

ENTREGA FINAL DE PRÁCTICAS CUIA



UNIVERSIDAD
DE GRANADA

ExploreAR – Aplicación Turística



- **Autor:** Rafael Luque Framit
- **Título:** ExploreAR – Aplicación turística
- **Asignatura:** Computación Ubicua e Inteligencia Ambiental

Índice

Descripción General	3
Interés del problema a resolver	3
Funcionalidades y Tecnologías involucradas	4
Reconocimiento facial	4
Reconocimiento de voz	4
Reproducción de voz	4
Realidad aumentada	5
Reconocimiento e identificación de imágenes	5
Requirements.txt.....	5
Manual de Usuario	6
Instalación y ejecución de la aplicación	6
Uso de la aplicación.....	6

Descripción General

ExploreAR es un proyecto que consiste en la realización de una aplicación enfocada al turismo en la ciudad de Málaga, España. Esta aplicación es apta para todos los públicos de todas las edades y adaptada para las personas que sean ciegas o sordas gracias a sus funcionalidades.

Esta aplicación realiza Interacción Hombre-Máquina implícita y Conciencia de contexto con funcionalidades como Reconocimiento de voz, Procesado de lenguaje natural, Realidad Aumentada (OpenCV) y Reconocimiento e identificación de imágenes (OpenCV). Está escrita íntegramente en Python y hace uso de diferentes librerías que se mencionarán más adelante en esta memoria.

Por un lado, la aplicación tiene identificación y registro de usuarios mediante reconocimiento facial, almacenando los datos y foto de los usuarios. La interacción entre la aplicación y el usuario es mediante comandos de voz. Una de las funcionalidades de la aplicación es que el usuario puede elegir que el programa le planifique una ruta en función del monumento que quiera visitar.

A través de una marca ArUco, el usuario puede escanearla en tiempo real mediante la cámara de su dispositivo y podrá observar mediante Realidad Aumentada un mapa del centro de la ciudad de Málaga con la ruta completa donde se encuentra hasta el monumento seleccionado, además, mediante comandos de voz la aplicación le irá diciendo la ruta completa.

Por otro lado, una vez el usuario esté delante del monumento la aplicación dispone de otra funcionalidad, la cual trata mediante la cámara de su dispositivo enfocar a dicho monumento, el sistema lo reconocerá y empezará a describir la información de dicho documento, tanto por comandos de voz como la información escrita en la pantalla.

Interés del problema a resolver

Esta aplicación turística es de gran utilidad ya que ayuda a las personas fomentando la exploración y descubrimiento de la ciudad de Málaga, de esta manera el usuario tiene más facilidades a la hora de hacer turismo ya que sin esta aplicación tendría que “buscarse la vida” a la hora de visitar monumentos y conseguir información de los mismos.

Por otro lado, esta aplicación mejora la experiencia del usuario ya que facilita la comunicación e interacción entre el sistema y el usuario. Gracias a su sistema de reconocimiento facial, el sistema siempre nos va a reconocer. Además, su funcionalidad de reconocimiento de voz y de reproducción de voz hace que la interacción entre el usuario y la máquina sea mucho más fluida y eficiente.

Por último, esta aplicación puede ser usada por todos los públicos de todas las edades, además de ser una aplicación integradora, ya que puede ser usada tanto por personas ciegas como por personas sordas.

Funcionalidades y Tecnologías involucradas

En esta aplicación se hace uso del reconocimiento facial, el reconocimiento de voz, la reproducción de voz, además tiene un papel muy importante la realidad aumentada y el reconocimiento e identificación de imágenes.

Reconocimiento facial

En primer lugar, la aplicación es capaz de llevar una identificación y registro de usuarios mediante el reconocimiento facial de la librería “face_recognition”. Gracias a esta librería se puede conseguir detectar si hay algún rostro en el frame y además calcular sus características gracias a su función “face_encodings”.

Si el usuario no está registrado aparece un recuadro rojo con la palabra “Desconocido”, este recuadro es capaz de seguir el movimiento de su cara. Una vez cierre la ventana el sistema se encargará de registrarlo si el usuario lo desea. El sistema le pedirá su nombre, su edad, y capturará una frame de su cara para almacenarla. El sistema almacenará en un archivo json los datos del usuario y un vector con las características del rostro del usuario.

En el caso de que el usuario si esté registrado aparecerá un recuadro verde con su nombre, este recuadro es capaz de seguir el movimiento de su cara. El sistema reconocerá que el usuario está registrado y una vez cierre la ventana podrá empezar a usar las funcionalidades de la aplicación.

Reconocimiento de voz

El usuario se podrá comunicar con la aplicación mediante comandos de voz, el sistema será capaz de recibirlos e identificarlos gracias a la librería “speech_recognition”.

Gracias a esta librería el sistema podrá abrir el micrófono de la máquina, recoger lo que el usuario diga en ese momento, y almacenarlo en una variable para posteriormente trabajar con ella.

En el caso de que el sistema no reconozca lo que se diga mostrará un mensaje de error.

Reproducción de voz

El sistema se va a comunicar con el usuario mediante comandos de voz. La aplicación es capaz de reproducir comandos de voz y además mostrar mensajes por pantalla de esta manera la interacción con el usuario es mucho más fluida y eficiente.

Esta funcionalidad se realiza gracias a la librería “gtts” para todo tipo de comandos de voz y gracias a la librería “mixer” de “pygame” para reproducir los mensajes relacionados con la información acerca de los monumentos esto es debido a que se debe mostrar la información por la cámara a la vez que se reproduce por audio.

Esta funcionalidad crea archivos de tipo mp3 en la carpeta tmp los cuales tendrán el mensaje que se desea reproducir en formato audio.

Realidad aumentada

La aplicación es capaz de llevar a cabo AR mediante un marcador ArUco el cual es escaneado en tiempo real mediante la cámara del dispositivo y sobrepone sobre ese marcador un mapa del centro de la ciudad de Málaga con la ruta desde donde se encuentra el usuario en ese momento hasta el monumento que el usuario desee llegar. Además, el sistema es capaz de indicar la ruta que debe seguir mediante comandos de voz con la funcionalidad mencionada anteriormente.

Esta funcionalidad es llevada a cabo gracias a la librería “cv2” de “OpenCV”. Con las funciones “detectMarkers”, “drawDetectedMarkers”, “findHomography”, “warpPerspective” y “fillConvexPoly” el sistema detecta si hay algún marcador ArUco en el frame y superpone la imagen del mapa con la ruta al monumento deseada.

Reconocimiento e identificación de imágenes

Una vez que el usuario ha llegado al monumento y esté delante de él podrá reconocerlo y obtener información sobre él. El sistema reconocerá mediante la captación de frames de la cámara que monumento es el que está enfocando, identificará dicho monumento y le preguntará al usuario si desea escuchar la información de dicho monumento.

Si el usuario desea obtener la información del monumento se le mostrará por pantalla, por la cámara y además se le irá reproduciendo por audio dicha información. Esto gracias a las funcionalidades mencionadas anteriormente.

Para llevar a cabo esta funcionalidad de reconocimiento e identificación de imágenes el sistema usa la librería “cv2” gracias al algoritmo “sift”, este algoritmo usa la función “detectAndCompute” para detectar keypoints y calcular los descriptores del frame actual. Realiza la comparación de los descriptores con los descriptores de los monumentos que ya tiene almacenados. Si alguno de ellos supera el umbral de coincidencias lo muestra y devuelve el nombre de ese monumento identificado.

Requirements.txt

La lista de dependencias que he usado para el desarrollo de mi aplicación son las siguientes:

```
numpy==1.26.4
requests==2.32.2
deepface==0.0.87
mediapipe==0.10.11
face-recognition==1.3.0
face-recognition-models==0.3.0
retina-face==0.0.15
gTTS==2.5.1
SpeechRecognition==3.10.1
folium==0.16.0
playsound==1.2.2
opencv-contrib-python==4.9.0.80
```



```
opencv-python==4.9.0.80
pydub==0.25.1
pygame==2.5.2
pillow==10.2.0
pyzbar==0.1.9
```

Manual de Usuario

Instalación y ejecución de la aplicación

En primer lugar, una vez tengamos el proyecto descompilado debemos instalarnos todas las dependencias necesarias para que la aplicación funcione. Para ello debemos ejecutar en la terminal el siguiente comando:

```
pip install -r requirements.txt
```

Tras esto podemos ejecutar la aplicación, para ello ejecutamos en la terminal el comando:

```
python main.py
```

Uso de la aplicación

En primer lugar, una vez iniciada la aplicación, se abrirá la cámara del dispositivo y nos reconocerá la cara. En el caso de que no estemos registrados el sistema nos preguntará si queremos registrarnos, en el caso de que queramos registrarnos nos preguntará nuestro nombre y nuestra edad y por último abrirá de nuevo la cámara del dispositivo para tomar captura de un frame y guardarlo como imagen de la cara del usuario. Tras esto podremos hacer uso de las siguientes funcionalidades de la aplicación. Para salir de esa ventana deberá presionar la letra “q”.

En el caso de que el usuario si esté registrado el sistema lo reconocerá mostrando un recuadro en verde con el nombre del usuario y seguidamente podrá hacer uso de las siguientes funcionalidades de la aplicación. Para salir de esa ventana deberá presionar la letra “q”.

El usuario podrá hacer uso de la aplicación con los siguientes comandos de voz:

- **ruta a monumento:** si el usuario dice este comando se le va a proporcionar una ruta hacia el monumento que el desee. En primer lugar, el sistema preguntará que monumento quiere visitar (Alcazaba, catedral, plaza de toros, teatro romano), el usuario responderá y a continuación se abrirá la cámara del dispositivo. Si la cámara detecta el marcador ArUco se superpondrá el mapa con la ruta hacia ese monumento y pulsando la tecla “a” podrá obtener la información de la ruta mediante reproducción de audio. Para salir de esta ventana deberá presionar la letra “q”.
- **reconocer monumento:** si el usuario dice este comando se abrirá la cámara del dispositivo, el sistema identificará el monumento al que se está enfocando. Para salir de esa ventana deberá presionar la letra “q”. A continuación, el sistema preguntará si desea obtener información sobre ese monumento, en el caso de que el usuario si quiera comenzará a reproducirse por audio la información y además se mostrará por cámara dicha información escrita.

- **ayuda:** si el usuario dice este comando el sistema mostrará y reproducirá la lista de comandos de voz que el usuario puede realizar aportando también una breve descripción de lo que realiza cada uno de ellos, entre los que se encuentran: “ruta a monumento, reconocer monumento, ayuda o salir”.
- **salir:** si el usuario dice este comando la aplicación se cerrará mostrando y reproduciendo un mensaje “Muchas gracias, hasta pronto.”.

A continuación, muestro una serie de capturas de pantalla de la ejecución de algunas funcionalidades de la aplicación:

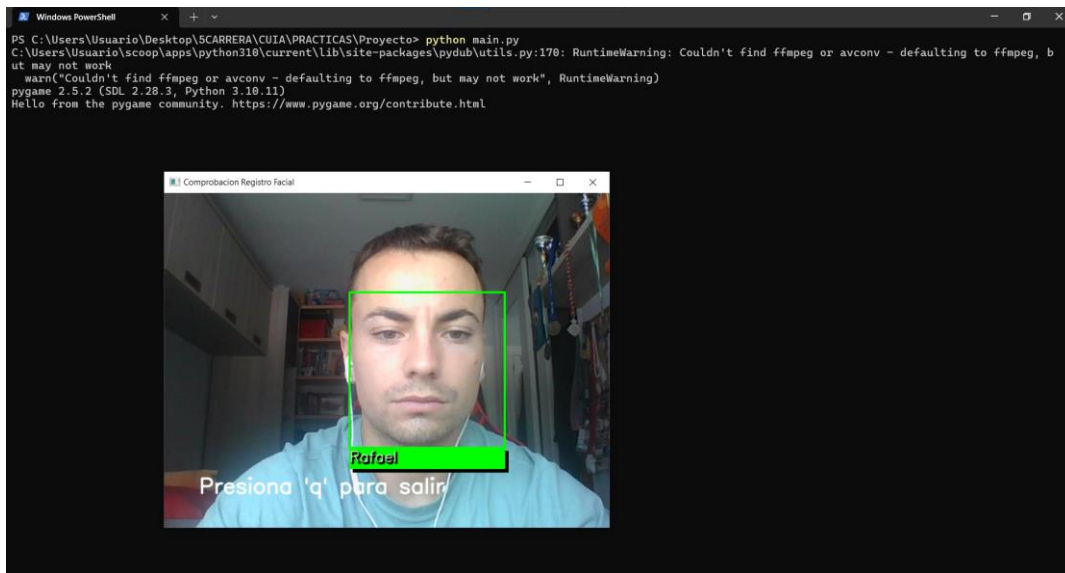


Figura 1: Registro facial

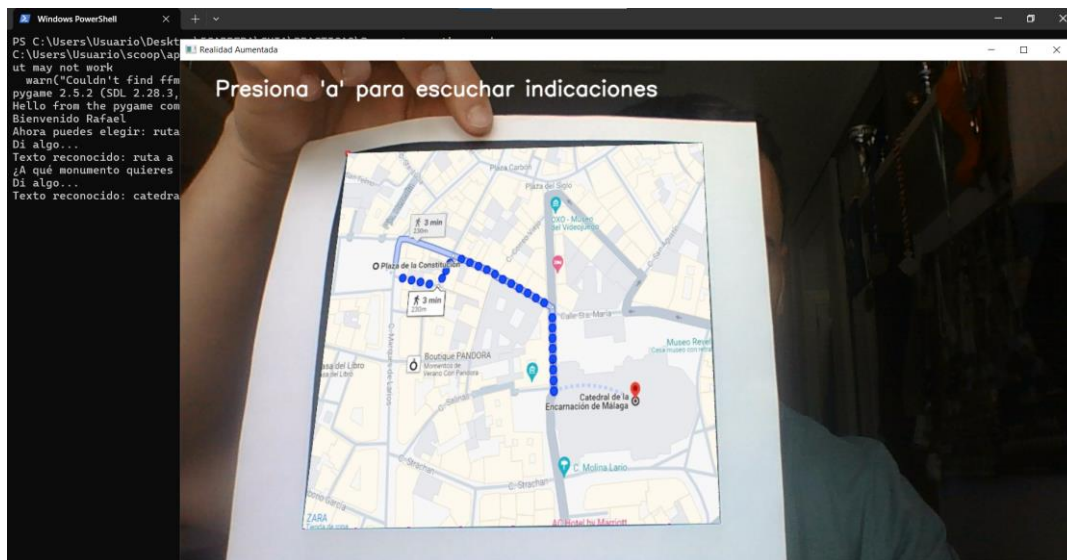


Figura 2: Ruta a monumento (Catedral)



Figura 3: Reconocer monumento (plaza de toros)

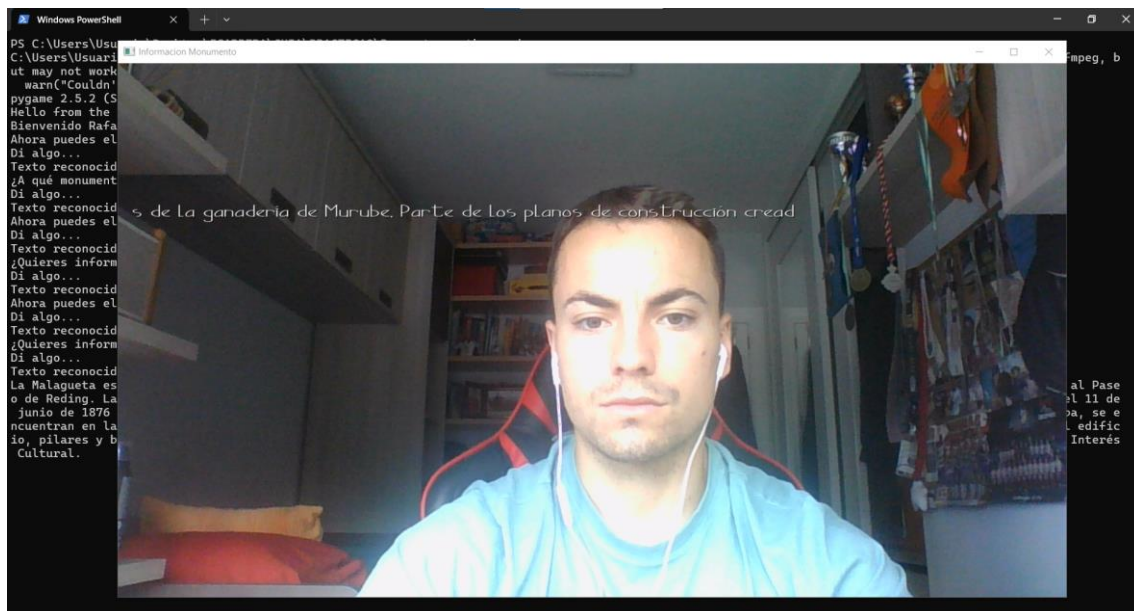


Figura 4: Obtener información del monumento (plaza de toros)