



Manual de Técnico – Sistema de Control de Cursos

Indice

| | |
|--|------|
| Introducción | 1. |
| Requisitos | 1.1 |
| Código del programa | 2 |
| Código del programa | 3 |
| Código del programa | 4 |
| Código del programa | 5 |
| Código del programa | 6 |
| Código del programa | 7 |
| Código del programa | 8 |
| Código del programa | 9 |
| Código del programa | 10 |
| Menú Principal | 10.1 |
| Descripción de funciones | 11 |
| Descripción de funciones parte 2 | 12 |
| Descripción de funciones parte 3 | 13 |
| Ejemplos de Usos | 14 |
| Conclusión | 15 |
| Bibliografía | 16 |

1. Introducción

Este programa lo tendría que ejecutar con Python para que funcione así mismo te traerá beneficios al momento de instalarlo y comenzar a usarlo el cual Este sistema permite registrar, gestionar y consultar información de cursos con sus notas.

Con él se pueden realizar operaciones como:

- Registrar nuevos cursos.
 - Consultar la lista de cursos guardados.
 - Calcular promedios generales.
 - Contar cursos aprobados y reprobados.
 - Buscar, actualizar y eliminar cursos.
 - Ordenar cursos por nombre o por nota.
 - Administrar solicitudes de revisión mediante una cola.
 - Revisar historial de cambios realizados.
-

2. Requisitos del Sistema

- **Software:** Instalar Python y para que se pueda ejecutar juntamente con el código para su uso respectivo luego o cualquier editor de código.
- **Conocimientos básicos** de manejo de menús en la consola para su uso
- **Datos de entrada:** Nombre del curso y nota asociada (0–100).

Se descarga Python y visual studio code así mismo se dará el link para que se descargue el programa juntamente con la extensión:

https://www.google.com/search?q=python&og=python&gs_lcrp=EgZjaHJvbWUyDAgAEEUYORixAxiABDINCAEQABiDARixAxiABDIGCAIQRRhAMgclAxAAGIAEMgYIBBBFGDwyBggFEEUYPDIGCAYQRRg9MgYIBxBFGEHSAQgyMTcxajBqN6gCCLACaFEF9Sd06hKJW6M&sourceid=chrome&ie=UTF-8



Ya teniendo descargado el programa de Python se instala también una extensión o programa secundario para poder ingresar el código el cual se describió anteriormente el cual es visual studio code el cual es:



Así mismo en el programa de visual studio code hay una opción donde se tiene que instalar una extensión para que pueda correr el programa el programa al momento de ingresar el código y hacer correr el programa que a continuación se le hará unas indicaciones al momento de ingresar los códigos.

Del cual se le dará el código para que se pueda ingresar al programa y se podrá ejecutar

```

# ===== GESTOR DE NOTAS ACADÉMICAS =====
# ----- DATOS INICIALES -----
def crear_curso(nombre, nota_sem1=None):
    notas = [None]*5
    if nota_sem1 is not None:
        notas[0] = round(nota_sem1,2)
    return {"nombre": nombre, "notas": notas}
cursos = [
    crear_curso("Diseño",90),
    crear_curso("Programación",100),
    crear_curso("Algoritmo",60),
    crear_curso("Precalculo",59),
    crear_curso("Matemática Discreta",62)]

historial = [] # pila de historial
cola_revisiones = {
    "Diseño": "Revisando",
    "Precalculo": "En proceso",
    "Programación": "Resuelto"}

# ----- FUNCIONES AUXILIARES -----
def ingresar_nota(semestre):
    while True:
        try:
            nota = float(input(f"Ingrese nota del semestre {semestre} (0-100): "))
            if 0 <= nota <= 100:
                return round(nota,2)
            else:
                print("Nota inválida. Debe estar entre 0 y 100.")
        except:
            print("Ingrese un número válido.")

def ingresar_semestre():
    while True:
        try:
            sem = int(input("Ingrese semestre (1-5): "))
            if 1 <= sem <= 5:
                return sem
            else:
                print("Semestre inválido.")

```

```

        except:
            print("Ingrese un número válido.")
def buscar_curso_lineal(nombre):
    for c in cursos:
        if c['nombre'].lower() == nombre.lower():
            return c
    return None
def calcular_promedio_curso(curso):
    notas_validas = [n for n in curso['notas'] if n is not None]
    if notas_validas:
        return round(sum(notas_validas)/len(notas_validas),2)
    return None

# ----- SUBMENÚ REGISTRAR CURSOS/NOTAS -----
def registrar_curso():
    while True:
        print("\n--- Submenú Registrar Curso/Notas ---")
        print("1. Nuevo curso y nueva nota")
        print("2. Ingresar nueva nota")
        print("3. Eliminar nota")
        print("4. Salir del submenú")
        opcion = input("Seleccione opción: ")
        if opcion == "1":
            nombre = input("Ingrese nombre del curso: ").strip()
            if any(c['nombre'].lower() == nombre.lower() for c in cursos):
                print("El curso ya existe.")
                continue
            nota = ingresar_nota(1)
            cursos.append(crear_curso(nombre,nota))
            historial.append(f"Nuevo curso '{nombre}' con nota {nota} registrada")
            print(f"Curso '{nombre}' registrado con nota {nota}.")

        elif opcion == "2":
            nombre = input("Ingrese nombre del curso: ").strip()
            curso = buscar_curso_lineal(nombre)
            if curso:
                semestre = ingresar_semestre()
                nota = ingresar_nota(semestre)
                historial.append(f"Se registró nota del curso {curso['nombre']} en semestre {semestre}: {nota}")
                curso['notas'][semestre-1] = nota

```

```

        print("Nota registrada/actualizada correctamente.")
    else:
        print("Curso no encontrado.")
elif opcion == "3":
    nombre = input("Ingrese nombre del curso: ").strip()
    curso = buscar_curso_lineal(nombre)
    if curso:
        semestre = ingresar_semestre()
        historial.append(f"Se eliminó nota del curso
{curso['nombre']} semestre {semestre} nota anterior
{curso['notas'][semestre-1]}")
        curso['notas'][semestre-1] = None
        print("Nota eliminada correctamente.")
    else:
        print("Curso no encontrado.")
elif opcion == "4":
    break
else:
    print("Opción inválida.")

# ----- MOSTRAR CURSOS -----
def mostrar_cursos():
    if not cursos:
        print("No hay cursos registrados.")
        return
    print("\nCursos y notas por semestres:")
    print(f"{'Curso':25} {'S1':>5} {'S2':>5} {'S3':>5} {'S4':>5} {'S5':>5}")
    for c in cursos:
        notas = [n if n is not None else "-" for n in c['notas']]
        print(f"{c['nombre']:25} {notas[0]:>5} {notas[1]:>5} {notas[2]:>5}
{notas[3]:>5} {notas[4]:>5}")
def calcular_promedio_general():
    if not cursos:
        print("No hay cursos.")
        return
    while True:
        print("\n--- Submenú Promedio General ---")
        print("1. Mostrar promedio de todos los cursos juntos")
        print("2. Mostrar promedio por curso")
        print("3. Salir submenú")
        opcion = input("Seleccione opción: ")
        # --- OPCIÓN 1: PROMEDIO GENERAL ---

```

```

if opcion == "1":
    todas_notas = []
    for c in cursos:
        todas_notas.extend([n for n in c['notas'] if n is not None])

    if todas_notas:
        promedio = round(sum(todas_notas)/len(todas_notas), 2)
        print(f"\nPromedio general de todos los cursos: {promedio}")

        # Mensaje motivacional según el promedio
        if promedio < 60:
            print(" Esfuérzate, ¡que lo lograrás!")
        elif 60 <= promedio < 70:
            print(" Sigue esforzándote, ¡vas por buen camino!")
        else:
            print(" Tu esfuerzo comienza a dar fruto, ¡sigue así!")
    else:
        print("No hay notas registradas.")

# --- OPCIÓN 2: PROMEDIO POR CURSO (en columnas) ---
elif opcion == "2":
    print("\n{:<25} {:>10} {:>15}".format("Curso", "Promedio",
"Estado"))
    print("-" * 52)
    for c in cursos:
        prom = calcular_promedio_curso(c)
        if prom is not None:
            estado = "Aprobado" if prom >= 60 else "Reprobado"
            print("{:<25} {:>10.2f} {:>15}".format(c['nombre'],
prom, estado))
        else:
            print("{:<25} {:>10} {:>15}".format(c['nombre'], "-",
"Sin notas"))
    print("-" * 52)

    elif opcion == "3":
        break
    else:
        print("Opción inválida.")

# ----- CONTAR APROBADOS/REPROBADOS -----

```



```

def contar_aprobados_reprobados():
    aprobados = 0
    reprobados = 0
    print("\n{:<25} {:>10} {:>15}".format("Curso", "Promedio", "Estado"))
    print("-"*52)
    for c in cursos:
        prom = calcular_promedio_curso(c)
        if prom is not None:
            estado = "Aprobado" if prom >= 60 else "Reprobado"
            print("{:<25} {:>10.2f} {:>15}".format(c['nombre'], prom,
estado))
            if prom >= 60:
                aprobados += 1
            else:
                reprobados += 1
        else:
            print("{:<25} {:>10} {:>15}".format(c['nombre'], "-", "Sin
notas"))

    print("-"*52)
    print(f"Cursos aprobados: {aprobados}")
    print(f"Cursos reprobados: {reprobados}")
    # ----- ACTUALIZAR NOTA -----
def actualizar_nota():
    nombre = input("Ingrese nombre del curso: ").strip()
    curso = buscar_curso_lineal(nombre)
    if curso:
        semestre = ingresar_semestre()
        nota = ingresar_nota(semestre)
        historial.append(f"Se corrigió nota curso {curso['nombre']} semestre
{semestre} de {curso['notas'][semestre-1]} a {nota}")
        curso['notas'][semestre-1] = nota
        print("Nota actualizada correctamente.")
    else:
        print("Curso no encontrado.")

# ----- ELIMINAR CURSO -----
def eliminar_curso():
    nombre = input("Ingrese nombre del curso a eliminar: ").strip()
    for i,c in enumerate(cursos):
        if c['nombre'].lower() == nombre.lower():
            cursos.pop(i)

```

```

        historial.append(f"Se eliminó el curso {nombre}")
        print("Curso eliminado exitosamente.")
        return
    print("Curso no encontrado.")

# ----- ORDENAMIENTO -----
def ordenar_por_nota():
    n = len(cursos)
    for i in range(n):
        for j in range(0, n-1-i):
            prom_j = calcular_promedio_curso(cursos[j]) or 0
            prom_j1 = calcular_promedio_curso(cursos[j+1]) or 0
            if prom_j < prom_j1:
                cursos[j], cursos[j+1] = cursos[j+1], cursos[j]
    print("Cursos ordenados por promedio (descendente):")
    mostrar_cursos()
def ordenar_por_nombre():
    for i in range(1, len(cursos)):
        key = cursos[i]
        j = i-1
        while j >= 0 and cursos[j]['nombre'].lower() > key['nombre'].lower():
            cursos[j+1] = cursos[j]
            j -= 1
        cursos[j+1] = key
    print("Cursos ordenados alfabéticamente:")
    mostrar_cursos()

# ----- BÚSQUEDA BINARIA -----
def buscar_curso_binaria():
    ordenar_por_nombre()
    nombre = input("Ingrese nombre del curso a buscar: ").strip().lower()
    low, high = 0, len(cursos)-1
    while low <= high:
        mid = (low + high)//2
        mid_name = cursos[mid]['nombre'].lower()
        if mid_name == nombre:
            prom = calcular_promedio_curso(cursos[mid])
            print(f"Curso encontrado: {cursos[mid]['nombre']} - Notas: {cursos[mid]['notas']} - Promedio: {prom}")
            return
        elif mid_name < nombre:
            low = mid + 1

```

```

        else:
            high = mid - 1
            print("Curso no encontrado.")
# ----- SIMULAR COLA -----
def simularCola():
    while True:
        nombre = input("Ingrese curso para revisión (o 'fin' para salir):
").strip()
        if nombre.lower() == "fin":
            break
        if nombre in cola_revisiones:
            cola_revisiones[nombre] = "Resuelto"
        else:
            cola_revisiones[nombre] = "En proceso"
        print("\nEstado actual de la cola de revisiones:")
        for c,estado in cola_revisiones.items():
            print(f"{c}: {estado}")
# ----- HISTORIAL -----
def mostrar_historial():
    if not historial:
        print("No hay historial de cambios.")
    else:
        print("\nHistorial de cambios recientes:")
        for i,c in enumerate(reversed(historial),1):
            print(f"{i}. {c}")
# ----- MENÚ PRINCIPAL -----
def menu():
    while True:
        print("\n===== GESTOR DE NOTAS ACADÉMICAS =====")
        print("1. Registrar nuevo curso")
        print("2. Mostrar todos los cursos y notas")
        print("3. Calcular promedio general")
        print("4. Contar cursos aprobados y reprobados")
        print("5. Buscar curso por nombre")
        print("6. Actualizar nota de un curso")
        print("7. Eliminar un curso")
        print("8. Ordenar cursos por nota")
        print("9. Ordenar cursos por nombre")
        print("10. Buscar curso por nombre")
        print("11. Simular cola de solicitudes de revisión")
        print("12. Mostrar historial de cambios")
        print("13. Salir")

```

```

opcion = input("Seleccione una opción: ")

if opcion=="1":
    registrar_curso()
elif opcion=="2":
    mostrar_cursos()
elif opcion=="3":
    calcular_promedio_general()
elif opcion=="4":
    contar_aprobados_reprobados()
elif opcion=="5":
    nombre = input("Ingrese nombre del curso: ").strip()
    curso = buscar_curso_lineal(nombre)
    if curso:
        prom = calcular_promedio_curso(curso)
        print(f"{curso['nombre']} - Notas: {curso['notas']} - Promedio: {prom}")
    else:
        print("Curso no encontrado.")
elif opcion=="6":
    actualizar_nota()
elif opcion=="7":
    eliminar_curso()
elif opcion=="8":
    ordenar_por_nota()
elif opcion=="9":
    ordenar_por_nombre()
elif opcion=="10":
    buscar_curso_binaria()
elif opcion=="11":
    simularCola()
elif opcion=="12":
    mostrar_historial()
elif opcion=="13":
    print("Gracias por usar el Gestor de Notas. ¡Vuelva pronto!")
    break
else:
    print("Opción inválida.")

# ----- EJECUCIÓN -----
menu()

```

Se dio los códigos para que se pueda ejecutar el programa en el cual se detalla a continuación algunas recomendaciones para evita que halla algún problema o error al ejecutar por lo cual se le da un pequeño resumen del menú y de cómo funciona cada sección del menú, Así mismo se le da algunos ejemplos acerca de las funciones y posibles errores.

3. Menú principal: El programa presenta un menú con las siguientes opciones:

1. Registrar Nuevo Curso
2. Mostrar todos los Cursos
3. Calcular Promedio General
4. Contar Cursos Aprobados y Reprobados
5. Buscar Curso por Nombre
6. Actualizar Notas de un Curso
7. Eliminar Curso
8. Ordenar Cursos por Nota
9. Ordenar Cursos por Nombre
10. Buscar Curso por Nombre
11. Simular Cola de Solicitudes de Revisión
12. Mostar Historial de Cambios
13. Salir

El usuario debe ingresar el número de la opción deseada.

4. Descripción de funciones principales

4.1 Registrar Nuevo Curso: Te trae un submenú del cual te consulta si quieres ingresar

1. nuevo curso con su nueva nota
2. ingresar nueva nota
3. eliminar nota
4. salir del submenú

```
--- Submenú Registrar Curso/Notas ---
1. Nuevo curso y nueva nota
2. Ingresar nueva nota
3. Eliminar nota
4. Salir del submenú
Seleccione opción: 1
Ingrese nombre del curso: fisica
Ingrese nota del semestre 1 (0-100): 54
Curso 'fisica' registrado con nota 54.0.
```

así mismo te dirá un mensaje que su curso y nota fue registrado y se ingresaran cuantas veces uno los ingreses a excepción de las notas que solo permite 5 semestre (unidades)

```
# ----- FUNCIONES AUXILIARES -----
def ingresar_nota(semestre):
    while True:
        try:
            nota = float(input(f"Ingrese nota del semestre {semestre} (0-100): "))
```

```

        if 0 <= nota <= 100:
            return round(nota,2)
        else:
            print("Nota inválida. Debe estar entre 0 y 100.")
    except:
        print("Ingrese un número válido.")
def ingresar_semestre():
    while True:
        try:
            sem = int(input("Ingrese semestre (1-5): "))
            if 1 <= sem <= 5:
                return sem
            else:
                print("Semestre inválido.")
        except:
            print("Ingrese un número válido.")

```

si uno ingresa mal el código o no le deja bien la tabulación le da sintaxis error y se lo remarca en el código como tenemos en el ejemplo de arriba

4.2 Mostrar todos los Cursos: En esta opción te muestra en forma de lista todos los cursos registrados junto con sus notas.

| Cursos y notas por semestres: | | | | | |
|-------------------------------|------|----|----|----|----|
| Curso | S1 | S2 | S3 | S4 | S5 |
| Diseño | 90 | - | - | - | - |
| Programación | 100 | - | - | - | - |
| Algoritmo | 60 | - | - | - | - |
| Precalculo | 59 | - | - | - | - |
| Matemática Discreta | 62 | - | - | - | - |
| fisica | 54.0 | - | - | - | - |

Así mismo para que salga con esa presentación se da a continuación ciertos códigos para que se tome de referencia es muy importante ver que se vea exactamente como se describe.

```

def mostrar_cursos():
    if not cursos:
        print("No hay cursos registrados.")
        return
    print("\nCursos y notas por semestres:")
    print(f"{'Curso':>25} {'S1':>5} {'S2':>5} {'S3':>5} {'S4':>5} {'S5':>5}")
    for c in cursos:
        notas = [n if n is not None else "-" for n in c['notas']]
        print(f"{c['nombre']:>25} {notas[0]:>5} {notas[1]:>5} {notas[2]:>5} {notas[3]:>5} {notas[4]:>5}")

```

4.3 Contar Cursos Aprobados y Reprobados: hay nuevamente un submenú y pregunta si quieres un promedio por todos los cursos o por cursos individual así mismo la opción de salir del submenú

| Curso | Promedio | Estado |
|---------------------|----------|-----------|
| ----- | ----- | ----- |
| Diseño | 90.00 | Aprobado |
| Programación | 100.00 | Aprobado |
| Algoritmo | 60.00 | Aprobado |
| Precalculo | 59.00 | Reprobado |
| Matemática Discreta | 62.00 | Aprobado |
| ----- | ----- | ----- |

```
--- Submenú Promedio General ---
1. Mostrar promedio de todos los cursos juntos
2. Mostrar promedio por curso
3. Salir submenú
```

4.4 Contar Cursos Aprobados y Reprobados: Cuenta cuántos cursos tienen nota mayor o igual a 60 (aprobados) y cuántos tienen nota menor (reprobados).

| Curso | Promedio | Estado |
|----------------------|----------|-----------|
| ----- | ----- | ----- |
| Diseño | 90.00 | Aprobado |
| Programación | 100.00 | Aprobado |
| Algoritmo | 60.00 | Aprobado |
| Precalculo | 59.00 | Reprobado |
| Matemática Discreta | 62.00 | Aprobado |
| ----- | ----- | ----- |
| Cursos aprobados: 4 | | |
| Cursos reprobados: 1 | | |

4.5 Buscar Curso por Nombre: Permite escribir el nombre de un curso y obtener su nota.

```
Seleccione una opción: 5
Ingrese nombre del curso: diseño
Diseño - Notas: [90, None, None, None, None] - Promedio: 90.0
```

4.6 Actualizar Notas de un Curso: pregunta el nombre del curso que semestre es (unidad) se ingresa la nueva nota y la actualiza.

```
Seleccione una opción: 6
Ingrese nombre del curso: diseño
Ingrese semestre (1-5): 1
Ingrese nota del semestre 1 (0-100): 89
Nota actualizada correctamente.
```

4.7 Eliminar Curso: Elimina un curso seleccionado de la lista.

```
Seleccione una opción: 7
Ingrese nombre del curso a eliminar: diseño
Curso eliminado exitosamente.
```

4.8 Ordenar Cursos por Nota: Muestra los cursos en orden descendente de nota.

| Cursos y notas por semestres: | | | | | |
|-------------------------------|-----|----|----|----|----|
| Curso | S1 | S2 | S3 | S4 | S5 |
| Programación | 100 | - | - | - | - |
| Matemática Discreta | 62 | - | - | - | - |
| Algoritmo | 60 | - | - | - | - |
| Precalculo | 59 | - | - | - | - |

4.9 Ordenar Cursos por Nombre: Muestra los cursos en orden alfabético (A–Z).

| Cursos y notas por semestres: | | | | | |
|-------------------------------|-----|----|----|----|----|
| Curso | S1 | S2 | S3 | S4 | S5 |
| Algoritmo | 60 | - | - | - | - |
| Matemática Discreta | 62 | - | - | - | - |
| Precalculo | 59 | - | - | - | - |
| Programación | 100 | - | - | - | - |

4.10 Buscar Curso por Nombre: se ingresa el nombre del curso luego te despliega el nombre buscado juntamente con las notas que se tengan ingresados

| Cursos y notas por semestres: | | | | | |
|--|-----|----|----|----|----|
| Curso | S1 | S2 | S3 | S4 | S5 |
| Algoritmo | 60 | - | - | - | - |
| Matemática Discreta | 62 | - | - | - | - |
| Precalculo | 59 | - | - | - | - |
| Programación | 100 | - | - | - | - |
| Ingrese nombre del curso a buscar: algoritmo | | | | | |
| Curso encontrado: Algoritmo - Notas: [60, None, None, None, None] - Promedio: 60.0 | | | | | |

4.11 Simular Cola de Solicitudes de Revisión: Administra solicitudes que los estudiantes envían para revisar su nota. Se guardan en una cola (FIFO: primero en entrar, primero en salir).

```
Estado actual de la cola de revisiones:
Diseño: Revisando
Precalculo: En proceso
Programación: Resuelto
algoritmo: En proceso
```

4.12 Mostrar Historial de Cambios: Muestra todas las modificaciones hechas las más recientes (registro, actualización, eliminación de cursos).

```
Historial de cambios recientes:
1. Se eliminó el curso diseño
2. Se corrigió nota curso Diseño semestre 1 de 90 a 89.0
```

4.13 Salir: Finaliza la ejecución del programa.

```
Seleccione una opción: 13
Gracias por usar el Gestor de Notas. ¡Vuelva pronto!
```


Como en este otro ejemplo si no se coloca los paréntesis en los lugares correctos tampoco podrá ejecutar el programa así mismo con las comillas, los dos puntos entre otras opciones que se tiene que tener cuidado con respecto a donde tiene que ir cada detalle

```
mostrar_historial()
elif opcion=="13":
    print("Gracias por usar el Gestor de Notas. ¡Vuelva pronto!")
    break
```

5. Ejemplo de uso

Entrada (usuario):

1. nuevo curso con su nueva nota: mate nota 80
2. ingresar nueva nota: mate 89
3. eliminar nota mate 80
4. salir del submenú

Salida (programa):

Curso registrado con éxito.

Entrada (usuario):

Seleccione menú una opción: 2 te desplegara todos los cursos registrados con notas

Salida (programa):

1. Matemáticas - Nota: 75
2. Fisica - Nota: 60

6. Recomendaciones

- Usar nombres de curso claros (ejemplo: "Lengua Española", no solo "Lengua").
- Ingresar notas entre 0 y 100.
- Revisar el historial para confirmar los cambios realizados.

7. Conclusión

La implementación de este programa es para que sea mas eficiente y con un buen control tanto para el docente como para el estudiante. Se le recomienda siempre hacer un mantenimiento al programa para que no se sature asi mismo como actualizaciones a futuras y siempre manter a la vanguardia el centro eduactivo, junto con las siguientes opciones que se muestran acontinuacion

- Registrar nuevos cursos.
- Consultar la lista de cursos guardados.
- Calcular promedios generales.
- Contar cursos aprobados y reprobados.
- Buscar, actualizar y eliminar cursos.
- Ordenar cursos por nombre o por nota.
- Administrar solicitudes de revisión mediante una cola.
- Revisar historial de cambios realizados.

Así mismo se recomiendo estar al pendiente de las nuevas actualizaciones del programa para que se vaya mejorando y siempre este a la vanguardia de cualquier otro lugar de aprendizaje y que este entre los mejores del país o de ser posible del mundo

8. Bibliografía

Creador del programa, El señor Alan Fuentes creada desde julio 2025