



Universidad Autónoma del Estado de México

Plantel UAEM Valle de Chalco

Avance del Proyecto Sistema de Seguridad IoT con ESP32-CAM y Sensor PIR

SOLORZANO PEÑA JOAHAN JAEI

ARROYO VILLALON CRISTIAN ADOLFO

GARCIA RAMIREZ IRVING SAID

9° Semestre

Grupo O0

Turno Vespertino

Materia: tecnologías computacionales II

PROF. José Luis Mora Santos

Universidad Autónoma de México: UAEM

1. Inicio del servidor

El servidor se ejecuta mediante el archivo app.py desde la terminal:

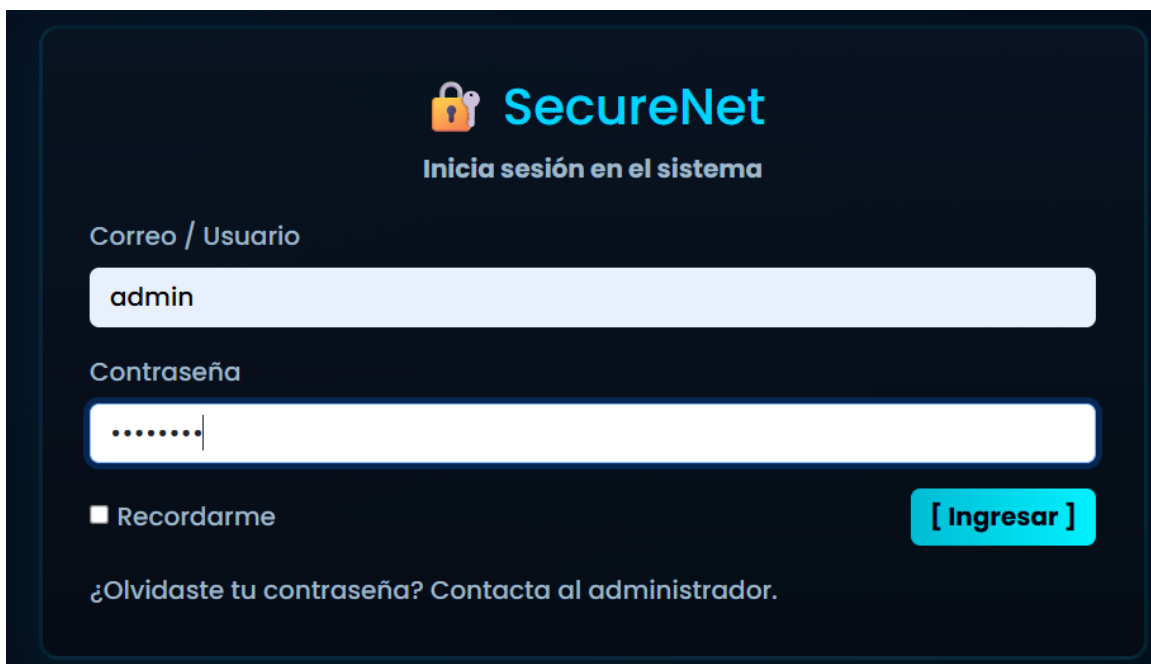
```
python app.py
```

En consola aparece el log indicando:

- Carga de rostros autorizados.
- Inicio del bot de Telegram.
- Servidor corriendo en `http://127.0.0.1:5000` y en la IP de la red local.



The image shows a login form for a system named "SecureNet". The form has a dark blue background with a lighter blue border. At the top, there is a logo consisting of a padlock and a key, followed by the text "SecureNet" in a large, bold, light blue font. Below the logo, the text "Inicia sesión en el sistema" is displayed in a smaller, white font. The form contains two input fields: "Correo / Usuario" and "Contraseña". The "Correo / Usuario" field has a placeholder text "Tu usuario". The "Contraseña" field has a placeholder text "Contraseña". Below the "Correo / Usuario" field, there is a checkbox labeled "Recordarme". To the right of the "Contraseña" field, there is a blue button with the text "[Ingresar]". At the bottom of the form, there is a link that says "¿Olvidaste tu contraseña? Contacta al administrador."



This image shows the same "SecureNet" login form as the previous one, but with the input fields filled out. The "Correo / Usuario" field now contains the text "admin". The "Contraseña" field contains a series of dots, indicating that the password is masked. The "Recordarme" checkbox remains unchecked. The "[Ingresar]" button is still present and visible. The link at the bottom, "¿Olvidaste tu contraseña? Contacta al administrador.", is also visible.

2. Acceso al dashboard web

Al abrir la dirección del servidor en el navegador, se muestra la pantalla de login. Tras ingresar usuario y contraseña válidos, se despliega el dashboard, donde se pueden observar estadísticas, intrusos detectados y rostros autorizados.

The image shows a web browser displaying the SecureNet dashboard and a terminal window below it.

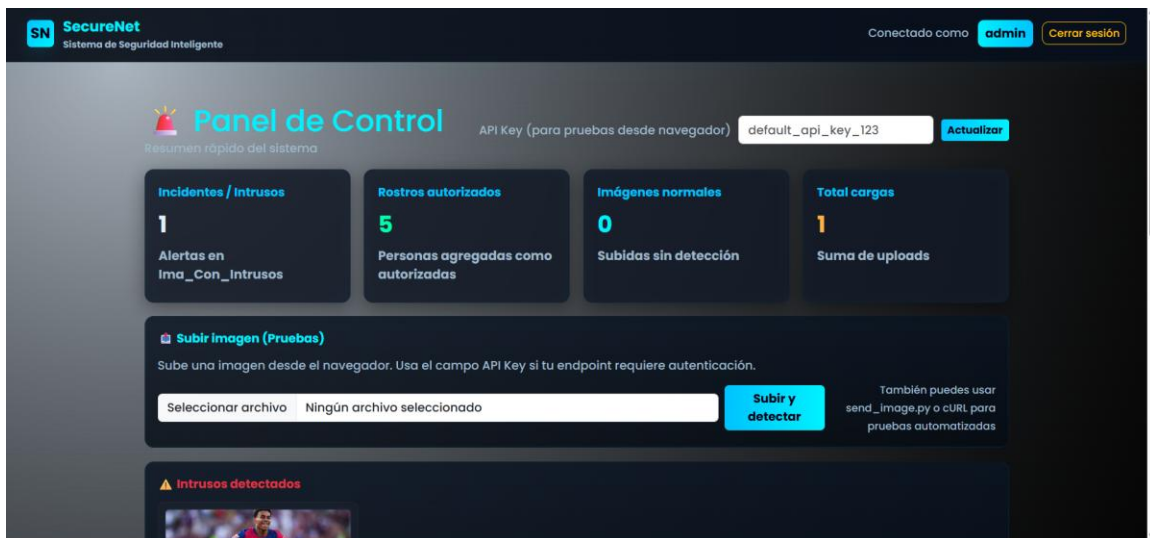
SecureNet Dashboard:

- Header: SecureNet Sistema de Seguridad Inteligente. Conectado como **admin**. [Cerrar sesión](#)
- Panel de Control: Resumen rápido del sistema. API Key (para pruebas desde navegador): [Actualizar](#)
- Statistics:
 - Incidentes / Intrusos:** 1. Alertas en `Ima_Con_Intrusos`
 - Rostros autorizados:** 5. Personas agregadas como autorizadas
 - Imágenes normales:** 0. Subidas sin detección
 - Total cargas:** 1. Suma de uploads
- Subir imagen (Pruebas): Sube una imagen desde el navegador. Usa el campo API Key si tu endpoint requiere autenticación.
 - Selección de archivo:
 - [Subir y detectar](#)
 - También puedes usar `send_image.py` o cURL para pruebas automatizadas
- Intrusos detectados:

Terminal Output:

```
PS C:\Users\JoahansP\Desktop\ProyectoCamara> & C:\Users\JoahansP\AppData\Local\Programs\Python\Python313\python.exe c:\Users\JoahansP\Desktop\ProyectoCamara\aut
h.py
Error creando usuario: UNIQUE constraint failed: users.username
No se pudo crear admin (quizá ya existe)
PS C:\Users\JoahansP\Desktop\ProyectoCamara> & C:\Users\JoahansP\AppData\Local\Programs\Python\Python313\python.exe c:\Users\JoahansP\Desktop\ProyectoCamara\app
.py
2025-09-23 01:42:12,331:757: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floatin
g-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_OEINNN OPTS=0'.
2025-09-23 01:42:19,017918: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floatin
g-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_OEINNN OPTS=0'.
WARNING:tensorflow:From C:\Users\JoahansP\AppData\Local\Programs\Python\Python313\Lib\site-packages\tf_keras\src\losses.py:2976: the name tf.losses.sparse_softm
ax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

2025-09-23 01:42:25,900 - Detecccion - INFO - Rostros autorizados cargados: 0
2025-09-23 01:42:26,037 - apscheduler.scheduler - INFO - Scheduler started
2025-09-23 01:42:27,778 - Servidor - INFO - Bot de Telegram iniciado correctamente
2025-09-23 01:42:27,785 - Servidor - INFO - Iniciando servidor en 0.0.0.0:5000
* Serving Flask app 'app'
* Debug mode: off
2025-09-23 01:42:27,980 - werkzeug - INFO - WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instea
d.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.100.7:5000
2025-09-23 01:42:27,988 - werkzeug - INFO - Press CTRL+C to quit
2025-09-23 01:42:31,608 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:31] "GET / HTTP/1.1" 200 -
2025-09-23 01:42:31,805 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:31] "GET /api/stats HTTP/1.1" 200 -
2025-09-23 01:42:31,840 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:31] "GET /static/pattern.png HTTP/1.1" 404 -
2025-09-23 01:42:31,921 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:31] "GET /api/alerts HTTP/1.1" 200 -
2025-09-23 01:42:32,169 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /api/images HTTP/1.1" 200 -
2025-09-23 01:42:32,316 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /alerts/20250923_003204_038021_yamal.jpg HTTP/1.1" 304 -
2025-09-23 01:42:32,394 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /api/authorized-faces HTTP/1.1" 200 -
2025-09-23 01:42:32,514 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /authorized/20250923_005526_559742_neymar2.jpg HTTP/1.1" 304 -
2025-09-23 01:42:32,515 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /authorized/20250923_005545_899293_neymar.jpg HTTP/1.1" 304 -
2025-09-23 01:42:32,518 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /authorized/descarga.jpg HTTP/1.1" 304 -
2025-09-23 01:42:32,526 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /authorized/ds_model_vggFace_detector_opencv_aligned_normalization_base_ex
pand_0.pk1 HTTP/1.1" 304 -
2025-09-23 01:42:32,538 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:32] "GET /authorized/messi.jpg HTTP/1.1" 304 -
2025-09-23 01:42:36,783 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:36] "GET /api/stats HTTP/1.1" 200 -
2025-09-23 01:42:36,818 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:36] "GET /api/alerts HTTP/1.1" 200 -
2025-09-23 01:42:36,835 - werkzeug - INFO - 127.0.0.1 - - [23/Sep/2025 01:42:36] "GET /api/images HTTP/1.1" 200 -
Ln 119, Col 1 Spaces: 4 UTF-8 CRF Python 3.13.7 (venv)
```

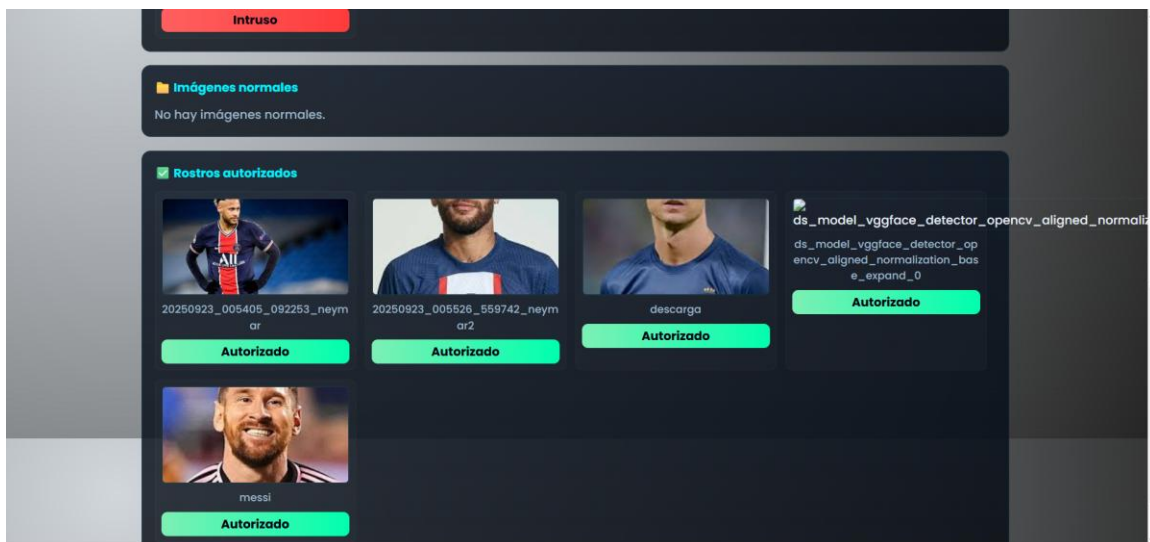


3. Subida de imágenes al servidor

El sistema permite enviar imágenes al endpoint /upload desde:

- El formulario web.
- Un script como send_image.py.
- Herramientas como Postman.

El servidor valida la imagen, la guarda temporalmente y ejecuta el reconocimiento facial.

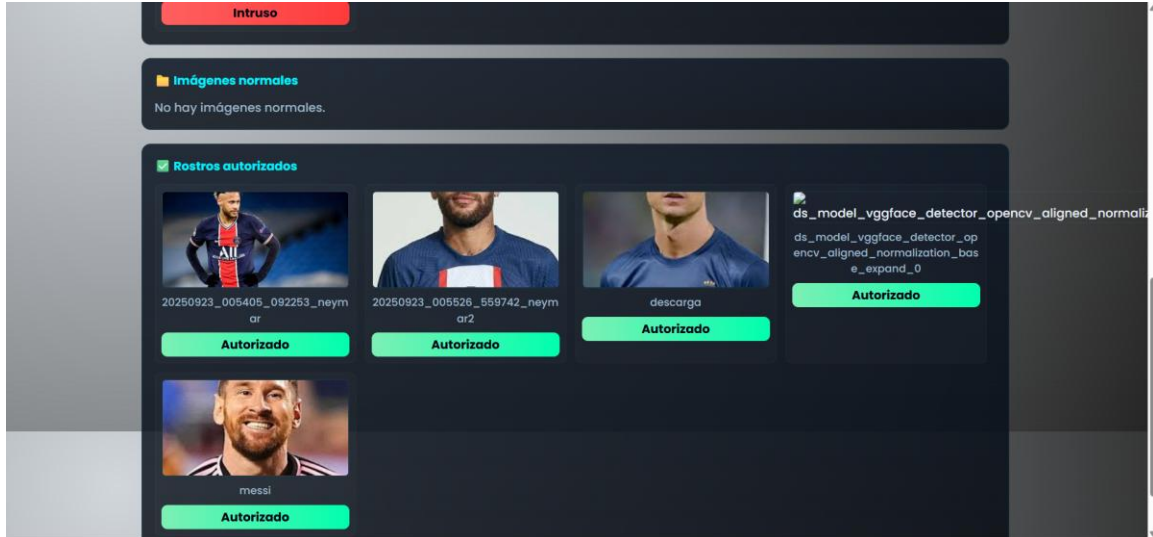
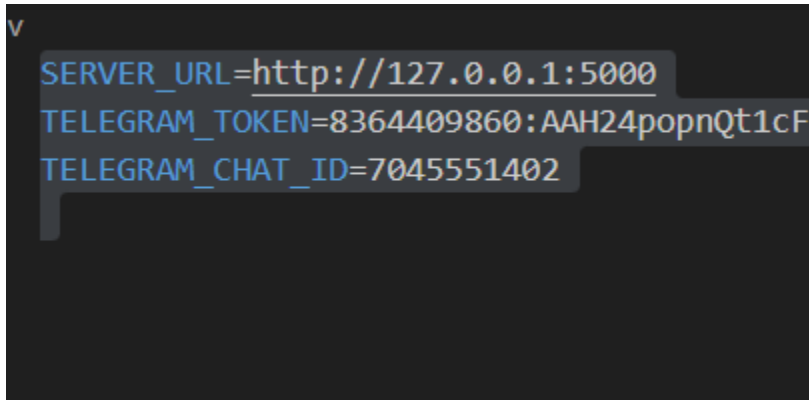


4. Procesamiento automático

El servidor clasifica las imágenes de la siguiente forma:

- Si la cara coincide con un rostro autorizado (ej. Messi, Cristiano): se marca como 'Autorizado' y queda en la galería normal.
- Si no coincide (ej. Neymar): se clasifica como 'Intruso', se mueve a la carpeta correspondiente y se envía alerta por Telegram.
- Si no se detecta rostro: el sistema responde 'no_faces'.

De esta forma, el sistema discrimina entre rostros conocidos y desconocidos.

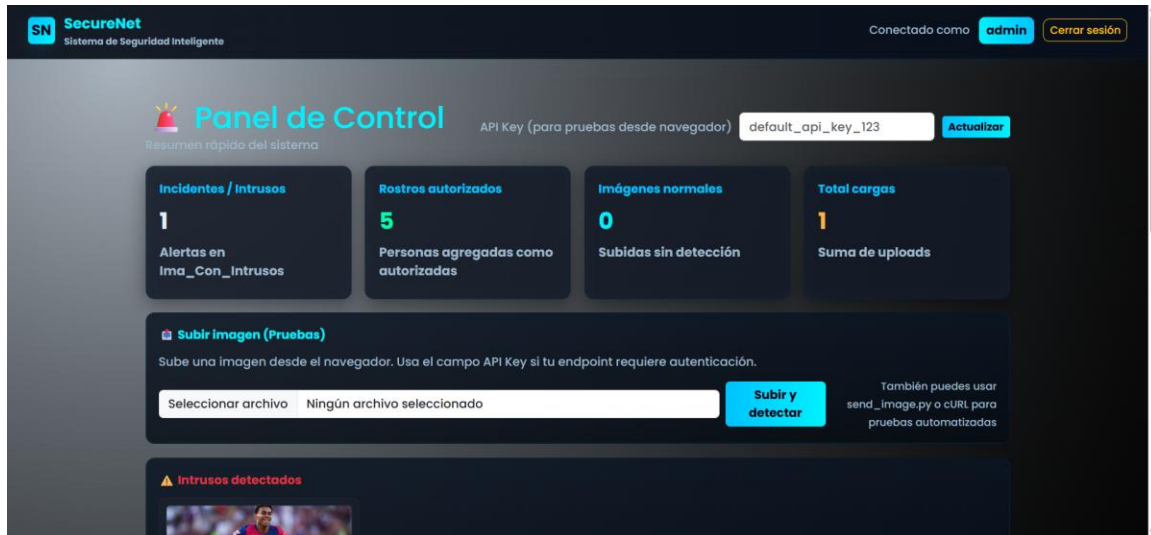


5. Actualización en el dashboard

El dashboard se actualiza automáticamente cada 5 segundos mostrando:

- Número total de imágenes procesadas.
- Número de intrusos detectados.
- Galería de rostros autorizados.
- Galerías de imágenes normales e intrusos.

Esto brinda al usuario un control visual en tiempo real.



6. Resultado del avance

Con este avance se demuestra que el servidor funciona correctamente en las funciones principales:

- Autenticación de usuario.
- Subida y validación de imágenes.
- Reconocimiento facial mediante DeepFace.
- Clasificación entre autorizados e intrusos.
- Notificación inmediata a través de Telegram.
- Visualización de resultados en el dashboard web.

El sistema está listo para continuar con fases de optimización y pruebas adicionales.

⚠ Intrusos detectados



20250923_003204_038021_yamal.jpg

Intruso

127.0.0.1:5000 dice

🔔 Intruso detectado y notificado a Telegram

Aceptar

⚠ Intrusos detectados



20250923_003204_038021_yamal.jpg

Intruso



20250923_022020_589092_con3.jpg

Intruso

2:21



CH

Chat id
bot



20250923_005405_092253_neymar a
autorizados.

12:54 AM



✗ Error al agregar 20250923
_005526_559742_neymar2 a
autorizados.

12:55 AM



⚠️ **20250923_022020_5890
92_con3.jpg** marcado como
intruso.

2:20 AM



Mensaje



2:17



Chat id
bot



19 de septiembre

/start 2:54 AM ✓✓

23 de septiembre



⚠️ **20250923_003204_03802
1_yamal.jpg** marcado como
intruso.

12:33 AM



❌ Error al agregar
20250923_005405_092253_neymar a
autorizados.

12:54 AM



Mensaje



2:20



CH

Chat id
bot



❌ Error al agregar 20250923_005526_559742_neymar2 a autorizados.

12:55 AM



🚩 Detección de Visitante

¿Es una persona autorizada?

2:20 AM



✅ Autorizar

❌ Intruso



Asignar nombre



Mensaje

