

MANUAL TECNICO

PARADIGMA APLICADO DE PROGRAMACIÓN

Programación modular

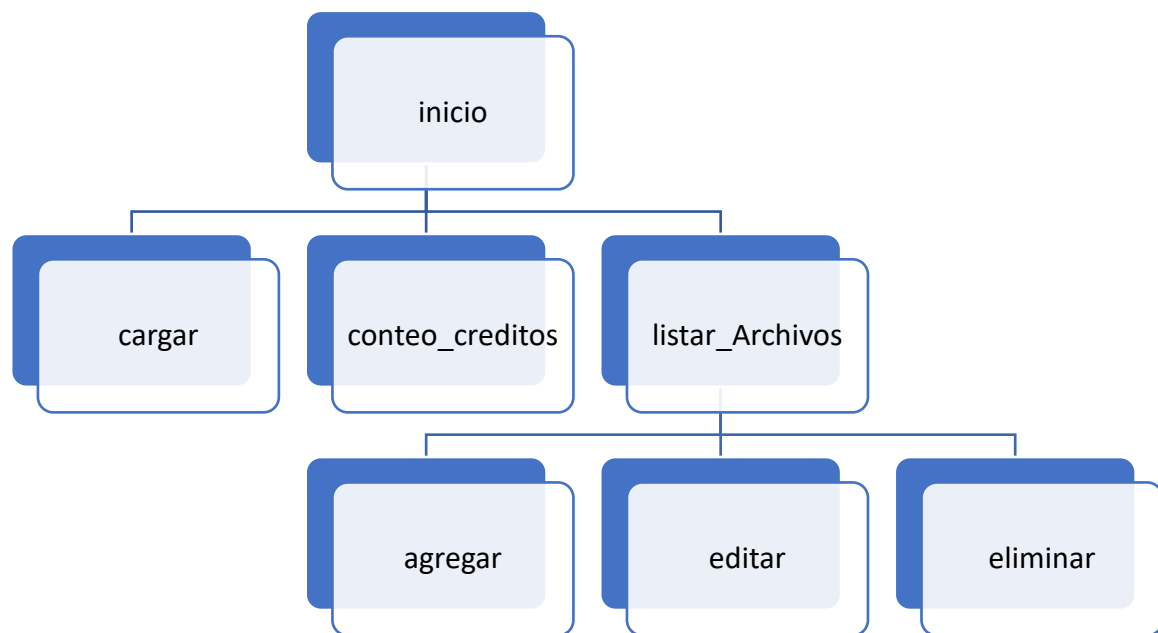
CONVENCIONES DE NOMENCLATURA

Método Limpiar

Este método limpia el treeview cada vez que se enlista ya que si no se limpia muestra de nuevo los valores al abrirlo.

```
def Limpiar():  
    tree.delete(*tree.get_children())
```

DIAGRAMA DE CLASES



MÉTODOS PRINCIPALES

Método mostrar_gestionar

Este método conecta el frame principal con el frame gestionar archivos para que se puedan conectar entre sí, ir y volver al como desee, de la misma forma esta

conectadas todas las ventanas. Este contiene un try except para el manejo de errores que pueda surgir al navegar entre frames.

```
def mostrar_gestionar():
    try:
        principal.grid_forget()
        gestionar_cursos.grid(row=0,column=0,columnspan=3,pady=0)
        from listar_Archivos import load_csv
        load_csv()
    except:
        tkinter.messagebox.showerror("Error","Cargue su archivo")
    return None
```

Método ventana cargar

Este método recibe la ventana como parámetro y configura todos los elementos que va a contener el nuevo frame. Todos los frames están configurados de la misma manera, únicamente cambian los elementos que contienen y las posiciones de los mismos.

```
def ventana_cargar(master, callback=None, args=(), kwargs={}):
    if callback is not None:
        callback = functools.partial(callback, *args, **kwargs)

    main_frame = Frame(master)

    Label(main_frame,text="
").grid(row=0,column=0, columnspan=3, pady=5)
    Label(main_frame,text="
").grid(row=4,column=0, columnspan=3, pady=5)
    ttk.Button(main_frame, text="Seleccionar", command=
abrirArchivo).grid(row=1,column=1,sticky=W+E, pady=5)
    ttk.Button(main_frame, text="Regresar",
command=callback).grid(row=3,column=1,sticky=W+E, pady=5)

    return main_frame
```

Método load_cvs

Este método lista los cursos cargados en un treeview mediante la lectura de archivos con la librería cvs luego almacena los valores en una lista.

```
def load_csv():
    Limpiar()
    import cargar
```

```

file=cargar.archivo
with open(file,encoding="utf8") as myfile:
    csvread = csv.reader(myfile, delimiter=',')
    i = 0
    global columna
    columna = []
    resultado=[]
    global lista
    lista=[]
    global sumas
    sumas=[]
    for fila in csvread:
        columna.append(fila[i])
        lista.append(fila)
    for element in columna:
        res = len(columna) - 1 - columna[::-1].index(element)
        if res not in resultado:
            resultado.append(res)
            sumas.append(lista[res])
    for r in range(len(resultado)):
        tree.insert("", 'end', values=lista[resultado[r]])
with open(file, "w", newline='',encoding="utf8") as myfile:
    csvwriter = csv.writer(myfile, delimiter=',')
    for row_id in range(len((lista))):
        row = lista[row_id]
        csvwriter.writerow(row)

```

Método abrirArchivo

Este método abre el archivo csv y lo lee si este es leído correctamente entra confirma el try, si no, manda un error por medio de un except.

```

def abrirArchivo():
    global archivo
    archivo = filedialog.askopenfilename(title = "Seleccionar Archivo",
filetypes = (("CSV Files","*.csv"),("all files",
                                                    " *.*")))
    #indica.configure(text="Archivo cargado correctamente")

    try:
        miArchivo = open(archivo, "r")
        lectura = miArchivo.readlines()
        tkinter.messagebox.showinfo("Informacion","Archivo cargado
correctamente")

```

```

except ValueError:
    tkinter.messagebox.showerror("Error", "Formato incorrecto")
    return None
except FileNotFoundError:
    tkinter.messagebox.showerror("Error", "Archivo dañado \n o no
seleccionado")
    return None

```

Método sabe_cvs

Este método sirve para la función agregar primero lee el archivo luego busca si ya existe el campo que se desea agregar, si este no existe pasa por una serie de validaciones de ingreso de datos para poder agregar un nuevo curso a la lista.

```

def save_csv():

    import listar_Archivos
    global tree1
    tree1=listar_Archivos.tree
    import cargar
    file=cargar.archivo
    with open(file, "w", newline='',encoding="utf8") as myfile:
        csvwriter = csv.writer(myfile, delimiter=',')
        colu=listar_Archivos.columna
        listt=listar_Archivos.lista
        if cod1.get() in colu:
            tkinter.messagebox.showerror("Error", "Ya existe el curso")
            for row_id in range(len(((tree1.get_children())))):
                row = listt[row_id]
                csvwriter.writerow(row)
        else:
            if cod1.get().isdigit() and opcion.get() == "1" or opcion.get()
== "0":
                if semestre.get().isdigit():
                    if estado.get()=="0" or estado.get()=="1"or
estado.get()=="-1":
                        if creditos.get().isdigit() and nombre.get() != "":
                            tree1.insert("", 'end',
values=[str(cod1.get()),str(nombre.get()), str(pre.get()),
str(opcion.get()),str(semestre.get()),str(credit
os.get()),str(estado.get()))]
                            listt.append([str(cod1.get()),str(nombre.get()),
str(pre.get()),

```

```

                                str(opcion.get()),str(semestre.get()),str(credit
os.get()),str(estado.get()))])
                                colu.append(cod1.get())
                                tkinter.messagebox.showinfo("Informacion", "Curso
agregado correctamente")
                                else:
                                    tkinter.messagebox.showerror("Error", "Datos
Ingresados Erroneamente")
                                else:
                                    tkinter.messagebox.showerror("Error", "Datos
Ingresados Erroneamente")
                                else:
                                    tkinter.messagebox.showerror("Error", "Datos Ingresados
Erroneamente")
                                else:
                                    tkinter.messagebox.showerror("Error", "Datos Ingresados
Erroneamente")
                                for row_id in range(len(listt)):
                                    row = listt[row_id]
                                    csvwriter.writerow(row)

```

Método obtener

Este método sirve para el requerimiento de mostrar datos obtiene los datos del código que ingresa el usuario y busca los valores del mismo y los carga en los entry por si se desea editar, para que esto suceda debe de validar que el código existe.

```

def obtener():
    import listar_Archivos
    listt=listar_Archivos.lista
    colu=listar_Archivos.columna
    if codigox.get() in colu:
        res = len(colu) - 1 - colu[::-1].index(codigox.get())
        global busqueda
        busqueda = listt[res]
        cod.set(busqueda[0])
        nombre.set(busqueda[1])
        pre.set(busqueda[2])
        opcion.set(busqueda[3])
        semestre.set(busqueda[4])
        creditos.set(busqueda[5])
        estado.set(busqueda[6])
        tkinter.messagebox.showinfo("Informacion", "Informacion precargada")
    else:

```

```
tkinter.messagebox.showerror("Error","El código no existe")
```

Método edit

Este método sirve para editar un curso recibiendo los valores que ingresa el usuario manualmente por medio de entry previamente haciendo validaciones del tipo de dato que se acepta en cada uno de los entrys. Si son correctos los valores estos se editan el curso que se selecciono.

```
def edit():
    import listar_Archivos
    global tree1
    tree1=listar_Archivos.tree
    import cargar
    file=cargar.archivo
    with open(file, "w", newline='',encoding="utf8") as myfile:
        csvwriter = csv.writer(myfile, delimiter=',')
        listt=listar_Archivos.lista
        if cod.get().isdigit() and opcion.get() == "1" or opcion.get() ==
"0":
            if semestre.get().isdigit():
                if estado.get()=="0" or estado.get()=="1"or estado.get()=="-
1":
                    if creditos.get().isdigit() and nombre.get() != "":
                        busqueda[1]=str(nombre.get())
                        busqueda[2]=str(pre.get())
                        busqueda[3]=str(opcion.get())
                        busqueda[4]=str(semestre.get())
                        busqueda[5]=str(creditos.get())
                        busqueda[6]=str(estado.get())
                        tkinter.messagebox.showinfo("Informacion","Curso
editado correctamente")
                    else:
                        tkinter.messagebox.showerror("Error","Datos
Ingresados Erroneamente")
                else:
                    tkinter.messagebox.showerror("Error","Datos Ingresados
Erroneamente")
            else:
                tkinter.messagebox.showerror("Error","Datos Ingresados
Erroneamente")
        else:
            tkinter.messagebox.showerror("Error","Datos Ingresados
Erroneamente")
        for row_id in range(len(listt)):
```

```
row = listt[row_id]  
csvwriter.writerow(row)
```

DESCRIPCIÓN

Tkinter: se eligió esta herramienta ya que cuenta con muchas opciones para el diseño de interfaces graficas.

Cvs: se eligió esta herramienta para poder desarrollar lógica de la lectura y escritura de archivos.