

Proyecto Final de Programación Orientada a Objetos
S1-2024

A continuación, se detallan los requerimientos del proyecto:

Requerimientos generales.

A continuación, se describen todas las consideraciones principales que se deberán tener en cuenta:

1. Modificar y hacer funcionar un aplicativo gráfico en Python 3 que usa la librería Tkinter con ttk y con la apariencia que se muestra en el **mockup** proporcionado (Anexo 1).
2. El propósito del programa es manejar la información de los cursos a los que se pueden inscribir los estudiantes de una universidad o instituto.
3. El aplicativo debe ser de tipo CRUD (Create, Read, Update, Delete) lo que significa que debe permitir:
 - Crear o insertar información.
 - Leer o consultar la información.
 - Actualizar o modificar la información. La única información modificable es la que aparece en la tabla **Inscritos**.
 - Eliminar o borrar la información. La única información que se puede eliminar es la que aparece en la tabla **Inscritos**.
4. Es obligatorio el uso de la base de datos SQLite3 para la persistencia de los datos.
5. Se debe crear un base de datos llamada **Inscripciones.db** (Anexo 2).
6. Se deben crear cuatro tablas en la base de datos llamadas **Alumnos, Carreras, Cursos e Inscritos**. La estructura de las tablas se anexa en este documento y es obligatorio definirlas de la misma forma dentro del sistema que se desarrolle (Anexo 3). Todas las tablas se deben poblar manualmente con datos para verificar el funcionamiento y la única tabla vacía y que se va poblando con el programa es la tabla Inscritos.
7. Es obligatorio usar el programa llamado **plantilla.py** (Anexo 4) que se proporciona para alcanzar los objetivos planteados.
8. El trabajo se debe realizar en grupo de acuerdo con los grupos ya definidos.
9. Solo un miembro del grupo realiza la sustentación, aunque en ese momento es obligatoria la presencia de todos los integrantes del grupo.
10. En cada clase presencial o virtual se destinará tiempo para resolver posibles dudas o inquietudes.
11. La entrega del proyecto se realiza a más tardar el día 20 de mayo del 2024 a las 11:59:59 p.m.
12. Las sustentaciones se realizarán los días 21 y 23 de mayo del 2024 en el horario de clase y seleccionará de forma aleatoria el grupo para sustentar.
13. Por lo anterior la presencia en ambas sesiones es obligatoria.

Requerimientos específicos.

A continuación, se describen todas las consideraciones específicas que se deberán tener en cuenta:

1. Al ejecutar la aplicación la ventana debe tener un icono personalizado y un título como se muestra en el mockup.
2. Al ejecutar el programa la ventana debe quedar centrada sin importar el tipo de pantalla.
3. El campo **No.Inscripción** es auto incremental y deberá mostrar el número de inscripción disponible.
4. El formato de fecha debe ser **dd/mm/aaaa**. La fecha que se solicita debe ser validada como fecha cierta, es decir, no pueden existir fechas como 29 de febrero sin validar que el año sea bisiesto. De igual forma se debe validar la cantidad de días que le corresponde a cada mes.
5. El combo box de identificación de los estudiantes debe ser cargado en tiempo de ejecución con las identificaciones de todos los estudiantes almacenados en la tabla **Estudiantes**.
6. Los campos Nombres y Apellidos se deben llenar en forma automática tan pronto se selecciona una identificación de estudiantes.
7. Los botones que se definen dentro del Mockup deben ser funcionales:
 - Botón Consultar:** se considera libre de acción, lo que implica que cada grupo lo puede usar para consultar información como estimen conveniente. Los datos de la consulta se deben mostrar mediante el uso del treeView que aparece en la pantalla, en forma de grilla.
 - Botón Grabar:** debe servir para grabar la información suministrada en las tablas de la base de datos. Mediante el uso de una ventana emergente se debe confirmar el éxito o fracaso de la acción
 - Botón Editar:** debe servir para modificar los datos almacenados en la base de datos y debe permitir seleccionar una tupla de la grilla de datos (componente TreeView) para que se puedan modificar en los campos de captura definidos en el mockup de la aplicación. Mediante el uso de una ventana emergente se debe confirmar el éxito o fracaso de la acción.
 - Botón Eliminar:** debe servir para eliminar datos de forma individual o debe permitir eliminar a un inscrito y **todos** sus cursos, confirmando en ambos casos su eliminación a través de una ventana emergente que muestre el éxito o fracaso de la acción.
 - Botón Cancelar:** debe servir para cancelar cualquier acción en curso y debe limpiar todos los campos de captura mostrándolos vacíos en la pantalla.
- Nota importante:**

En el mockup el treeView es una muestra de lo que puede ser.
8. Las operaciones sobre la base de datos se deben realizar mediante sentencias SQL.

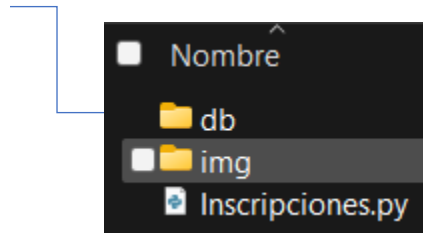
Restricciones.

Adicional a lo expresado en los ítems anteriores se debe tener en cuenta:

1. La estructura de directorios debe ser:

Un directorio principal llamado

Inscripciones



2. En el sub directorio **db** debe de estar la base de datos llamada **Inscripciones.db**
3. En el sub directorio **img** debe estar el icono de la aplicación
4. En el directorio **Inscripciones** debe estar la **plantilla** modificada con el nombre final de **Inscripciones.py**
5. La entrega del programa debe hacerse en formato comprimido tipo ZIP y no se admite ningún otro formato como RAR, TAR, TARGZ o cualquier otro

Nota Importante:

No pueden crearse programas adicionales o módulos para ser importados, sólo debe existir un único programa llamado Incripciones.py

Criterios de evaluación:

1. Para la evaluación deben estar presentes todos los integrantes del equipo.
2. El programa debe funcionar sin errores para que se le pueda otorgar una nota.
3. El programa debe cumplir con todos los requerimientos expresados en este documento.
4. Es posible adicionar funcionalidad, pero esta debe trabajar sin errores para que la nota se vea afectada.
5. Las fechas son finales y no admiten extensiones o prorrogas.

Anexo 1. Mockup de la aplicación

Inscripciones de Materias y Cursos

No.Inscripción

Id Alumno:

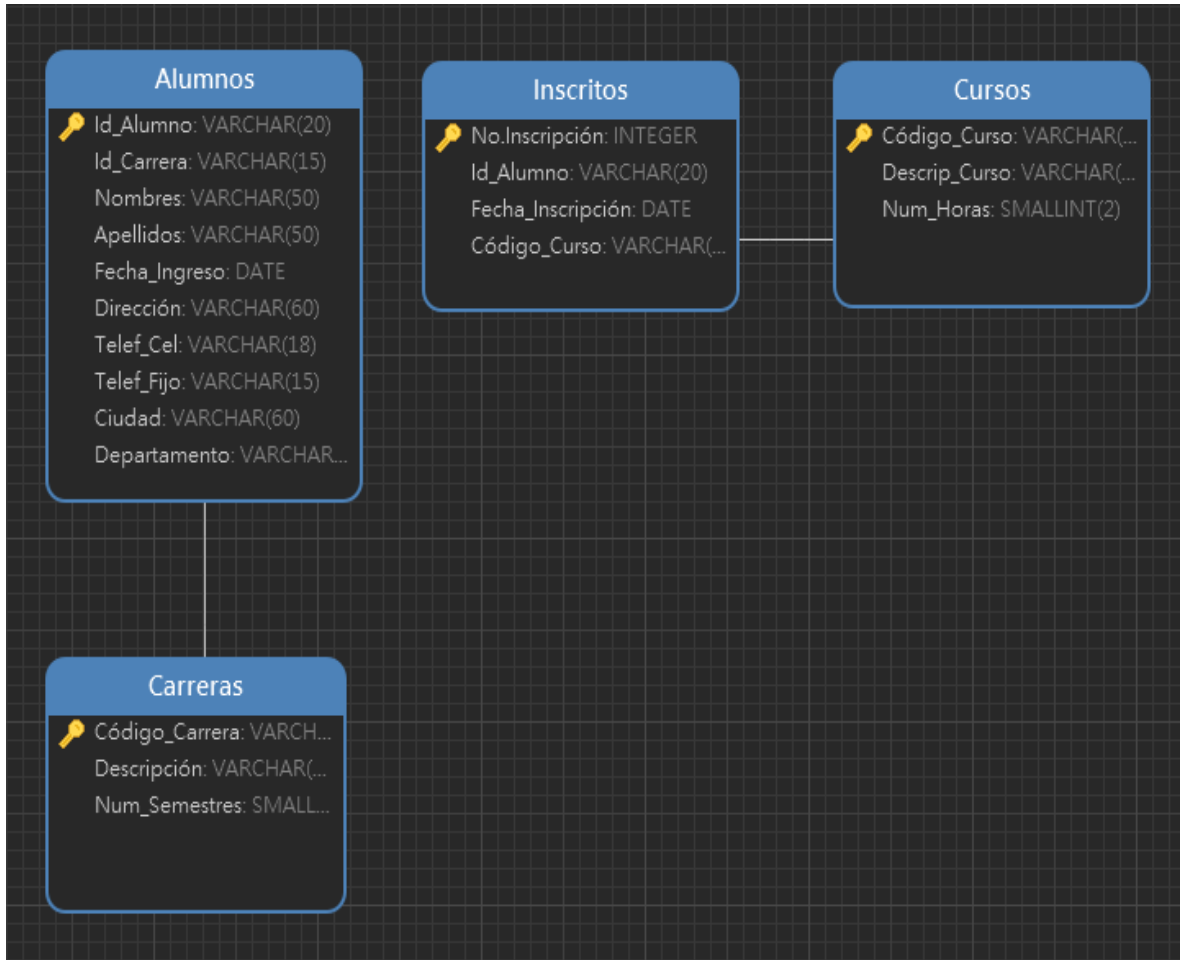
Fecha:

Nombre(s):

Apellido(s):

Curso	Descripción
-------	-------------

Anexo 2. Diseño de la Base de Datos – Modelo E-R



Anexo 3. Estructura de las tablas de datos

Alumnos			
	Field	Type	Extra
P	Id_Alumno	VARCHAR(20)	
	Id_Carrera	VARCHAR(15)	
	Nombres	VARCHAR(50)	Allow Null
	Apellidos	VARCHAR(50)	Allow Null
	Fecha_Ingreso	DATE	
	Dirección	VARCHAR(60)	Allow Null
	Telef_Cel	VARCHAR(18)	Allow Null
	Telef_Fijo	VARCHAR(15)	Allow Null
	Ciudad	VARCHAR(60)	Allow Null
	Departamento	VARCHAR(60)	Allow Null

Carreras			
	Field	Type	Extra
P	Código_Carrera	VARCHAR(10)	
	Descripción	VARCHAR(100)	Allow Null
	Num_Semestres	SMALLINT(2)	Allow Null

Cursos			
	Field	Type	Extra
P	Código_Curso	VARCHAR(20)	
	Descrip_Curso	VARCHAR(60)	Allow Null
	Num_Horas	SMALLINT(2)	Allow Null

Inscritos			
	Field	Type	Extra
P	No.Inscripción	INTEGER	Auto Increment
	Id_Alumno	VARCHAR(20)	
	Fecha_Inscripción	DATE	
	Código_Curso	VARCHAR(20)	Allow Null

Anexo 4. Código en Python.

Se proporciona el código a mejorar, pero se debe tener en cuenta la compatibilidad con Word por si se copia directamente desde aquí. El código en Python o **Plantilla.py** se publica en el Drive del curso y se recomienda descargarlo desde el sitio.

```
# !/usr/bin/python3
# -*- coding: utf-8 -*-
import tkinter as tk
import tkinter.ttk as ttk
import sqlite3

class Incripciones:
    def __init__(self, master=None):
        # Vewntan principal
        self.db_name = 'Incripciones.db'
        self.win = tk.Tk(master)
        self.win.configure(background="#f7f9fd", height=600, width=800)
        self.win.geometry("800x600")
        self.win.resizable(False, False)
        self.win.title("Incripciones de Materias y Cursos")
        # Crea los frames
        self.frm_1 = tk.Frame(self.win, name="frm_1")
        self.frm_1.configure(background="#f7f9fd", height=600, width=800)
        self.lblNoInscripcion = ttk.Label(self.frm_1,
name="lblnoinscripcion")
        self.lblNoInscripcion.configure(background="#f7f9fd",font="{Arial}
11 {bold}",
justify="left",state="normal",
takefocus=False,text='No.Inscripción
')

        #Label No. Inscripción
        self.lblNoInscripcion.place(anchor="nw", x=670, y=20)
        #Entry No. Inscripción
        self.num_Inscripcion = ttk.Entry(self.frm_1, name="num_inscripcion")
        self.num_Inscripcion.configure(justify="right")
        self.num_Inscripcion.place(anchor="nw", width=100, x=672, y=50)

        #Label Fecha
        self.lblFecha = ttk.Label(self.frm_1, name="lblfecha")
        self.lblFecha.configure(background="#f7f9fd", text='Fecha:')
        self.lblFecha.place(anchor="nw", x=630, y=85)
        #Entry Fecha
```

```
self.fecha = ttk.Entry(self.frm_1, name="fecha")
self.fecha.configure(justify="center")
self.fecha.place(anchor="nw", width=90, x=680, y=85)
#Label Alumno
self.lblIdAlumno = ttk.Label(self.frm_1, name="lblidalumno")
self.lblIdAlumno.configure(background="#f7f9fd", text='Id Alumno:')
self.lblIdAlumno.place(anchor="nw", x=30, y=85)
#Combobox Alumno
self.cmbx_Id_Alumno = ttk.Combobox(self.frm_1,
name="cmbx_id_alumno")
self.cmbx_Id_Alumno.place(anchor="nw", width=112, x=110, y=85)
#Label Alumno
self.lblNombres = ttk.Label(self.frm_1, name="lblnombres")
self.lblNombres.configure(background="#f7f9fd", text='Nombre(s):')
self.lblNombres.place(anchor="nw", x=30, y=140)
#Entry Alumno
self.nombres = ttk.Entry(self.frm_1, name="nombres")
self.nombres.place(anchor="nw", width=200, x=110, y=140)
#Label Apellidos
self.lblApellidos = ttk.Label(self.frm_1, name="lblapellidos")
self.lblApellidos.configure(background="#f7f9fd", text='Apellido(s):'
)

self.lblApellidos.place(anchor="nw", x=400, y=140)
#Entry Apellidos
self.apellidos = ttk.Entry(self.frm_1, name="apellidos")
self.apellidos.place(anchor="nw", width=200, x=485, y=140)

''' Botones de la Aplicación'''
#Botón Guardar
self.btnGuardar = ttk.Button(self.frm_1, name="btnguardar")
self.btnGuardar.configure(text='Guardar')
self.btnGuardar.place(anchor="nw", x=200, y=220)
#Botón Editar
self.btnEditar = ttk.Button(self.frm_1, name="btncargar")
self.btnEditar.configure(text='Editar')
self.btnEditar.place(anchor="nw", x=300, y=220)
#Botón Eliminar
self.btnEliminar = ttk.Button(self.frm_1, name="btneliminar")
self.btnEliminar.configure(text='Eliminar')
self.btnEliminar.place(anchor="nw", x=400, y=220)
#Botón Cancelar
self.btnCancelar = ttk.Button(self.frm_1, name="btncancelar")
self.btnCancelar.configure(text='Cancelar')
self.btnCancelar.place(anchor="nw", x=500, y=220)
```



```
#Separador
separator1 = ttk.Separator(self.frm_1)
separator1.configure(orient="horizontal")
separator1.place(anchor="nw", width=796, x=2, y=260)

''' Treeview de la Aplicación'''
#Treeview
self.tView = ttk.Treeview(self.frm_1, name="tview")
self.tView.configure(selectmode="extended")
#Columnas del Treeview
self.tView_cols = ['tV_descripción']
self.tView_dcols = ['tV_descripción']
self.tView.configure(columns=self.tView_cols,displaycolumns=self.tView_dcols)
self.tView.column("#0",anchor="w",stretch=True,width=10,minwidth=10)
self.tView.column("tV_descripción",anchor="w",stretch=True,width=200,minwidth=50)
#Cabeceras
self.tView.heading("#0", anchor="w", text='Curso')
self.tView.heading("tV_descripción", anchor="w", text='Descripción')
self.tView.place(anchor="nw", height=300, width=790, x=4, y=280)
#Scrollbars
self.scroll_H = ttk.Scrollbar(self.tView,name="scroll_h")
self.scroll_H.configure(orient="horizontal")
self.scroll_H.place(anchor="s", height=12, width=1534, x=15, y=595)
self.scroll_Y = ttk.Scrollbar(self.tView, name="scroll_y")
self.scroll_Y.configure(orient="vertical")
self.scroll_Y.place(anchor="s", height=275, width=12, x=790, y=582)
self.frm_1.pack(side="top")
self.frm_1.pack_propagate(0)

# Main widget
self.mainwindow = self.win

def run(self):
    self.mainwindow.mainloop()

''' A partir de este punto se deben incluir las funciones
para el manejo de la base de datos '''

if __name__ == "__main__":
    app = Inscripciones()
    app.run()
```

Proyecto Final de Programación Orientada a Objetos
S1-2024

Suerte!!