

# PLATAFORMA DE GESTIÓN Y EVALUACIÓN DE CURSOS ONLINE

MANUAL DE USUARIO



Alejandro de León - 1502425  
Carlos Cholutío - 1517925

## Descripción General

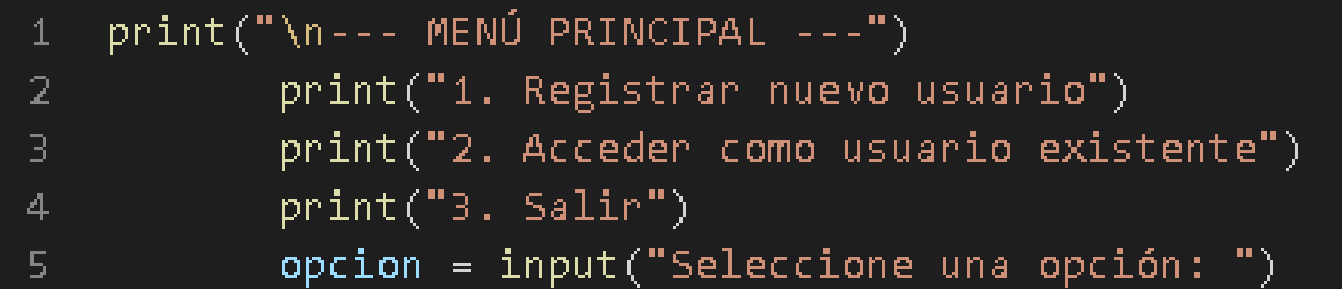
Esta plataforma permite gestionar cursos, usuarios (estudiantes e instructores) y evaluaciones de forma automatizada. El sistema soporta registro de usuarios, inscripción en cursos, creación de evaluaciones (exámenes, tareas, proyectos), registro de calificaciones y reportes de estudiantes con bajo rendimiento.



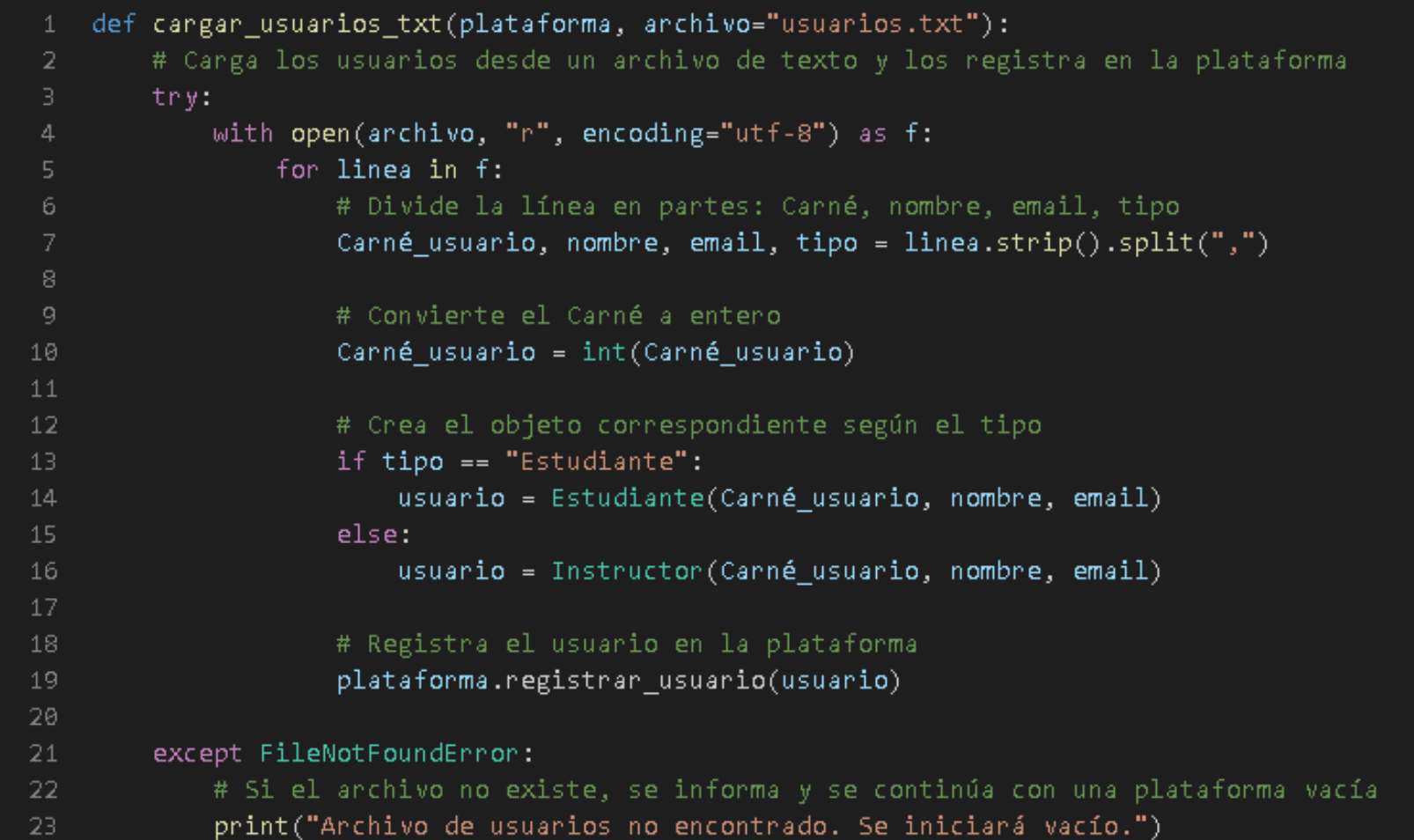
# Inicio del Sistema

**Al ejecutar el programa, se realizan dos acciones automáticas:**

1. Se carga el archivo usuarios.txt para recuperar usuarios registrados.
2. Se abre el menú principal para interactuar con el sistema



```
1 print("\n--- MENÚ PRINCIPAL ---")
2     print("1. Registrar nuevo usuario")
3     print("2. Acceder como usuario existente")
4     print("3. Salir")
5     opcion = input("Seleccione una opción: ")
```



```
1 def cargar_usuarios_txt(plataforma, archivo="usuarios.txt"):
2     # Carga los usuarios desde un archivo de texto y los registra en la plataforma
3     try:
4         with open(archivo, "r", encoding="utf-8") as f:
5             for linea in f:
6                 # Divide la línea en partes: Carné, nombre, email, tipo
7                 Carné_usuario, nombre, email, tipo = linea.strip().split(",")
8
9                 # Convierte el Carné a entero
10                Carné_usuario = int(Carné_usuario)
11
12                # Crea el objeto correspondiente según el tipo
13                if tipo == "Estudiante":
14                    usuario = Estudiante(Carné_usuario, nombre, email)
15                else:
16                    usuario = Instructor(Carné_usuario, nombre, email)
17
18                # Registra el usuario en la plataforma
19                plataforma.registrar_usuario(usuario)
20
21    except FileNotFoundError:
22        # Si el archivo no existe, se informa y se continúa con una plataforma vacía
23        print("Archivo de usuarios no encontrado. Se iniciará vacío.")
```





# Menú Principal

## 1. Registrar nuevo usuario

- El sistema solicita si el usuario es Estudiante (E) o Instructor (I).
- Luego pide:
  - Carné (solo números)
  - Nombre (sin números ni símbolos)
  - Correo electrónico (debe contener @ y .)
- El usuario se guarda en usuarios.txt.

## 2. Acceder como usuario existente

- Se solicita el Carné.
- Si el Carné existe, se abre el menú correspondiente según el tipo de usuario.

## 3. Salir

- Se guarda la información de cursos en cursos.txt.
- Se guarda el historial de notas en notas.txt.
- El sistema finaliza.




```
1  print("\n--- MENÚ PRINCIPAL ---")
2      print("1. Registrar nuevo usuario")
3      print("2. Acceder como usuario existente")
4      print("3. Salir")
5      opcion = input("Seleccione una opción: ")
```





# Menú Estudiante

- **Ver cursos inscritos:** muestra los nombres de los cursos en los que está inscrito.
- **Ver calificaciones:** muestra las notas asignadas por curso.
- **Ver perfil:** muestra todos los datos del estudiante.
- **Salir:** regresa al menú principal.



```
1 print(f"\n--- MENÚ ESTUDIANTE ({estudiante.nombre}) ---")
2     print("1. Ver cursos inscritos")
3     print("2. Ver calificaciones")
4     print("3. Ver perfil")
5     print("4. Salir")
6     opcion = input("Seleccione una opción: ")
```



# Menú Instructor

- **1. Crear curso**
  - Solicita código y nombre del curso.
  - Asocia el curso al instructor y lo guarda en la plataforma.
- **2. Crear evaluación**
  - Solicita código del curso, nombre de la evaluación y tipo:
    - Examen (requiere duración)
    - Tarea (requiere fecha de entrega)
    - Proyecto (requiere porcentaje)
  - La evaluación se asocia al curso.
- **3. Asignar calificación**
  - Selecciona curso y evaluación.
  - Solicita Carné del estudiante y la nota.
  - La nota se valida y se guarda en el historial (notas.txt).
- **4. Inscribir estudiante en curso**
  - Solicita Carné del estudiante y código del curso.
  - Inscribe al estudiante si el curso pertenece al instructor.
- **5. Ver cursos impartidos**
  - Muestra los nombres de los cursos que imparte el instructor.
- **6. Ver perfil**
  - Muestra todos los datos del instructor.
- **7. Ver historial de notas**
  - Muestra todas las notas registradas en el sistema.
- **8. Generar reporte de promedio bajo**
  - Solicita código del curso.
  - Muestra estudiantes con promedio menor a 60 (o el umbral que se defina).
- **9. Salir**
  - Regresa al menú principal.



```
1  print(f"\n--- MENÚ INSTRUCTOR ({instructor.nombre}) ---")
2      print("1. Crear curso")
3      print("2. Crear evaluación")
4      print("3. Asignar calificación a estudiante")
5      print("4. Inscribir estudiante en curso")
6      print("5. Ver cursos impartidos")
7      print("6. Ver perfil")
8      print("7. Ver historial de notas")
9      print("8. Generar reporte de promedio bajo")
10     print("9. Salir")
11     opcion = input("Seleccione una opción: ")
```







# Archivos Generados

- **usuarios.txt:** contiene Carné, nombre, email y tipo de usuario.
- **cursos.txt:** contiene información detallada de cada curso.
- **notas.txt:** contiene el historial completo de calificaciones.



cursos.txt



notas.txt




usuarios.txt





# Validaciones Importantes

- Carné y código de curso deben ser numéricos.
- Nombres no deben contener números ni símbolos.
- Correos deben tener @ y ..
- Las notas deben estar entre 0 y 100.



```
1  # =====
2  # VALIDACIONES DE ENTRADA
3  # =====
4  def pedir_Carné():
5  def pedir_nombre():
6  def pedir_email():
7  def pedir_codigo():
8  def pedir_nombre_curso():
9  def pedir_nombre_evaluacion():
10 def pedir_tipo_evaluacion():
11 def validar_nota(nota):
12 def registrar_historial(estudiante, evaluacion, nota):
13 def asignar_nota_a_estudiante(evaluacion, estudiante, nota):
14 def tipo_de_evaluacion(evaluacion):
15 def reporte_promedio_bajo(curso, umbral=60):
```







# ¡Gracias por su atención!

MANUAL DE USUARIO



PROYECTO NO.01

