de listas y tuplas proporcionan habilidades esenciales para organizar y manipular datos de manera efectiva en programas Python.

### **2. Integración de Bases de Datos:**

- La capacidad para trabajar con bases de datos SQLite3 representa un paso importante en el manejo de datos de manera persistente, permitiendo la creación de aplicaciones más robustas y funcionales.

### **3. Desarrollo de Interfaces Gráficas:**

- La adquisición de habilidades en Tkinter posibilita la creación de interfaces gráficas de usuario (GUI), mejorando la experiencia del usuario y facilitando la interactividad en las aplicaciones.

### **4. Aplicación Práctica en Desarrollo de Software:**

- La combinación de las habilidades anteriores ha llevado a la creación exitosa de una aplicación funcional que utiliza tanto estructuras de datos como bases de datos, demostrando la capacidad para desarrollar soluciones completas y prácticas.

Realmente hemos aprendido un nuevo lenguaje de programación, saber relacionarlo con una base de datos y poder interactuar como usuario realizando nuestro primer proyecto de programación, siendo en mi caso una aplicación para un estudio contable, aplicando todo lo dado en el año y investigando aparte sobre como adornarlo un poco, para que esta experiencia primeriza demuestre el compromiso del alumno con el curso.