

## **Sistema de detección de placas vehiculares**

### **Objetivo general**

Desarrollar e implementar un sistema automatizado de detección y reconocimiento de placas vehiculares en estacionamientos, con el fin de identificar y registrar en tiempo real a los vehículos que ocupan espacios no autorizados con el fin de fortalecer el control, la trazabilidad y la correcta utilización de áreas asignadas.

### **Descripción del problema**

El problema se presenta en los estacionamientos donde, pese a contar con señalizaciones visibles y espacios específicamente designados para ciertos usuarios, es común que algunos conductores utilicen estos lugares sin contar con la autorización para hacerlo. Esto afecta directamente a las personas que requieren de estos espacios por motivos de movilidad o acceso prioritario, sino que también puede provocar conflictos entre otras personas usando el estacionamiento, así como también puede servir de incentivo para que otras personas continúen con estas conductas.

Tomando como ejemplo el Tecnológico de México Campus Culiacán, podemos observar que algunas personas tienden a acaparar espacios para discapacitados, estacionarse en aceras o lugares los cuales no están designados como estacionamiento, incluso llegando a tapar espacios donde hay tránsito de coches.

La gestión tradicional de este tipo de infracciones depende de la supervisión manual por parte del personal encargado del estacionamiento, lo cual representa un proceso limitado, reactivo y sujeto a errores humanos. La naturaleza dinámica de estos espacios hace que sea difícil mantener un monitoreo constante y preciso, lo que ocasiona que muchas conductas irregulares pasen desapercibidas o no sean atendidas debidamente.

### **Justificación**

Ante este problema, surge la necesidad de implementar un sistema automatizado de detección y reconocimiento de placas vehiculares que permita identificar, registrar y documentar de manera inmediata los casos en que un vehículo ocupa un espacio no autorizado. Con todo esto se busca llevar de una manera más eficiente y controlada la trazabilidad de las infracciones, optimizar los procesos de supervisión y reforzar el correcto uso de los espacios designados.

### **Modelo empleado para el proyecto**

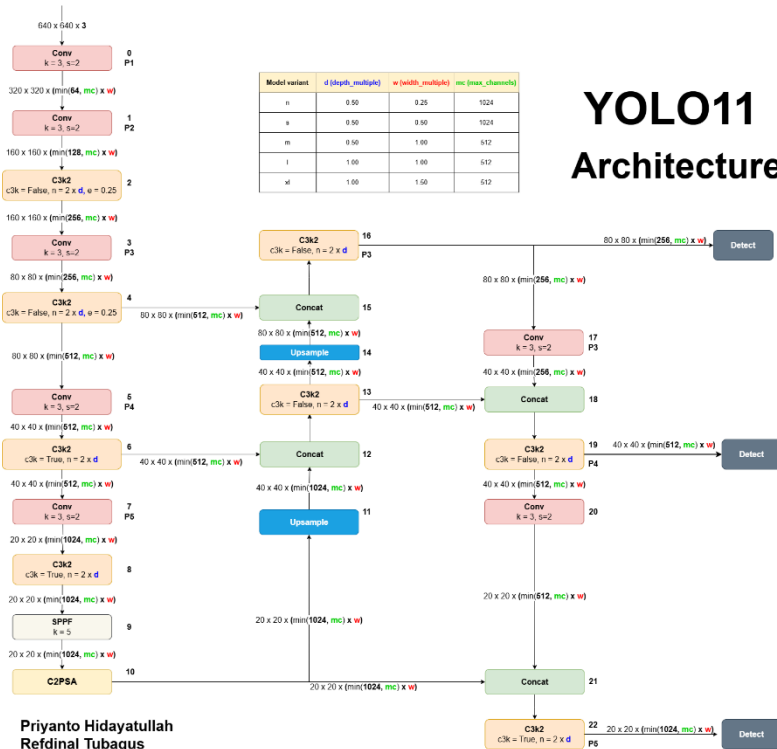
La arquitectura empleada fue YOLO (You Only Look Once). Es un modelo de detección de objetos que analiza toda la imagen de una sola vez, emplea una CNN (red neuronal convolucional) para llevar a cabo este proceso. Funciona de manera que toma la imagen y la divide en una rejilla, donde cada celda predice si hay un objeto, su ubicación mediante celdas delimitadoras (o bounding boxes) y la probabilidad de pertenecer a cierta clase. Usa “anchor boxes” para ajustar el tamaño de los objetos y después aplica un filtro NMS (Non-Maximum Suppression o

Supresión No Máxima) para eliminar las celdas duplicadas y quedarse con las más precisas.

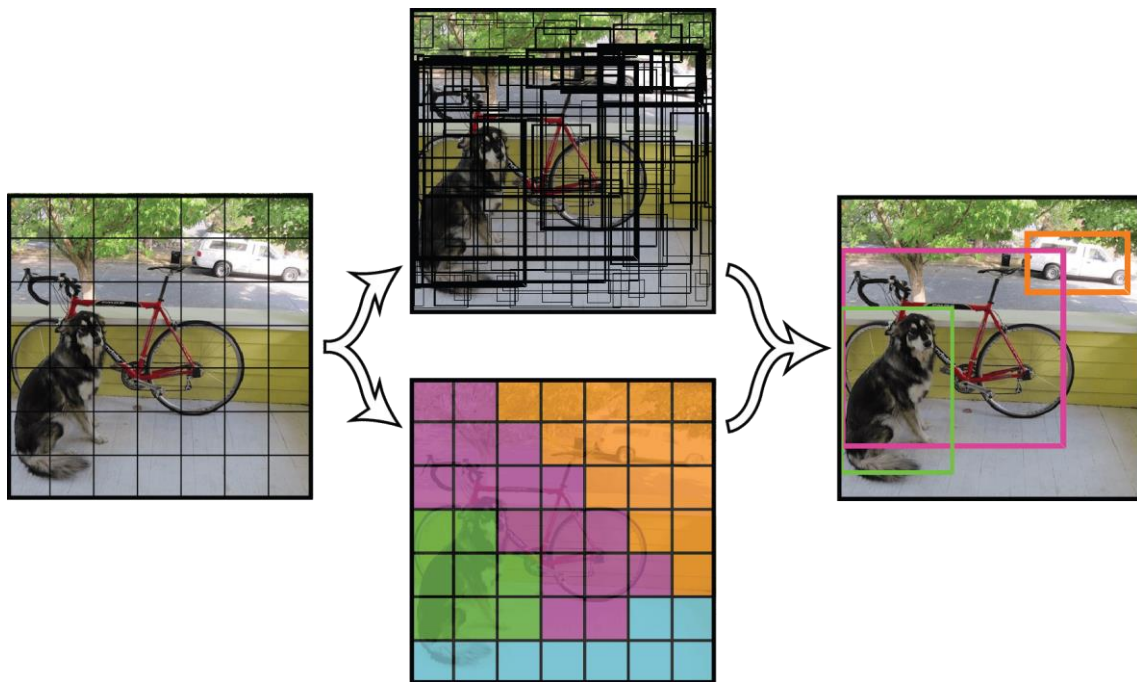
Hay tres componentes principales en la arquitectura YOLO:

- **Columna vertebral:** Sirve como extracción de características básicas y utiliza esta CNN para convertir los datos de la imagen original en un mapa de características de múltiples escalas.
- **Cuello:** Es un paso de procesamiento intermedio que aumenta la expresión de característica a través de capas especiales que agregan y fortalecen características de varias escalas.
- **Cabeza:** Es un mecanismo de predicción que genera la posición y clasificación del objeto final basándose en un mapa de características refinado.

Aquí podemos observar la arquitectura de YOLOv11 a partir del archivo yolo11.yaml en el github de ultralytics.

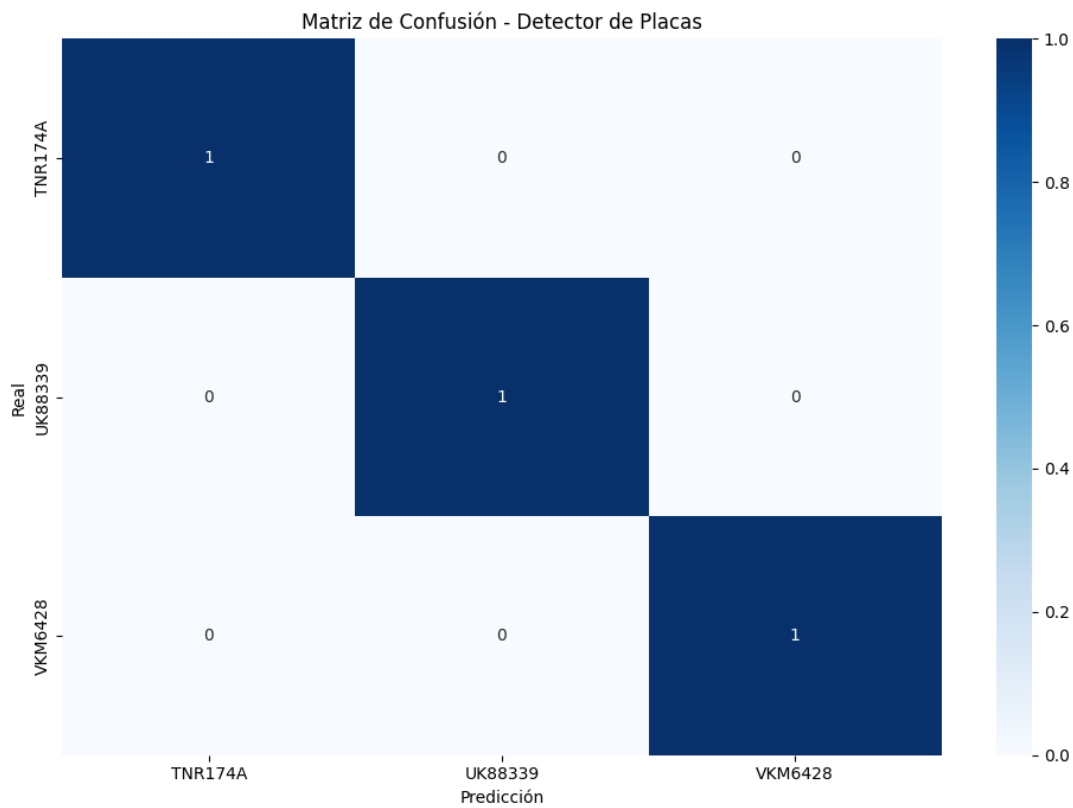


Aquí se puede apreciar la aplicación del filtro NMS.



## Prueba con el modelo YOLO

Se realizó una prueba de matriz de confusión que permite evaluar el rendimiento del modelo al comparar las predicciones del modelo con los valores reales



```

MATRIZ DE CONFUSIÓN
      TNR174A  UK88339  VKM6428
TNR174A      1      0      0
UK88339      0      1      0
VKM6428      0      0      1
REPORTE DE CLASIFICACIÓN
      precision  recall  f1-score  support
      TNR174A      1.00    1.00    1.00      1
      UK88339      1.00    1.00    1.00      1
      VKM6428      1.00    1.00    1.00      1
      accuracy
      macro avg      1.00    1.00    1.00      3
      weighted avg    1.00    1.00    1.00      3
  
```

El resultado fue bueno: solo que la prueba fue con solo 3 imágenes para poder detectar las placas y extraer el texto

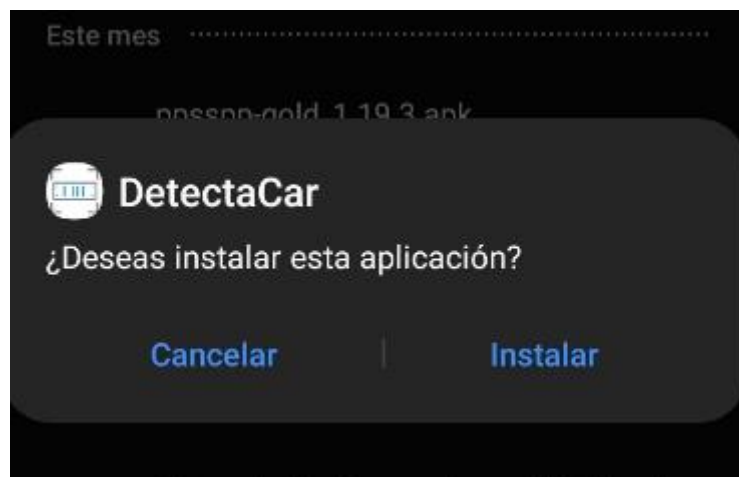
# Sistema de detección de placas vehiculares: Manual de usuario

## Instalación

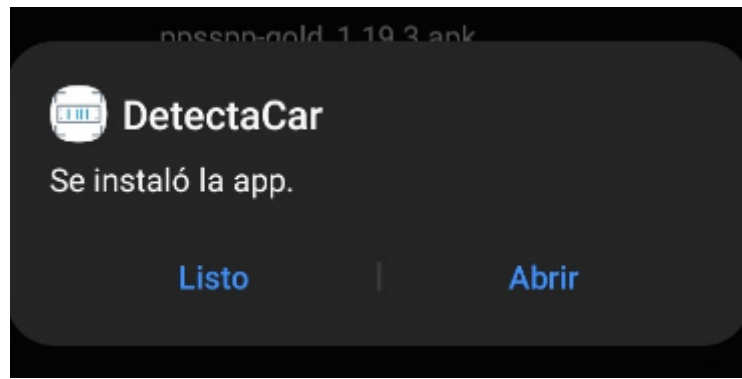
**IMPORTANTE:** Para poder instalar el APK en su dispositivo Android debe de tener la versión de Android 11 (como mínimo) una versión inferior a esa no instalara la aplicación



Primero, hemos de ingresar al repositorio del proyecto y descargar el apk desde el enlace presentado en el README.

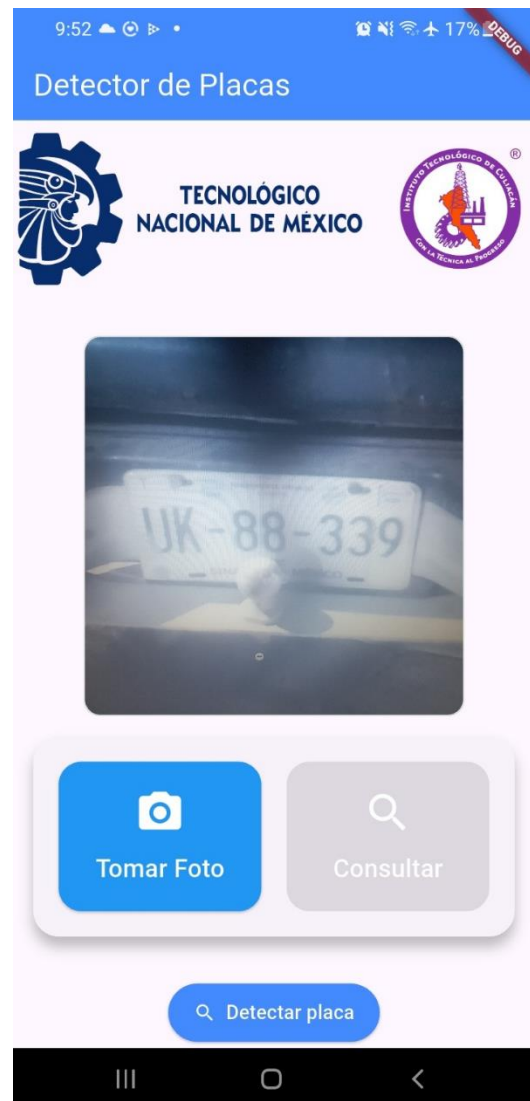
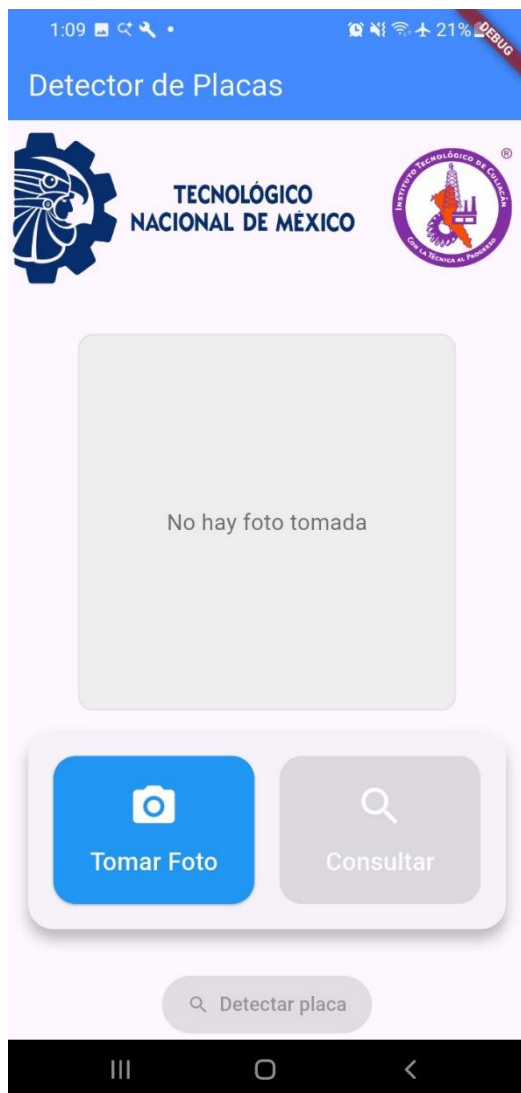


Siguiente a la descarga, hemos de hacer clic en el archivo descargado, e instalar la aplicación.



Una vez instalada, daremos clic en abrir, para ahora empezar a usar la aplicación.

## Uso del sistema

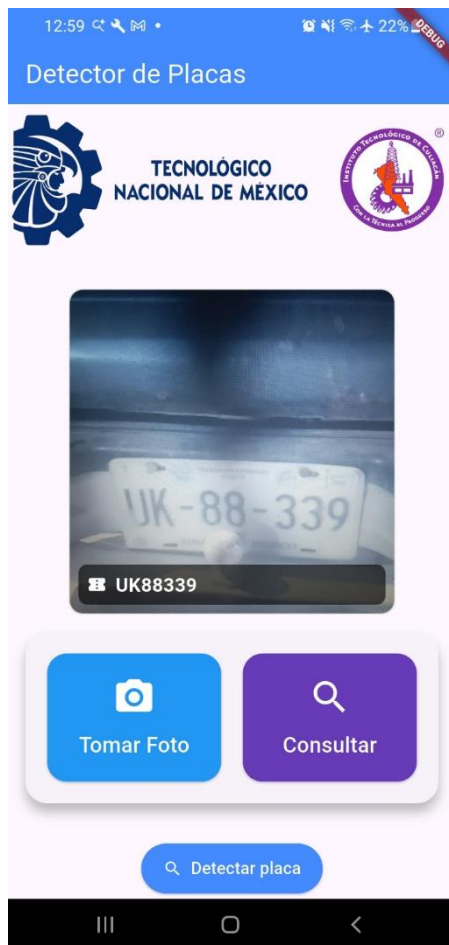


Recién entrando a la aplicación “DetectaCar”, nos encontraremos con dos botones, uno para tomar la foto de la placa, otro para detectarla, y finalmente consultar con la base de datos la placa. Necesitamos primero tomar una foto.

**IMPORTANTE:** Se tiene que aceptar los permisos de cámara y ubicación para que pueda funcionar correctamente la aplicación.



Una vez tomada la foto, se habilitará el siguiente botón de “Detectar placa”, pulsamos este para que trabaje nuestro modelo de IA.



Una vez detectada la placa, se desbloqueará el siguiente botón, donde se consultará con nuestra base de datos el número de las placas para así obtener los datos de la persona propietaria del coche.

Al consultar, nos muestra los datos del alumno, junto con dos botones, cerrar y enviar correos. Al presionar en enviar correos, serán notificados tanto el alumno responsable de la incidencia como del departamento correspondiente a este.

## **Solución de problemas comunes**

### **1. Detección de placa**

Es posible que la aplicación no detecte la placa. Esto podría deberse a varios factores, entre ellos la cámara. Deberemos de asegurarnos que la cámara esté limpia y la foto sea clara, tomándola desde una distancia corta donde sea apreciable la placa completa y que los caracteres sean claros. Otro problema podría ser la conectividad. La aplicación manda la foto al servidor donde nuestro modelo de IA se encargará de detectar la placa y tomar la información de esta. Por ello es necesario verificar que contamos con una conexión a internet estable.

### **2. Caracteres erróneos**

Como todo sistema, el nuestro no está exento de errores y es posible que el modelo llegue a leer los caracteres de la placa de manera incorrecta. Para ello nos debemos de cerciorar que la imagen que hayamos tomado sea clara, tomando en cuenta factores que afecten la visibilidad de nuestra imagen como clima, hora del día, iluminación, entre otras.

### **3. Errores de consulta**

Si se llega a presentar un error en la consulta, puede que sea debido a dos factores; Caracteres erróneos, lo cual ya fue explicado en el punto anterior; problemas en el servidor, que pueden llegar a ser cosas como problemas de conectividad con este o problemas para encontrar la consulta, posiblemente siendo el problema la ausencia de los datos dentro de nuestra base de datos.

#### 4. Error al conectar con los servidores

Si se llega a presentar este error, hay dos variables que tenemos que tomar en cuenta: nuestra conexión a internet y el estado del servidor. Primeramente, hemos de determinar si nuestro dispositivo

### **Detalles técnicos y manual de instalación**

#### **Instalación de librerías**

Se requieren las siguientes librerías para que el sistema funcione correctamente:

fastapi, sqlalchemy, pydantic, ultralytics, paddleocr, cv2 y numpy.

Para la instalación de estas librerías, usaremos el comando “Python -m pip install (escribir aquí el nombre de la librería)”

```
python -m pip install "fastapi[standard]"
```

```
python -m pip install SQLAlchemy
```

```
python -m pip install paddlepaddle (necesaria para paddleocr)
```

```
python -m pip install paddleocr (posee características básicas de reconocimiento de texto)
```

```
python -m pip install opencv-python
```

```
python -m pip install numpy
```

Una vez instaladas todas las librerías, descargaremos el proyecto de github (Es importante aclarar que github no posee

una opción para descargar carpetas directamente, por lo que se necesita un tercero, ya sea una web o programa para poder hacerlo). Una vez descargados todos los archivos necesarios, podemos pasar a configurar y modificar el proyecto de manera que se pueda aplicar a diferentes contextos.

## Conexión a la base de datos

En nuestro proyecto, entraremos al archivo en la dirección “Backend/app/db/database.py” y modificaremos la línea 4.

```
2 from sqlalchemy.orm import Sessionmaker, declarative_base
3 # configuracion de la base de datos sqlite
4 DATABASE_URL = "sqlite+aiosqlite:///./app/db/placas.db"
5
```

Agregaremos el URL de nuestra base de datos para hacer la conexión y poder consultar los datos dentro de esta.

Es importante aclarar que necesitamos modificar el formato de nuestra aplicación para que esta pueda acceder a los datos de nuestra base de datos. Para esto, necesitamos entrar a la dirección “Backend/app/models”, aquí encontraremos dos archivos: alumnos.py y placavehicular.py, modificaremos los formatos de estas de acuerdo a nuestra base.

```
7 class Alumno(Base):
8     __tablename__ = "alumnos"
9
10     matricula_id = Column(String, primary_key=True)
11     nombre = Column(String, nullable=False)
12     ape_paterno = Column(String, nullable=False)
13     ape_materno = Column(String, nullable=False)
14     correo = Column(String, nullable=False)
15     celular = Column(String, nullable=False)
16     carrera = Column(String, nullable=False)
17     num_incidencias = Column(Integer, nullable=False, default="0")
18     placas = relationship("PlacaVehicular", back_populates="alumno")
```

Aquí podemos apreciar el formato de alumnos.py.

	matricula_id	nombre	ape_paterno	ape_materno	correo	celular	carrera	num_incidentes
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	A0123456	Juan	Pérez	López	121170474@culiacan.tecnm.mx	5512345678	Ingeniería en Sistemas	0

Aquí muestro el formato de la base de datos empleada para comprobar el funcionamiento del proyecto.

## Modificación de formato de correo

Ahora, me enfocaré en los correos. Entraremos a “/Backend/app/app/services/correo\_service.py”, aquí veremos todo lo relacionado con el formato de los correos y la manera en la que estos son enviados.

```
#-----
remite = ""#<-- Poner aqui el correo del remitente
password = "" #<-- poner aqui la contraseña o contraseña de aplicacion (si tiene verificacion en 2 pasos))
destinatario_institucional = "" #<-- poner aqui el correo institucional que recibira los reportes
#-----
```

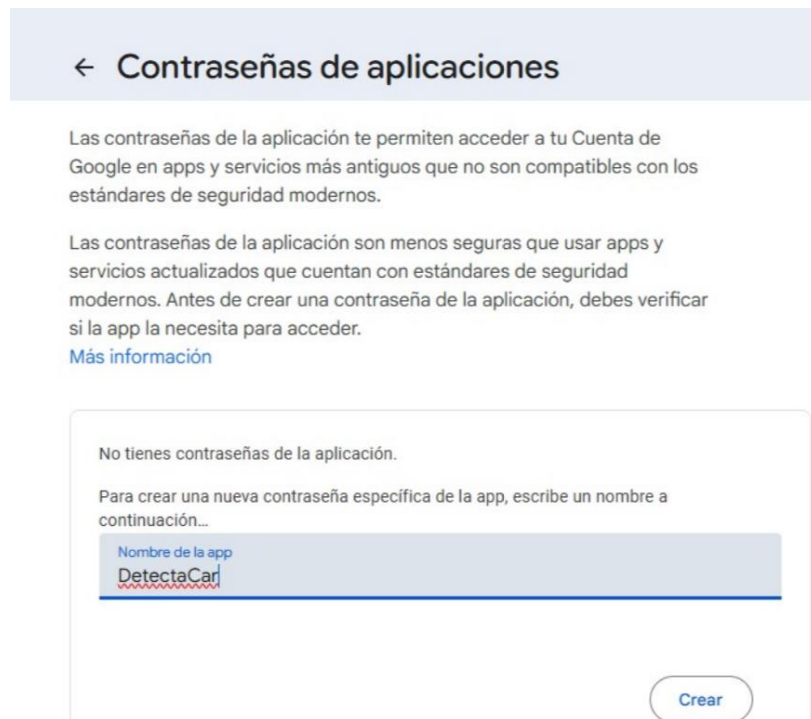
Hemos de crear un correo exclusivo para el envío de los correos, ya que estos manejan información sensible de los estudiantes.

En remitente escribiremos el correo designado. En password agregaremos la contraseña del mismo. En caso de que el correo posea verificación en dos pasos (recomendado para mayor seguridad), será necesario crear una contraseña de aplicación.

Para la creación de nuestra contraseña de aplicación, hemos de ingresar a la siguiente liga:

<https://myaccount.google.com/apppasswords>

Aquí, nos encontraremos con la siguiente pantalla



← **Contraseñas de aplicaciones**

Las contraseñas de la aplicación te permiten acceder a tu Cuenta de Google en apps y servicios más antiguos que no son compatibles con los estándares de seguridad modernos.

Las contraseñas de la aplicación son menos seguras que usar apps y servicios actualizados que cuentan con estándares de seguridad modernos. Antes de crear una contraseña de la aplicación, debes verificar si la app la necesita para acceder.

[Más información](#)

No tienes contraseñas de la aplicación.

Para crear una nueva contraseña específica de la app, escribe un nombre a continuación...

Nombre de la app  
DetectaCar

Crear

Aquí, iremos a la sección donde dice “No tienes contraseñas de la aplicación”, escribiremos el nombre de nuestra app y le daremos a crear.



**Contraseña de aplicación generada**

Tu contraseña de aplicación para el dispositivo

**dfjf nsah zbix hxdq**

**Instrucciones de uso**

Ve a la configuración de tu cuenta de Google en la aplicación o el dispositivo que quieres configurar. Ingresa la contraseña de 16 caracteres que aparece arriba para reemplazar la anterior.

Al igual que la contraseña normal, esta contraseña de la aplicación otorga acceso completo a tu cuenta de Google. Como no es necesario que la recuerdes, no la escribas ni la compartas con nadie.

Se nos dará esta contraseña, es la que debemos de agregar a nuestro apartado de “password” en el código.

Si se quiere emplear este proyecto en un ámbito diferente, se pueden modificar el mensaje enviado al estudiante en esta función dentro del mismo módulo de correo\_service.py.

```
107
108     # Genera el cuerpo del correo para el estudiante con ayuda de los datos recibidos
109     # dependiendo del numero de incidencias genera un mensaje diferente
110     def mensajecorreoEstudiante(correo_request: CorreoRequest):
111
```

De igual manera, más adelante podemos encontrar el mismo formato para el mensaje que se envía al departamento de la institución.

```
    # Genera el cuerpo del correo para la institucion con ayuda de los datos recibidos
    # dependiendo del numero de incidencias genera un mensaje diferente
    def mensajecorreoInstitucion(correo_request: CorreoRequest):
        if correo_request.num_incidencias >= 3:
```

Cabe destacar que podemos modificar el número máximo de incidencias, donde si este límite es excedido, podemos mandar un correo hablando sobre la sanción hacia el infractor, y mientras no sea excedido, será notificado de la infracción y se le dará una advertencia.

Link del repositorio:

[https://github.com/JSarevalo25/TopicosdeIA/tree/main/U4/Detector\\_Placas\\_Vehiculares](https://github.com/JSarevalo25/TopicosdeIA/tree/main/U4/Detector_Placas_Vehiculares)

APK: [https://itculiacanedu-my.sharepoint.com/:u:/g/personal/l21170314\\_culiacan\\_tecn\\_m\\_mx/IQA0T7U4m6JtR5SpQ\\_ABO21XAWHBpvy01EJOG8uOKkmRu-E?e=wfMXag](https://itculiacanedu-my.sharepoint.com/:u:/g/personal/l21170314_culiacan_tecn_m_mx/IQA0T7U4m6JtR5SpQ_ABO21XAWHBpvy01EJOG8uOKkmRu-E?e=wfMXag)

Video-Demostración: [https://itculiacanedu-my.sharepoint.com/:v:/g/personal/l21170314\\_culiacan\\_tecn\\_m\\_mx/IQAM27IX\\_g6wSJe7fXwZF7TzAaHWsqQ7E8O9E3RK\\_Rblo2w?nav=eyJyZWZlcnJhbEluZm8iOmsicmVmZXJyYWxBcHAI](https://itculiacanedu-my.sharepoint.com/:v:/g/personal/l21170314_culiacan_tecn_m_mx/IQAM27IX_g6wSJe7fXwZF7TzAaHWsqQ7E8O9E3RK_Rblo2w?nav=eyJyZWZlcnJhbEluZm8iOmsicmVmZXJyYWxBcHAI)



[OiJPbmVEcml2ZUZvckJ1c2luZXNzliwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6lidlYilsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=oacXf7](#)