



MANUAL TÉCNICO

POR:

Juan Diego Vinueza Olayes
Sebastián Alejandro León Jácome
Jeremy Gabriel Cobos Villalba
Fernando Alexander Cárdenas Chulco

Carrera:

Ingeniería Industrial

Asignatura:

Lógica de Programación

Fecha de Entrega:

23/06/2025

Contenido

<i>Descripción General del sistema</i>	2
<i>Requerimientos Técnicos:</i>	2
<i>Aplicaciones y bibliotecas usadas para la creación de nuestro sistema</i>	3
<i>Estructura de archivo de proyectos:</i>	3
<i>Lógica y arquitectura del código:</i>	3

Índice de gráficos

<i>Menú principal</i>	7
<i>Registro asistencia manual</i>	7
<i>registro asistencia QR</i>	8
<i>lector QR ventana</i>	8
<i>registro de asistencias</i>	9
<i>archivo CSV</i>	10

Sistema de Registro de Asistencia en Python

Descripción General del sistema

El sistema de registro de asistencia desarrollado en Python por nuestro grupo es una solución sencilla, funcional y eficiente para el control de entrada de usuarios a una institución, clase o evento. A través de una interfaz por consola, permite a los usuarios registrar su asistencia de manera manual o automática usando códigos QR. Los datos quedan almacenados en un archivo CSV que puede ser consultado posteriormente por los administradores o cualquier personal docente autorizado.

Su propósito principal es automatizar el proceso de toma de asistencia, reducir errores humanos y mejorar la organización de registros. El sistema también permite la visualización de los registros bajo credenciales de acceso, agregando un nivel básico de seguridad y su llamativo uso de QR para el registro la hace una gran novedad y muy útil, en cualquier caso.

Requerimientos Técnicos:

Software: Los requerimientos que usamos en el código es Python en la versión 3.10 o superior a esa y un editor de código que en este caso sería Visual Studio Code

Librerías: las librerías necesarias que usted necesitara para hacer funcionar este código se instalan colocando en el terminal: `pip install pandas opencv-python numpy`

Hardware: Los recursos físicos que necesita es una computadora con sistema operativo Windows ya que este código está destinado a abrir y encontrar los archivos con las vías de

Windows y también es necesario para que la opción 2 funcione (Código QR) una cámara web funcional puede ser integrada o externa.

Aplicaciones y bibliotecas usadas para la creación de nuestro sistema

El lenguaje de programación que usamos es Python en el entorno de desarrollo Visual Studio Code y las librerías que utilizamos son:

- **pandas:** Permite manejar estructuras de datos, leer y escribir archivos CSV con facilidad.
- **opencv-python (cv2):** Utilizada para acceder a la cámara y detectar códigos QR.
- **datetime:** Se utiliza para capturar y registrar la fecha y la hora exacta en la que se realiza una asistencia.
- **os:** Proporciona acceso a funcionalidades del sistema operativo como rutas de archivos.
- **platform:** Detecta el sistema operativo en uso para abrir archivos con la aplicación predeterminada.
- **subprocess:** Ejecuta comandos del sistema para abrir el archivo CSV dependiendo del sistema operativo.
- **numpy:** Dependencia requerida por OpenCV para operaciones numéricas.

Estructura de archivo de proyectos:

Descripción

control_asistencia.py: Archivo principal que contiene todo el código fuente del sistema

asistencia.csv: Archivo generado automáticamente para almacenar los registros de asistencia

Adicionalmente, el proyecto va a incluir documentos de soporte como el manual de usuario y presentación

Lógica y arquitectura del código:

El código está dividido funcionalmente en tres secciones principales: registro manual, registro por QR y visualización de registros. A continuación, se detalla cada parte:

```
import pandas as pd
import os
from datetime import datetime
import cv2
```

```
import numpy as np
import platform
import subprocess
```

¿Para qué sirve?

- pandas: para leer y escribir archivos .csv.
- os: para verificar si el archivo existe y manejar rutas.
- datetime: para registrar la fecha y hora de la asistencia.
- cv2: para usar la cámara y detectar códigos QR (de OpenCV).
- numpy: lo necesita cv2 internamente (aunque tú no lo usas directamente).
- platform y subprocess: para abrir el archivo dependiendo del sistema operativo.

```
archivo_csv = 'asistencia.csv'
```

sirve para definir el nombre del archivo donde se guarda la asistencia.

```
if not os.path.exists(archivo_csv):
    df = pd.DataFrame(columns=['Nombre', 'Fecha', 'Hora'])
    df.to_csv(archivo_csv, index=False)
```

Sirve para crear el archivo asistencia.csv con las columnas adecuadas si aún no existe.

```
def registrar_asistencia_manual(nombre):
    ahora = datetime.now()
```

sirve para guardar la fecha y hora actual.

```
    fecha = ahora.strftime("%Y-%m-%d")
    hora = ahora.strftime("%H:%M:%S")
```

Convierte la fecha y hora a texto legible como por ejemplo 2025-06-05, 14:22:01.

```
df = pd.read_csv(archivo_csv)
```

Abre el archivo asistencia.csv y carga su contenido en memoria como tabla.

```
ya_registrado = ((df['Nombre'] == nombre) & (df['Fecha'] == fecha)).any()
```

compara con nombre y fecha si esa persona ya se registró hoy.

```
if not ya_registrado:
    nuevo = {'Nombre': nombre, 'Fecha': fecha, 'Hora': hora}
    df = pd.concat([df, pd.DataFrame([nuevo])], ignore_index=True)
    df.to_csv(archivo_csv, index=False)
    print(f"Asistencia registrada para {nombre} a las {hora}")
```

Si no está registrado aún crea un diccionario con el nombre la fecha y la hora de la persona y posteriormente agrega ese registro a la tabla, para después agregarlo al archivo csv y terminar mostrándole la confirmación al usuario

```
else:
    print(f"{nombre} ya registró asistencia hoy.")
```

Si se encuentra igual el nombre y la fecha muestra al usuario que la asistencia ya se registro el día de hoy

```
def registrar_asistencia_qr():  
    cap = cv2.VideoCapture(0)
```

define variable de registrar asistencia y cap indica que se abra la cámara y muestre un video para poder indicar el código QR a la cámara

```
    detector = cv2.QRCodeDetector()
```

Prepara un detector de códigos QR para poder identificarlo

```
    while True:  
        _, frame = cap.read()
```

Lee todos los frames de la cámara en tiempo real, en otras palabras, captura cada imagen que este en la cámara.

```
    datos, puntos, _ = detector.detectAndDecode(frame)
```

lee todas las capturas intentando buscar un código QR y decodificarlo.

```
    cv2.imshow("Lector QR", frame)
```

proyecta lo que esta leyendo la imagen de la cámara en una ventana para que no haya confusión de a donde apuntar el QR

```
    if datos:  
        nombre = datos.strip()  
        print(f"QR detectado: {nombre}")  
        registrar_asistencia_manual(nombre)  
        break
```

Si logro detectar algo se guarda el texto decodificado en la variable nombre e imprime QR detectado para informar al usuario que el QR ha sido identificado y llama a la variable registrar asistencia manual para volver a hacer todo el proceso de guardado y el break termina el bucle

```
    if cv2.waitKey(1) == ord('q'):  
        break
```

Indica que si el usuario presiona la tecla "q" puede salir de la ventana y de la opción 2 sin necesidad de hacer escanear un código QR y después break termina el bucle

```
    cap.release()  
    cv2.destroyAllWindows()
```

Esto quiere decir que va a liberar la cámara y cierra la ventana que mostraba la imagen de la misma

```
def ver_lista_asistencias():
```

```

usuario = input("Usuario: ")
while usuario != "UserUIDE":
    usuario = input("Usuario incorrecto, pruebe otra vez: ")

```

Define una nueva variable ver lista de asistencias en donde se crea una un login para el cual se crea un usuario que es UserUIDE y se crea un bucle mientras el usuario no sea el mismo no se puede acceder.

```

contraseña = input("Contraseña: ")
while contraseña != "Uide.asu.123":
    contraseña = input("Contraseña incorrecta, pruebe otra vez: ")

```

Funciona de la misma manera que el usuario para mas seguridad y la contraseña es Uide.asu.123

```

print("Acceso permitido. Lista de asistencias:\n")
df = pd.read_csv(archivo_csv)
print(df)

```

Informa que el usuario ha colocado bien el login y manda un mensaje de confirmación para después proceder a buscar el archivo csv y enseña todo el registro de asistencias.

```

ruta = os.path.abspath(archivo_csv)
os.startfile(ruta)

```

crea una variable ruta que busca la ruta completa al archivo CSV y lo que hace os es abrir este documento

```

print("Bienvenidos al registro de asistencia")

print("1 - Registro de asistencia manual")
print("2 - Registro de asistencia con código QR")
print("3 - Ver la lista de asistencias")

opcion = input("Elija una opción (1-3): ")

if opcion == "1":
    print("Registro manual seleccionado")
    nombre = input("Ingrese su nombre completo: ").strip()
    registrar_asistencia_manual(nombre)

elif opcion == "2":
    print("Registro por código QR seleccionado")
    registrar_asistencia_qr()

elif opcion == "3":
    print("Acceso a la lista de asistencias")
    ver_lista_asistencias()

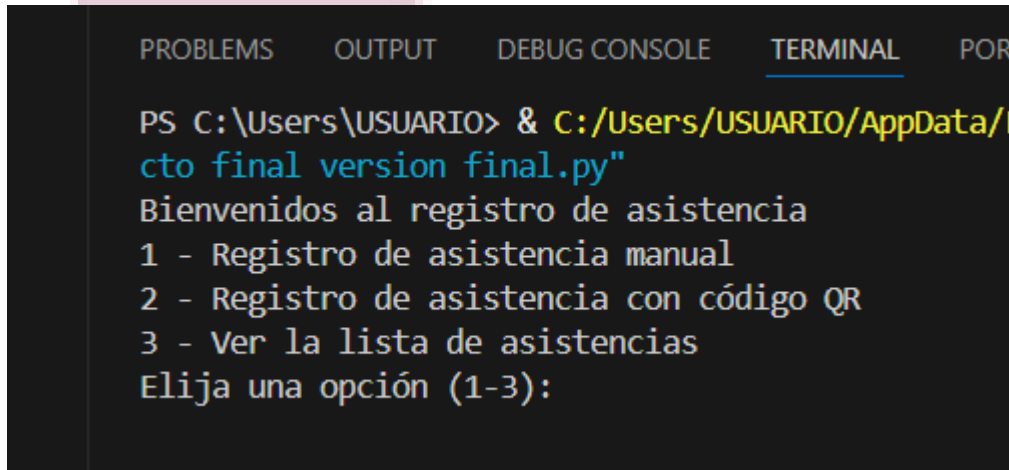
else:
    print("Ingrese una opción válida (1, 2 o 3)")

```

Realiza todo el menú, incluyendo la bienvenida y generando las opciones y para cada opción que el usuario elija va llamando a la variable definida anteriormente y al final si el usuario no escoge una opción correcta imprime que no es válida y cierra el programa

Glosario

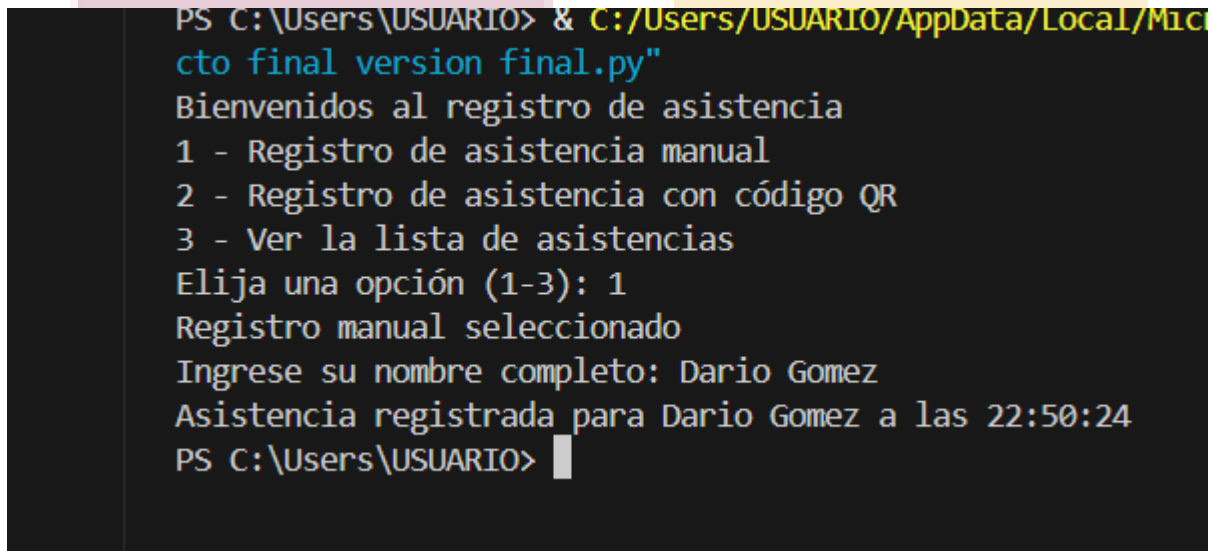
Menú principal:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POR
PS C:\Users\USUARIO> & C:/Users/USUARIO/AppData/Local/Microsoft/Windows/PowerShell/Scripts/ctofinal version final.py"
Bienvenidos al registro de asistencia
1 - Registro de asistencia manual
2 - Registro de asistencia con código QR
3 - Ver la lista de asistencias
Elija una opción (1-3):
```

Ilustración 1 Menú principal

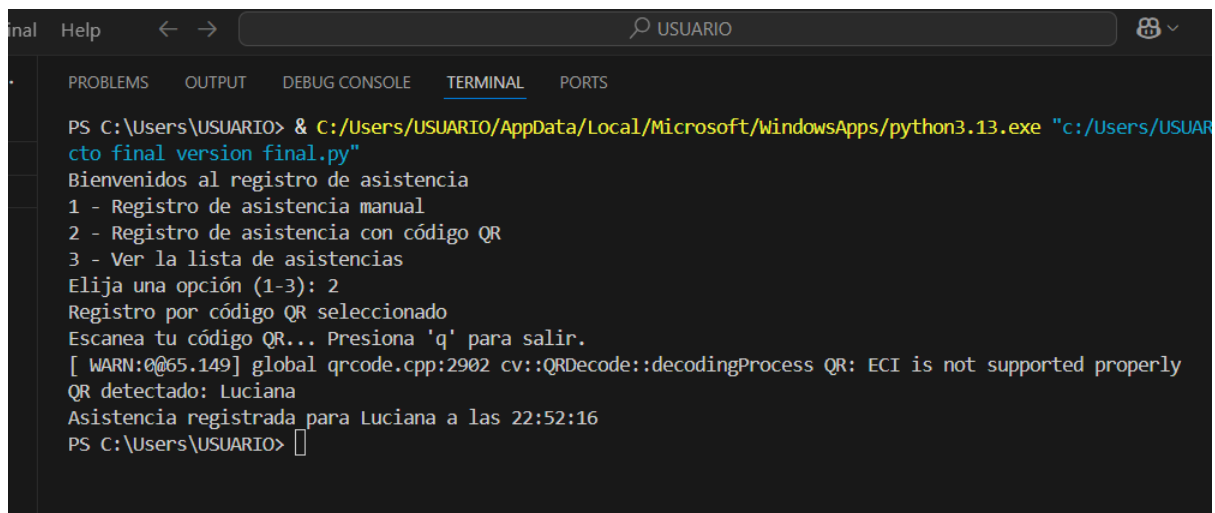
Registro de asistencia manual:



```
PS C:\Users\USUARIO> & C:/Users/USUARIO/AppData/Local/Microsoft/Windows/PowerShell/Scripts/ctofinal version final.py"
Bienvenidos al registro de asistencia
1 - Registro de asistencia manual
2 - Registro de asistencia con código QR
3 - Ver la lista de asistencias
Elija una opción (1-3): 1
Registro manual seleccionado
Ingrese su nombre completo: Dario Gomez
Asistencia registrada para Dario Gomez a las 22:50:24
PS C:\Users\USUARIO>
```

Ilustración 2 Registro asistencia manual

Registro de asistencia QR:



```
PS C:\Users\USUARIO> & C:/Users/USUARIO/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/USUARIO/ctofinal/version final.py"
Bienvenidos al registro de asistencia
1 - Registro de asistencia manual
2 - Registro de asistencia con código QR
3 - Ver la lista de asistencias
Elija una opción (1-3): 2
Registro por código QR seleccionado
Escanea tu código QR... Presiona 'q' para salir.
[ WARN:0@65.149] global qrcode.cpp:2902 cv::QRDecode::decodingProcess QR: ECI is not supported properly
QR detectado: Luciana
Asistencia registrada para Luciana a las 22:52:16
PS C:\Users\USUARIO>
```

Ilustración 3 registro asistencia QR

Ventana Lector QR:

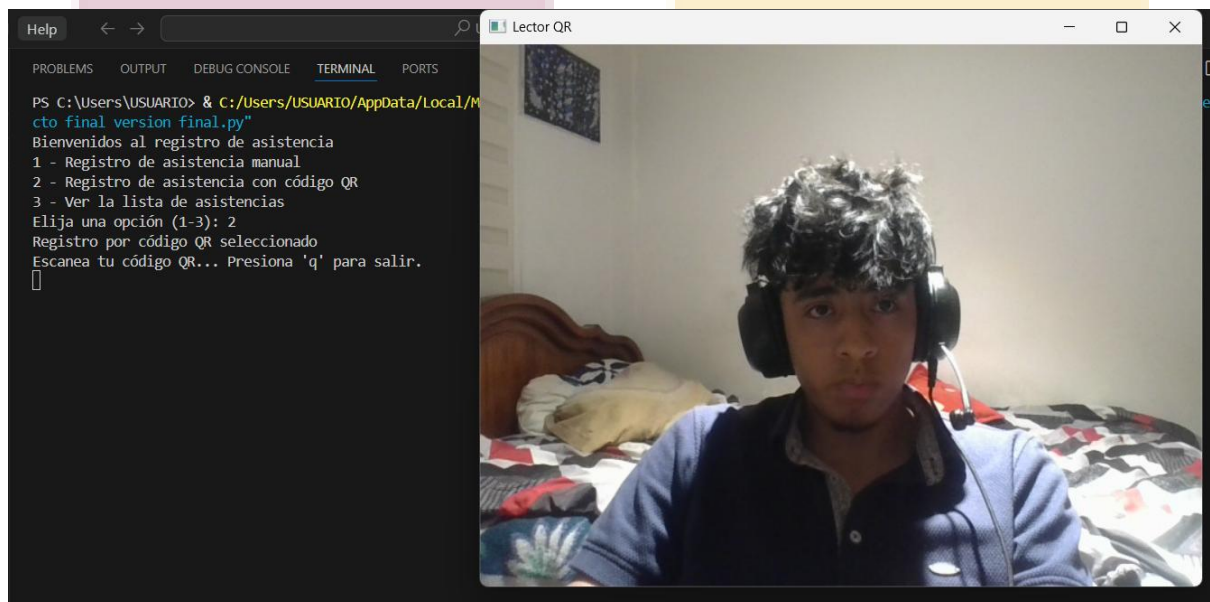


Ilustración 4 lector QR ventana

Registro de asistencias:

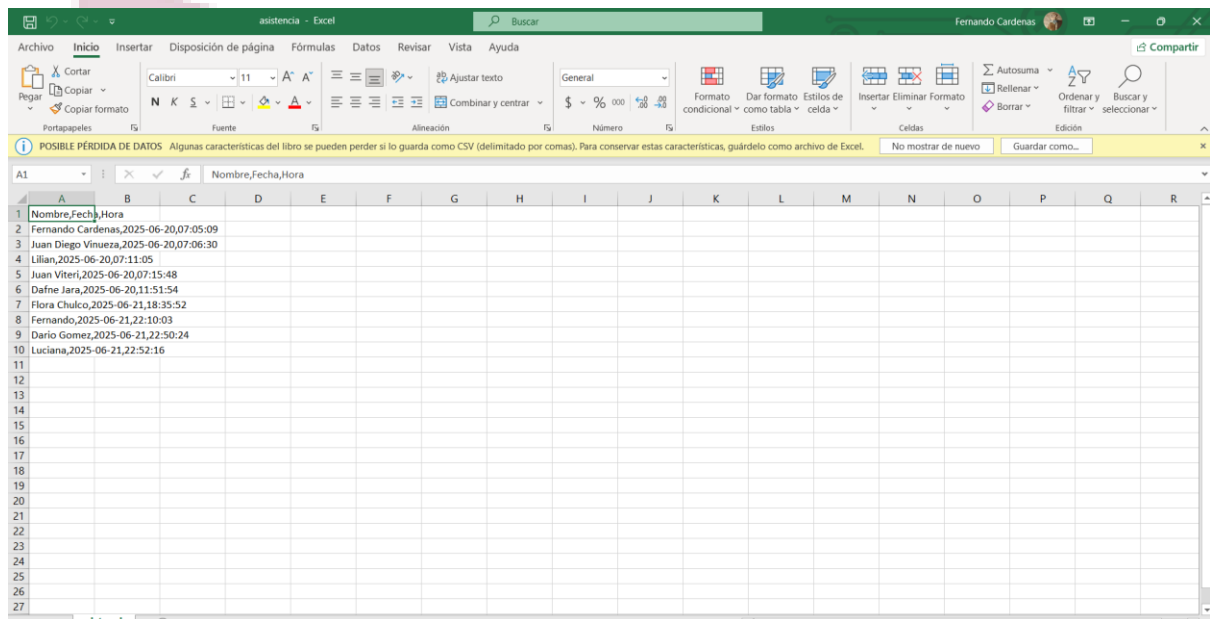
```
.. PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\USUARIO> & C:/Users/USUARIO/AppData/Local/M
cto final version final.py"
Bienvenidos al registro de asistencia
1 - Registro de asistencia manual
2 - Registro de asistencia con código QR
3 - Ver la lista de asistencias
Elija una opción (1-3): 3
Acceso a la lista de asistencias
Usuario: Uide.asu
Usuario incorrecto, pruebe otra vez: UserUIDE
Contraseña: daw
Contraseña incorrecta, pruebe otra vez: Uide.asu.123
Acceso permitido. Lista de asistencias:

      Nombre      Fecha      Hora
0  Fernando Cardenas  2025-06-20  07:05:09
1  Juan Diego Vinueza  2025-06-20  07:06:30
2      Lilian  2025-06-20  07:11:05
3    Juan Viteri  2025-06-20  07:15:48
4    Dafne Jara  2025-06-20  11:51:54
5    Flora Chulco  2025-06-21  18:35:52
6      Fernando  2025-06-21  22:10:03
7    Dario Gomez  2025-06-21  22:50:24
8      Luciana  2025-06-21  22:52:16
PS C:\Users\USUARIO> 
```

Ilustración 5 registro de asistencias

Archivo Csv:



POSSIBLE PÉRDIDA DE DATOS Algunas características del libro se pueden perder si lo guarda como CSV (delimitado por comas). Para conservar estas características, guárdelo como archivo de Excel. No mostrar de nuevo Guardar como...

Nombre, Fecha, Hora
Fernando Cardenas, 2025-06-20, 07:05:09
Juan Diego Vinuesa, 2025-06-20, 07:06:30
Lilian, 2025-06-20, 07:11:05
Juan Viteri, 2025-06-20, 07:15:48
Dafne Jara, 2025-06-20, 11:51:54
Flora Chulco, 2025-06-21, 18:35:52
Fernando, 2025-06-21, 22:10:03
Dario Gomez, 2025-06-21, 22:50:24
Luciana, 2025-06-21, 22:52:16

Ilustración 6 archivo CSV

Autor y soporte

Desarrollado por: Fernando Cárdenas

Correo de contacto: fecardenasch@uide.edu.ec

Fecha: 23/06/2025

Para soporte adicional, se recomienda consultar la documentación de las librerías utilizadas, en especial:

- <https://pandas.pydata.org/>
- <https://opencv.org/>
- <https://docs.python.org/3/>