



# MANUAL TECNICO

Programa para leer y manipular cursos

## [ENCABEZADO]

El presente manual describe las funcionalidades, métodos, herramientas y validaciones que se utilizaron para la realización de la aplicación.

## [PRINCIPIO, TÉCNICA O PARADIGMA APLICADO DE PROGRAMACIÓN]

Programación orientada a objetos

- Polimorfismo

- Modularidad

Paradigma Imperativo

- Programación modular

## [CONVENCIONES DE NOMENCLATURA]

[DECLARACION DE VARIABLES, METODOS O FUNCIONES]

```
es_variable=""
```

```
es_variable=0
```

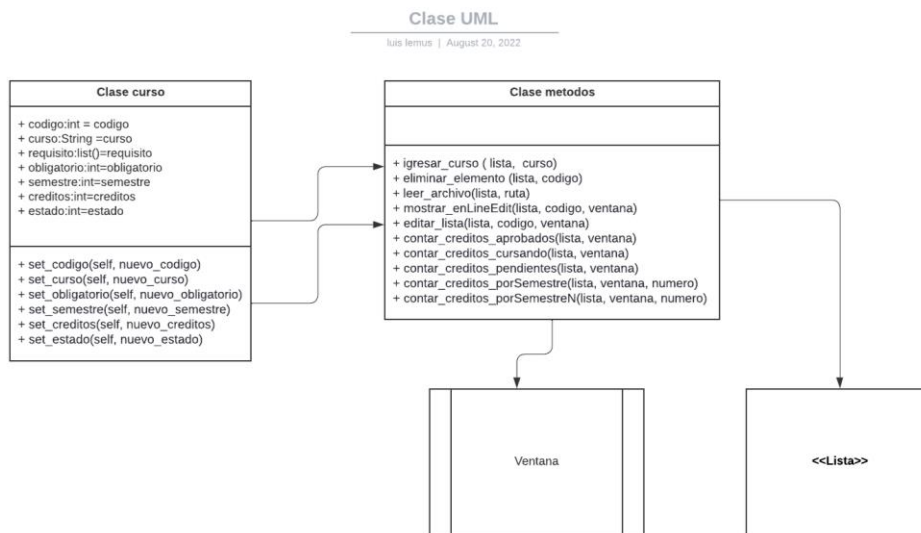
```
def es_metodo():
```

```
    print("hola mundo")
```

```
def es_funcion():
    es_variable="hola mundo"

    return es_variable
```

## [DIAGRAMA DE CLASES]



## [MÉTODOS PRINCIPALES]

- Ingresar curso

Descripción: A partir de una lista vacía y validando si el elemento existe ingresa el nuevo curso.

```
class metodos:
    def ingresar_curso ( lista, curso):
        numero=int(curso.codigo)
        existe = False

        if not lista:
            lista.append(curso)

        else:
            for x in lista:
                nux=int(x.codigo)

                if nux == numero:
                    existe = True
                    lista.remove(x)
                    lista.append(curso)
                    break

            if existe == False:
                lista.append(curso)
```

### Eliminar elemento

Descripción: A partir de una lista no vacía y

Validando que el curso exista se elimina dicho curso.

```
def eliminar_elemento (lista, codigo):
    numero=int(codigo)
    existe = False

    if not lista:
        messagebox.showerror("error","no se puede eliminar elementos vacios ")
    else:
        for x in lista:
            nux=int(x.codigo)

            if nux == numero:
                existe = True
                lista.remove(x)
                break

        if existe == False:
            messagebox.showerror("error","el codigo del curso que desea eliminar no esta en la lista")
```

## Leer archivo

Descripción: A partir de una ruta y una lista para almacenar los cursos llenamos la lista concursos recorriendo por medio de un ciclo for el objeto del archivo.

```
def leer_archivo(lista, ruta):
    with open(ruta, encoding='utf-8') as f :
        reader = csv.reader(f)
        messagebox.showinfo("mensaje", "se leyo correctamente el archivo")
        filas = ""
        requisito=[]
        for row in reader :

            try:
                curs = row[1]
                codigo = row[0]
                requisito = row[2]
                obligatorio = int(row[3])

                semestre = int(row[4])
                creditos = int(row[5])
                estado = int(row[6])
                cur = curso(codigo, curs, requisito, obligatorio, semestre, creditos, estado )
                igresar_curso(lista, cur)
                messagebox.showinfo("mensaje", "se agrego el contenido del archivo correctamente a la lista")
            except:
                messagebox.showerror("error", "has escrito un parametro fuera del tipo")
```

## [REQUERIMIENTO O FUNCIONALIDAD ESPECIFICA]

### [DESCRIPCIÓN]

```
from tkinter import filedialog
from clase_curso import *
from metodos import*
from PyQt5 import QtCore
from PyQt5 import QtWidgets, uic
```

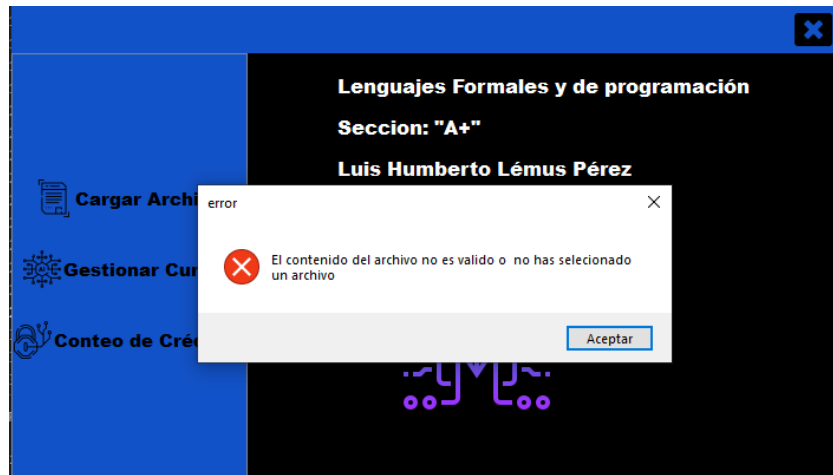
Se usó librerías de tkinter por la facilidad para generar diálogos ya sea de leer archivos como mensajes de error

Se usó librerías de PyQt5 para el manejo de widgets y la aplicación de interfaces graficas necesarias para la visualización del usuario.

[INTERFACES PRINCIPALES]

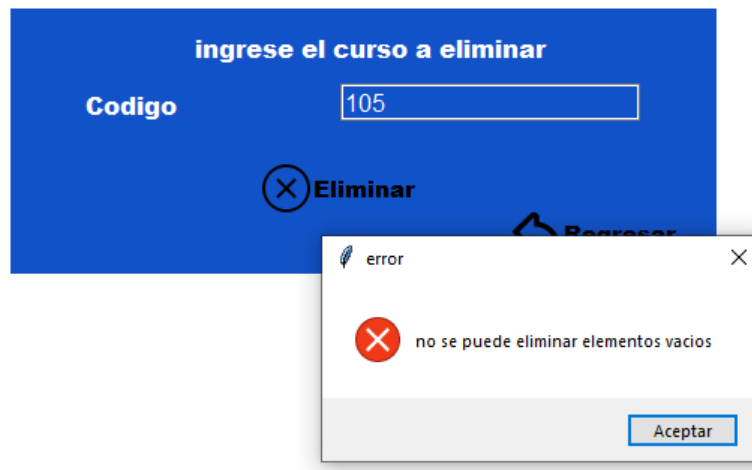
[VERIFICACIÓN DE VALIDACIONES]

[VALIDACION #1]



[Error al cerrar el dialogo de búsqueda del archivo o un archivo incorrecto]

[VALIDACION #2]



[Error si el código no existe o no hay elementos en la lista]

### [VALIDACION #3]

The screenshot shows a web application interface with a dark background. At the top, there's a section titled "Mostrar credits" with sub-sections: "Creditos Aprobados:", "Créditos Cursando:", and "Créditos Pendientes:". Below these is a blue button labeled "Actualizar". Further down, there's a form with the label "Ingrese el semestre" followed by a text input field and a blue button labeled "validar". Below this, the same sub-sections are repeated, followed by another "Ingrese el semestre" input field. At the bottom, there's a section for "Creditos obligatorios:". An error dialog box is open in the foreground, titled "error", with a red 'X' icon and the message "no se puede contar creditos vacios". It has an "Aceptar" button.

[Error el cual indica que no se pueden contar elementos que no existen en la lista o escribió un parámetro no valido]

### [VALIDACION #4]

The screenshot shows a web application interface with a dark background. At the top, there's a section titled "Ingresa el codigo a editar" followed by a text input field and a blue button labeled "Validar". Below this, there's a form with labels: "Codigo", "Nombre", "Pre Requisito", "Opcionalidad", "Semestre", "Creditos", and "Estado", each followed by a text input field. At the bottom, there's a blue button labeled "Editar" and a blue button labeled "Regresar" with a back arrow icon. An error dialog box is open in the foreground, titled "error", with a red 'X' icon and the message "has escrito un parametro fuera del tipo o el codigo no existe". It has an "Aceptar" button.

[Error que indica que no se pueden buscar elementos que no existen o lo que se ingresó no es válido]