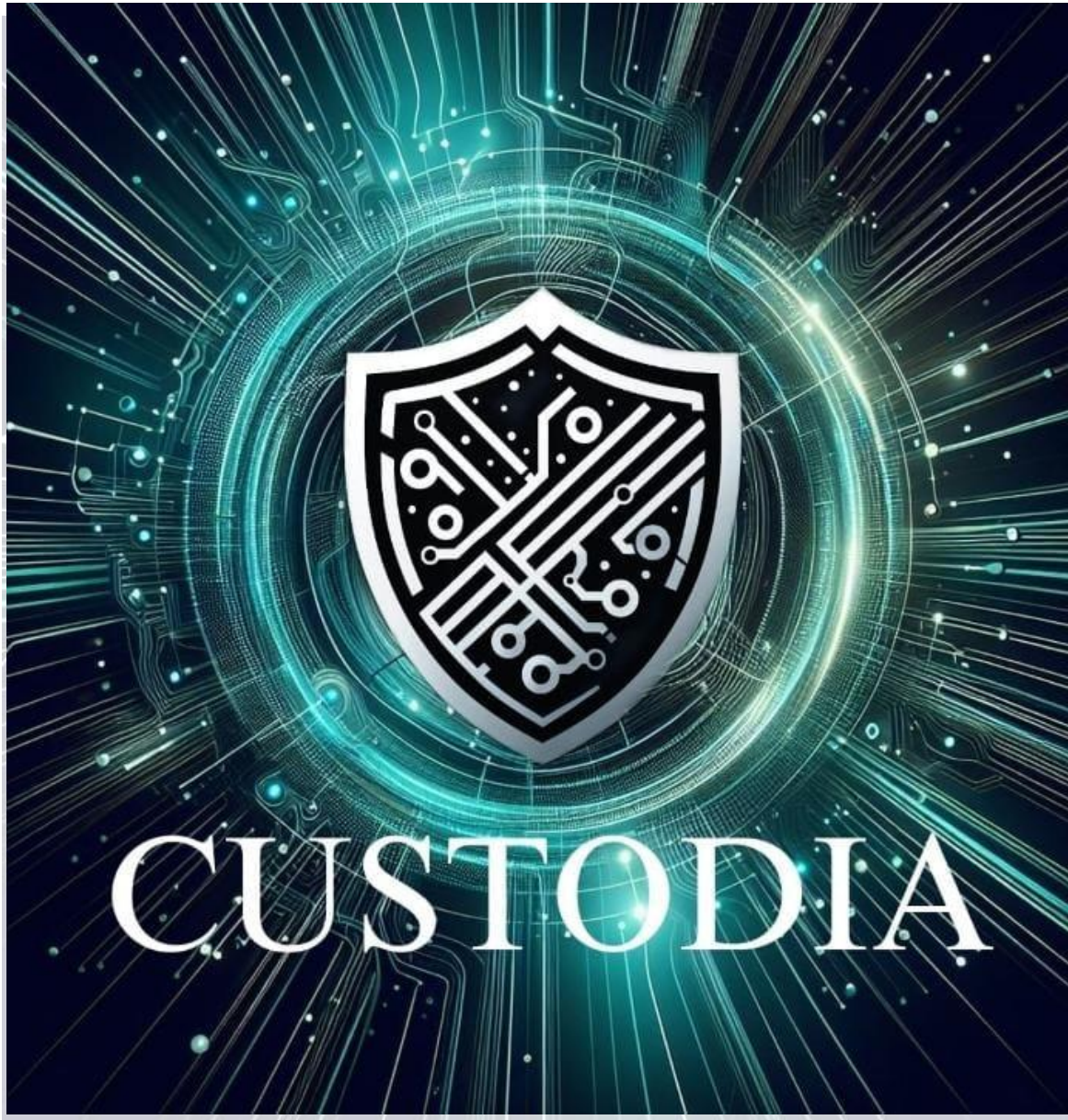


# SEGURIDAD PUBLICA EN TU COMONIDAD

EQUPO HACKERS: JONATHAN LEAL, ALAN MAYORCA, DARA PEREZ,  
LUIS TACANO, ANTONIO NAVARRETE.





## SEGURIDAD PÚBLICA EN MI COMUNIDAD

La seguridad es un tema que exige mucha atención por parte de los órganos que representan la seguridad pública de la nación, así como de la población en cierto porcentaje.

Según Citizens' Council for Public Security and Criminal Justice, *think tank* con sede en México y amplio reconocimiento en EUA publican una revista cada año a las urbes que se encuentran amenazadas seriamente en cuanto a inseguridad pública se refiere. En su última publicación se llegó a la conclusión de que México es el país más inseguro de América Latina lo cual generó una gran preocupación por parte de las autoridades máximas del país.

Dentro de México existen estados con mayor conflicto de inseguridad de los cuales el estado de Morelos ocupa el primer lugar con mayor enfoque en municipios como Cuautla, Yauteppec, Ayala, etc.

La problemática impuesta por la autoridad de Ayala es realizar una propuesta donde por medio de tecnología e innovación se pueda combatir y disminuir en un grado considerable la inseguridad pública del municipio y posteriormente en el resto del estado y más.

# CUSTODIA





## **Descripción de la propuesta de solución y Metodología**

Implementación de un sistema de vigilancia aérea no tripulada basado en drones con tecnología avanzada.

1.Drones de Vigilancia: Utilizar drones equipados con cámaras de alta resolución, y capacidades de transmisión en tiempo real para patrullar áreas críticas. Estos drones pueden llegar a lugares de difícil acceso y proporcionar una visión completa del terreno.

2.Inteligencia Artificial para Análisis de Imágenes: Implementar algoritmos que pueden identificar patrones sospechosos, actividades delictivas y movimientos inusuales, permitiendo una respuesta más rápida por parte de las autoridades.

3.Despliegue Estratégico: Utilizar datos recopilados por los drones para realizar despliegues policiales estratégicos, aumentando la efectividad de las operaciones.

4.Colaboración con la Comunidad: Involucrar a la comunidad en la operación de los drones a través de programas de capacitación. Esto no solo aumentaría la participación ciudadana, sino que también podría proporcionar información valiosa desde el nivel local.

5.Concienciación y Prevención: Utilizar los drones para difundir mensajes de prevención y concienciación en comunidades afectadas, promoviendo la seguridad y disuadiendo actividades delictivas.

Esta solución aprovecha la tecnología de drones y la inteligencia artificial para mejorar la vigilancia, la detección temprana y la respuesta rápida, al tiempo que fomenta la colaboración activa de la comunidad en la lucha contra la inseguridad.



## **Temas normativos**

- **NOM-107-SCT3-2019 Que establece los requerimientos para operar un sistema de aeronave pilotada a distancia (RPAS) en el espacio aéreo mexicano.**

Para que los drones puedan operar, la norma requiere:

Distancias necesarias de vuelo, partiendo de supuestos específicos. Por ejemplo, exige que la persona que opere el dron lo haga con una distancia de separación de al menos 9.2 kilómetros de cualquier aeródromo, y a una distancia de separación de al menos 0.900 kilómetros de cualquier helipuerto, sin importar el tamaño.

Obligaciones de piloto de drones, entre los que destacan:

1. No operar la nave si no se hace de manera segura.

Esto significa que la persona usuaria drones, no debe dejar caer o arrojar objetos o materiales que puedan causar daño a cualquier persona o propiedad. “Manera segura” también refiere a la necesidad de una inspección pre vuelo.

La Norma agrega que todas las personas involucradas en la operación de un dron deberán recibir una sesión informativa que incluya las condiciones de operación, procedimientos de emergencia y contingencia, así como los deberes responsabilidades y posibles riesgos.

2. No operar en zonas prohibidas que son detalladas en el Manual de Publicación de Información Aeronáutica.

3. Mantener el control de la trayectoria de la aeronave en todo momento

4. No operar la aeronave de forma negligente que ponga en riesgo la vida de otras personas.

5. No operar desde vehículos en movimiento

6. No operar más de un dron al mismo tiempo





## Responsabilidades.

1. Responsabilidad por accidentes, daños o lesiones causados por la operación del dron.
2. Responsable sobre el uso de información obtenida durante la operación de la aeronave.
3. No operar la aeronave en estado de ebriedad o bajo los efectos de estupefacientes o similares.

### Requerimientos específicos para cada tipo de dron

La norma establece requerimientos y limitaciones para aeronaves de distintos tamaños y para distintas finalidades.

Drones en la Legislación Mexicana. (2020, octubre). seguridad integral artículo19. Recuperado 22 de noviembre de 2023, de [https://seguridadintegral.articulo19.org/wp-content/uploads/2020/10/art19\\_2020\\_infografia-Drones.pdf](https://seguridadintegral.articulo19.org/wp-content/uploads/2020/10/art19_2020_infografia-Drones.pdf)

CUSTODIA



## Herramientas tecnológicas para el desarrollo

Entre algunas de las herramientas que utilizaremos para poder lograr nuestro objetivo son:

**Drones:** Los drones son una de las herramientas principales bastante útiles, que nos ayudará a poder alcanzar nuestro objetivo de aumentar la seguridad, ya que estos pueden facilitar en gran medida el recorrer grandes áreas, en el menor tiempo posible, facilitando así el poder encontrar a personas.

**Internet:** El internet es una de las herramientas principales que no puede faltar, ya que, gracias a este podremos difundir o transmitir información de forma rápida, para lograr tomar las medidas necesarias de acuerdo a la situación que se pueda presentar.

**Redes sociales:** Las redes sociales actualmente juegan un papel muy importante a la hora de difundir información a la comunidad, lo cual nos puede facilitar transmitir información útil a la comunidad.

**Inteligencia artificial:** Aunque no sea demasiado su uso, es bastante útil a la hora de automatizar ciertos procesos, como el proceso de reconocimiento de rostros, ya que haría más fácil para el piloto lograr procesar la información, que si lo hiciera de forma manual.



CUSTODIA





## Especificación técnica

Empezaremos importando las librerías que se utilizarán para el proceso de reconocimiento facial:

### Que se cargaran la imagen al analizar:

```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")

faces = face_recognition.face_locations(image)

for face in faces:

    facial_features = face_recognition.face_encodings(image, [face])

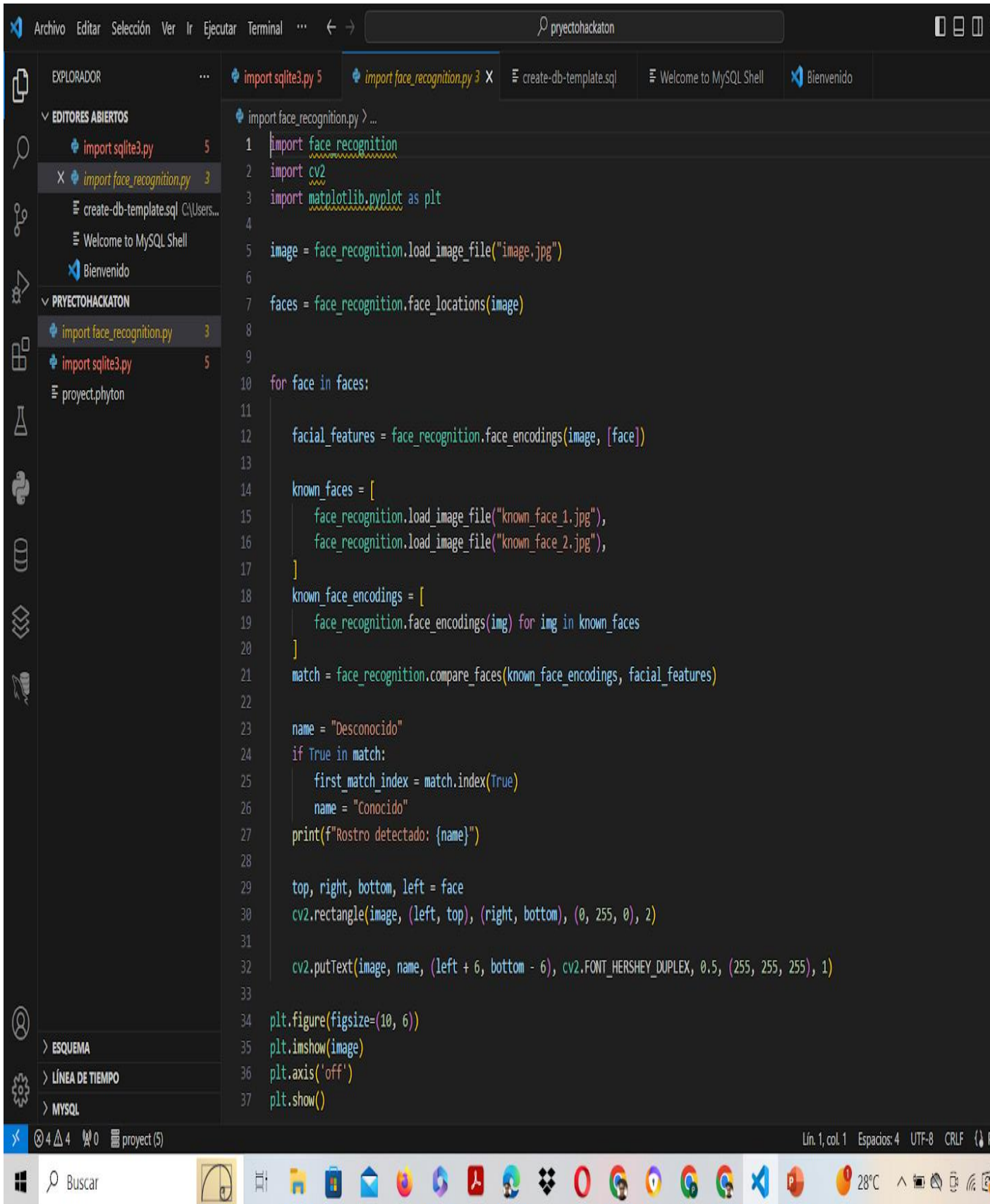
    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```



The image shows a screenshot of a Visual Studio Code editor window. The left sidebar contains the Explorer, Search, and Run and Debug views. The Explorer view shows a project named 'PRYECTO HACKATON' with files 'import face\_recognition.py' and 'import sqlite3.py'. The main editor area displays the 'import face\_recognition.py' file, which contains the following Python code:

```
1 import face_recognition
2 import cv2
3 import matplotlib.pyplot as plt
4
5 image = face_recognition.load_image_file("image.jpg")
6
7 faces = face_recognition.face_locations(image)
8
9
10 for face in faces:
11
12     facial_features = face_recognition.face_encodings(image, [face])
13
14     known_faces = [
15         face_recognition.load_image_file("known_face_1.jpg"),
16         face_recognition.load_image_file("known_face_2.jpg"),
17     ]
18     known_face_encodings = [
19         face_recognition.face_encodings(img) for img in known_faces
20     ]
21     match = face_recognition.compare_faces(known_face_encodings, facial_features)
22
23     name = "Desconocido"
24     if True in match:
25         first_match_index = match.index(True)
26         name = "Conocido"
27     print(f"Rostro detectado: {name}")
28
29     top, right, bottom, left = face
30     cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)
31
32     cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)
33
34 plt.figure(figsize=(10, 6))
35 plt.imshow(image)
36 plt.axis('off')
37 plt.show()
```

The bottom status bar shows the file encoding as UTF-8 and the line/col/char count as 37/1/1.





## Se detectarán los rostros que contenga la imagen:

```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")

faces = face_recognition.face_locations(image)

for face in faces:

    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```



**Después se procesa cada rostro detectado en la imagen extrayendo las características faciales de cada rostro.**

**Se comparan las características faciales con una base de datos que contenga imágenes ya conocidas.**

```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")

faces = face_recognition.face_locations(image)

for face in faces:

    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

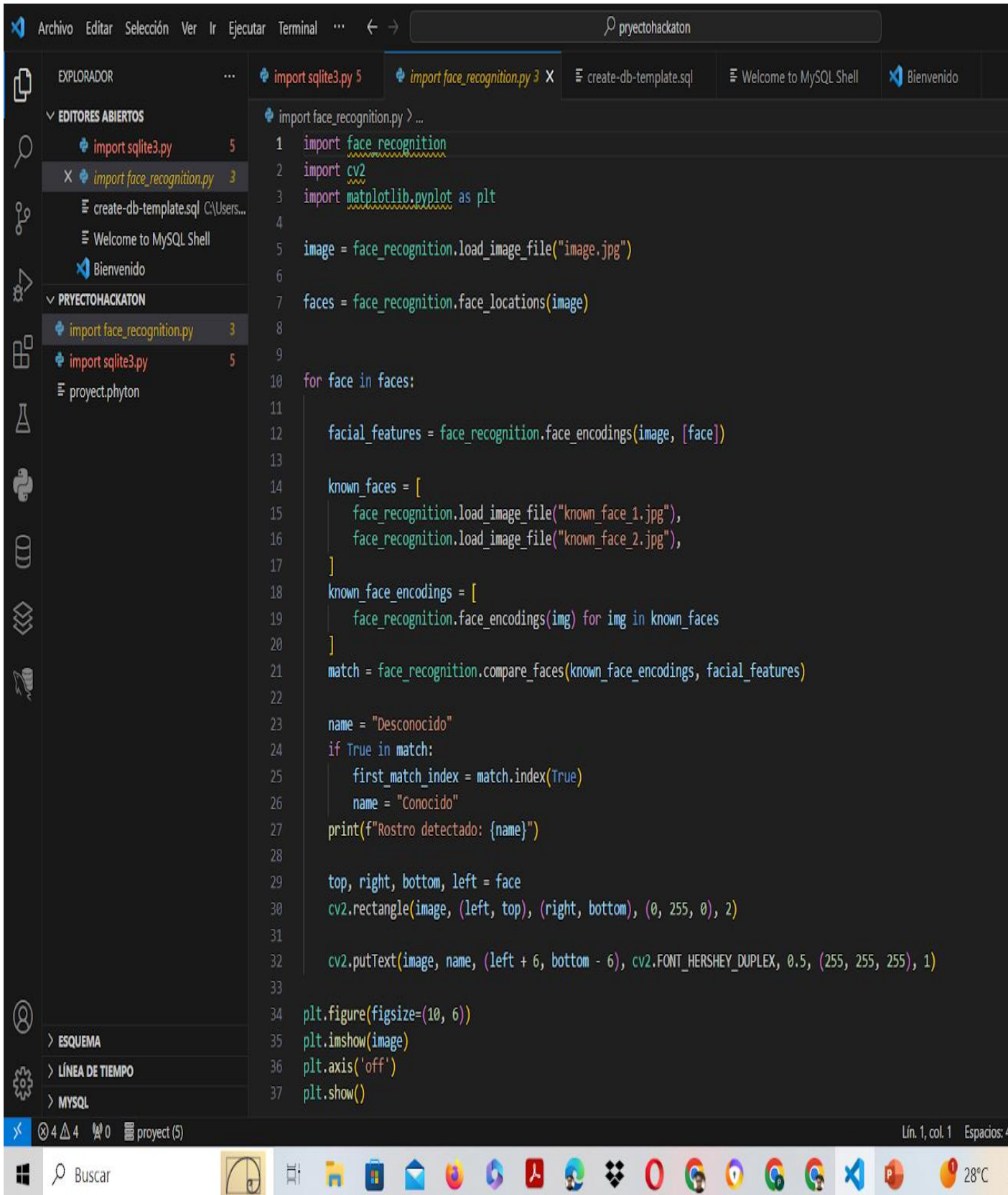
    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```





```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")

faces = face_recognition.face_locations(image)

for face in faces:

    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

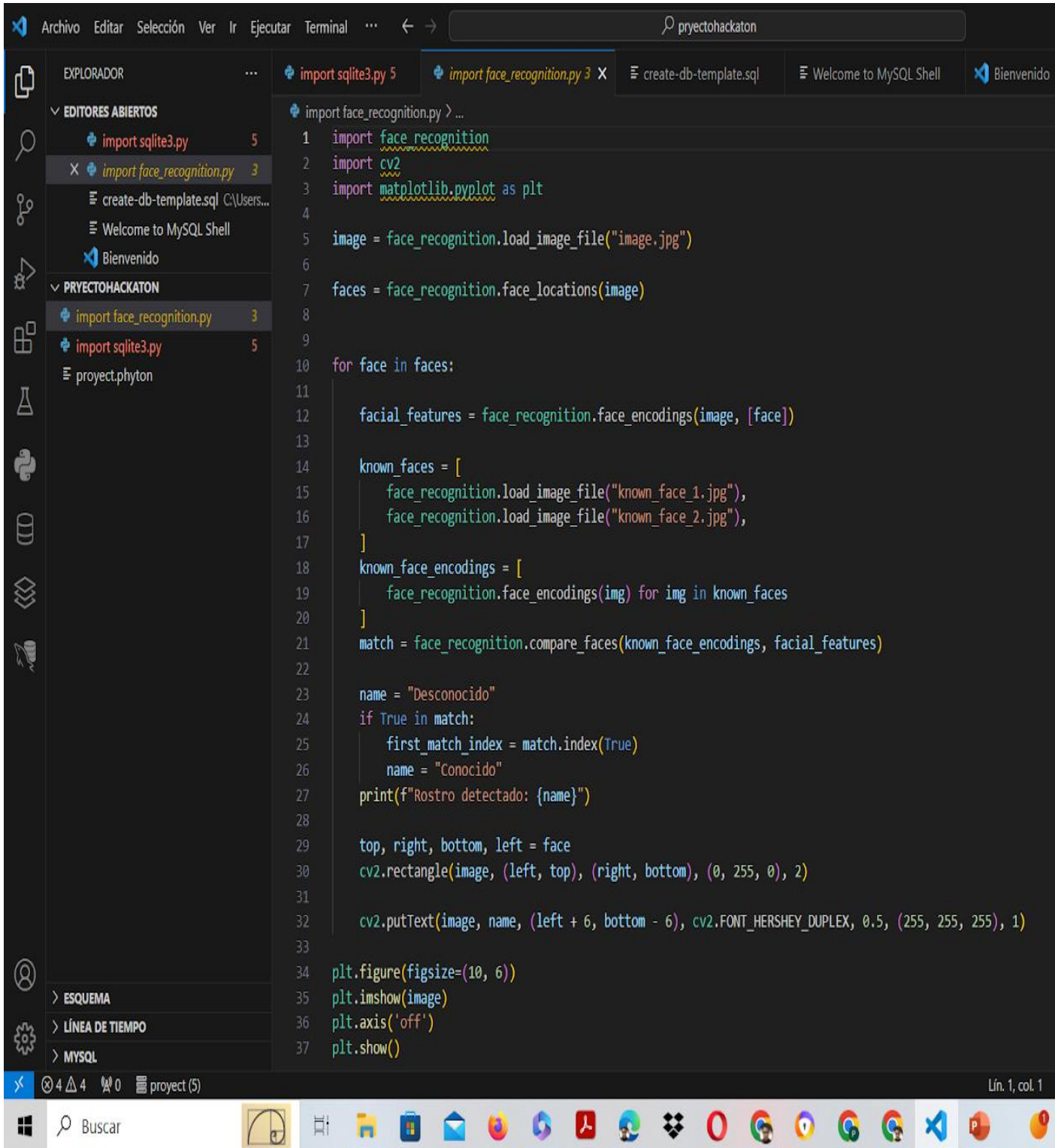
    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```



## E imprime el resultado de la comparación.



```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")

faces = face_recognition.face_locations(image)

for face in faces:

    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```





**Y se le pedirá que dibuje un rectángulo en el rostro detectado.**

```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")
faces = face_recognition.face_locations(image)

for face in faces:
    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

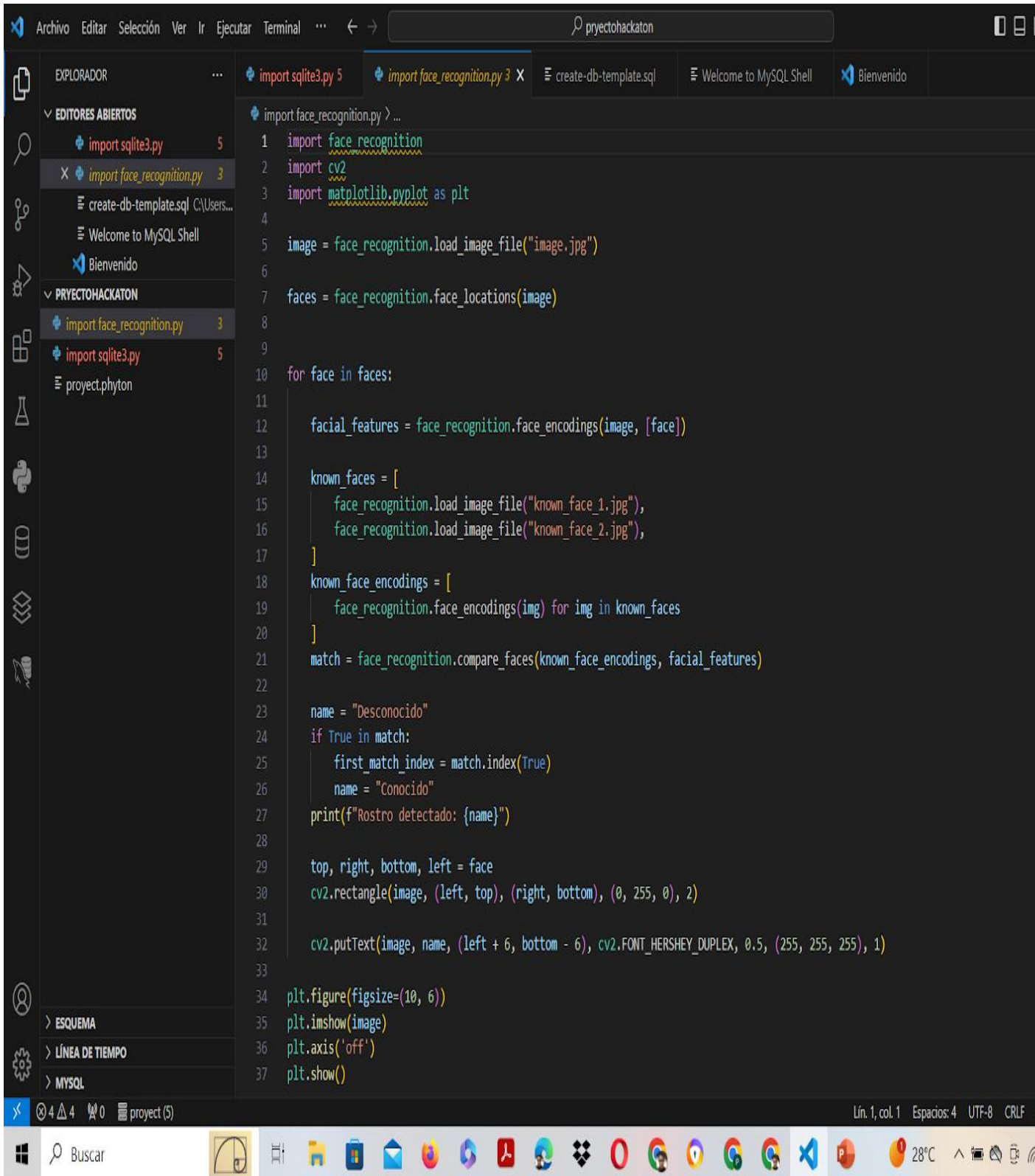
    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```

**Luego le pediremos que escriba el nombre del rostro detectado.**



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'PRYECTO HACKATON' with files 'import face\_recognition.py' (3 lines) and 'import sqlite3.py' (5 lines). The Editor sidebar shows 'EDITORES ABIERTOS' with 'import face\_recognition.py' (3 lines) and 'import sqlite3.py' (5 lines). The main editor area displays the code for 'import face\_recognition.py'.

```
import face_recognition.py > ...
1 import face_recognition
2 import cv2
3 import matplotlib.pyplot as plt
4
5 image = face_recognition.load_image_file("image.jpg")
6
7 faces = face_recognition.face_locations(image)
8
9
10 for face in faces:
11
12     facial_features = face_recognition.face_encodings(image, [face])
13
14     known_faces = [
15         face_recognition.load_image_file("known_face_1.jpg"),
16         face_recognition.load_image_file("known_face_2.jpg"),
17     ]
18     known_face_encodings = [
19         face_recognition.face_encodings(img) for img in known_faces
20     ]
21     match = face_recognition.compare_faces(known_face_encodings, facial_features)
22
23     name = "Desconocido"
24     if True in match:
25         first_match_index = match.index(True)
26         name = "Conocido"
27     print(f"Rostro detectado: {name}")
28
29     top, right, bottom, left = face
30     cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)
31
32     cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)
33
34 plt.figure(figsize=(10, 6))
35 plt.imshow(image)
36 plt.axis('off')
37 plt.show()
```

The status bar at the bottom shows 'Lín. 1, col. 1 Espacios: 4 UTF-8 CRLF' and a search bar with 'Buscar'.





```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")

faces = face_recognition.face_locations(image)

for face in faces:

    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```



**Y para finalizar le pediremos que nos muestre los rostros identificados.**

```
import face_recognition
import cv2
import matplotlib.pyplot as plt

image = face_recognition.load_image_file("image.jpg")
faces = face_recognition.face_locations(image)

for face in faces:
    facial_features = face_recognition.face_encodings(image, [face])

    known_faces = [
        face_recognition.load_image_file("known_face_1.jpg"),
        face_recognition.load_image_file("known_face_2.jpg"),
    ]
    known_face_encodings = [
        face_recognition.face_encodings(img) for img in known_faces
    ]
    match = face_recognition.compare_faces(known_face_encodings, facial_features)

    name = "Desconocido"
    if True in match:
        first_match_index = match.index(True)
        name = "Conocido"
    print(f"Rostro detectado: {name}")

    top, right, bottom, left = face
    cv2.rectangle(image, (left, top), (right, bottom), (0, 255, 0), 2)

    cv2.putText(image, name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.5, (255, 255, 255), 1)

plt.figure(figsize=(10, 6))
plt.imshow(image)
plt.axis('off')
plt.show()
```





## Implementación de las ODS

**ODS 16:** Para promover sociedades pacíficas, justas e inclusivas se busca implementar campañas de concientización debido a que estudios realizados por la INEGI muestran que las personas no están preparadas para reaccionar a ciertos tipos de inseguridades (enfrentamientos armados, robos, homicidios, etc.). Así como también hay una observación a estos estudios en donde se puede llegar a la conclusión preocupante que ante cualquier delito directo o indirecto a un ciudadano no es reportado por la población dando un porcentaje bajo a lo que se representa como "Delitos de Prevención y Reacción Policía" en los datos de la INEGI publicados por la COESPO.

SÍNTESIS ESTADÍSTICA MUNICIPAL 2019. (2019, octubre). coespo morelos. Recuperado 22 de noviembre de 2023, de [https://coespo.morelos.gob.mx/images/Datos\\_municipales/2020/AYALA2019.pdf](https://coespo.morelos.gob.mx/images/Datos_municipales/2020/AYALA2019.pdf)

**ODS 17:** Para fortalecer medios de implementación es necesario nuevas tecnologías para combatir los casos de delincuencia en la comunidad de Ayala para proteger a los ciudadanos, mientras se resguardan la vida de los policías.

La ventaja de los drones es que, ofrecen seguridad en los procedimientos policiales, así como también pueden supervisar áreas grandes de manera rápida y eficaz, sin la necesidad de incrementar la fuerza de trabajo y/o arriesgar demasiado la vida de los oficiales.

**16** PAZ, JUSTICIA  
E INSTITUCIONES  
SÓLIDAS



ALIANZAS PARA LOGRAR  
LOS OBJETIVOS





## Análisis FODA

<b>Fortalezas</b> <ul style="list-style-type: none"><li>• Mayor seguridad a la población.</li><li>• Reducción de riesgos a agentes policiacos.</li><li>• Alta confiabilidad.</li></ul>	<b>Oportunidades</b> <ul style="list-style-type: none"><li>• Mejorar la calidad de vida y seguridad.</li><li>• Llegar a una población más extensa.</li><li>• Interés por organismos de mayor impacto.</li></ul>
<b>Debilidades</b> <ul style="list-style-type: none"><li>• Resistencia por parte de la población al aceptar nuevas tecnologías.</li><li>• Tiempo de carga del dron, debe tenerse un control y un personal que se encargue de que el prototipo esté listo.</li></ul>	<b>Amenazas</b> <ul style="list-style-type: none"><li>• Colaboración o participación de la población en campañas y toma de acciones.</li><li>• Empresas que busquen lucrar con el mismo objetivo.</li></ul>





### **Misión:**

Facilitar y fortalecer la seguridad pública a través de la implementación responsable y eficiente de tecnología dron con reconocimiento facial y de área. Nuestra misión es proporcionar herramientas avanzadas que ayuden a las fuerzas del orden a prevenir y responder de manera ágil a situaciones críticas, garantizando la protección de la comunidad y el respeto a los derechos individuales.

### **Visión:**

Ser líderes en la innovación tecnológica para la seguridad pública, construyendo un entorno donde la aplicación ética y transparente de la tecnología dron contribuya de manera significativa a la reducción de la delincuencia y al mantenimiento de entornos seguros. Aspiramos a ser reconocidos por nuestra contribución al bienestar de la sociedad y por establecer nuevos estándares en seguridad ciudadana.

### **Valores:**

1. **Ética:** Guiamos cada acción con principios éticos sólidos, asegurando que la tecnología se utilice de manera justa y respetuosa de los derechos individuales.
2. **Innovación:** Buscamos constantemente formas de mejorar y evolucionar, adoptando nuevas tecnologías y enfoques para ofrecer soluciones de seguridad más efectivas.
3. **Responsabilidad Social:** Nos comprometemos a utilizar la tecnología para el bien común, contribuyendo activamente a la seguridad y el bienestar de las comunidades a las que servimos.
4. **Transparencia:** Fomentamos la transparencia en todas nuestras operaciones, desde el desarrollo de la tecnología hasta su implementación, para construir la confianza de la comunidad y las autoridades.



5. Colaboración: Valoramos las asociaciones sólidas con fuerzas del orden, gobiernos locales y otras partes interesadas para abordar de manera efectiva los desafíos de seguridad pública.

6. Respeto a la Privacidad: Reconocemos y protegemos el derecho a la privacidad de las personas, implementando salvaguardias y prácticas que equilibren la seguridad con la protección de datos individuales.

Estos principios fundamentales guían nuestro compromiso de aprovechar la tecnología de drones con reconocimiento facial y de área para mejorar la seguridad pública de manera ética y efectiva.





# Modelo Canvas "CUSTODIA"

Aliados clave	Actividades clave	Propuesta de valor	Relación con los clientes	Segmentación de clientes
Como aliados clave se busca trabajar de la mano primeramente con el gobierno del municipio de Ayala y posterior al gobierno de otras entidades federativas así como organismos privados donde el proyecto puede adecuarse a las necesidades de distintas organizaciones (seguridad privada, seguridad de inteligencia, aplicaciones militares, etc.).	1.Producir las unidades que se requieran buscando la eficiencia y una inversión adecuada. 2.Generar campañas de concientización para la población que va dirigido el proyecto.	Debido a la inseguridad generada por grupos delictivos, <b>CUSTODIA</b> brinda la solución más eficiente y segura para la población así como para los agentes de seguridad pública.  Al ser un proyecto que cumple o contempla todas las normas y adecuaciones en la región logramos destacar como una propuesta eficiente, de bajo riesgo y de mayor rendimiento para la seguridad publica del municipio de Ayala.	Capacitación administrada por personal sumamente calificado a los agentes de seguridad pública. Aclaración de dudas e implementación de programas de ayuda a personas que sufren de algún tipo de violencia o extorsión	Nuestra propuesta de valor va dirigida principalmente al gobierno del municipio de Ayala. Como colaboradores potenciales también puede ser adecuada a demás regiones u organizaciones de seguridad privada.
	Recursos clave		Canales	
	1.Recursos humanos: proveedores, ingenieros implicados, piloto especialista en drones.  2.Recursos físicos: Drones, controles, zonas de prueba  3.Recursos intelectuales : Conocimiento de los ingenieros con el software		<ul style="list-style-type: none"><li>• Campañas publicitarias.</li><li>• Casetas de atención de seguridad mas cercanas.</li><li>• Publicaciones en revistas citios web.</li><li>• Conferencias.</li><li>• Medios digitales como redes o foros de discusión.</li><li>• Colaboraciones con personalidades que generan impacto</li></ul>	
Estructura de costos			Fuentes de ingresos	
Servicios de marketing: \$2500 Drone BINDEN GD93 \$3499 Costos indirectos Pago mensual al piloto especializado \$12764 Zona de prácticas \$			Recursos financieros: Financiamiento gubernamental, presupuesto de egresos de la federación (PEF). CONACYT (Consejo Nacional de Ciencia y Tecnología) INEGI (Instituto Nacional de Estadística y Geografía)	

