

- Facultad de Ingeniería
- Escuela de Ciencias y Sistemas
- Lenguajes Formales y de Programación, 2er. Semestre 2022.



Manual Técnico

PRACTICA 1

LENGUAJES FORMALES Y DE PROGRAMACION

GENESIS NAHOMI APARICIO ACAN

carne :20211329

Índice

descripcion del programa	01
logica del programa	02
frame	03
Clase menuU	03
main	04
fun1	04
func2	05
elimini	05
mostrar	06
listar	06
agrgr	06
edit	07
conte	07
logica	09
Clase esqueleto	10
imprecion	11
prueba	12
optener	13
edi	14
most	14
eliC	15
cont1	16
boton1	16
boton2	

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

Descripcion de la practica

Muchos estudiantes de ingeniería a veces se retrasan Avance profesional para ignorar las condiciones previas y las restricciones crediticias. requerido por algunos cursos. por esto mismo se pidio la implementación de software de tipo visual, que facilita la presentación de información relacionada con el programa escolar. El programa fue realizado enel lenguaje de programacion python, usando VisualStudioCode .Cuenta con una interfaz amigable con el usuario para que este pueda interactuar sin problemas con el programa asi mostrar y facilitar la información del programa de estudio . Un formulario visual de cada uno de ellos para ver qué cursos llevar durante un semestre o escuela de vacaciones.

¿paradigma que se utilizo?

para esta practica se utilizo
mayormente POO (Programacion
orientado a objetos

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

logica del programa

se utilizaron tresmodulos interfaz , logica y main para realizar el programa . A continuación se presenta una breve descripción de las clases, métodos y funciones que fueron implementadas para la realización del programa :

- main (modulo donde inicia el programa)
- interfaz (modulo donde se encuentra la clase cuerpo la cual contiene la parte visual de lo que conlleva el sistema)
- logica (modulo que contiene el funcionamiento interno del programa)

Main

Este modulo es donde se manda a llamar al modulo interfaz y a su clase cuerpo para que el programa comience a funcionar se usa para ejecutar algún código solo si el archivo se ejecutó directamente y no se importó.

```
from frame import Cuerpo

if __name__ == '__main__':
    Cuerpo()
```

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

Frame

En este modulo es donde se crea la clase cuerpo utilizando la libreria de Tkinter de python , que representa la interfaz que esta conformada por una ventana de menu principal la cual muestra el inicio del programa para que el usuario cargue archivos con la extencion csv que contengan cursos , lea estos y los modifique .

```
from tkinter import ttk
from tkinter import *
from tkinter.ttk import Combobox
from tkinter import filedialog
from logica import Cuerpo
class MenuU:
    global scanner

    scanner=Cuerpo()

    def __init__(self): ...

    def func1(self): ...

    def browseFiles(self): ...

    def coca(self): ...

    def func2(self): ...

    def ELIMINI(self): ...

    def eliof(self): ...

    def ele(self): ...

    def mostrar(self): ...

    def ent(self): ...
    def ell(self): ...

    def LISTAR(self): ...
```

- **Universidad de San Carlos de Guatemala**
- **Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería**
- **Lenguajes Formales y de Programación, 2er. Semestre 2022.**

Clase MenuU

la clase Cuerpo de este modulo consta de una variable global y de aproximadamente 23 metodos siendo estos los cuales conforman toda la parte visual con la que el usuario puede interactuar.

los metodos que contiene esta clase son :

- **`__init__`**

es el metodo del constructor el cual contiene el menu de inicio del programa teniendo en ella tres label los cuales muestran el texto que contiene los datos de el programador y cuatro botones que tienen las funciones del programa cargar archivo, gestionar curso, conteo de creditos y salir del programa

```
def __init__(self):
    self.menu1 = Tk()
    self.menu1.title("PRACTICA 1")
    self.menu1.geometry("400x400")
    frame = Frame(self.menu1)
    frame.configure(bg='aquamarine')
    frame.place(x=0, y=0, width=500, height=500)

    texto1 = Label(frame, text="Lab. Lenguajes Formales y de Programacion",bg='aquamarine',fg = "violetRed4")
    texto1.place(x=5, y=20, width=300, height=20)
    texto1.configure(font=("Helvetica",10))
    texto2 = Label(frame, text="Nombre: Genesis Nahomi Aparicio Acan",bg='aquamarine',fg = "violetRed4")
    texto2.place(x=5, y=50, width=280, height=20)
    texto2.configure(font=("Helvetica",10))
    texto3 = Label(frame, text="Carne del Estudiante: 202113293",bg='aquamarine',fg = "violetRed4")
    texto3.configure(font=("Helvetica",10))
    texto3.place(x=5, y=80, width=230, height=20)
```

- **Universidad de San Carlos de Guatemala**
- **Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería**
- **Lenguajes Formales y de Programación, 2er. Semestre 2022.**

• **Func1**

Este metodo contiene la funcion de el boton cargar archivos el cual cuenta con Filechooser para escoger el archivo el cual contiene los cursos que el usuario quiere añadir al programa tener en cuenta que el programa solo puede abrir archivos con la extencion .csv si es otra extencion el programa mostrara un mensaje de error

• **func2**

el metodo fuc2 es el que contienen la funcion del segundo boton de el menu de inicio (gestionar cursos) usa la libreria Tkinter para mostrar una nueva ventana en pantalla ,la cual contiene diferentes botones los cuales conectana a difrentes funciones del programa siendo estos los botones de mostrar,agregar editar y eliminar cursos y un boton de salir el cual tienen la funcion de regresar al menu principal

• **ELIMINI**

la funcion de este metodo es mostrar una nueva ventana con la cual se puede ingresar el codigo de el curso y eliminar este mismo. esta funciona con la libreria Tkinter. ademas de usar un .destroy() para ocultar la ventana del metodo anterior , los botones de este metodo tienen como comandos los metodos eliof y ele

```
def ELIMINI(self):
    self.elimi = Tk()
    self.elimi.title("eliminar curso")
    self.elimi.geometry("400x200")
    self.elimi.configure(bg='aquamarine2')
    self.menu3.destroy()
    self.cod = Label(self.elimi, text="Codigo de Curso",bg='aquamarine2',fg = "violetRed4")
    self.cod.place(x=10, y=40, width=150, height=20)
    self.cod.configure(font=("Helvetica",11))
    self.rut1 = Entry(self.elimi)
    self.rut1.place(x=150, y=40, width=200, height=20)
    Button(self.elimi, text="regresar",bg= "bisque3",fg="DeepSkyBlue4",command=self.ele).place(x=280, y=150, width=100, height=30)
    Button(self.elimi, text="Seleccionar",bg= "bisque3",fg="DeepSkyBlue4",command=self.eliof).place(x=160, y=80, width=80, height=30)
```

- **Universidad de San Carlos de Guatemala**
- **Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería**
- **Lenguajes Formales y de Programación, 2er. Semestre 2022.**

- **mostrar**

el metodo mostrar usa la libreria Tkinter para crear una ventana diferente a las anteriores , ademas de usar un .destroy() para ocultar la ventana del metodo func2 cuenta con varios label ,una caja de texto la cual recibe el codigo del curso a mostrar y dos botones los cuales su funcion es el mostrar los datos de un curso en espesifico atravez de los diferentes label y regresar al menu anterior

- **LISTAR**

este metodo al igual que la mayoria de los metodos de la clase menuU usa la libreria Tkinter para mostrar la interfaz , este cuenta solo con una tabla usando ttk.Treeview(),conformada por columnas , filas , encavezados y un scroll que hace posible el moverse atravez de ella . esta tabla no es editable solo muestra los datos de los diferentes cursos que se agregan al sistema de forma manual o por medio de un archivo .

```
def LISTAR(self):
    self.listar = Tk()
    self.listar.title("listar Cursos")
    self.listar.geometry("560x300")
    self.listar.configure(bg='aquamarine2')
    self.menu3.destroy()
    Button(self.listar, text="regresar",bg= "bisque3",fg="DeepSkyBlue4",command=self.eles).place(x=420, y=260, width=120, height=30)
    columns =('#1', '#2', '#3', '#4', '#5', '#6')
    self.mita=ttk.Treeview(self.listar,columns=columns ,height=8)
    self.mita.grid(sticky="W")
    self.mita.column('#0',width=70, anchor=CENTER)
    self.mita.heading('#6',text="Estado", anchor=CENTER)
    scanner.prueba(mitad=self.mita)
    self.mita.pack(side='left')
    ejscrollbar= ttk.Scrollbar(self.listar,orient=VERTICAL,command=self.mita.yview)
    ejscrollbar.pack(side='right',fill='y')
    self.mita.configure(yscrollcommand=ejscrollbar.set)
```

- **agrrr**

este metodo muestra en pantalla varias cajas de texto asi como label para que el usuario pueda ingresar los datos de algun curso y agregarlo al sistema atravez de el boton agregar que tiene en el , este manda a llamar a otro metodo el cual ayuda a que los datos se guarden en la lista y el programa

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

• EDIT

El siguiente metodo cuenta con la misma interaccion con el usuario que la ventana agrgr , a diferencia que su funcion principal es el de enviar los datos por los que se cambiara algun curso , si este existe atravez del boton editar . a su vez cuenta con un boton regresar que elimina la ventana dle metodo y regresa al menu del metodo func2

```
def EDIT(self):
    self.edit= Tk()
    self.edit.title("editar")
    self.edit.geometry("450x450")
    self.edit.configure(bg='aquamarine2')
    self.menu3.destroy()
    #-----
    self.cd = Label(self.edit, text="Codigo",bg='aquamarine2',fg = "violetRed4")
    self.cd.place(x=20, y=50, width=50, height=20)
    self.cd.configure(font=("Helvetica",11))
    self.ent11 = Entry(self.edit)
    self.ent11.place(x=140, y=50, width=250, height=20)
    #-----
    self.name = Label(self.edit, text="Nombre",bg='aquamarine2',fg = "violetRed4")
    self.name.place(x=20, y=85, width=50, height=20)
    self.name.configure(font=("Helvetica",11))
    self.ent22 = Entry(self.edit)
    self.ent22.place(x=140, y=85, width=250, height=20)
    #-----
    self.pre = Label(self.edit, text="Pre Requisito",bg='aquamarine2',fg = "violetRed4")
    self.pre.place(x=20, y=125, width=90, height=20)
    self.pre.configure(font=("Helvetica",11))
    self.ent33 = Entry(self.edit)
```

• Conte

este es el ultimo metodo de la clase menuU que hace uso de la libreria Tkinter para mostrar una ventana al usuario , esta conectada con el comando del boton conteo de creditos del metodo __init__ su funcion es mostrar en pantalla los creditos de los cursos aprobados pendientes y cursados de los semestres que se ingresen al sistema , a su vez este muestra en pantalla los cursos obligatorios de un rango de semestres al igual que muestra los creditos totales de cada uno de los semestres

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

los metodos siguientes tienen la misma funcion de hacer que la parte logica del programa funcione atravez de los parametros de las diferentes clases, tomando valores de las diferentes entradas , de la ruta del archivo , etc

- eliof
- ent
- entradas
- sepa
- coma1
- coma2
- browseFiles

```
def sepa(self):
    scanner.coint1(poo=self.crec1,pol=self.crep1,pou=self.cre1)
def coma1(self):
    scanner.boton1(pool=self.sem2,sem1=self.se)
```

los diferentes metodos a continuacion su funcion es la misma llamar al metodo que contiene algun menu del programa ocultando y /o destrullendo el metodo donde se encontraba la anterior ventana

- ele
- el1
- eles
- elem
- elemt
- coca

```
def elem(self):
    self.agr.destroy()
    self.func2()
```

```
def coca(self):
    self.menu2.destroy()
    self.__init__()
```

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

logica

En este modulo es donde esta toda la logica y el funcionamiento que hace que el el modulo frame funcione , cuenta con una unica clase llamada esqueleto, tiene 9 metodos ademas de usar la libreria Tkinter para mostrar mensajes de error, advertencia y/o informacion

```
from tkinter import END, messagebox

class Cuerpo :
    global lis
    lis=[]
    global litt

    def prueba(self,mitad): ...

    def optener(self,n1,n2,n3,n4,n5,n6,n7): ...

    def edi(self,n11,n22,n33,n44,n55,n66,n77): ...

    def most(self,no,muu,mee,maa,mii,paa,pee): ...

    def eliC (self,elw): ...

    def impresion(self,explorar): ...

    def coint1(self,poo,pol,pou): ...

    def boton1(self,sem11,poo1): ...

    def boton2(self,sem12,poo2): ...

Cuerpo()
```

Clase Esqueleto

la clase Cuerpo de este modulo consta de dos variables globales y una lista la cual es la lista principal que el programa maneja para la todos los metodos de esta clase

los metodos que contiene esta clase son :

- Impresion

este metodo tiene como parametro principal explorar el cual al iniciar el programa y abrir el archivo con extencion .csv contendra la ruta de este , cuenta con un *try* el cual hace que habra el archivo atraves del parametro mostrar tomandolo solo para leer con *"r"* y leyendo este linea por linea usando un *while* ,este archivo se parsea separandolo por cada *split ","* para al final incertarlo en una lista la cual es global *lis* , tambien cuenta con un *for* el cual su funcion es el leer cada elemento de la lista y comparar si existe un elemento repetido en la posicion *[i][0]* (el codigo del curso).Asi este borrarse de la lista ya que no se permiten cursos con codigo repetido mostrando al final un mensaje de informacion que el archivo se cargo al sistema correctamente, una vez optenida la informacion se cierra el archivo . Este metodo consta tambien de dos *except* que muestran un mensaje de error si el archivo que se desea abrir es de una extencion distinta o si no se logra abrir un archivo

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

```
def impresion(self,explorar):
    try:
        archivo = open(explorar, 'r',encoding = "utf-8" )
        contenido = archivo.readline()
        while contenido != '':
            contenido = archivo.readline()
            tmp = contenido.split(",")
            lis.append(tmp)
            for p in lis:
                O=1
                for i in range(len(lis)):
                    if lis[i][0]==p[0]:
                        if O==2:
                            lis.remove(p)
                        O+=1
            messagebox.showinfo("cargar archivo", "ARCHIVO CARGADO CORRECTAMENTE")
            archivo.close()
    except UnicodeDecodeError:
        messagebox.showerror("Error", "el archivo no es compatible")
    except FileNotFoundError:
        messagebox.showerror("ERROR", "no se encontro el archivo")
```

prueba

Este metodo es que te tiene el funcionamiento de la tabla que se encuentra en el metodo *listar* de la *clase menuU*, contiene un *try* con un *if* dentro de el el cual si la lista principal *lis* se encuentra vacia(que no existe ningun archivo cargado al programa), los cursos que se agregen de forma manual se muestren antes de los cursos cargados atraves del archivo *.csv* a la tabla del *metodo listar* del modulo *frame*, si esto no es cierto y se tienen los datos del archivo cargado en la lista principal se incertaran en la tabla primero asi los cursos que se agregen de forma manual luego se agregaran al final de la tablita , el metodo recorre cada elemento de la lista con un *for* tomando de rango la lista principal *lis*, recorriendo desde *0* hasta *6* para incertar los datos de manera ordenada en las columnas correspondientes . si no existe ningun curso cargado o agregado de forma manual en el programa este mostrara un *mensaje de error* atraves del *except* de que no se encontro curso alguno

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

```
def prueba(self, mitad):
    try:
        if lis == []:
            lis.append(self.litt)
            cantid = len(lis)
            for i in range(cantid):
                mitad.insert("", END, text=lis[i][0], values=(lis[i][1], lis[i][2], lis[i][3], lis[i][4], lis[i][5], lis[i][6]))
        else:
            cantid = len(lis)
            for i in range(cantid-1):
                mitad.insert("", END, text=lis[i][0], values=(lis[i][1], lis[i][2], lis[i][3], lis[i][4], lis[i][5], lis[i][6]))
    except AttributeError:
        messagebox.showerror("ERROR", "no se encontro ningun curso")
```

• **optener**

El presente metodo tiene la funcion de obtener y guardar los datos del *metodo agrgr*, este metodo consta de 7 *parametros* los cuales toman el valor de las diferentes cajas de entradas del metodo agrgr atravez de *.get()* los cuales son incertados en una lista llamada *litt*, el metodo consta de varios *if* los cuales no permiten el ingreso erroneo del estado del cursos o de la opcionalidad de este, a demas de no permitir agregar algun dato si existen cajas de entradas vacias o si la la entrada del codigo esta vacia. si todo esto se cumple comienza otra *validacion*, si existen cursos antes cargados al sistema o si no existian estos antes, si los cursos ya fueron agregados al sistema el metodo incertara la lista *litt* (la cual contiene el nuevo curso) al final de la lista principal atraves de un *for* y un *if* que valida si el codigo ya existe o no, dejando el curso agregado al final de la lista principal si esta tiene cursos agregados atraves del archivo anteriormente. Si no existen cursos cargados antes el programa pasara y este curso se agregara en el metodo prueba

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

• edi

este consta de 7 parametros los cuales toman el valor de las entradas de un metodo de el modulo frame, siendo espesificamente del metodo edit. al igual que el metodo optener incerta los datos optenidos en una lista de la misma forma, contiene varios if para validar la escritura de los datos restringiendo el ingresar su estado y su opcionalidad . la diferencia de este metodo con el anterior es de que el curso debe de existir (el codigo debe de existir)para que este sea editado , de lo contrario se mostrara un mensaje de error si no existe o si aun no hay cursos en el sistema a los cuales poder editar

```
def edi(self,n11,n22,n33,n44,n55,n66,n77):
    litto=[]

    nu11=n11.get()
    nu22=n22.get()
    ...

    litto.append(nu55)
    litto.append(nu66)

    if nu11 == "":
        messagebox.showerror("ERROR", "escriba el codigo del curso ") #shi
    elif nu11!=" " and nu22=="":
        messagebox.showerror("ERROR", "llene los demas espacios")#shi
    ...

else:
    contar=0
    conteo1=len(lis)
    if lis==[]:
        messagebox.showerror('editar',"no existen cursos para editar")
        pass
    else:
        for i in range (conteo1-1):
            if nu11== lis[i][0]:
                conteo1=len(lis)
                lis.remove(lis[i])
                lis.insert(conteo1-2,litto)
                print(lis)
                messagebox.showinfo('editar',"su curso a sido editado")
                break
```

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

- **most**

la funcion de este metodo es mostrar en pantalla los datos de un curso en espesifico, consta de 7 parametros los cuales tomara el valor de los label y la entrada de el metodo mostrar de la clase menuU .Este metodo contiene varias validacion.Mientras la caja de entrada de codigo este vacia mostrara un mensaje de error, al contrario este usara un *for* para tomar de *rango* la lista principal y hacer un conteo buscando el codigo que se escribe, al igual que los datos que contiene dicho codigo hasta igualar el codigo de entrada con la lista principal con `lis[i][0]` , cuando estos sean iguales los datos se incertaran en los diferentes labels mostrando en pantalla todos o referente al codigo de curso escrito

- **eliC**

el metodo eliC tiene la funcion de obtener el codigo del curso atravez de el paremetro elw que toma el valor de la entrada de el *metodo elimini* de la *clase menuU* , contiene un *if* que muestra un mensaje si no se a ingresado el codigo del curso , de lo contrario el programa valida un *for* que tiene de rango la lista principal, en donde iguala el codigo optenido con la posicion `[i][0]` del codigo de esta ,la cual al encontrarla eliminara por completo la lista `lis[i]`. *mostrando un mensaje al haberse eliminado el curso*

```
def eliC (self,elw):
    mu=elw.get()
    cantid=len(lis)
    if mu == "":
        messagebox.showerror("ERROR", "escriba el codigo del curso ")
    else:
        for i in range(cantid-1):
            if mu==lis[i][0]:
                lis.remove(lis[i])
                messagebox.showinfo("eliminar curso", "SU CURSO SE ELIMINO CORRECTAMENTE")
                print(lis)
                break
```


- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

los siguientes 3 metodos estan conectados con el metodo conte de la clase menuU que permiten mostrar en pantalla los creditos que contienen los cursos subidos y/o agregados al programa.

- **coint1**

este metodo contiene 3 parametros(**poo,pol,pou**) los cuales tomaran el valor de los labels de el metodo contar para mostrar los creditos aprobados, cursados y pendientes del usuario. atravez de validaciones de **if** de la posicion **[i][6]** de la lista si la pocicion **lis [i][6]** es igual a **1**, los creditos se sumaran y se mostraran en creditos cursados. Si la pocicion lis **[i][6]** es igual a **-1**, los creditos se sumaran y se mostraran en creditos pendientes , si la pocicion **lis [i][6]** es igual a **0**, los creditos se sumaran y se mostraran en creditos aprobados.

```
def coint1(self,poo,pol,pou):  
  
    lita=[]  
    lito=[]  
    liu=[]  
  
    cantid=len(lis)  
    if lis==[]:  
        messagebox.showerror("ERROR", "NO EXISTEN CURSO AUN ")  
        pass  
    else:  
        for i in range(cantid-1):  
            if "1" in lis[i][6] and '-1' not in lis[i][6]:  
                lita.append(lis[i][5])  
  
                l3 = sum(int(ele) for ele in lita)  
  
                poo.configure(text=l3,font=("Helvetica",11))
```

...

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.

• Boton1

este metodo contiene 2 pametros, uno tomara el valor de seleccion del boton combobox y el restante el valor de el label para mostrar la informacion, tiene un for que hace un conteo desde 0 hasta el rango de la lista validando principalmente el valor que se le incerta al **combo box** para hacer su recorrido por esta e ir igualando el semestre elegido , asi hacer una sumatoria de todos los creditos de este .

```
def boton1(self,sem11,pool):
    sel1=sem11.get()
    litaa=[]
    cantid=len(lis)

    for i in range(cantid-1):
        if sel1 ==lis[i][4]:
            litaa.append(lis[i][5])
            l33 = sum(int(ele) for ele in litaa)
            pool.configure(text=l33,font=("Helvetica",11))
```

• Boton2

este metodo contiene 2 pametros, uno tomara el valor de seleccion del boton **combobox** y el restante el valor de el label para mostrar la informacion, tiene un **for** que hace un conteo **0** hasta el rango de la lista validando principalmente el valor que se le incerta al **combobox** para hacer su recorrido por la lista igualando el semestre elegido, la unica diferencia de el **metodo boton1** es que este valida si en la lista de cursos que se a ingresado solo se muestren los creditos de los cursos obligatorios . Agregando cada valor a una nueva lista y sumando los valores cada que se hace el conteo para asi mostrarlos en la pantalla .

- Universidad de San Carlos de Guatemala
- Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
- Lenguajes Formales y de Programación, 2er. Semestre 2022.



GENESIS NAHOMI APARICIO ACAN

carne :20211329