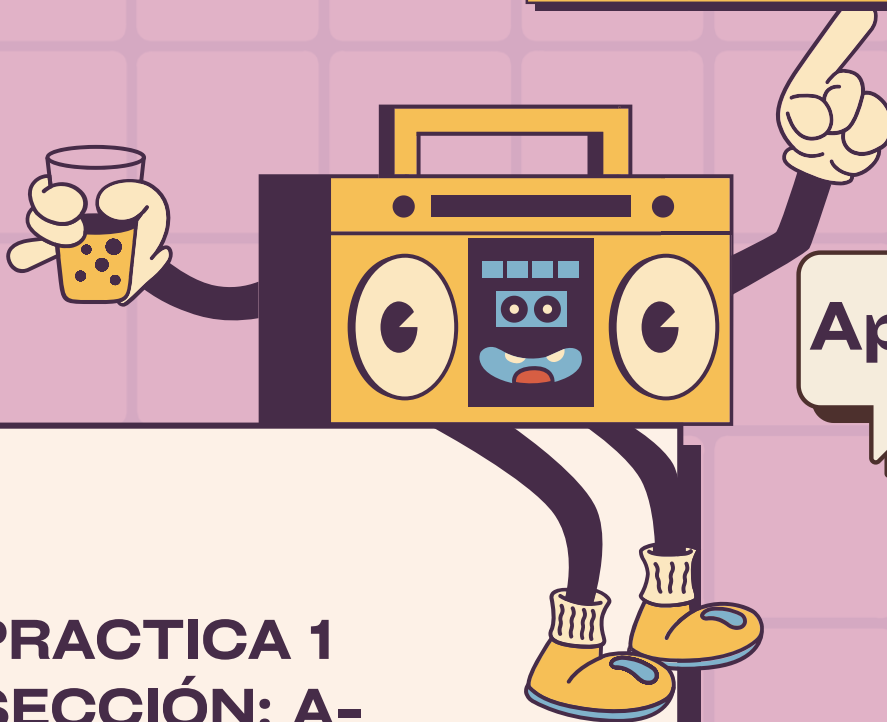


MANUAL TECNICO

Lenguajes Formales
y de Programación



App-Courses

PRACTICA 1
SECCIÓN: A-
CARNET: 201900647
NOMBRE: EDUARDO
JOSUÉ GONZÁLEZ
CIFUENTES

INDICE

INTRODUCCIÓN.....	1
OBJETIVOS	1
DIRIGIDO	2
ESPECIFICACIÓN TÉCNICA	2
LÓGICA DEL PROGRAMA	3

I. Introducción

La aplicación App-Courses es un programa que sirve para analizar archivos de tipo lfp, el cual fue desarrollada con Python para ser específico la versión 3.10.5, mas adelante se explicara como funcionan los diferentes métodos que la componen de una manera lógica.

II. Objetivos

- Dar a conocer las funciones a nivel interno del programa para que a otros desarrolladores se les haga fácil el entender el funcionamiento.
- Así mismo dar a una herramienta más para el análisis de información de un archivo lfp y el poder manejar de la mejor manera la información analizada,

III. Dirigido

Este manual está dirigido para todos los desarrolladores que estén interesados en cómo se desarrolló ésta aplicación.

IV. Especificación Técnica

1.Requisitos de Hardware

- Computadora de escritorio o portátil.
- Mínimo 4gb de RAM.
- i-3 de 4ta generación en adelante.
- Mínimo 10 GB de almacenamiento en el Disco Duro.

2.Requisitos de Software

- Windows 7 o superior.
- Python 3.10.5
- Visual Code

V. Lógica del programa

Clase curso

En ésta clar hace que se guarden los objetos de tipo curso

```
class Curso:
    def __init__(self, idCourse, NameCourse, PreRequisite, Optionality, Semester, Credits, Status):
        self.idCourse = idCourse
        self.NameCourse = NameCourse
        self.PreRequisite = PreRequisite
        self.Optionality = Optionality
        self.Semester = Semester
        self.Credits = Credits
        self.Status = Status
        self.Counter=0

    def __repr__(self):
        return str(self.__dict__)

    def getIdCourse(self):
        return self.idCourse
    def setIdCourse(self, idCourse):
        self.idCourse=idCourse

    def getNameCourse(self):
        return self.NameCourse
    def setNameCourse(self, NameCourse):
        self.NameCourse=NameCourse

    def getPreRequisite(self):
        return self.PreRequisite
    def setPreRequisite(self, PreRequisite):
        self.PreRequisite=PreRequisite

    def getOptionality(self):
        return self.Optionality
    def setOptionality(self, Optionality):
        self.Optionality=Optionality

    def getSemester(self):
        return self.Semester
    def setSemester(self, Semester):
        self.Semester=Semester

    def getCredits(self):
        return self.Credits
    def setCredits(self, Credits):
        self.Credits=Credits

    def getStatus(self):
        return self.Status
    def setStatus(self, Status):
        self.Status=Status
```

Método readFile

Este método sirve para poder analizar el archivo lfp y poder guardar en una lista de objetos cada curso.

```
def readFile(txt):
    global ContaCursos
    global ListaCursos
    print("xd",txt)
    if txt == "":
        mensaje=messagebox.showwarning("Archivos","No se ha cargado ningún archivo")
        return mensaje
    else:
        partsOfnameFile=txt.split('.')
        print(partsOfnameFile)
        print("longitud",len(partsOfnameFile))
        pos1=partsOfnameFile[0]
        if os.path.exists(str(pos1)+".lfp"):
            if len(partsOfnameFile)!=2:
                messagebox.showwarning("Archivos","Ingresa un archivo correcto")
            else:
                if partsOfnameFile[1] == 'lfp':
                    file=""
                    if file=="":
                        file =open(txt,"r", encoding="utf-8")
                        line=file.read()
                        file.close()
                        #print(line)
                        print("")
                        newLine = line.split("\n")
                        for c in newLine:
                            if c != "":
                                niuline= c.split(",")
                                idcourseaux=0
                                namecourseaux=""
                                prerequisiteaux=0
                                optinalityaux=0
                                semesteraux=0
                                creditsaux=0
                                statusaux=0
                                idcourseaux=niuline[0]
                                namecourseaux = niuline[1]
                                prerequisiteaux = niuline[2]
                                optinalityaux =niuline[3]
                                semesteraux=niuline[4]
                                creditsaux =niuline[5]
                                statusaux = niuline[6]
                                if(checkCourse(idcourseaux)==False):
                                    Cursonew=Curso(idcourseaux,namecourseaux,
                                    prerequisiteaux,optinalityaux,semesteraux,creditsaux,statusaux)
                                    Cursonew.Counter=ContaCursos
                                    ListaCursos.append(Cursonew)
                                    ContaCursos+=1
                                print(ListaCursos)
                                for r in ListaCursos:
                                    print(
                                        "
                                        print("Curso No. ", str(r.Counter))
                                        print(" ID Curso: " +r.idCourse + " |Curso: "+r.NameCourse+
                                        " |PreRequisito: " +r.PreRequisite+" |Opcionalidad: " +r.Optionality+ " |Semestre: " + r
                                        .Semester + " |Creditos: " + r.Credits + " |Estados: " + r.Status)
                                        print("Hay " + str(ContaCursos) + " Cursos")
                                        print("")
                                        print("")
                                        messagebox.showinfo("Archivos","Archivos Cargados")
                                else:
                                    messagebox.showinfo("Archivos","Ya Cargo un archivo")
                                else:
                                    messagebox.showwarning("Archivos","Ingresa un archivo correcto")
                                    print("Ingresa un archivo correcto :( ")
                                else:
                                    messagebox.showwarning("Archivos","Ingresa un archivo correcto")
```

Métodos para la interfaz gráfica

Éste es un ejemplo del método que se utilizó para crear la primer ventana de la interfaz gráfica.

```
def window_mainMenu():
    global wndw_menu
    #Configuración ventana menu principal
    wndw_menu = tk.Tk()
    wndw_menu.title("App-Courses")
    wndw_menu.resizable(0,0)
    wndw_menu.iconbitmap("img/menu.ico")
    wndw_menu.geometry("720x480")
    wndw_menu.config(bg="#48F1D8")
    wndw_menu.config(bd=30)
    wndw_menu.config(relief="groove")
    #Configuraciones Labels
    labelNameCourse=Label(wndw_menu,text="Curso: Lenguajes Formales y de Programación",
fg="black",font=("Courier 15 bold"),bg="#48F1D8")
    labelNameCourse.place(x=80,y=20)

    labelNameStudent=Label(wndw_menu,text="Nombre: Eduardo Josué González Cifuentes",fg=
"black",font=("Courier 15 bold"),bg="#48F1D8")
    labelNameStudent.place(x=80,y=45)

    labelNameidstudent=Label(wndw_menu,text="Carnet: 201900647      Sección: A- ",fg=
"black",font=("Courier 15 bold"),bg="#48F1D8")
    labelNameidstudent.place(x=80,y=70)
    #Configuración Botones
    buttonFileUpload=Button(wndw_menu,text="Cargar Archivo",width=20,bg="#48F1D8", font=
("Courier 15 bold"),relief="groove",bd=8,command=window_fileupload)
    buttonFileUpload.place(x=220,y=120)

    buttonManageCourse=Button(wndw_menu,text="Gestionar Cursos",width=20,bg="#48F1D8",
font=("Courier 15 bold"),relief="groove",bd=8,command=goToManageMenuFromMain)
    buttonManageCourse.place(x=220,y=180)

    buttonCreditCount=Button(wndw_menu,text="Conteo de Créditos",width=20,bg="#48F1D8",
font=("Courier 15 bold"),relief="groove",bd=8,command=window_creditcount)
    buttonCreditCount.place(x=220,y=240)

    buttonMainMenuOut=Button(wndw_menu,text="Salir",width=20,bg="#48F1D8", font=(
"Courier 15 bold"),relief="groove",bd=8,command=exit)
    buttonMainMenuOut.place(x=220,y=300)

    wndw_menu.mainloop()
```

Los siguientes métodos se realizaron de la misma manera que el método para la pantalla principal de la aplicación.

```
> def window_fileupload(): ...
```

```
> def window_managecourse(): ...
```

```
> def window_listcouse(): ...
```

```
> def window_showcourse(): ...
```

```
> def window_addcourse(): ...
```

```
> def window_editcourse(): ...
```

```
> def window_deleteCourse(): ...
```

```
> def window_creditcount(): ...
```

Métodos para la funcionalidad de los botones

Éstos métodos sirvieron para el uso de los botones dentro de la interfaz gráfica, para poder regresar a otra ventana y para poder hacer funcionar métodos lógicos.


```
def getBackMainMenuFromManage():  
    global wndw_managecourse  
    wndw_managecourse.destroy()  
    window_mainMenu()
```

Los demás métodos de los botones fueron realizados de la misma manera que el anterior.

```
> def getBackMainMenuFromFileUpload(): ...  
  
> def getBackMainMenuFromCreditCount(): ...  
  
> def getBackManageCourseFromListCourse(): ...  
  
> def getBackManageCourseFromShowCourse(): ...  
  
> def getBackManageCourseFromAddCourse(): ...  
  
> def getBackManageCourseFromEditCourse(): ...  
  
> def getBackManageCourseFromDeleteCourse(): ...  
  
> def goToManageMenuFromMain(): ...
```

Método showCourse

Éste método se utilizó para poder mostrar realizar una búsqueda por medio del id del curso y mostrarlo con sus atributos.

```
def showCourse(idgg):
    global textboxCourseName
    global textboxCoursePreRequis
    global textboxCourseSemester
    global textboxCourseOptionality
    global textboxCourseCredits
    global textboxCourseStatus
    foundd = False
    for e in ListaCursos:
        if idgg==e.idCourse:
            textboxCourseName.configure(text=e.NameCourse)
            textboxCoursePreRequis.configure(text=e.PreRequisite)
            textboxCourseSemester.configure(text=e.Semester)
            textboxCourseOptionality.configure(text=e.Optionality)
            textboxCourseCredits.configure(text=e.Credits)
            textboxCourseStatus.configure(text=e.Status)
            foundd=True
            print(" ID Curso: " + e.idCourse + " |Curso: "+e.NameCourse+
" |PreRequisito: " + e.PreRequisite+" |Opcionalidad: " + e.Optionality+ " |Semestre: " + e
.Semester + " |Creditos: " + e.Credits + " |Estados: " + e.Status)
            if (foundd==True):
                messagebox.showinfo("Curso", "Curso Encontrado :)")
            else:
                messagebox.showinfo("Curso", "Curso No Encontrado :'v'")
```

Método addCourse

Este método se utilizó para poder agregar un nuevo curso a la lista de cursos.

```
def addCourse(niuIdCourse):
    global ContaCursos
    global ListaCursos
    addgg = False
    global textboxCourseidadd
    global textboxCourseNameadd
    global textboxCoursePreRequisadd
    global textboxCourseSemesteradd
    global textboxCourseOptionalityadd
    global textboxCourseCreditsadd
    global textboxCourseStatusadd
    if( (textboxCourseidadd.get() and textboxCourseNameadd.get() and
textboxCourseSemesteradd.get() and textboxCourseOptionalityadd.get() and
textboxCourseCreditsadd.get() and textboxCourseStatusadd.get()) != ""):
        if checkCourse(textboxCourseidadd.get())==False:
            niuCourse= Curso(textboxCourseidadd.get(),textboxCourseNameadd.get(),
textboxCoursePreRequisadd.get(),textboxCourseOptionalityadd.get(),
textboxCourseSemesteradd.get(),textboxCourseCreditsadd.get(),textboxCourseStatusadd.get
())
            ListaCursos.append(niuCourse)
            ContaCursos+=1
            addgg=True
            for e in ListaCursos:
                print(" ID Curso: " +str(e.idCourse) + " |Curso: "+str(e.NameCourse)+
" |PreRequisito: " +str(e.PreRequisite)+" |Opcionalidad: " +str(e.Optionality)+
" |Semestre: " + str(e.Semester )+ " |Creditos: " + str(e.Credits) + " |Estados: " +str
(e.Status))
            print("Cursos ahora: ",ContaCursos)
        else:
            pass
    else:
        messagebox.showinfo("Curso", "Los Campos están vacios")
```

Método checkCourse

Este método se utilizó para validar si un curso existe por medio de su id.

```
def checkCourse(idgg):  
    global ListaCursos  
    for e in ListaCursos:  
        if(e.idCourse==idgg):  
            return True  
    return False
```

Método editCourse

Este método se utilizó para poder editar un curso que esté en la lista.

```
def editCourse(idxd):  
    global ListaCursos  
    editgg = False  
    global textboxEditCourseided  
    global textboxEditCourseameed  
    global textboxEditCoursePreRequised  
    global textboxEditCourseSemestered  
    global textboxEditCourseOptionaltyed  
    global textboxEditCourseCreditsed  
    global textboxEditCourseStatused  
    if( (textboxEditCourseided.get() and textboxEditCourseameed.get() and  
        textboxEditCourseSemestered.get() and textboxEditCourseOptionaltyed.get() and  
        textboxEditCourseCreditsed.get() and textboxEditCourseStatused.get()) != ""):  
        if (checkCourse(textboxEditCourseided.get())==True):  
            for r in ListaCursos:  
                if(str(r.idCourse)==textboxEditCourseided.get()):  
                    r.idCourse = textboxEditCourseided.get()  
                    r.NameCourse = textboxEditCourseameed.get()  
                    r.PreRequisite = textboxEditCoursePreRequised.get()  
                    r.Optionality = textboxEditCourseOptionaltyed.get()  
                    r.Semester = textboxEditCourseSemestered.get()  
                    r.Credits = textboxEditCourseCreditsed.get()  
                    r.Status = textboxEditCourseStatused.get()  
                    editgg=True  
            else:  
                pass  
        else:  
            messagebox.showinfo("Curso", "Los Campos están vacíos")  
  
    if (editgg==True):  
        messagebox.showinfo("Curso", "Curso Editado :)")  
    else:  
        messagebox.showinfo("Curso", "Ingrese un id válido")
```

Método deleteCourse

Éste método se utilizó para poder eliminar un curso que esté en la lista.

```
def deleteCourse(idd):
    global ListaCursos
    global ContaCursos
    global textboxidCourseDel
    dele = False
    if (checkCourse(textboxidCourseDel.get())==True
    ):
        for i in ListaCursos:
            if(i.idCourse==str(textboxidCourseDel.
            get())):
                ListaCursos.remove(i)
                ContaCursos-=1
                dele=True
                print("Cursos Ahora: ",ContaCursos)
            else:
                pass
        if (dele==True):
            messagebox.showinfo("Curso",
            "Curso Eliminado :)")
        else:
            messagebox.showinfo("Curso",
            "Ingrese un id válido")
```

Método CrediCountNsemester

Éste método se utilizó para hacer la suma de los créditos tanto aprobados, pendientes y cursando de un semestre N.

Método CredicountNSemesterObligatory

Éste método se utilizó para hacer la sumatoria de los créditos de los cursos obligatorios hasta semestre N en pocas palabras la suma acumulada hasta semestre N. (se puso un pantallazo del principio del código ya que tiene la misma lógica pero es un código muy extenso),

```
def CredicountNSemesterObligatory():
    global semester
    global sendDataCredN
    global ListaCursos
    sumObligatory1=0
    sumObligatory2=0
    sumObligatory3=0
    sumObligatory4=0
    sumObligatory5=0
    sumObligatory6=0
    sumObligatory7=0
    sumObligatory8=0
    sumObligatory9=0
    sumObligatory10=0

    if(semester.get()=="1"):
        for e in ListaCursos:
            if(e.Semester=="1"):
                if(e.Optionality=="1"):
                    credit =int(e.Credits)
                    sumObligatory1=sumObligatory1+credit
                    sendDataCredN.set(sumObligatory1)
                print("")
                print("Los créditos hasta el semestre " + semester.get() + " son: " + str(
sumObligatory1))
                sumObligatory1=0
            elif(semester.get()=="2"):
                for e in ListaCursos:
                    if(e.Semester=="1"):
                        if(e.Optionality=="1"):
                            credit =int(e.Credits)
                            sumObligatory1=sumObligatory1+credit
                        elif(e.Semester=="2"):
                            if(e.Optionality=="1"):
                                credit =int(e.Credits)
                                sumObligatory2=sumObligatory2+credit
                            tot=sumObligatory1+sumObligatory2
                            sendDataCredN.set(tot)
                        print("")
                        print("Semestre 1: " + str(sumObligatory1))
                        print("Semestre 2: " + str(sumObligatory2))
                        print("Los créditos hasta el semestre " + semester.get() + " son: " + str(tot
                    ))
                sumObligatory1=0
                sumObligatory2=0
            elif(semester.get()=="3"):
                for e in ListaCursos:
                    if(e.Semester=="1"):
                        if(e.Optionality=="1"):
                            credit =int(e.Credits)
                            sumObligatory1=sumObligatory1+credit
                        elif(e.Semester=="2"):
                            if(e.Optionality=="1"):
                                credit =int(e.Credits)
                                sumObligatory2=sumObligatory2+credit
                        elif(e.Semester=="3"):
                            if(e.Optionality=="1"):
                                credit =int(e.Credits)
                                sumObligatory3=sumObligatory3+credit
                            tot=sumObligatory1+sumObligatory2+sumObligatory3
                            sendDataCredN.set(tot)
                        print("")
                        print("Semestre 1: " + str(sumObligatory1))
                        print("Semestre 2: " + str(sumObligatory2))
                        print("Semestre 3: " + str(sumObligatory3))
                        print("Los créditos hasta el semestre " + semester.get() + " son: " + str(tot
                    ))
                sumObligatory1=0
                sumObligatory2=0
                sumObligatory3=0
```