

Universidad Rafael Landívar
Facultad de Ingeniería
Laboratorio de Pensamiento Computacional, Sección 02
Docente: Ing. Luis Pedro Ovalle

PROYECTO No. 02 – B
“PROGRAMA FUNCIONAL”

Beltran Tello, Carlos Rodrigo – 1002324

Guatemala, 14 de abril de 2024

Codificación del Programa

```
#IMPORTAR LIBRERIAS
import re #Regex
from datetime import datetime #Datetime

#INICIALIZAR VARIABLES

#--Crear Variables Globales--
def inicializador():
    global opcion, opcion_reporte, opcion_reporte, opcion_alumno, id_curso,
    nombre_curso, horario_curso, salon_curso, catedratico_curso, notaAgregarCurso,
    totalAlumnos_curso, notaAcumulada, cursos, alumnos_curso, carne_alumno,
    nombre_alumno
    global fechaNacimiento_alumno, notaAlumno_curso, alumnos, notas_Alumno,
    cursosAlumno, notasAlumno, media_curso, estadoid_CursoCreado,
    estadoid_CursoEditado, estado_carneAlumnoCreado, estado_carneAlumnoEditado,
    estado_fechaAlumnoCreado
    global estado_fechaAlumnoEditado, estado_idCursoBuscado,
    estado_CarneAlumnoBuscado, estado_notaAlumno, estado_Opcionb, estado_Opcionc,
    estado_InformacionCurso, estado_InformacionAlumno, i, j, k, l, m, n, p, q, r, s

# -- Opciones --
opcion = ''
opcion_reporte = ''
opcion_alumno = ''

#--Cursos--
id_curso = ""
nombre_curso = ""
horario_curso = ""
salon_curso = ""
catedratico_curso = ""
notaAgregarCurso = 0.0
totalAlumnos_curso = 0
notaAcumulada = 0.0
cursos = [["1", "Pensamiento Computacional (Práctica)", "9:50 - 11:40", "T-203",
"Ing. Luis Pedro Ovalle"],
          ["2", "Variable Compleja", "8:30 - 9:50", "D-15", "Dr. Mario
Ranferi"]]

alumnos_curso = [["1", "Pensamiento Computacional (Práctica)", "1", "38.75",
["1125022"], ["Maria Andreé Lopez Lozano"], ["38.75"]],
                  ["2", "Variable Compleja", "2", "160.00", ["1002324",
"1125022"], ["Carlos Rodrigo Beltran Tello", "Maria Andreé Lopez Lozano"],
["80.00", "80.00"]]] #Se inicializa el array para un curso
```

```

#--Alumnos--
carne_alumno = ""
nombre_alumno = ""
fechaNacimiento_alumno = datetime.now() #Se inicializa la variable con la fecha
de hoy
notaAlumno_curso = 0.0
alumnos = [["1002324", "Carlos Rodrigo Beltran Tello", "01-11-2005"], #Se
inicializa el arrya con 2 alumnos
            ["1125022", "Maria Andreé Lopez Lozano", "22-07-2003"]]
notas_Alumno = [["1002324", ["2"], ["Variable Compleja"], ["80.00"]],
                ["1125022", ["1", "2"], ["Pensamiento Computacional (Práctica)",
"Variable Compleja"], ["39.75", "80.00"]]] #Se inicializa el array
cursosAlumno = []
notasAlumno = []

#--Reportes--
media_curso = 0.0

#Verificaciones
estadoid_CursoCreado = True
estadoid_CursoEditado = True
estado_carneAlumnoCreado = True
estado_carneAlumnoEditado = True
estado_fechaAlumnoCreado = True
estado_fechaAlumnoEditado = True
estado_idCursoBuscado = True
estado_CarneAlumnoBuscado = True
estado_notaAlumno = True
estado_Opcionb = True
estado_Opcionc = True
estado_InformacionCurso = True
estado_InformacionAlumno = True

#--Contadores--
i = 0
j = 0
k = 0
l = 0
m = 0
n = 0
p = 0
q = 0
r = 0
s = 0

```

#FUNCIONES

--Opción 1 (Crear Curso)--

```
def CrearCurso ():

    print("\n¡CREAR CURSO!")

    #Inicializar Variables
    estadoid_CursoCreado = True
    estado_InformacionCurso = True
    id_curso = ""
    nombre_curso = ""
    horario_curso = ""
    salon_curso = ""
    catedratico_curso = ""
    i = 0
    j = 0
    k = 0

    print("\nCURSOS EXISTENTES")
    print("| ID | | Nombre | | Horario | | Salón | | Catedrático |")

    cursosOrdenado = sorted(cursos, key = lambda x: x[0])

    #Mostrar Cursos Existentes
    for i in range (len(cursosOrdenado)):
        textoImprimir = ""

        for j in range(len(cursosOrdenado[0])):
            textoImprimir += "|" + cursosOrdenado[i][j] + " | "

        print(textoImprimir)

    #Verificar el ID del Curso
    while estadoid_CursoCreado:

        id_curso = input("\nIngrese el ID del Curso a Crear -> ")

        for k in range(0, len(cursos)):
            if(id_curso == cursos[k][0]):
                print("EROR. Ya existe un curso con el ID ingresado.")
                estadoid_CursoCreado = True
                break
            else:
```

```

        estadoid_CursoCreado = False

#Verificar la Información del Curso
while estado_InformacionCurso:
    nombre_curso = input("Ingrese el nombre del Curso -> ")
    horario_curso = input("Ingrese el horario del Curso -> ")
    salon_curso = input("Ingrese el salón donde se impartirá
el Curso -> ")
    catedratico_curso = input("Ingrese el nombre del
catedrático que impartirá el curso -> ")

    if ((re.fullmatch(r"[A-Za-zÑñÁáÉíÓóÚú¿?() ]{1,500}",
nombre_curso))):

        #Agregar Curso en las Matrices Correspondientes
        cursos.append([id_curso, nombre_curso, horario_curso,
salon_curso, catedratico_curso])
        alumnos_curso.append([id_curso, nombre_curso, "0",
"0.0", [], [], []])

        print("CURSO CREADO")

        estado_InformacionCurso = False
    else:
        print("Los datos ingresados no cumplen el formato
correcto. Intente de Nuevo.")

#--Opcion 2 (Editar Curso)--
def EditarCurso ():

    print("\n|EDITAR CURSO!")

    #Inicializar Variables
    estadoid_CursoEditado = True
    estado_InformacionCurso = True
    id_curso = ""
    nombre_curso = ""
    horario_curso = ""
    salon_curso = ""
    catedratico_curso = ""
    textoImprimir = ""
    id_cursosAlumno = [] #Variable Local
    nombres_cursosAlumno = [] #Variable Local
    i = 0
    j = 0

```

```

k = 0

print("\nCURSOS EXISTENTES")
print("| ID | | Nombre | | Horario | | Salón | | Catedrático |")

cursosOrdenado = sorted(cursos, key = lambda x: x[0])

#Mostrar Cursos Existentes
for i in range (len(cursosOrdenado)):
    textoImprimir = ""

    for j in range(len(cursosOrdenado[0])):
        textoImprimir += "| " + cursosOrdenado[i][j] + " | "

    print(textoImprimir)

#Verificar ID del Curso
while estadoid_CursoEditado:

    id_curso = input("\nIngrese el ID del Curso a Editar -> ")

    for k in range(len(cursos)):

        if (cursos[k][0] == id_curso):

            #Verificar Información del Curso
            while estado_InformacionCurso:

                print("\nINFORMACIÓN A EDITAR")
                print()

                nombre_curso = input("Ingrese el nombre del Curso -> ")
                horario_curso = input("Ingrese el horario del Curso -> ")
                salon_curso = input("Ingrese el salón donde se impartirá
el Curso -> ")

                catedratico_curso = input("Ingrese el nombre del
catedrático que impartirá el curso -> ")

                if ((re.fullmatch(r"[A-Za-zÑñÁáÉíÓóÚú?() ]{1,100}",
nombre_curso))):

                    cursos[k][1] = nombre_curso
                    cursos[k][2] = horario_curso
                    cursos[k][3] = salon_curso
                    cursos[k][4] = catedratico_curso

```

```

        alumnos_curso[k][1] = nombre_curso

        for l in range(len(notas_Alumno)):
            id_cursosAlumno = notas_Alumno[l][1]
            nombres_cursosAlumno = notas_Alumno[l][2]
            for m in range (len(id_cursosAlumno)):
                if (id_curso == id_cursosAlumno[m]):
                    nombres_cursosAlumno[m] = nombre_curso

            notas_Alumno[l][1] = id_cursosAlumno
            notas_Alumno[l][2] = nombres_cursosAlumno

        print("CURSO EDITADO")

        estado_InformacionCurso = False
        estadoid_CursoEditado = False
    else:
        print("Los datos ingresados no cumplen el formato
correcto. Intente de Nuevo.")

        if (estadoid_CursoEditado):
            print("El curso no Existe. Inténtelo de Nuevo.")

#--Opcion 3 (Control de Alumnos)--
def ControlAlumnos ():

    print("\n¡CONTROL ALUMNOS!")

    #Inicializar Variables
    opcion_alumno = ""
    estado_carneAlumnoCreado = True
    estado_carneAlumnoEditado = True
    estado_fechaAlumnoCreado = True
    estado_fechaAlumnoEditado = True
    estado_CarneAlumnoBuscado = True
    estado_InformacionAlumno = True
    carne_alumno = ""
    nombre_alumno = ""
    textoImprimir = ""
    carnesCurso_Edicion = []
    alumnosCurso_Edicion = []
    fechaNacimiento_alumno = datetime.now()
    i = 0
    j = 0

```

```

k = 0

opcion_alumno = input("\n¿Desea Crear o Editar un Alumno? C: Crear / E:
Editar -> ").upper()

#Crear Alumno
if(opcion_alumno == 'C'):

    print("\nALUMNOS EXISTENTES")
    print("| Carné | | Nombre | | Fecha de Nacimiento |")

    #Mostrar Alumnos Existentes
    for i in range (len(alumnos)):
        textoImprimir = ""

        for j in range(len(alumnos[0])):
            textoImprimir += "|" + alumnos[i][j] + " | "

        print(textoImprimir)

    #Verificar el Carné del Alumno
    while estado_carneAlumnoCreado:

        carne_alumno = input("\nIngrese el Carné del Alumno a Crear -> ")

        for i in range(0, len(alumnos)):

            if(carne_alumno == alumnos[i][0]):
                print("EROR. Ya existe un alumno con el Carné
ingresado.")

                estado_carneAlumnoCreado= True
                break
            else:
                estado_carneAlumnoCreado = False

    #Verificar la Información del Alumno
    while estado_InformacionAlumno:

        nombre_alumno = input("Ingrese el nombre del Alumno -> ")

        if ((re.fullmatch(r"[A-Za-zÑáéíóúÁÉÍÓÚ ]{1,100}",
nombre_alumno))):

            while estado_fechaAlumnoCreado:

```



```

        fechaNacimiento_alumno = input("Ingrese la fecha
de nacimiento del alumno (dd/mm/aaaa) -> ")

        if (datetime.strptime(fechaNacimiento_alumno,
'%d/%m/%Y')):

            if (datetime.strptime(fechaNacimiento_alumno,
'%d/%m/%Y') < datetime.now()):

                fechaNacimiento_alumno =
datetime.strptime(fechaNacimiento_alumno, '%d/%m/%Y').strftime('%d-%m-%Y')
                estado_fechaAlumnoCreado = False
                estado_InformacionAlumno = False
            else:
                print("Ingrese una fecha Correcta.")
            else:
                print("Ingrese la fecha de Nacimiento del
Alumno en el formato indicado.")

        alumnos.append([carne_alumno, nombre_alumno,
fechaNacimiento_alumno])
        notas_Alumno.append([carne_alumno, [], [], []])

        print("\nALUMNO CREADO")

    else:
        print("El nombre ingresado no cumple con el formato
correcto. Intente de Nuevo.")

#Editar Alumno
elif(opcion_alumno == 'E'):
    print("\nALUMNOS EXISTENTES")
    print("| Carné | | Nombre | | Fecha de Nacimiento |")

#Mostrar Alumnos Existentes
for i in range (len(alumnos)):
    textoImprimir = ""

    for j in range(len(alumnos[0])):
        textoImprimir += "| " + alumnos[i][j] + " | "

    print(textoImprimir)

#Verificar Carné del Alumno
while estado_carneAlumnoEditado:

```

```

carne_alumno = input("\nIngrese el Carné del Alumno a Editar -> ")

for k in range(len(alumnos)):

    if (alumnos[k][0] == carne_alumno):

        #Verificar Información del Curso
        while estado_CarneAlumnoBuscado:

            print("\nINFORMACIÓN A EDITAR")
            print()

            nombre_alumno = input("Ingrese el nombre del Alumno -> ")

            if ((re.fullmatch(r"[A-Za-zÑÁÉÍÓÚÁÉÍÓÚ ]{1,100}",
nombre_alumno))):

                #Verificar la Fecha de Nacimiento del Alumno
                while estado_fechaAlumnoEditado:

                    fechaNacimiento_alumno = input("Ingrese la fecha
de nacimiento del alumno (dd/mm/aaaa) -> ")

                    if (datetime.strptime(fechaNacimiento_alumno,
'%d/%m/%Y')):

                        if (datetime.strptime(fechaNacimiento_alumno,
'%d/%m/%Y') < datetime.now()):

                            fechaNacimiento_alumno =
datetime.strptime(fechaNacimiento_alumno, '%d/%m/%Y').strftime('%d-%m-%Y')

                            alumnos[k][1] = nombre_alumno
                            alumnos[k][2] = fechaNacimiento_alumno

                            estado_fechaAlumnoEditado = False
                            estado_carneAlumnoEditado = False
                            estado_CarneAlumnoBuscado = False
                            estado_InformacionAlumno = False

                            #FALTA EDITAR LA MATRIZ DE NOTAS_ALUMNO

                            for l in range(len(alumnos_curso)):

                                carnesCurso_Edicion =
alumnos_curso[l][4]

```

```

alumnos_curso[1][5]
alumnosCurso_Edicion =

for m in
range(len(carnesCurso_Edicion)):

    if (carnesCurso_Edicion[m] ==
alumnosCurso_Edicion[m] =
nombre_alumno

alumnos_curso[1][4] =
alumnos_curso[1][5] =
carnesCurso_Edicion
alumnosCurso_Edicion

        print("\nALUMNO EDITADO")
    else:
        print("Ingrese una fecha Correcta.")
    else:
        print("Ingrese la fecha de Nacimiento del
Alumno en el formato indicado.")

        else:
            print("Los datos ingresados no cumplen el formato
correcto. Intente de Nuevo.")

        if (estado_carneAlumnoEditado):
            print("El Alumno no Existe. Inténtelo de Nuevo.")
        else:
            print("ERROR. Opción Inválida.")

#--Opcion 4 (Módulo de Calificación de Cursos)--
def CalificacionCursos ():

    print("\n¡CALIFICACIÓN DE CURSOS!")

    #Inicializar Variables
    id_curso = ""
    carne_alumno = ""
    textoImprimir = ""
    notaAlumno_curso = 0.0
    totalAlumnos_curso = 0.0
    notaAgregarCurso = 0.0
    estado_idCursoBuscado = True

```

```

estado_CarneAlumnoBuscado = True
estado_notaAlumno = True
cursosAlumno = []
notasAlumno = []
i = 0
j = 0
k = 0
l = 0
m = 0
n = 0
p = 0
q = 0

print("\nCURSOS EXISTENTES")
print("| ID | | Nombre | | Horario | | Salón | | Catedrático |")

#Mostrar Cursos Existentes
for i in range (len(cursos)):
    textoImprimir = ""

    for j in range(len(cursos[0])):
        textoImprimir += "|" + cursos[i][j] + " | "

    print(textoImprimir)

print("\nALUMNOS EXISTENTES")
print("| Carné | | Nombre | | Fecha de Nacimiento |")

#Mostrar Alumnos Existentes
for k in range (len(alumnos)):
    textoImprimir = ""

    for l in range(len(alumnos[0])):
        textoImprimir += "|" + alumnos[k][l] + " | "

    print(textoImprimir)

#Validar ID del Curso
while estado_idCursoBuscado:

    id_curso = input("\n\nIngrese el ID del Curso -> ")

    for m in range(len(cursos)):
        if (cursos[m][0] == id_curso):

```

```

estado_idCursoBuscado = False

#Valir Carné del Alumno
while (estado_CarneAlumnoBuscado):

    carne_alumno = input("Ingrese el carné del Alumno -> ")

    for n in range(len(alumnos)):

        if (alumnos[n][0] == carne_alumno):

            estado_CarneAlumnoBuscado = False

            #Validar Nota Alumno
            while (estado_notaAlumno):

                notaAlumno_curso = float(input("\nIngrese la nota
del Alumno -> "))

                if (notaAlumno_curso >= 0.0 and notaAlumno_curso
<= 100.00):

                    estado_notaAlumno = False

                    for q in range(len(notas_Alumno)):

                        if (notas_Alumno[q][0] == carne_alumno):

                            cursosAlumno = notas_Alumno[q][1]
                            notasAlumno = notas_Alumno[q][3]

                            for r in range(len(cursosAlumno)):

                                if (id_curso == cursosAlumno[r]):

                                    #Actualizar nota alumno en
caso exista

                                    notaAgregarCurso =
float(alumnos_curso[m][3])

                                    notaAgregarCurso =
notaAgregarCurso - float(notasAlumno[r])

                                    notaAgregarCurso =
notaAgregarCurso + notaAlumno_curso

```

```

notasAlumno[r] =
str(notaAlumno_curso)

alumnos_curso[m][3] =
alumnos_curso[m][6][r] =

notas_Alumno[q][3] =

estado_notaAlumno = False

print("NOTA AGREGADA CON
ÉXITO")

#Agregar alumno a un curso, junto a
su nota

if(id_curso not in cursosAlumno):

    totalAlumnos_curso =
    totalAlumnos_curso += 1

    notaAgregarCurso =
    notaAgregarCurso =

    notasAlumno.append(str(notaAlumno
_curso))

    notas_Alumno[q][1].append(id_curs
o)
    notas_Alumno[q][2].append(cursos[
m][1])

    notas_Alumno[q][3] = notasAlumno

    alumnos_curso[m][2] =
    alumnos_curso[m][3] =
    alumnos_curso[m][4].append(carne_
alumno) #LO SEPARA

```

```

                                alumnos_curso[m][5].append(alumno
s[n][1]))#LO SEPARA
                                alumnos_curso[m][6].append(str(no
taAlumno_curso))

                                estado_notaAlumno = False

                                print("NOTA AGREGADA CON ÉXITO")
                                else:
                                    print("Ingrese una Nota Correcta.")

                                if (estado_CarneAlumnoBuscado):
                                    print("El alumno no Existe. Intentelo de Nuevo.")

                                if (estado_idCursoBuscado == True):
                                    print("El curso no Existe. Intente de Nuevo.")

#--Opcion 5 (Reportes)--
def Reportes ():

    print("\n¡REPORTES!")
    #Inicializar Variables
    opcion_reporte = ''
    id_curso = ""
    carne_alumno = ""
    estado_Opcionb = True
    estado_Opcionc = True
    media_curso = 0.0
    totalAlumnos_curso = 0
    notaAcumulada = 0.0
    i = 0
    j = 0
    k = 0
    l = 0
    m = 0

    #Mostrar Menú
    print("\nMENÚ - REPORTES")
    print("\ta. Listado de Cursos", "\tb. Estudiantes de un Curso", "\tc. Notas
Estudiante", "\td. Nota Media por Curso", sep="\n")

    opcion_reporte = input("\nIngrese una opción de Reporte -> ")

    match (opcion_reporte):

```

```

#Listado de Cursos de Manera Descednete
case 'a':
    print()
    print("| ID | | Nombre Curso | | Cantidad Alumnos|")

    cursosOrdenado = sorted(alumnos_curso, key = lambda x: x[2])

    for i in range(len(alumnos_curso)):
        print(f"| {cursosOrdenado[-(i+1)][0]} | | {cursosOrdenado[-(i+1)][1]} | | {cursosOrdenado[-(i+1)][2]} |")

#Notas para un Curso
case 'b':
    print("\nCURSOS EXISTENTES")
    print("| ID | | Nombre | | Horario | | Salón | | Catedrático |")

    cursosOrdenado = sorted(cursos, key = lambda x: x[0])

#Mostrar Cursos Existentes
for i in range (len(cursosOrdenado)):
    textoImprimir = ""

    for j in range(len(cursosOrdenado[0])):
        textoImprimir += "|" + cursosOrdenado[i][j] + " | "

    print(textoImprimir)

while (estado_Opcionb):

    id_curso = input("\nIngrese el ID del Curso -> ")

    for k in range(len(cursos)):

        if (id_curso == cursos[k][0]):

            for l in range(len(alumnos_curso)):
                if (alumnos_curso[l][0] == id_curso):
                    estado_Opcionb = False

            print()
            print(f"\nCurso: {alumnos_curso[l][1]}")
            print("\n| Carné | | Alumno | | Nota |")

            for m in range(len(alumnos_curso[l][4])):

```



```

                                print(f"| {alumnos_curso[1][4][m]} | |
{alumnos_curso[1][5][m]} | | {alumnos_curso[1][6][m]} |")

        if (estado_Opcionb):
            print("El Curso no Existe. Intente de Nuevo.")

#Notas de un Alumno
case 'c':

    print("\nALUMNOS EXISTENTES")
    print("| Carné | | Nombre | | Fecha de Nacimiento |")

#Mostrar Alumnos Existentes
for i in range (len(alumnos)):
    textoImprimir = ""

    for j in range(len(alumnos[0])):
        textoImprimir += "| " + alumnos[i][j] + " | "

    print(textoImprimir)

while (estado_Opcionc):

    carne_alumno = input("\nIngrese el carné del Alumno -> ")

    for k in range(len(alumnos)):

        if (alumnos[k][0] == carne_alumno):

            for l in range(len(notas_Alumno)):

                if (notas_Alumno[l][0] == carne_alumno):

                    estado_Opcionc = False

                    print(f"\nNombre Alumno: {alumnos[k][1]}")

                    print("| ID Curso | | Nombre Curso | | Nota |")

                    for m in range(len(notas_Alumno[l][1])):

                        print(f"| {notas_Alumno[l][1][m]} | |
{notas_Alumno[l][2][m]} | | {notas_Alumno[l][3][m]} |")

                    if (estado_Opcionc):

```

```

        print("El Alumno no Existe. Intente de Nuevo.")

#Nota Media por Curso
case 'd':

    print()
    print("| ID | | Nombre | | Nota Media |")

    for i in range(len(alumnos_curso)):

        totalAlumnos_curso = int(alumnos_curso[i][2])

        if (totalAlumnos_curso == 0):
            media_curso = 0
            continue

        notaAcumulada = float(alumnos_curso[i][3])

        media_curso = round(notaAcumulada / totalAlumnos_curso, 2)

        print(f"| {alumnos_curso[i][0]} | | {alumnos_curso[i][1]} | | 
{media_curso} |")

#Caso default
case _:

    print("\nOpción Inválida.")

#--Opcion 6 (Salir)--
def Salir ():
    print("Muchas Gracias por hacer uso del sistema de control de notas de la
Universidad Rafael Landivar. Esperamos tu experiencia haya sido la mejor. Ten un
hermoso día :)")
    print("CONTROL DE NOTAS UNIVERSIDAD RAFAEL LANDÍVAR (URL)")

while True:

    #Imprimir Menú de Opciones
    print("\nMENU DE OPCIONES: ")
    print("\t1. Crear Curso", "\t2. Editar Curso", "\t3. Control de Alumnos",
"\t4. Módulo de Calificación de Cursos", "\t5. Reportes", "\t6. Salir", sep="\n")

    opcion = input("\nIngrese una opción -> ")

```

```
match (opcion):

    #Opcion 1 -- Crear Curso
    case '1':
        CrearCurso()

    #Opción 2 -- Editar Curso
    case '2':
        EditarCurso()

    #Opción 3 -- Control Alumnos (Crear y Editar)
    case '3':
        ControlAlumnos()

    #Opción 4 -- Módulo de Calificación de Cursos
    case '4':
        CalificacionCursos()

    #Opción 5 -- Reportes
    case '5':
        Reportes()

    #Opción 6 -- Salir (Terminar Programa)
    case '6':
        Salir()
        break

    #Caso default
    case _:
        print("Opción Inválida. Intentelo de Nuevo")
```