

# A. Manual de usuario

En este anexo se presentará un manual de usuario que detallará el uso de la aplicación y su interfaz.

## A.1. Interfaz de usuario

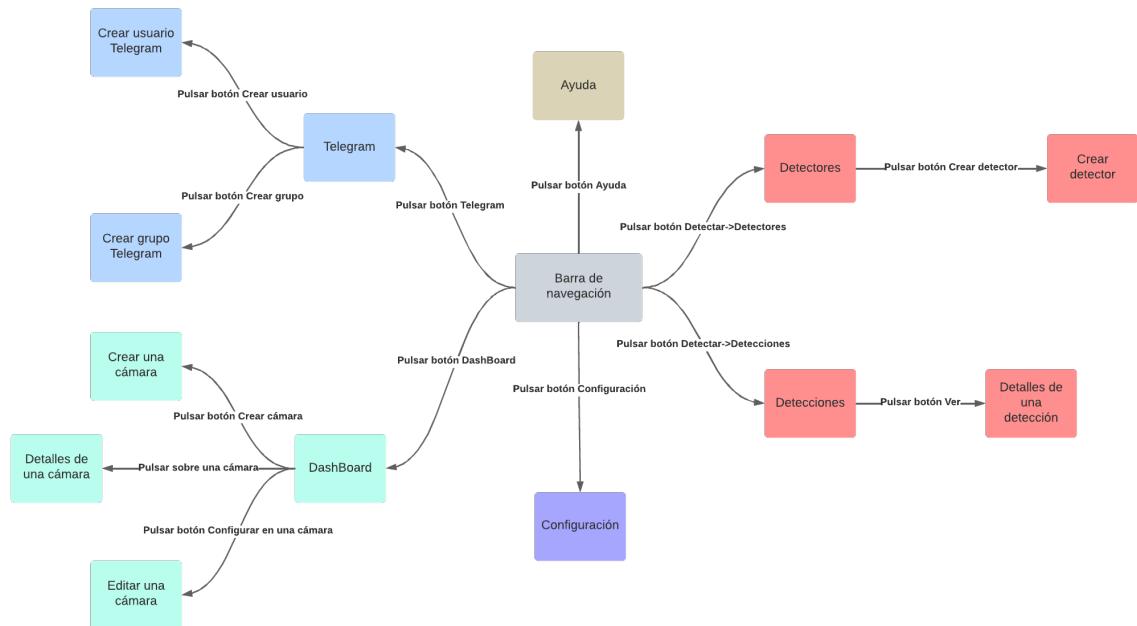


Figura A.1: Mapa conceptual de la interfaz

En esta sección mostraremos y explicaremos las distintas pantallas con las que el usuario se encontrará durante el uso de la aplicación.

### A.1.1. Dashboard

Esta es la pantalla principal de la aplicación y en la que el usuario pasará el mayor tiempo. Desde aquí el usuario puede acceder al vídeo en directo de todas las cámaras y encontrará botones para crear o editar cámaras.

En caso de no tener ninguna cámara cargada en la aplicación , la pantalla se mostrará vacía con el mensaje *No se ha cargado ninguna cámara todavía.*

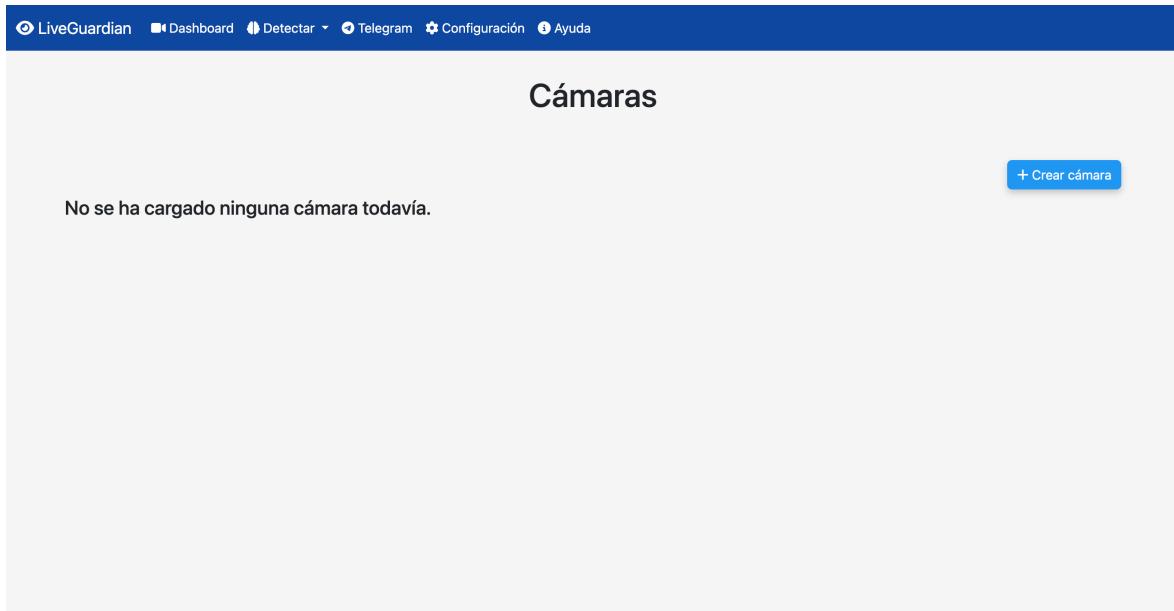


Figura A.2: Dashboard sin cámaras

Si la base de datos contiene cámaras, el usuario podrá acceder a previsualizaciones en tiempo real de todas las cámaras.

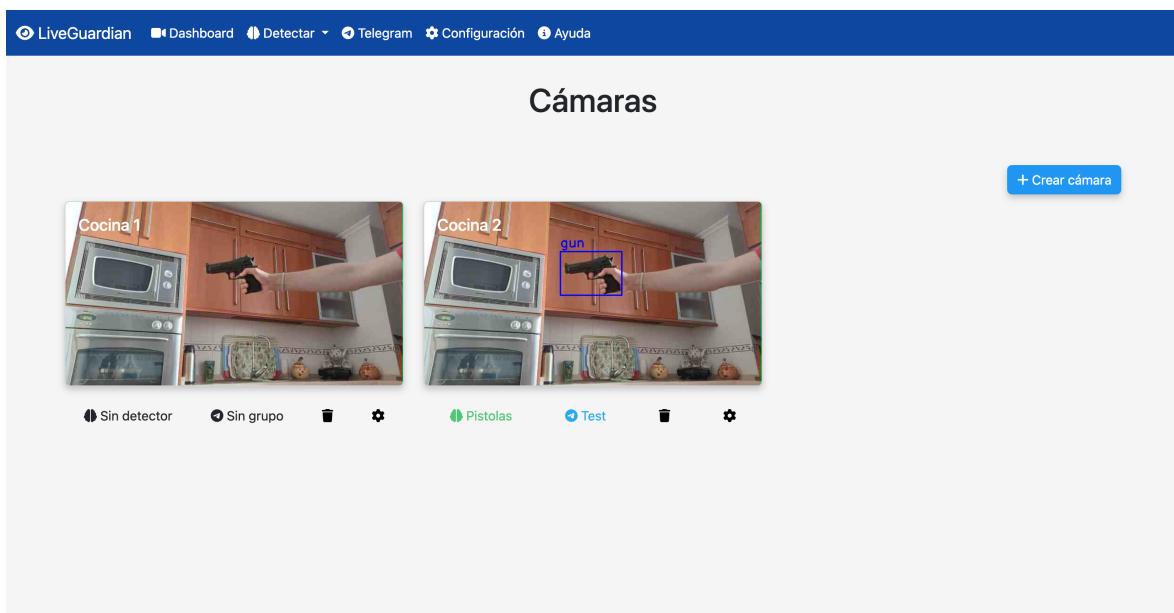


Figura A.3: Dashboard con cámaras

Desde la previsualización, obtenemos información acerca del detector empleado, el grupo de Telegram al que está asignada la cámara y botones para eliminarla o configurarla. Además, si se hace clic sobre la imagen, es posible acceder a una vista detallada de la cámara.

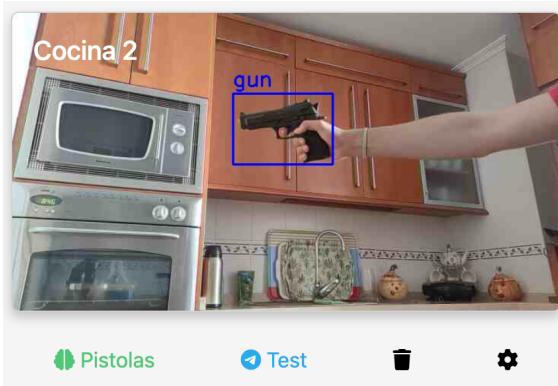


Figura A.4: Previsualización cámara

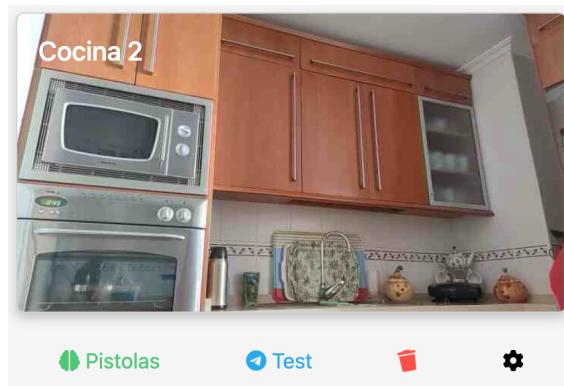


Figura A.5: Animación borrar

Si alguna cámara presenta problemas o pierde la conexión, la aplicación mostrará automáticamente una imagen de error en dicha cámara, junto con un contador que indica el tiempo transcurrido desde que comenzó a fallar. Si la cámara vuelve a estar operativa, retomará la transmisión de imágenes y la detección de objetos.

Figura A.6: Cámara fallando

### A.1.2. Detalles de una cámara

Al hacer clic en la previsualización de una cámara, el usuario será dirigido a los detalles de la misma. En esta pantalla, podrá observar la cámara en tamaño ampliado y tendrá a su disposición varias opciones en la esquina superior izquierda de la imagen.

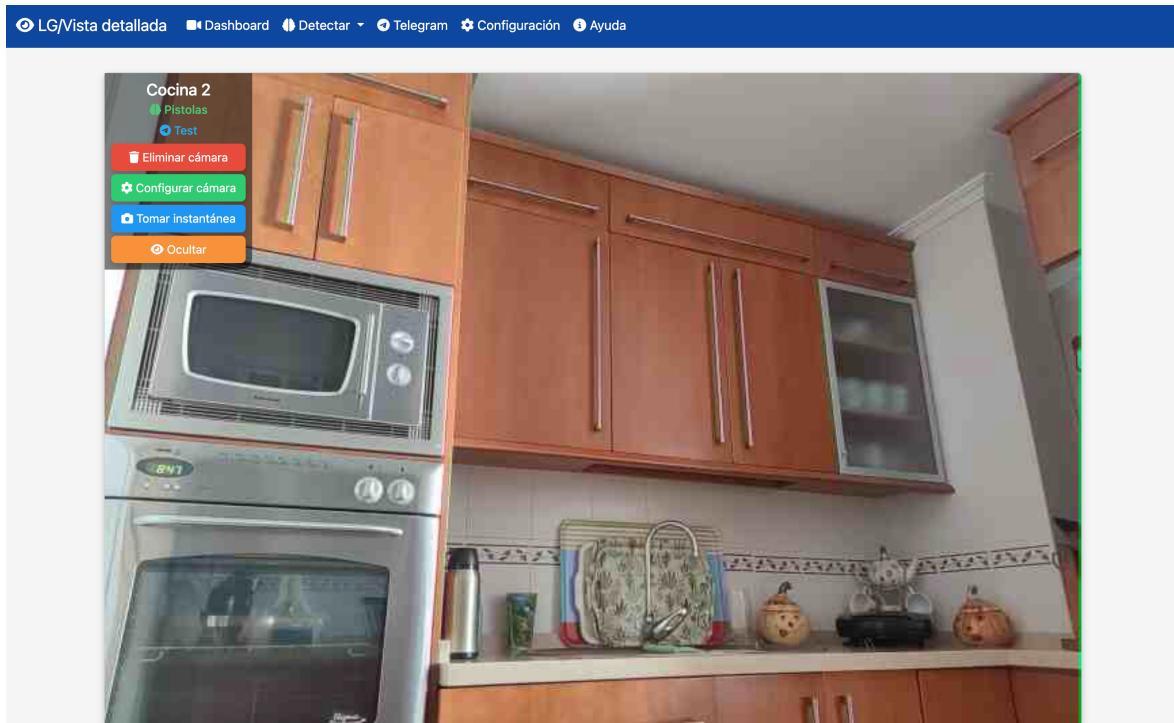


Figura A.7: Detalles cámara

En el menú situado en la esquina superior izquierda, el usuario puede acceder a información detallada de la cámara, como su nombre, el detector utilizado y el grupo asignado, así como a varias opciones, como eliminarla o configurarla, tomar una instantánea (esta instantánea se guardará como una detección sin objetos) y ocultar el menú.



Figura A.9: Panel oculto

Figura A.8: Panel desplegado

### A.1.3. Creación/Edición de una cámara

En esta pantalla, el usuario tiene la posibilidad de crear/editar una cámara ingresando su nombre, detector, URL de la cámara y el grupo de Telegram para recibir

alertas en caso de detección. En caso de no seleccionar ningún detector, la cámara simplemene mostrará imagen en tiempo real sin realizar inferencia.

The screenshot shows the 'LG/Crear cámara' (Create Camera) page. At the top, there is a navigation bar with links: LG/Crear cámara, Dashboard, Detectar, Telegram, Configuración, and Ayuda. The main title 'LG/Crear cámara' is centered above a form. The form fields include:

- Nombre de la cámara: Salón de casa
- Seleccione un detector: -----
- Url de la cámara: http://192.168.1.36:8080/video
- Seleccionar grupos: Test

A blue 'Crear' (Create) button is located at the bottom right of the form area.

Figura A.10: Creación de una cámara

The screenshot shows the 'LG/Editar cámara' (Edit Camera) page. At the top, there is a navigation bar with links: LG/Editar cámara, Dashboard, Detectar, Telegram, Configuración, and Ayuda. The main title 'LG/Editar cámara' is centered above a form. The form fields are identical to the creation form:

- Nombre de la cámara: Cocina
- Seleccione un detector: -----
- Url de la cámara: http://192.168.1.36:8080/video
- Seleccionar grupos: Test

A blue 'Crear' (Create) button is located at the bottom right of the form area.

Figura A.11: Editar una cámara

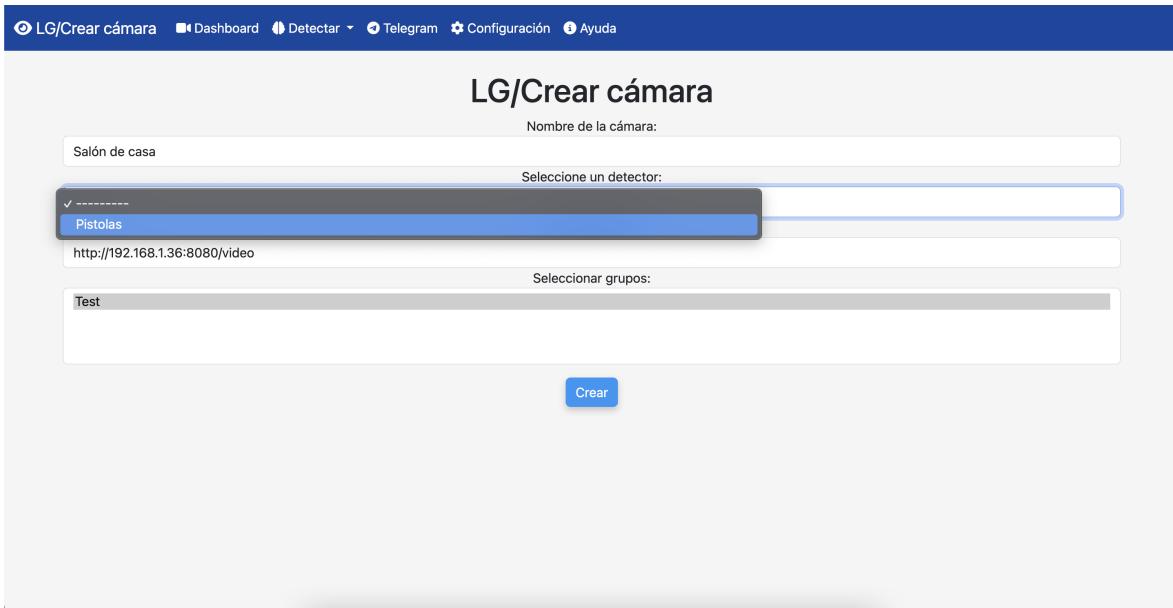


Figura A.12: Selector de detector en formulario cámara

#### A.1.4. Configuración

En esta pantalla, el usuario tiene la opción de modificar la configuración de la aplicación. Entre las opciones disponibles se incluyen: el número de fotogramas que deben contener una detección de objetos para ser registrada, el tamaño del historial de los últimos fotogramas con o sin detecciones, el intervalo de fotogramas en el cual se procesa la detección de objetos y el token del bot de Telegram que se utilizará para enviar las alertas.

- **Intervalo de fotogramas:** número de fotogramas en el cual se procesa la detección de objetos. Cuanto mayor sea este intervalo, menos fotogramas pasarán por el detector y más libre quedará este.
- **Tamaño del historial:** número de fotogramas anteriores considerados para enviar detecciones.
- **Número de fotogramas para detectar:** umbral de fotogramas con detección requerido en el historial para registrar una detección de objetos.
- **Token bot Telegram:** código que se utiliza para identificar y autenticar un bot en la API de Telegram.

Configuración

Intervalo de fotogramas	3 Este intervalo indica cada cuantos fotogramas será uno introducido en el detector.
Tamaño del historial	40 Número de fotogramas almacenados (un número mayor puede producir retraso en las detecciones)
Nº de fotogramas para detectar	10 Número de fotogramas necesarios en el historial para producirse una detección.
Token bot Telegram	5282233910:AAG_mddkn8zdw_lip-n1zQX_gSURiBLomC0 Token del bot de Telegram utilizado en el sistema.

**Guardar**

Figura A.13: Configuración de la aplicación

### A.1.5. Detectores

En esta pantalla, el usuario puede visualizar una lista con todos los detectores de objetos subidos a la aplicación, además de gestionar los mismos, creando o eliminando detectores.

Detectores

Eliminar	Nombre	Modelo
	Pistolas	models/dataSetGrande.pth

**+ Crear detector**

Figura A.14: Lista de detectores

### A.1.6. Crear detección

En esta pantalla el usuario puede subir sus modelos entrenados para la detección de objetos. Tan solo tiene que asignar un nombre al detector y subir el archivo con su modelo. Este modelo debe de ser un archivo de un modelo entrenado con Icevision, ya que necesitamos sus metadatos para inicializar la arquitectura del modelo.

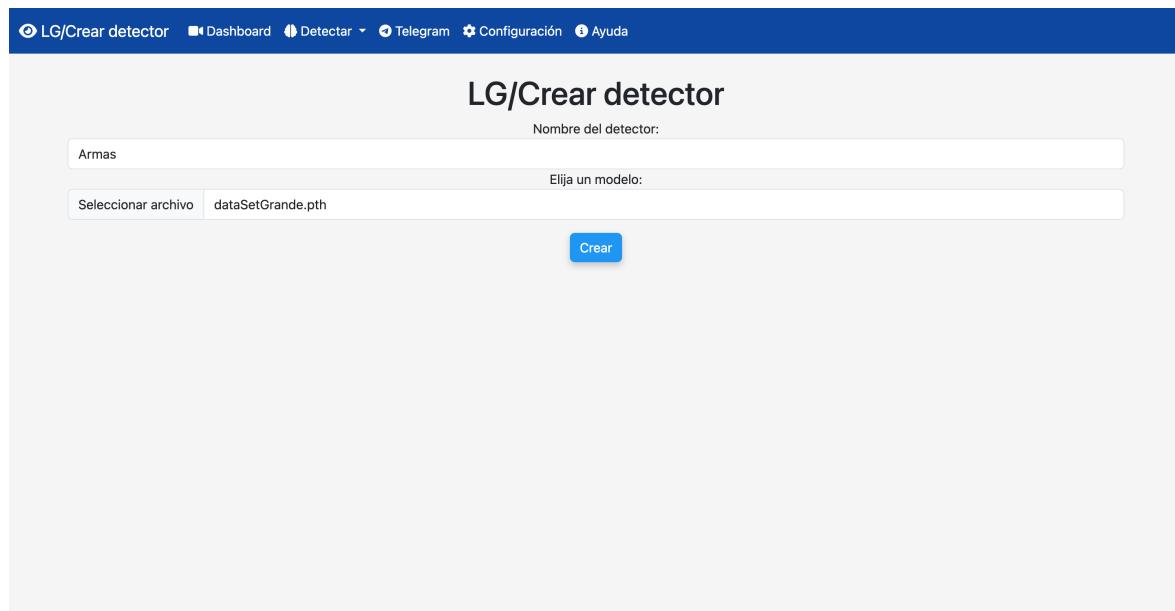


Figura A.15: Subir un detector

### A.1.7. Detecciones

En esta pantalla, el usuario puede ver un resumen de todas las detecciones ocurridas en una lista y tiene la opción de acceder a cada una de ellas para obtener una vista más detallada.

Detecciones				
①	🔗 Id	📷 Cámara	📅 Fecha	🔫 Detector
<a href="#">Ver</a>	10	Cocina 2	1 de mayo de 2023 a las 18:22	Pistolas
<a href="#">Ver</a>	9	Cocina 2	1 de mayo de 2023 a las 18:21	Pistolas

Figura A.16: Lista de detecciones

### A.1.8. Detección detallada

Cuando el usuario hace clic en el botón *Ver* en la lista de detecciones, accede a una vista detallada que muestra información adicional sobre la detección, como los objetos detectados, las probabilidades de cada objeto y una imagen del momento en que se detectó. Además, el usuario tiene la opción de eliminar la detección y su imagen.

LG/Vista detallada				
Dashboard				
				
<b>Detección #10</b> <ul style="list-style-type: none"> <li>📷 Cámara: Cocina 2</li> <li>🔫 Detector: Pistolas</li> <li>📦 Items: ['gun']</li> <li>✖ Probabilidades: [ 0.8669 ]</li> <li>📅 Fecha: 1 de mayo de 2023 a las 18:22</li> </ul>				
<input type="button" value="Eliminar detección"/>				

Figura A.17: Detalles de una detección

### A.1.9. Telegram

En esta pantalla el usuario puede gestionar de forma directa los Usuarios y Grupos de Telegram, permitiendo eliminar existentes o crear nuevos.

The screenshot shows a dashboard with the following sections:

- Groups:** A table with one entry: "Test" (Nombre), "1" (Nº usuarios). Includes a "Crear grupo" button.
- Users:** A table with one entry: "José Joaquín Virtudes Castro" (Nombre), "900224881" (Chat Id), "Test" (Grupo). Includes a "Crear usuario" button.

Figura A.18: Usuarios y grupos de Telegram

### A.1.10. Crear usuario de Telegram

En esta pantalla, el usuario puede crear un usuario de Telegram. Para ello, debe proporcionar un nombre, el ID del chat de Telegram y el grupo al que se asignará dicho usuario.

LG/Crear usuario

Nombre del usuario:

José Joaquín Virtudes Castro

Id del chat:

900224881

Grupos:

Test

Crear

Figura A.19: Crear usuario de Telegram

### A.1.11. Crear grupo de Telegram

Para crear un grupo de Telegram, el usuario tan solo tendrá que indicar un nombre.

LG/Crear grupo

Nombre del grupo:

Seguridad interna

Crear

Figura A.20: Crear grupo de Telegram

### A.1.12. Mensaje de Telegram

Una vez que el usuario ha configurado un grupo y un detector en una cámara, en caso de que se produzca una detección, el usuario de Telegram recibirá un mensaje estructurado con la siguiente forma:

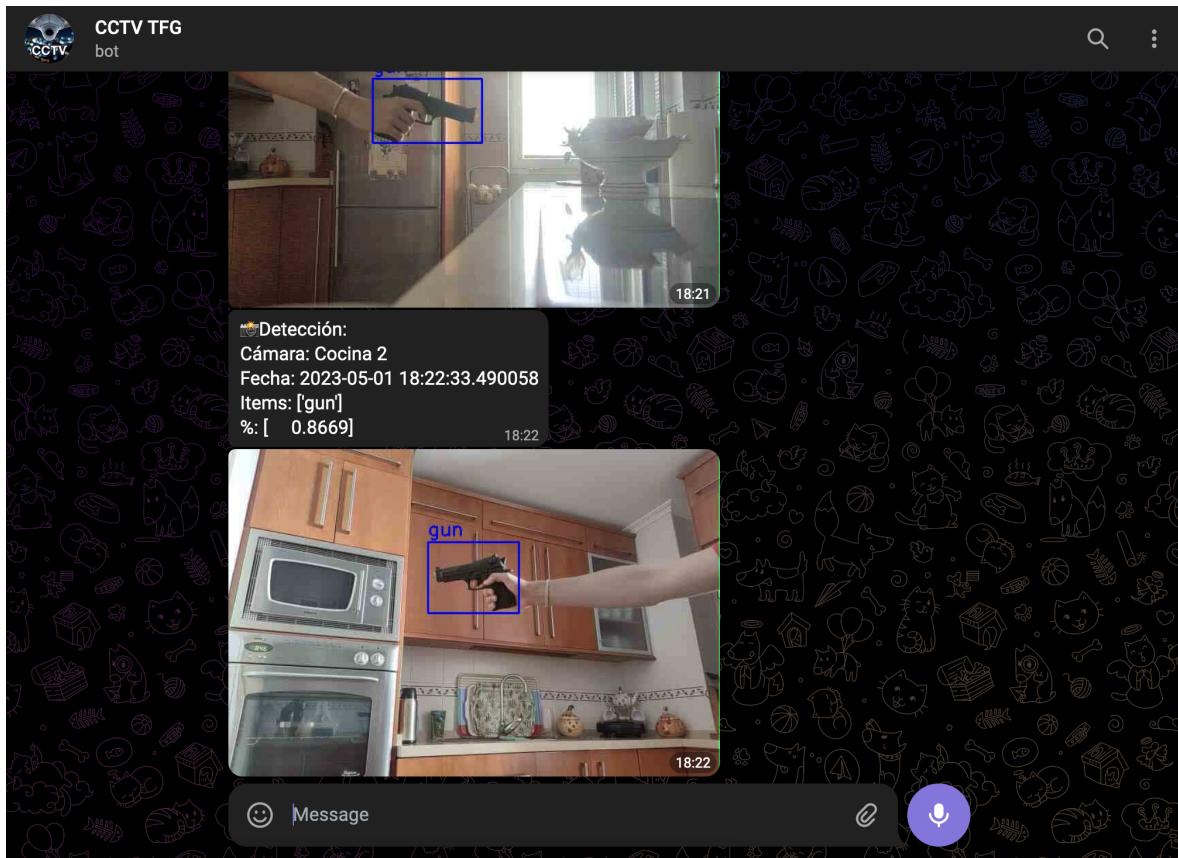


Figura A.21: Chat Telegram

# B. Instalación

---

En este anexo, se proporcionarán las instrucciones necesarias para instalar y poner en marcha las herramientas necesarias para la ejecución de la aplicación. Se presentarán los pasos necesarios para instalar Django, IceVision y OpenCV. Además se hablará del proceso de creación y configuración de un bot de Telegram.

## B.0.1. Versiones

Para este proyecto, se han utilizado los siguientes paquetes y herramientas:

- Python 3.9.13
- OpenCV 4.6.0.66
- Django 4.1.3
- numpy 1.22.3
- Icevision 0.12.0
- pyTelegramBotApi 4.7.0
- yolo5-icevision 6.0.0
- torchvision 0.11.1
- torch 1.10.0
- fastai 2.5.6
- Pillow 8.4.0
- mmcv-full 1.3.17
- CUDA 11.4

Como veremos más adelante, dependiendo del entorno y los drivers del sistema, las versiones de estos paquetes puede cambiar. Los paquetes que mayormente se verán afectados serán torch, torchvision, mmcv-full y CUDA.

## B.0.2. Python

Si se encuentra en un sistema Windows, puede descargar Python desde el sitio web oficial o desde la tienda de Microsoft. En MacOS puede obtenerlo desde el sitio web oficial o mediante gestores de paquetes como Homebrew. En Linux, puede instalar Python desde el gestor de paquetes que proporcione su distribución.

Una vez contamos con Python instalado en nuestro sistema, es recomendable crear un entorno virtual para el proyecto. Un entorno virtual es un espacio aislado donde puedes instalar paquetes sin afectar a la instalación Python del sistema. Para crear un entorno virtual, se abrirá una terminal y ejecutarán los siguientes comandos:

---

```
1 #!/bin/bash
2 # Creamos el entorno virtual
3 python -m venv liveguardian
4 # Activamos el entorno virtual
5 source liveguardian/bin/activate
```

---

Extracto de código B.1: Crear entorno virtual

### B.0.3. CUDA

CUDA es un framework para computación paralela que permite aprovechar el poder de procesamiento de tarjetas gráficas NVIDIA.

Durante este apartado explicaremos la instalación de CUDA para Ubuntu (será similar para cualquier otra distribución linux). Para Windows, sugerimos acudir al sitio web oficial de NVIDIA para descargar el instalador de CUDA Toolkit. En el caso de macOS, se debe tener en cuenta que NVIDIA CUDA Toolkit no está oficialmente soportado.

Antes de instalar CUDA Toolkit tenemos que tener instalados los drivers de NVIDIA en nuestro sistema:

---

```
1 #!/bin/bash
2 # Cambiar <version> por una compatible con el CUDA Toolkit a instalar.
3 sudo apt update
4 sudo apt install nvidia-driver-<version>
```

---

Extracto de código B.2: Instalar drivers NVIDIA

La forma más sencilla para instalar CUDA Toolkit en Ubuntu es partiendo de los repositorios oficiales de la distribución:

---

```
1 #!/bin/bash
2 sudo apt update
3 sudo apt install nvidia-cuda-toolkit
```

---

Extracto de código B.3: Instalar CUDA Toolkit

Una vez instalado, podemos verificar la instalación comprobando la versión:

---

```
1 #!/bin/bash
2 nvcc --version
```

---

Extracto de código B.4: Comprobar versión instalada

## B.0.4. IceVision

Dependiendo de la versión instalada de CUDA o si se desea realizar inferencia mediante CPU, se tendrán que instalar distintas versiones de torch y torchvision:

---

```
1 #!/bin/bash
2
3 # CUDA 10.2
4 pip install torch==1.10.0+cu102 torchvision==0.11.1+cu102 -f https://
   download.pytorch.org/whl/torch_stable.html
5 # CUDA 11.1
6 pip install torch==1.10.0+cu111 torchvision==0.11.1+cu111 -f https://
   download.pytorch.org/whl/torch_stable.html
7 # CPU
8 pip install torch==1.10.0+cpu torchvision==0.11.1+cpu -f https://download
   .pytorch.org/whl/cpu/torch_stable.html
```

---

Extracto de código B.5: Instalar torch y torchvision

De forma opcional podemos instalar **mmcv-full** para ampliar la cantidad de modelos soportados por nuestro sistema. En nuestro caso, para correr nuestro modelo YOLOV5 no será necesaria la instalación de este paquete. Dependiendo de la versión de CUDA o si desea realizar inferencia mediante CPU, se tendrá que instalar una versión distinta:

---

```
1 #!/bin/bash
2
3 # CUDA 10.2
4 pip install mmcv-full==1.3.17 -f https://download.openmmlab.com/mmcv/dist
   /cu102/torch1.10.0/index.html
5 pip install mmdet==2.17.0
6 # CUDA 11.1
7 pip install mmcv-full==1.3.17 -f https://download.openmmlab.com/mmcv/dist
   /cu111/torch1.10.0/index.html
8 pip install mmdet==2.17.0
9 # CPU
10 pip install mmcv-full==1.3.17 -f https://download.openmmlab.com/mmcv/dist
    /cpu/torch1.10.0/index.html
11 pip install mmdet==2.17.0
```

---

Extracto de código B.6: Instalación opcional Icevision

Una vez instaladas las dependencias, procedemos a instalar Icevision. En este proyecto optaremos por su versión completa y estable:

---

```
1 #!/bin/bash
2 pip install icevision[all]
```

---

Extracto de código B.7: Instalación Icevision

## B.0.5. Django

Una vez tenemos creado y activado nuestro entorno virtual Python, la instalación de Django es tan sencilla como:

---

```
1 #!/bin/bash
2 pip install Django
```

---

Extracto de código B.8: Instalar Django

## B.0.6. Pillow

Necesitaremos esta librería para poder inyectar las imágenes en tiempo real a los modelos:

---

```
1 #!/bin/bash
2 pip install pillow
```

---

Extracto de código B.9: Instalar Pillow

## B.0.7. OpenCV

OpenCV será el encargado de procesar las imágenes de las fuentes en tiempo real, además de realizar transformaciones a las mismas:

---

```
1 #!/bin/bash
2 pip install opencv-python
```

---

Extracto de código B.10: Instalar openCV

## B.0.8. pyTelegramBotApi

---

```
1 #!/bin/bash
2 pip install pyTelegramBotAPI
```

---

Extracto de código B.11: Instalar pyTelegramBotAPI

El resto de paquetes mencionados son instalados como dependencias de los instalados en este manual. En caso de no instalarse alguno de estos, el proceso de instalación es similar al seguido anteriormente.

## B.1. Puesta en marcha

Una vez tenemos toda la instalación necesaria para ejecutar nuestra aplicación, podemos empezar con los preparativos para un primer inicio.

Antes de iniciar la aplicación, será necesario inicializar la base de datos. Para esto nos situaremos en el directorio principal de la aplicación, donde encontraremos el fichero llamado **manage.py** y ejecutaremos los siguientes comandos (es necesario tener activado el entorno virtual mencionado en la instalación):

---

```
1 #!/bin/bash
2 python manage.py makemigrations web
3 python manage.py migrate web
```

---

Extracto de código B.12: Inicializar base de datos

Una vez que hayamos configurado la base de datos, necesitaremos hacer algunos cambios en el archivo **settings.py**. Allí encontraremos dos campos que tenemos que editar: **ALLOWED\_HOSTS** y **DEBUG**.

En el primer campo, añadiremos en forma de string la IP del host de la aplicación, en el segundo, en caso de ejecutar la aplicación en un entorno de producción, **DEBUG** de True a False para desactivar los mensajes de desarrollo.

---

```
1 # settings.py
2 DEBUG = True
3 ALLOWED_HOSTS = ['Ip del Host']
```

---

Extracto de código B.13: Editar **settings.py**

Finalmente, iniciar la aplicación es tan sencillo como ejecutar el siguiente comando con la IP del host y el puerto en el que se desea desplegar la app:

---

```
1 #!/bin/bash
2 python manage.py runserver <ip_host>:<puerto>
```

---

Extracto de código B.14: Inicializar base de datos