


Reporte de resultados e instalación

Este documento describe paso a paso el proceso para implementar y operar el sistema LauCows, un proyecto de deep learning diseñado para la detección y monitoreo de vacas. Se proporcionan instrucciones para la configuración del entorno y ejecución del modelo, así como la supervisión del sistema en tiempo real. Está dirigido a desarrolladores, técnicos y operadores que buscan poner en funcionamiento el sistema con los recursos especificados, asegurando un despliegue exitoso. Además en el documento se incluyen resultados que se obtuvieron durante el desarrollo del proyecto. Los requisitos previos para la instalación del proyecto se lista a continuación:

- Hardware
 - Raspberry Pi 3
- Software
 - Sistema Operativo: Linux o Windows (con WSL)
- Herramientas necesarias:
 - Python 3.8 o superior
 - Git
 - Gestor de paquetes: pip

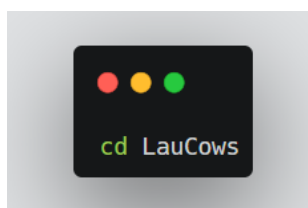
Esta guía toma lugar considerando que ya se ha conectado a la raspberry por medio de SSH o bien se está usando directamente y se encuentra dentro de la terminal.

El primer paso para la instalación y descargar el proyecto con los archivos de código y scripts necesarios. El repositorio se encuentra alojado en github y se puede descargar de la siguiente forma. <https://github.com/Amatt2-B/LauCows.git>

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The text 'git clone https://github.com/Amatt2-B/LauCows.git' is displayed in a light green monospace font.

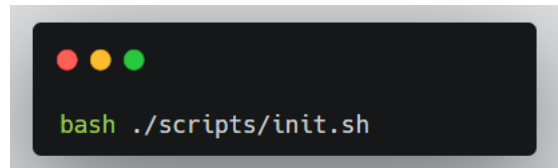
```
git clone https://github.com/Amatt2-B/LauCows.git
```

Una vez finalizado, se debe de acceder a la carpeta contenedora

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The text 'cd LauCows' is displayed in a light green monospace font.

```
cd LauCows
```

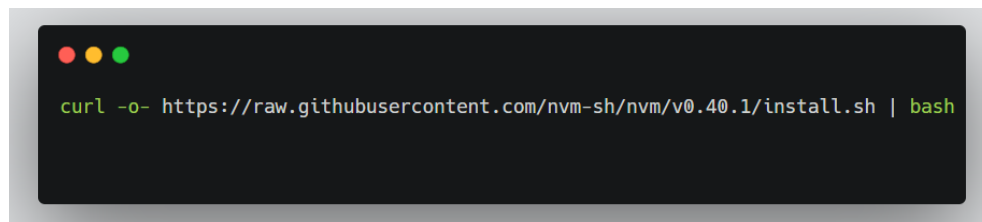
Ahora, la instalación del proyecto está configurada para que se realice a partir de la ejecución de un solo script. Este script contempla la instalación de algunos paquetes externos que son requeridos para el proyecto, así como la ejecución de los scripts necesarios



```
bash ./scripts/init.sh
```

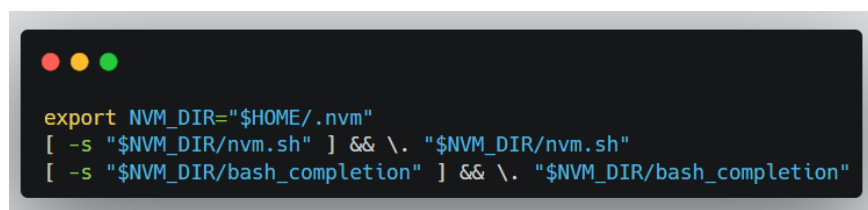
A continuación, se detalla los pasos que realiza el script de instalación en caso de que se requiera hacer manualmente y para conocer los paquetes a instalar así como la ejecución de scripts.

Primero se descarga y ejecuta el script de instalación de nvm (Node Version Manager) desde su repositorio oficial en GitHub nvm es una herramienta que facilita la instalación, actualización y administración de versiones de Node.js necesario para la ejecución del servidor del proyecto



```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
```

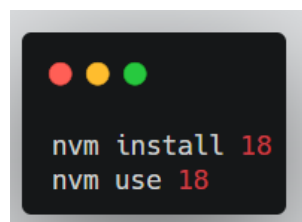
Ahora se configuran las variables de entorno necesarias para que nvm funcione en la terminal asegurando que los comandos de nvm estén disponibles en la sesión actual del shell



```
export NVM_DIR="$HOME/.nvm"  
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
```

Este comando utiliza nvm para descargar e instalar la versión 18 de Node.js que es la que utiliza el proyecto, sin embargo se puede seleccionar alguna otra.

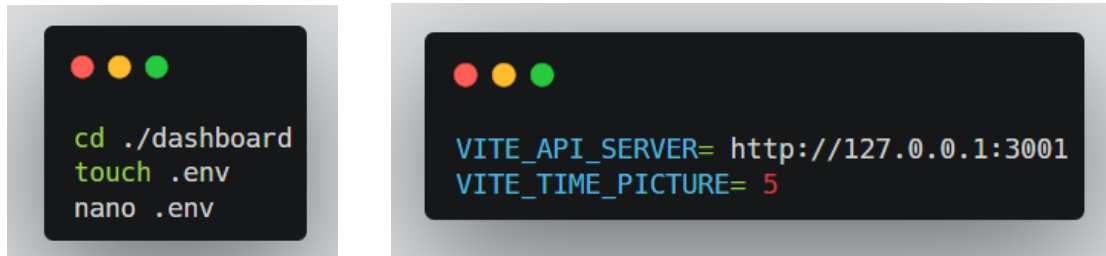
Seguido a esto, usar el comando para seleccionar la versión para estar usando.



```
nvm install 18  
nvm use 18
```

Ahora lo que sigue es crear los archivos de entorno para el dashboard. En este se definen los valores para la URL del servidor y el tiempo en el que cada foto es tomada.

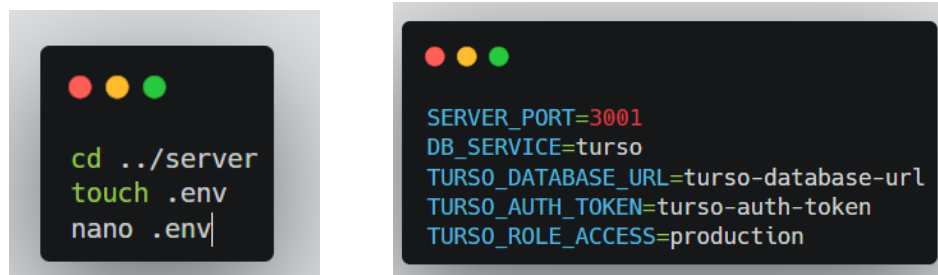
Para esto se debe acceder a la carpeta de dashboard, crear el archivo y editarlo, agregando los valores de la imagen de la derecha.



```
cd ./dashboard
touch .env
nano .env
```

```
VITE_API_SERVER= http://127.0.0.1:3001
VITE_TIME_PICTURE= 5
```

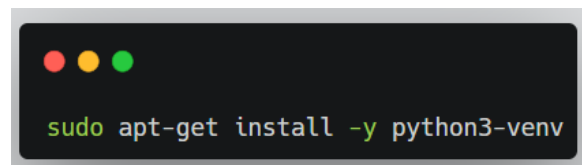
Se realiza el mismo procedimiento pero para el servidor.



```
cd ../server
touch .env
nano .env
```

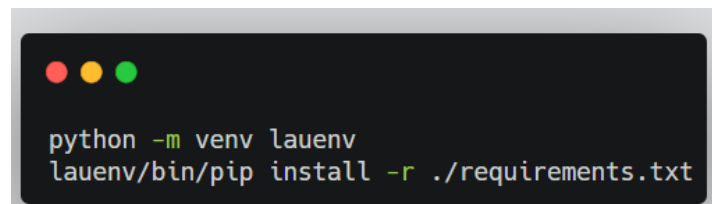
```
SERVER_PORT=3001
DB_SERVICE=turso
TURSO_DATABASE_URL=turso-database-url
TURSO_AUTH_TOKEN=turso-auth-token
TURSO_ROLE_ACCESS=production
```

Ahora, se requiere tener un entorno de ejecución de python para ejecutar el proyecto, para esto primero se instala python3-venv si es que aún no se tiene instalado.



```
sudo apt-get install -y python3-venv
```

Una vez terminada la instalación se crea el entorno virtual e instalan las dependencias del proyecto



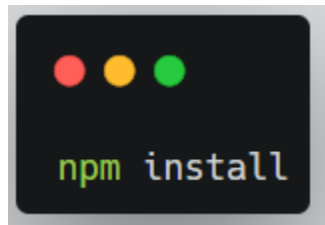
```
python -m venv lauens
lauens/bin/pip install -r ./requirements.txt
```

Se lista a continuación los paquetes a instalar por si se prefiere instalar manualmente uno por uno:

- croniter
- google-api-python-client

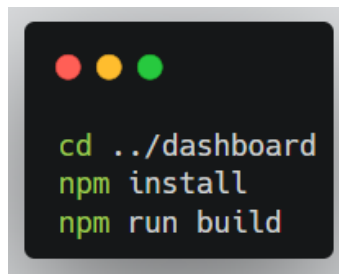
- nest-asyncio
- opencv-python
- request
- ultralytics
- ultralytics-thop
- websockets
- Numpy

Ahora lo que sigue, es instalar las dependencias del server



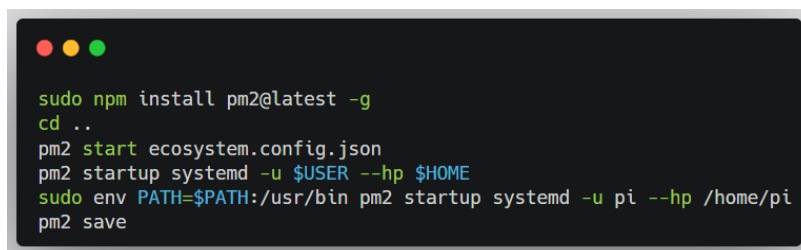
```
npm install
```

A continuación es necesario hacer lo mismo con el dashboard, pero además se necesita hacer la exportación del dashboard



```
cd ../dashboard  
npm install  
npm run build
```

Ahora, en este punto se tiene todo listo para poder ejecutar el dashboard, server y el modelo de detección. Pero es necesario usar un gestor de procesos para monitorear la ejecución del proyecto y manejar los procesos que se crean. Para ello usaremos el gestor de procesos de PM2. Con el siguiente código se instala la herramienta, ejecuta el proyecto y define la configuración para que se ejecute automáticamente si la raspberry se reinicia.



```
sudo npm install pm2@latest -g  
cd ..  
pm2 start ecosystem.config.json  
pm2 startup systemd -u $USER --hp $HOME  
sudo env PATH=$PATH:/usr/bin pm2 startup systemd -u pi --hp /home/pi  
pm2 save
```

Después de un momento el proyecto va a estar disponible para su uso, para acceder al dashboard se puede acceder a través del navegador en la siguiente URL:

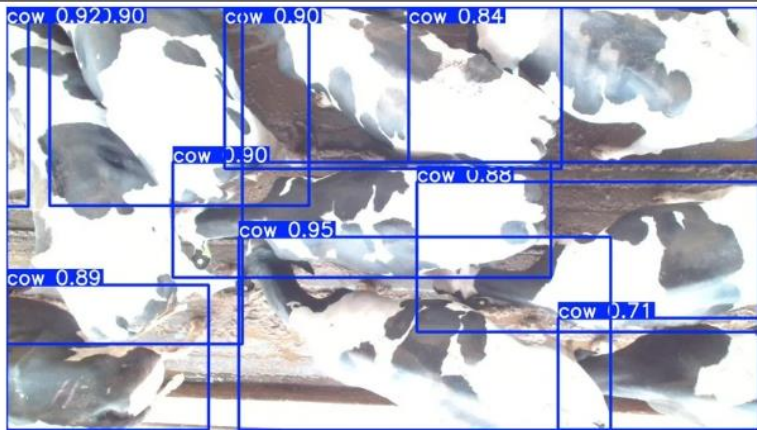
<http://localhost:5173>

Por parte del modelo se estará ejecutando el script que cada cierto tiempo va a tomar fotos las cuales serán procesadas por el modelo y guardar la predicción final para que se pueda observar en el dashboard.


Resultados

El modelo diurno y el modelo nocturno tienen enfoques especializados para la detección de vacas en condiciones específicas de iluminación. El modelo diurno alcanza una precisión del 96.39%, destacándose en condiciones de luz óptima, como las que suelen encontrarse durante el día. Es ideal para aplicaciones donde las condiciones de iluminación son constantes y favorables. Por otro lado, el modelo nocturno logra un 92.9% de precisión, siendo efectivo en situaciones de baja luz, como las que se presentan en la noche o en áreas con iluminación limitada, ofreciendo una solución confiable en estas circunstancias.

Ambos modelos cumplen funciones esenciales dentro de sus respectivos contextos. Sin embargo, al enfocarse en condiciones específicas, su rendimiento puede verse afectado cuando las condiciones de iluminación cambian. Esto resalta la importancia del modelo híbrido, que combina las fortalezas de ambos para operar eficientemente en ambientes con iluminación variable, ofreciendo una solución más integral y versátil para la detección de vacas en diferentes escenarios.

MAÑANA (06:40 AM)	RESULTADOS
	<p>Número de vacas en la foto:</p> <ul style="list-style-type: none">• Real: 9• Predicción: 9 <p>Condición de alta iluminación con una gran cantidad de vacas esperando en la fila.</p>

TARDE (18:25 PM)



cow 0.89 cow 0.89 cow 0.68 cow 0.77
cow 0.66 cow 0.66
cow 0.92 cow 0.84
cow 0.86 cow 0.83
cow 0.94


RESULTADOS

Número de vacas en la foto:

- Real: 11
- Predicción: 11

Condición de iluminación neutra con una gran cantidad de vacas esperando en la fila.

NOCHE (02:10 AM)



cow 0.92 cow 0.68 cow 0.31 cow 0.65
cow 0.73
cow 0.79 cow 0.81
cow 0.87

RESULTADOS

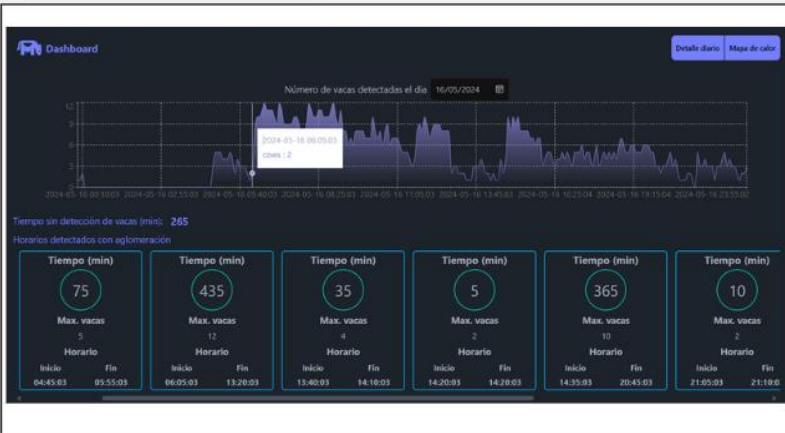
Número de vacas en la foto:

- Real: 7
- Predicción: 8

Condición de baja iluminación con una cantidad moderada de vacas esperando en la fila.

El modelo enfrentó desafíos específicos en momentos clave, como identificar únicamente una parte del cuerpo de una vaca y clasificarla incorrectamente como un animal completo. Aunque mantiene un error general de detección cercano al 5%, en esta imagen en particular presentó un falso positivo, clasificando incorrectamente un objeto como una vaca, lo que representa un 14% de error dentro de la muestra evaluada.

MONITOREO DIARIO



Dashboard

Número de vacas detectadas el día: 16/05/2024

2024-05-16 00:00:00 2024-05-16 02:00:00 2024-05-16 04:00:00 2024-05-16 06:00:00 2024-05-16 08:00:00 2024-05-16 10:00:00 2024-05-16 12:00:00 2024-05-16 14:00:00 2024-05-16 16:00:00 2024-05-16 18:00:00 2024-05-16 20:00:00 2024-05-16 22:00:00

Tiempo sin detección de vacas (min): 265

Horarios detectados con egimeración

Tiempo (min)	Tiempo (min)	Tiempo (min)	Tiempo (min)	Tiempo (min)	Tiempo (min)
75	435	35	5	365	10
Max. vacas	Max. vacas	Max. vacas	Max. vacas	Max. vacas	Max. vacas
5	12	4	2	10	2
Horario	Horario	Horario	Horario	Horario	Horario
Inicio Fin	Inicio Fin	Inicio Fin	Inicio Fin	Inicio Fin	Inicio Fin
04:45:03 05:55:03	06:05:03 13:20:03	13:40:03 14:10:03	14:20:03 14:28:03	14:35:03 20:45:03	21:05:03 21:10:03

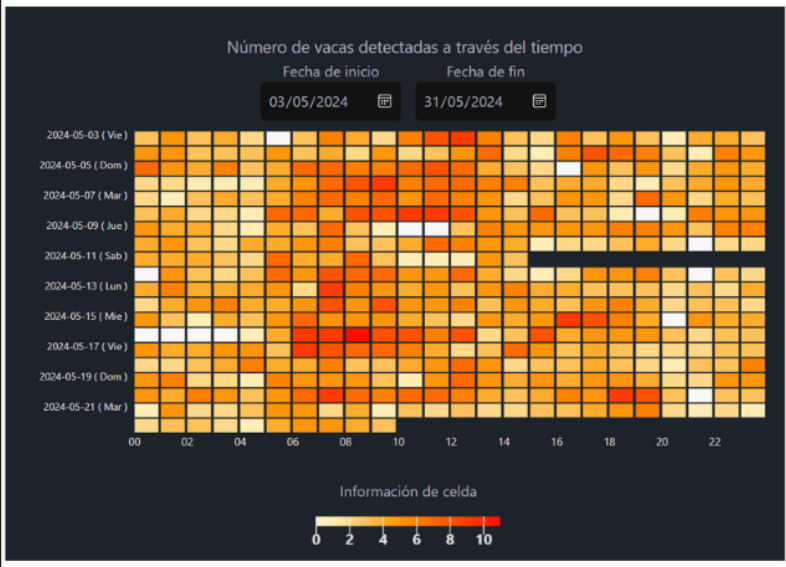
DESCRIPCIÓN

Visualiza la distribución de la cantidad de vacas registradas cada cinco minutos a lo largo de un día completo.

Además, muestra alertas indicando cuándo se detectó el máximo número de vacas y calcula el tiempo transcurrido hasta que la cantidad se redujo a dos vacas en la fila.

MAPA DE CALOR

DESCRIPCIÓN



El mapa de calor ilustra la distribución de la cantidad de vacas, facilitando el análisis de las tendencias del ganado durante el proceso de ordeño.

Este análisis puede ajustarse para abarcar un rango específico de días, desde el inicio hasta el final del periodo seleccionado.