

**Universidad** Mariano Gálvez de Guatemala

**Sede** San Juan Sacatepéquez

**Faculta** Ingeniería en sistemas

**Docente** Miguel Catalán

**Curso** Algoritmo



José Edy Camey Boror

Sección B

7590-25-9906

17/10/2025

# Manual Técnico

Es una forma de explicar y visualizar las funciones que se encuentran disponibles dentro del código Python, se muestran los códigos generales del sistema operativo y de funciones básicas, que sea una guía por si desean hacer modificaciones u otras acciones

- **Requisitos**

1. Debes tener un computador o un dispositivo que pueda ejecutar archivos de formato Python
2. Luego debes tener instalado el programa (un compilador de Python)
3. Luego guiarte de este manual para saber cómo funciona el archivo.py

- mostrare el siguiente menú en en el código PY

```
#aqui esta el menu de revision del proyecto
while True:
    print("          MENÚ PRINCIPAL          ")
    print("1. Registrar curso nuevo y nota")
    print("2. Mostrar todas las notas")
    print("3. Calcular promedio general")
    print("4. Buscar nota por curso")
    print("5. Eliminacion de curso")
    print("6. Actualizar la nota de un curso")
    print("7. Mostrar cursos aprobados")
    print("8. historial de cursos")
    print("9. Solicitud de Revicion de Cursos")
    print("10. Ordenacion de cursos en Burbuja")
    print("11. Ordenacion de cursos en insercion")
    print("12. Salir")
    opcion = input("Elige una opción: ")
```

## Muestra de las condiciones en el menú con funciona de ELIF

```
if opcion == "1":
    registrar_curso_y_nota()
elif opcion == "2":
    mostrar_todas_las_notas()
elif opcion == "3":
    calcular_promedio_general()
elif opcion == "4":
    buscar_nota_por_curso()
elif opcion == "5":
    eliminar_curso()
elif opcion == "6":
    actualizar_nota()
elif opcion == "7":
    cursos_aprobados()
elif opcion == "8":
    historial_curso()
elif opcion == "9":
    cola_revision()
elif opcion == "10":
    cursos_ordenados = ordenar_burbuja(cursos)
    print("\nCursos ordenados por nota de menor a mayor:")
    for c in cursos_ordenados:
        print(f"{c['curso']}: {c['nota']}")
    print()
elif opcion == "11":
    cursos_ordenados = ordenar_insercion(cursos)
    print("Cursos ordenados por nombre:")
    for c in cursos_ordenados:
        print(f"{c['curso']}: {c['nota']}")
    print(" ")
elif opcion == "12":
    print("Saliendo...")
    print(" ")
    break
else:
    print("Opción no válida.\n")
    print(" ")
```

## 1. def registrar\_curso\_y\_nota():

- La función Permitirá registrar el curso que desees y te dará una lista sugerida con cursos ya registrados, donde tendrás la opción de colocar la nota y guardar el curso registrado

```
def registrar_curso_y_nota():
    for i, elem in enumerate(mi_lista_cursos, start=1):
        print(i, ".", elem)

    try:
        indice = int(input("Selecciona el número del curso: "))
        if 1 <= indice <= len(mi_lista_cursos):
            curso = mi_lista_cursos[indice-1]
        else:
            print("Número fuera de lo establecido.\n")
            continue
    except ValueError:
        print("Debes ingresar un número válido.\n")
        continue

    else:
        curso = input("Ingrese el curso: ").strip()
        if not curso:
            print("El curso no puede estar vacío.\n")
            continue

    try:
        nota = float(input("Ingrese la nota del curso (0 a 100): "))
        if 0 <= nota <= 100:
            historial.append(curso) # guardamos en el historial
            cursos.append({"curso": curso, "nota": nota})
            print("Registro exitoso .....")
            print("Nombre del Curso", curso, "registrado con la nota de ", nota, "\n")

        else:
            print("Error: la nota debe estar entre 0 y 100.\n")
            continue
    except ValueError:
        print("Error: debes ingresar un número válido.\n")
        continue

    opcion = input("¿Registrar otro curso? (S/N): ").strip().lower()
    if opcion != "s":
        break
```

## 2. def mostrar\_todas\_las\_notas():

- Función que Permitirá visualizar las notas registradas que el usuario a colocado
- Mostrará todos los cursos con sus notas correspondiente

```
def mostrar_todas_las_notas():
    if cursos:
        print("\nRegistro de notas por curso:")
        for i, c in enumerate(cursos, start=1):
            print(f"{i}. {c['curso']}:{c['nota']}")
        print()
    else:
        print("No se han registrado cursos.\n")
```

### 3. def calcular\_promedio\_general():

- la función Permitirá ver todas las notas y calculará el promedio general de todas las notas

```
def calcular_promedio_general():
    if cursos:
        promedio = sum(c["nota"] for c in cursos) / len(cursos)
        print(" El promedio general es: \n", promedio)
    else:
        print("No fue posible calcular el promedio (no hay cursos registrados).\n")
```

### 4. def buscar\_nota\_por\_curso():

- La función Dará la opción de buscar el curso
- Te mostrara el curso y la nota registrada

```
def buscar_nota_por_curso():
    if not cursos:
        print("No has registrado cursos.\n")
        return
    curso_buscar = input("Ingrese el curso a buscar: ")
    for c in cursos:
        if c["curso"].lower() == curso_buscar.lower():
            print(f"Registro encontrado: {c['curso']}, Nota: {c['nota']}\n")
            return
    print("Curso no encontrado.\n")
```

### 5. def eliminar\_curso():

- La función Te dará la opción de eliminar un curso , el que desees
- Te pide el nombre y lo elimina

```
def eliminar_curso():
    if not cursos:
        print("No hay cursos para eliminar.\n")
        return
    eliminar = input("Nombre del curso a eliminar: ")
    for i in range(len(cursos)):
        if cursos[i]["curso"].lower() == eliminar.lower():
            cursos.pop(i)
            print("Curso", eliminar, " eliminando.\n")
            return
    print("No se encontró ese curso.\n")
```

## 6. def historial\_curso():

- Mostrará los nombre de los cursos que se llevan en un registro
- Te pedirá la opción y mostrara el Historial

```
historial = []

def historial_curso():
    if historial:
        print("\nHistorial de cursos:")
        for i, curso in enumerate(historial, start=1):
            print(i, ".", curso)
    else:
        print("El historial está vacío.\n")
```

## 7. def cola\_revision():

- La función Te pedirá un curso registrado
- Te mostrara como una simulación de cola de revisión de cursos

```
def cola_revision():
    cola = [] # Lista para simular la cola de cursos

    # Ingreso de cursos que va a Revisar
    while True:
        curso = input("Ingrese curso para revisión y 'fin' para terminar:\n> ")
        if curso.lower() == "fin":
            break
        # Agregar al final de la cola
        cola.append(curso)

    # Procesar la cola que esta mandando el usuario
    print("\nProcesando las solicitudes:")
    while cola:
        curso_actual = cola.pop(0) # Sacar el primer elemento (FIFO)
        print("Revisando el curso de: ", curso_actual)
```

## 8. def ordenar\_burbuja(cursos):

- La función Ordena los cursos conforme estén registrado de menor a mayor nota
- Busca este orden debido al modularidad de Burbuja

```
# aqui comienza la opcion de ordenado de Burbuja que de notas
def ordenar_burbuja(cursos):
    n = len(cursos)
    for i in range(n):
        for h in range(0, n-i-1):
            if cursos[h]["nota"] > cursos[h+1]["nota"]:
                cursos[h], cursos[h+1] = cursos[h+1], cursos[h]
    return cursos
```

## 9. def ordenar\_insercion(cursos):

- Ordena conforme estén los cursos de forma alfabética
- Conforme estén en el abecedario

```
def ordenar_insercion(cursos):
    n = len(cursos)
    for i in range(1, n):
        clave = cursos[i]
        m = i - 1
        # Comparar los nombres de los cursos alfabéticamente
        while m >= 0 and cursos[m]["curso"].lower() > clave["curso"].lower():
            cursos[m + 1] = cursos[m]
            m -= 1
        cursos[m + 1] = clave
        print(" ")
    return cursos
print(" ")
```

## 10. def cursos\_aprobados():

- Muestra al momento de pedir los cursos aprobados y reprobados
- Los muestra en la consola

```
def cursos_aprobados():
    print("\n--- Cursos Aprobados ---")
    aprobados = [c for c in cursos if c['nota'] >= 61]
    if aprobados:
        for c in aprobados:
            print("Curso:", c['curso'], "Nota:", c['nota'])
    else:
        print("Ningún curso aprobado.")
    print()

# esta opción va permitir al usuario actualizar su curso final y su historial final
```

## 11. def actualizar\_nota():

- La función actualiza el sistema de notas y lo modifica acoplándolo al cliente
- Conforme estén quiera la nota

```
def actualizar_nota():
    if not cursos:
        print("\nNo has registrado cursos aún.\n")
        return

    print("\n Cursos registrados:")
    for i, c in enumerate(cursos, 1):
        print(f"{i}. {c['curso']} (Nota actual: {c['nota']})")

    curso_buscar = input("\nIngresa el nombre de curso que quieres actualizar:\n> ").strip()
    encontrado = False

    for c in cursos:
        if c["curso"].lower() == curso_buscar.lower():
            try:
                nota_actualizada = float(input("Ingresa la nueva nota para el curso:\n> "))
                c["nota"] = nota_actualizada
                print(f"\nLa nota del curso {c['curso']} ha sido actualizada a {nota_actualizada}.\n")
                encontrado = True
                break
            except ValueError:
                print(" Error: la nota debe ser un número válido.\n")
                return

    if not encontrado:
        print(f" No se encontró el curso '{curso_buscar}'.\n")
```

## 12. Salir

- Finalizara todo el programa

```
elif opcion == "12":
    print("Saliendo...")
    print("")
    break
else:
    print("Opción no válida.\n")
    print("")
```



## Funciones generales

- almacena todos los cursos registrados y sus notas.
- Cada función está separada una de otra.
- El menú principal contiene un ciclo while que hace que el programa termine hasta que el usuario elija salir.
- Explicación del uso de listas, pilas, colas, etc.
- Lista cursos: permite almacenar y recorrer los cursos registrados.
- (LIFO) pila: registra los cambios del sistema (registro, actualización y eliminación).
- (FIFO) cola: simula solicitudes de revisión que se atienden en el orden de llegada.
- Justificación de los algoritmos de ordenamiento
- Burbuja: se utilizó para ordenar los cursos por notas de una forma simple ya que es ideal para listas pequeñas.
- Inserción: se utilizó para ordenar los cursos de manera alfabéticamente por nombre ya que es eficiente para listar parcialmente ordenadas.

## Pseudocodigo General de muestra del Menu

INICIO

DEFINIR opcion COMO CADENA

DEFINIR cursos COMO LISTA VACÍA

REPETIR

IMPRIMIR "=====

IMPRIMIR "      MENÚ PRINCIPAL      "

IMPRIMIR "=====

IMPRIMIR "1. Registrar curso y nota"

IMPRIMIR "2. Mostrar todas las notas"

IMPRIMIR "3. Calcular promedio general"

IMPRIMIR "4. Buscar nota por curso"

IMPRIMIR "5. Eliminar curso"

IMPRIMIR "6. Actualizar nota"

IMPRIMIR "7. Mostrar cursos aprobados"

IMPRIMIR "8. Historial de cursos (PILA)"

IMPRIMIR "9. Cola de revisión"

IMPRIMIR "10. Ordenar cursos por nota (Burbuja)"

IMPRIMIR "11. Ordenar cursos por nombre (Inserción)"

IMPRIMIR "12. Salir"

IMPRIMIR "=====

IMPRIMIR "Seleccione una opción:"

LEER opcion

SEGÚN opcion HACER

CASO "1":

    LLAMAR registrar\_curso\_y\_nota()

    FIN CASO

CASO "2":

    LLAMAR mostrar\_todas\_las\_notas()

    FIN CASO

CASO "3":

    LLAMAR calcular\_promedio\_general()

    FIN CASO

CASO "4":

    LLAMAR buscar\_nota\_por\_curso()

    FIN CASO

CASO "5":

    LLAMAR eliminar\_curso()

    FIN CASO

CASO "6":

    LLAMAR actualizar\_nota()

    FIN CASO

CASO "7":

    LLAMAR cursos\_aprobados()

    FIN CASO

CASO "8":

```

    LLAMAR historial_curso()
    FIN CASO

CASO "9":
    LLAMAR cola_revision()
    FIN CASO
CASO "10":
    cursos_ordenados ← LLAMAR ordenar_burbuja(cursos)
    IMPRIMIR "Cursos ordenados por nota de menor a mayor:"
    PARA CADA c EN cursos_ordenados HACER
        IMPRIMIR c.curso, ":", c.nota
    FIN PARA
    FIN CASO
CASO "11":
    cursos_ordenados ← LLAMAR ordenar_insercion(cursos)
    IMPRIMIR "Cursos ordenados por nombre:"
    PARA CADA c EN cursos_ordenados HACER
        IMPRIMIR c.curso, ":", c.nota
    FIN PARA
    FIN CASO
CASO "12":
    IMPRIMIR "Saliendo del sistema..."
    SALIR DEL BUCLE
    FIN CASO
DE OTRO MODO:
    IMPRIMIR "Opción no válida. Intente nuevamente."
FIN SEGÚN
HASTA QUE opcion = "12"
FIN

```

Hasta Aquí El manual Tecnico, Espero sea Compresivo

saludos