

Universidad Nacional Autónoma de México

Facultad de ingeniería.

Estructura de Datos y
Algoritmos 2

Prof. Gerardo Tovar Tapia

Equipo 1

INTRODUCCIÒN

En este trabajo se llevará a cabo la creación de un interfaz con el objetivo de simular la reinscripción de los alumnos de la facultad de ingeniería de la UNAM.

El código está basado en la programación orientada a objetos, ya que el objeto contiene toda la información que permite definirlo, en este programa se usa la clase Alumno para ir almacenando la información de cada alumno en un objeto de esta clase, a su vez, este objeto dispone de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos. Esta comunicación favorece el cambio de estado en los propios objetos y lleva a tratarlos como unidades indivisibles, en las que no se separa el estado y el comportamiento.

El programa cuenta con una interfaz gráfica, esto simplifica el funcionamiento permitiendo a los usuarios que no están familiarizados con la programación, ser capaces de utilizar la máquina en una variedad de maneras.

- **Programa Principal:**

El programa es un sistema de inscripción de materias, escrito en Python 2.6, cuenta con una interfaz gráfica haciendo uso de la librería TKinter, y basándonos en el paradigma de orientado a objetos, ya que utilizamos una clase para almacenar la información de los estudiantes con sus respectivos métodos.

El acceso a nuestra propia cuenta dentro de la plataforma de inscripción, es con el único requerimiento de que esta persona ya este registrada en la base de datos (documento txt).

Se implementó una función la cual nos proporciona clave única de inscripción con la cual podremos ingresar a la plataforma junto con nuestro número de cuenta, esto con el motivo de aumentar la seguridad para los alumnos, ya que solo ese alumno tendrá el conocimiento de su “Clave de inscripción”, esto se logró gracias a la creación de nuestra propia función “Hash”

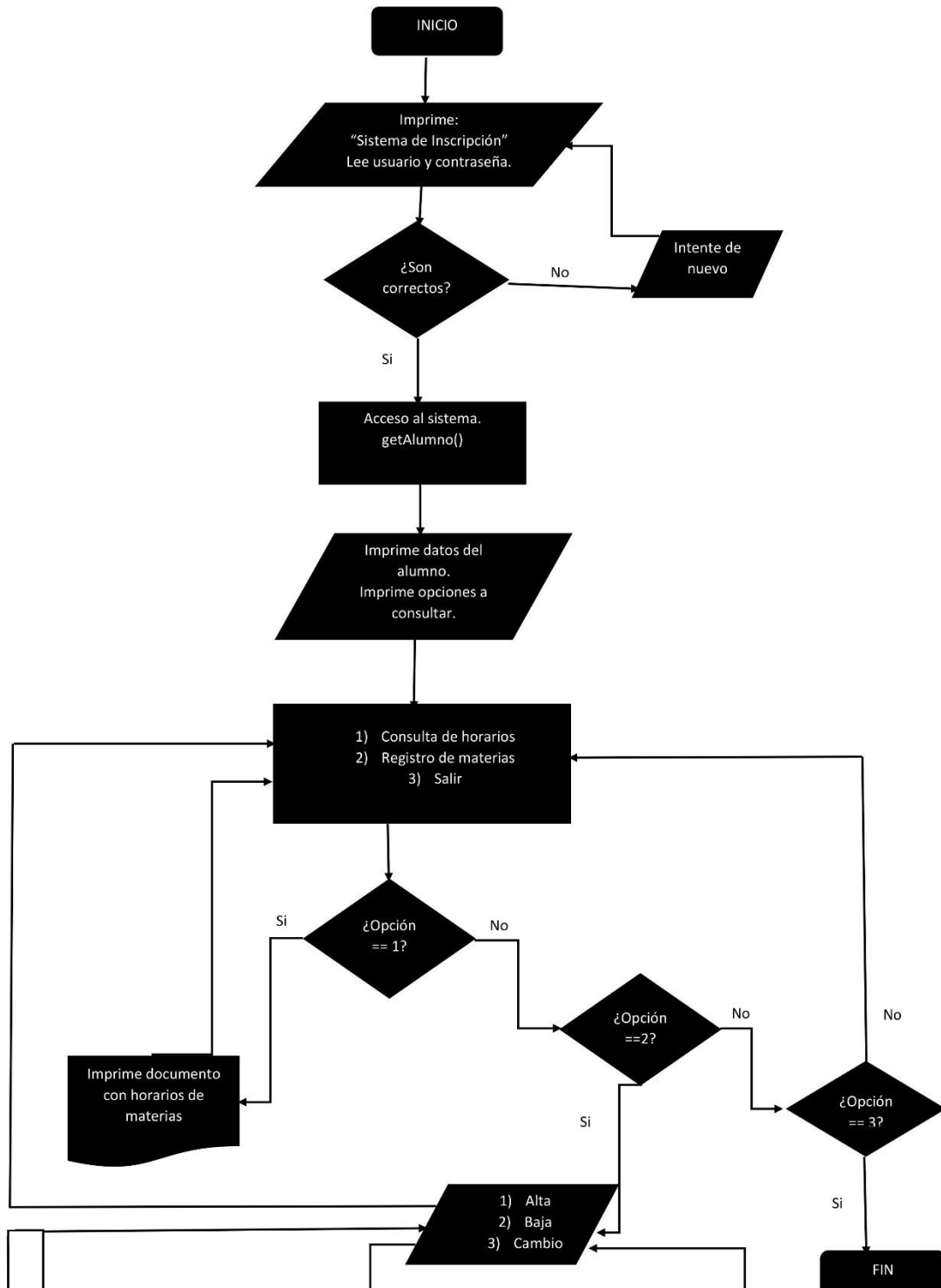
Antes ingresar tiene que asegurarse de escribir sus datos de manera correcta ya que si cualquiera de ambos está mal, aparecerá un mensaje señalando que la información que proporcionó no es correcta.

Posteriormente que haya entrado aparecerán los datos del alumno como su nombre, número de cuenta y el promedio general que lleva hasta ese momento, también encontrará un menú en el que podrá ingresar, quitar o cambiar materias según su conveniencia, además de poder consultar su horario en el cual aparecerá información como el nombre de la materia, clave, profesor, grupo y vacantes.

Si lo que el usuario desea es dar de alta una disciplina en su horario actual, deberá elegir la opción de “Ingresar Materia”, así mismo se puede remover seleccionando “Quitar Materia”, por último se puede reemplazar una asignatura con otra eligiendo “Cambiar Materia” (introduciendo la materia y grupo).

Una vez que se haya terminado de usar la plataforma, se tiene que seleccionar la opción de “Salir” y la información se guardará y actualizará de manera correcta.

Diagrama de flujo:



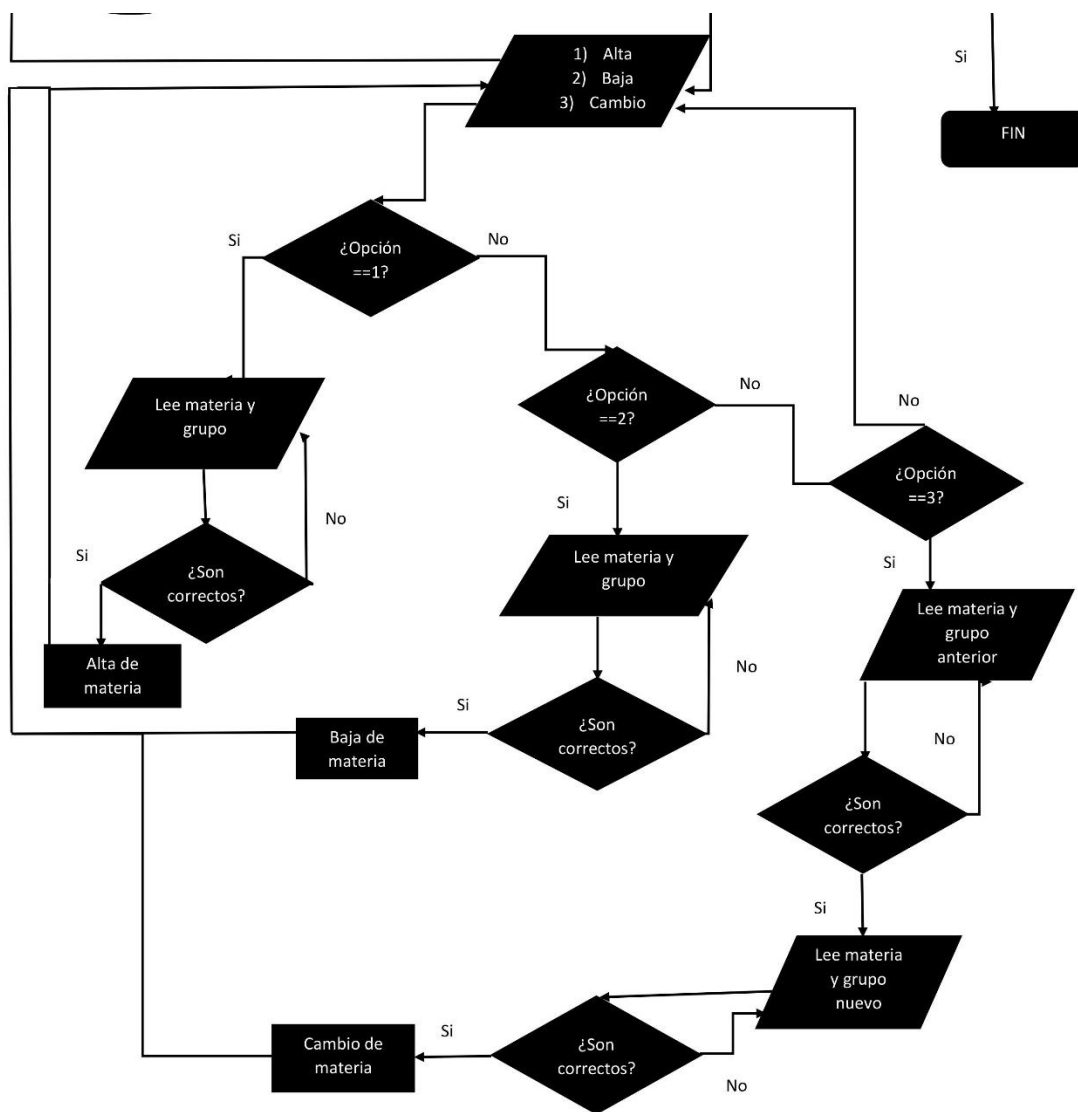
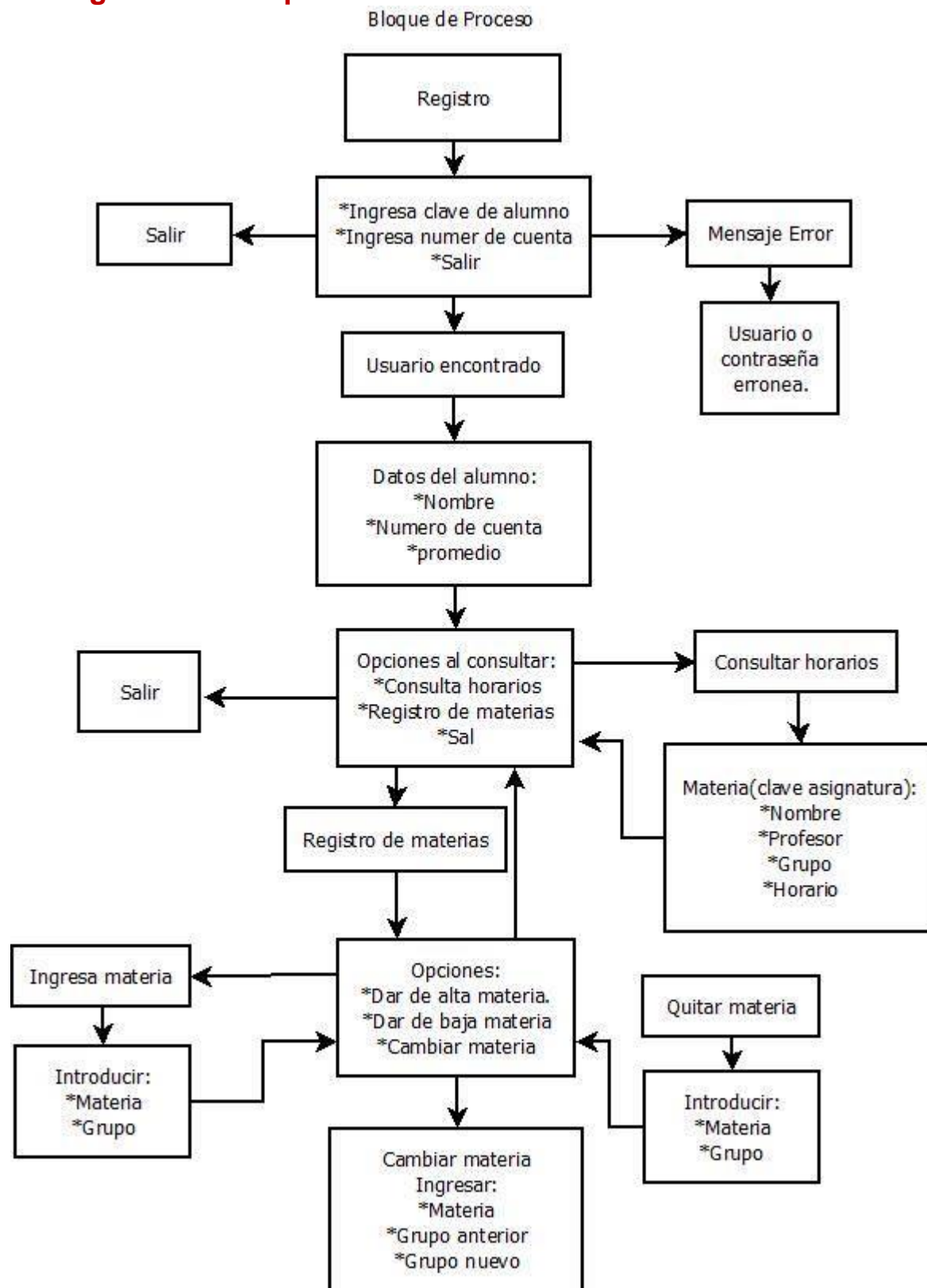


Diagrama de bloques:



Código del Programa:

```
Proyecto.py x
1  #!/usr/bin/env python
2  #-*- coding: utf-8 -*-
3
4  from Tkinter import *
5  import tkFont
6  import tkMessageBox
7
8  """Función ingresar crea la ventana principal, donde despliega primero los datos del
9  alumno, nombre, cuenta y promedio, con los
10 botones de Consulta horarios, Registra Materias, y Cerrar Sesión"""
11 def ingresar():
12     val = validaAcesso() #Llama a la funcion valida acceso, para poder ingresar al
13     sistema
14     if val == True:
15         font=tkFont.Font(family="times new roman")
16         global opcion1,opcion2,opcion3,opcion4,ventana2
17         opcion1 = IntVar()
18         opcion2 = IntVar()
19         opcion3 = IntVar()
20         opcion4 = IntVar()
21         ventana.destroy()
22         ventana2 = Tk()
23         ventana2.title("Menú principal")
24         ventana2.geometry("600x400")
25         ventana2.config(bg="#ffffff")
26         EtiquetaEncontrado = Label(ventana2,text="Usuario Encontrado",bg="#ffffcc",
27                                     font=font).pack()
28         NombreLa = Label(ventana2,text="Nombre del Alumno: "+alumno_encontrado[0],bg=
29                             "#ffffcc",font=font).place(x=10,y=20)
30         NumCla = Label(ventana2,text="Numero de Cuenta: "+str(Numero_Cuenta.get()),bg=
31                             "#ffffcc",font=font).place(x=10,y=40)
32         PromLa = Label(ventana2,text="Promedio: "+str(alumno_encontrado[2]),bg=
33                             "#ffffcc",font=font).place(x=10,y=60)
34         ConsultaHorBu = Button(ventana2,text="Consulta Horarios",bg="#ff9966",command=
35                                 Consulta_Horario,borderwidth=2,padx=20,cursor="hand2",textvariable=opcion1
36                                 ,font=font).place(x=200,y=150,anchor=W)
37         RegisMatBu = Button(ventana2,text="Registra Materias",command=Registra_Materia
38                               ,bg="#ff9966",borderwidth=2,padx=20,textvariable=opcion2,font=font,cursor=
39                               "hand2").place(x=200,y=190,anchor=W)
40         RegresarBu = Button(ventana2,text="Cerrar Sesion",bg="#ff9966",command=
41                               Regresar_Login,borderwidth=2,padx=20,textvariable=opcion3,font=font,cursor
42                               ="hand2").place(x=215,y=230,anchor=W)
43         SalirBu = Button(ventana2,text="Salir",borderwidth=2,padx=20,command=Salir,bg=
44                               "#ff9966",textvariable=opcion4,font=font,cursor="hand2").place(x=255,y=270
45                               )
46
47     else:
48         tkMessageBox.showerror("AVISO","No. de Cuenta o Clave inválidas") #Si es
49         falso arroja una ventana con la leyenda correspondiente
50
51 """Función hash para generar la clave unica de cada usuario para ingresar al sistema;
52 le aplica el modulo de 3 al numero de cuenta
53 y despues lo concatena con los ultimos tres digitos de su numero de cuenta"""
54 def hash(Numero_Cuenta):
55     nc=[]
56     p1=str(Numero_Cuenta)
57     p2=str(Numero_Cuenta%3)
58     nc=str(p1[len(p1)-4:len(p1)])
59     con= p2+nc
60     clave=int(con)
61     return clave
62
63 """Función Salir, es un comando para destruir la ventana"""
64 def Salir():
65     ventana2.destroy()
```



```
Proyecto.py x
50
51 """Funcion Registra_Materia, abre una ventana para dar de alta la materia, contiene
una etiqueta "Registro" y
52 cuadros texto para ingresar el grupo y clave a dar de alta"""
53 def Registra_Materia():
54     global ventana_Registra, grupo, clave_grupo, nuevo_grupo, nueva_clave
55     grupo = IntVar()
56     clave_grupo = IntVar()
57     nuevo_grupo = IntVar()
58     nueva_clave = IntVar()
59     ventana_Registra = Tk()
60     ventana_Registra.title("Registro")
61     ventana_Registra.geometry("600x400")
62     GrupoLabe = Label(ventana_Registra, text="Grupo: ").place(x=65, y=40)
63     GrupoEn = Entry(ventana_Registra, textvariable=grupo).place(x=115, y=40)
64     ClaveMaLabe = Label(ventana_Registra, text="Clave de Grupo: ").place(x=10, y=60)
65     ClaveMaEn = Entry(ventana_Registra, textvariable=clave_grupo).place(x=115, y=60)
66     AltaBu = Button(ventana_Registra, text="Alta", command=alta, borderwidth=3, cursor="
hand2").place(x=30, y=350)
67
68 """Función alta, es un comando utilizado en la interfaz grafica para dar de alta la
materia"""
69 def alta(grupo, clave_grupo):
70     print("Se dio de alta: "+grupo+" "+clave_grupo)
71
72 """Funcion Regresar_Login, al momento de cerrar la sesion se muestra el aviso de
sesion cerrada, se destruye la ventana y se
73 regresa a la pantalla del Login"""
74 def Regresar_Login():
75     tkMessageBox.showinfo("AVISO", "Sesión Cerrada")
76     ventana2.destroy()
77     main()
78
79 """Esta funcion solo cierra o destruye la ventana de la consulta de horarios"""
80 def Cerrar_Horario():
81     ventana_Horario.destroy()
82
83 """Función Consulta_Horario, abre el documento de texto Materias.txt, y lo lee línea
a línea para imprimirlo en la ventana"""
84 def Consulta_Horario():
85     global ventana_Horario
86     i=30
87     manf = open("Materias.txt")
88     ventana_Horario = Tk()
89     ventana_Horario.title("Horarios")
90     ventana_Horario.geometry("600x400")
91     NomProfe = Label(ventana_Horario, text="Profesor").place(x=80, y=30)
92     NomMate = Label(ventana_Horario, text="Materia").place(x=250, y=30)
93     NomHora = Label(ventana_Horario, text="Horario").place(x=350, y=30)
94     NomVacan = Label(ventana_Horario, text="Vacantes").place(x=460, y=30)
95     for linea in manf:
96         HorarioLa = Label(ventana_Horario, text=linea).place(x=20, y=i)
97         i+=20
98     RegresaBu = Button(ventana_Horario, text="Cerrar", command=Cerrar_Horario,
borderwidth=3, cursor="hand2").place(x=270, y=350)
99
100 def validaAcesso():
101     global alumno_encontrado
102     clave = hash(Numero_Cuenta.get())
103     print(clave)
104     for i in a:
105         #Dentro de toda la lista de alumnos busca la coincidencia del número de
cuenta.
106         if i.numCuenta==str(Numero_Cuenta.get()) and clave == Clave_Alumno.get():
107             alumno_encontrado = i.GetAlumno()
108             #Una vez encontrada la cuenta se comprueba si la cuenta es correcta.
109             return True
110     return False
111
112 #Al dar enter, llama a la función ingresar()
113 def onEnter(event):
114     ingresar()
115
116 """Función Login, crea la ventana principal con las etiquetas correspondientes a
Número de cuenta y Clave, con sus respectivos cuadros de texto"""
117 def Login(ventana, Numero_Cuenta, Clave_Alumno):
118     font = tkFont.Font(family="times new roman", size="36")
119     font2 = tkFont.Font(family="times new roman")
120     LoginLabel = Label(ventana, text="LOGIN", bg = "#caad00", font=font).place(x=215, y=50
)
121     NumCuen = Label(ventana, text="No. de Cuenta: ", bg="#caad00", font=font2).place(x=
100, y=150)
122     NumCuenEn = Entry(ventana, textvariable = Numero_Cuenta)
123     NumCuenEn.delete(0, END)
124     NumCuenEn.bind('<Return>', onEnter)
125     NumCuenEn.place(x=210, y=150)
126     ClaveAlu = Label(ventana, text="Clave de Alumno: ", bg="#caad00", font=font2).place(x
=85, y=170)
127     ClaveAluEn = Entry(ventana, textvariable = Clave_Alumno)
```



```

127 ClaveAluEn = Entry(ventana, textvariable = Clave_Alumno)
128 ClaveAluEn.delete(0, END)
129 ClaveAluEn.bind('<Return>', onEnter) #Se llama a la función onEnter, para capturar
    los datos al dar enter
130 ClaveAluEn.place(x=210, y=170)
131 IngresarB = Button(ventana, text="Ingresar", command=ingresar, bg="#FFCC66",
    borderwidth=3, activebackground="#FFCC00", cursor="hand2", font=font2).place(x=
    250, y=200)
132
133 class Alumno:
134     nombre=""
135     numCuenta=""
136     claveAcceso=""
137     materias=[]
138     calificaciones=[]
139     promedio=0
140     #Constructor de la clase Alumno.
141     """Recibe nombre, número de cuenta y las materias con sus calificaciones"""
142     def __init__(self, n, nu, ma, cal):
143         self.nombre=n
144         self.numCuenta=nu
145         self.materias=ma
146         self.calificaciones=cal
147
148     #Agrega una materia al arreglo de materias del alumno
149     def addMaterias(self, mat):
150         self.materias.append(mat)
151
152     #Recibe el arreglo de calificaciones y después de calcularlo retorna el promedio.
153     def getPromedio(self, mat):
154         p=0
155         if len(mat)!=0:
156             for i in range(len(mat)):
157                 p+=float(mat[i])
158             p=p/(len(mat))
159         return p
160
161     #Muestra la información del alumno-->nombre, número de cuenta y su promedio.
162     def GetAlumno(self):
163         lista_alumno = []
164
165         lista_alumno.append(self.nombre)
166         lista_alumno.append(self.numCuenta)
167         lista_alumno.append(self.getPromedio(self.calificaciones))
168         return lista_alumno
169
170 def Capturar():
171     total=[]
172     #Abre el archivo(al estar terminado será el registro de alumnos)
173     doc=open("Prueba.txt")
174     #Iguala línea a la lectura de una línea del documento.
175     linea=doc.readline()
176     #Mientras línea contenga algo dentro se ejecutara todo el código.
177     while linea!='':
178         if linea!=None:
179             #Convierte línea en un arreglo de cadenas llamado "palabras"
180             palabras = linea.split()
181             #Si el tamaño de la lista "palabras" es mayor a 0 accede al if.
182             if len(palabras)>0:
183                 """Aquí busca que el primer elemento de la lista sea "Nombre:" si es
184                 así
185                 comienza a conseguir valores para crear un alumno el cual se guardara
186                 en
187                 otra lista llamada "total" la cual contendrá todos los alumnos que
188                 están
189                 dentro del documento que contiene a los alumnos"""
190                 if palabras[0]=='Nombre:':
191                     nom=""
192                     mat=[]
193                     cal=[]
194                     for i in range(1, len(palabras)):
195                         #Se guarda el nombre del alumno sacado del documento en una
196                         cadena 'nom'
197                         nom+=palabras[i]
198                         nom+=" "
199                     linea=doc.readline()
200                     palabras=linea.split()
201                     #Guarda el número de cuenta sacado del documento en una variable
202                     'cuenta'
203                     cuenta=palabras[1]
204                     linea=doc.readline()
205                     palabras=linea.split()
206                     #Guarda todas las materias sacadas del documento y las guarda en
207                     una lista 'mat'
208                     for i in range(1, len(palabras)):
209                         mat.append(palabras[i])

```

```

202     for i in range(1,len(palabras)):
203         mat.append(palabras[i])
204     linea=doc.readline()
205     palabras=linea.split()
206     #Guarda todas las calificaciones sacadas del documento y las
    guarda en
    #una lista 'cal'
207     for i in range(1,len(palabras)):
208         cal.append(palabras[i])
209     #Crea un alumno con los parametros obtenidos anteriormente y los
    guarda en
    #una lista 'total'
210     total.append(Alumno(nom,cuenta,mat,cal))
211     #Continúa leyendo línea a línea
212     linea=doc.readline()
213     #Retorna la lista con todas los alumnos localizados en el documento.
214     return total
215
216
217
218
219 def main():
220     global ventana,Numero_Cuenta,Clave_Alumno,a
221     a=Capturar()
222     ventana = Tk()
223     ventana.title("Registro")
224     ventana.geometry("600x400")
225     ventana.config(bg="#caad00")
226     Numero_Cuenta = IntVar()
227     Clave_Alumno = IntVar()
228     Login(ventana,Numero_Cuenta,Clave_Alumno)
229     ventana.mainloop()
230
231 main()

```

Interfaz gráfica:

Pantalla principal donde el usuario deberá proporcionar su “número de cuenta” y la “clave única de inscripción” para poder acceder a la plataforma y así comenzar el proceso de su reinscripción al siguiente semestre, esta pantalla también desplegará un mensaje de “Numero de cuenta o clave incorrecta” así se requiera hasta que el usuario proporcione estos dos rubros correctamente.

The image shows a graphical user interface for a login system. The window is titled "Registro" and has a yellow background. In the center, the word "LOGIN" is displayed in a large, bold, black serif font. Below this, there are two input fields. The first is labeled "No. de Cuenta:" and the second is labeled "Clave de Alumno:". Both labels are in a black serif font. At the bottom center of the window, there is a yellow button with a black border and the text "Ingresar" in a black serif font.

Si se ingresa algo incorrecto, se despliega la ventana de incorrecto

Registro

LOGIN

No. de Cuenta: 456

Clave de Alumno: 456

Ingresar

AVISO

No. de Cuenta o Clave inválidas

OK

Si se ingresa el número de cuenta y clave correctos, se despliega la ventana de menú principal

Registro

×

LOGIN

No. de Cuenta: 314317128

Clave de Alumno: 7128

Ingresar

Menú principal

×

Usuario Encontrado

Nombre del Alumno: Isaias Hernandez Guatemala

Numero de Cuenta: 314317128

Promedio: 9.33333333333

Consulta Horarios

Registra Materias

Cerrar Sesion

Salir

Dentro de la plataforma el usuario cuenta con varias opciones de las cuales elegir según sean sus necesidades (Solo se cuentan con los botones ya que aún no se tiene el vínculo entre estos botones y su parte del código).

Al poner el cursor por arriba de los botones, estos se colorean. Al consultar horario se despliega otra ventana.

Menú principal

Usuario Encontrado

Nombre del Alumno: Isaias Hernandez Guatemala

Numero de Cuenta: 314317128

Promedio: 9.3333333333

Consulta Horarios

Registra Materias

Cerrar Sesion

Salir

Horarios			
Profesor	Materia	Horario	Vacantes
123 Hernandez Lopez Miguel	Algebra	Mar.Juev 7-9	34
122 Santiago Huerta José	Algebra	Mar.Juev 9-11	10
134 Jimenez Medrano Luz	Calculol	Mar.Jue 9-11	15
145 Cortina Meza Arturo	EDA II	Lun.Mier.Jue 9-11	45
167 Higareda Mendoza Alan	Mecanica	Mar.Jue 7-9	18
189 Pedroza Sanchez Miguel	Cultura	Vie 9-11	20
Cerrar			

Al registrar materia, y para dar de alta se despliega otra ventana.

Menú principal

Usuario Encontrado

Nombre del Alumno: Isaias Hernandez Guatemala

Numero de Cuenta: 314317128

Promedio: 9.33333333333

Consulta Horarios

Registra Materias

Cerrar Sesion

Salir

Registro

Grupo:

Clave de Grupo:

Alta

Al hacer click en cerrar sesión, despliega otra ventana con el aviso correspondiente y vuelve a la ventana del Login. Se cuenta con un botón de cerrar sesión que no solo sirve para cerrar el programa, sino que también guarda todas las modificaciones hechas en ese perfil. Y para la comodidad del usuario y saber que se encuentra en el perfil correcto agregamos en la parte superior de la pantalla los datos generales del usuario con el que se ha ingresado a la plataforma.

