

# Sistema de monitorización para el control y estudio del bienestar de animales de laboratorio mediante una infraestructura de bajo coste

Isabel Cebollada Gracia

[i.cebollada.2018@alumnos.urjc.es](mailto:i.cebollada.2018@alumnos.urjc.es)



Trabajo fin de grado

27 de Julio de 2022

1. En esta presentación voy a hablar sobre...
2. En primer lugar...



(CC) Isabel Cebollada Gracia

*Este trabajo se entrega bajo licencia CC BY-NC-SA.  
Usted es libre de (a) compartir: copiar y redistribuir el material en  
cualquier medio o formato; y (b) adaptar: remezclar, transformar  
y crear a partir del material. El licenciador no puede revocar estas  
libertades mientras cumpla con los términos de la licencia.*

1. La presentación esta dividida en cuatro partes.

## Contenidos

1 Introducción

2 Objetivos

3 Plataforma de desarrollo

4 Desarrollo del sistema

5 Conclusiones

# *Introducción*

# Introducción a la robótica

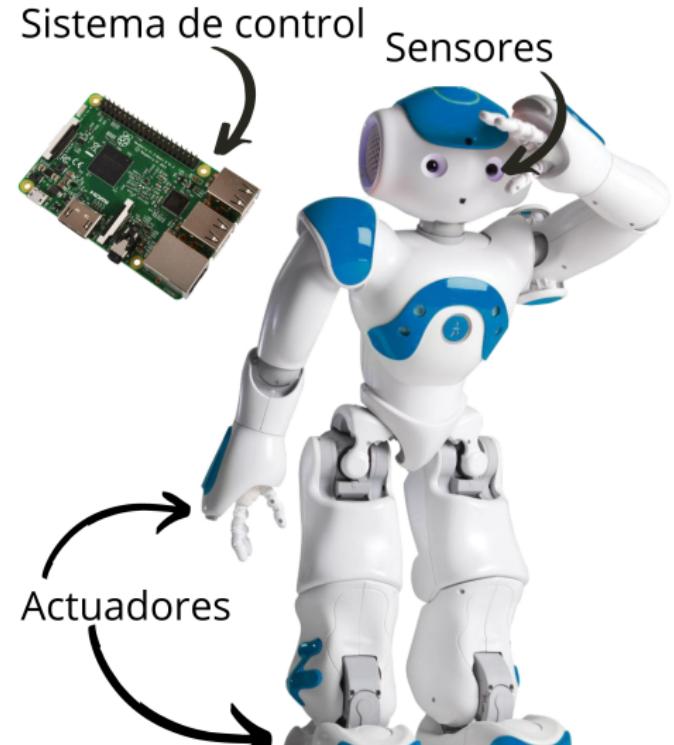
## ¿Qué es la robótica?

- Campo de investigación muy importante que continua en desarrollo.
- Objetivo: Realizar trabajos **aburridos** o **pesados** para el humano.

1. La robótica es un campo de investigación muy importante en la actualidad que continua en desarrollo, cuyo objetivo es realizar trabajos que resultan pesados, aburridos o sucios para el ser humano.

# Definición de robot

¿Qué es un robot?

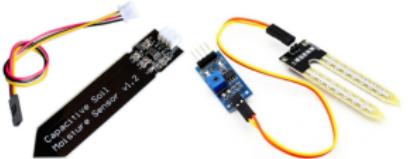


1. A lo largo de los años el concepto de robótica ha ido variando, así como la definición de robot. Actualmente, se entiende por robot a cualquier dispositivo dotado por sensores, actuadores y un sistema de control.

# Algunos tipos de sensores



Sensores de temperatura



Sensores de humedad



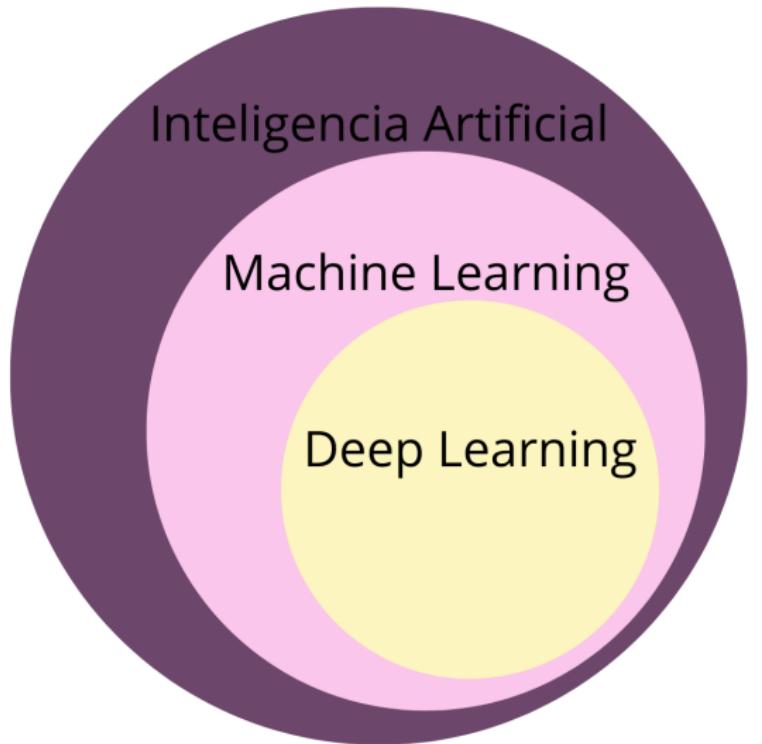
Sensores de ultrasonidos



Sensores de visión

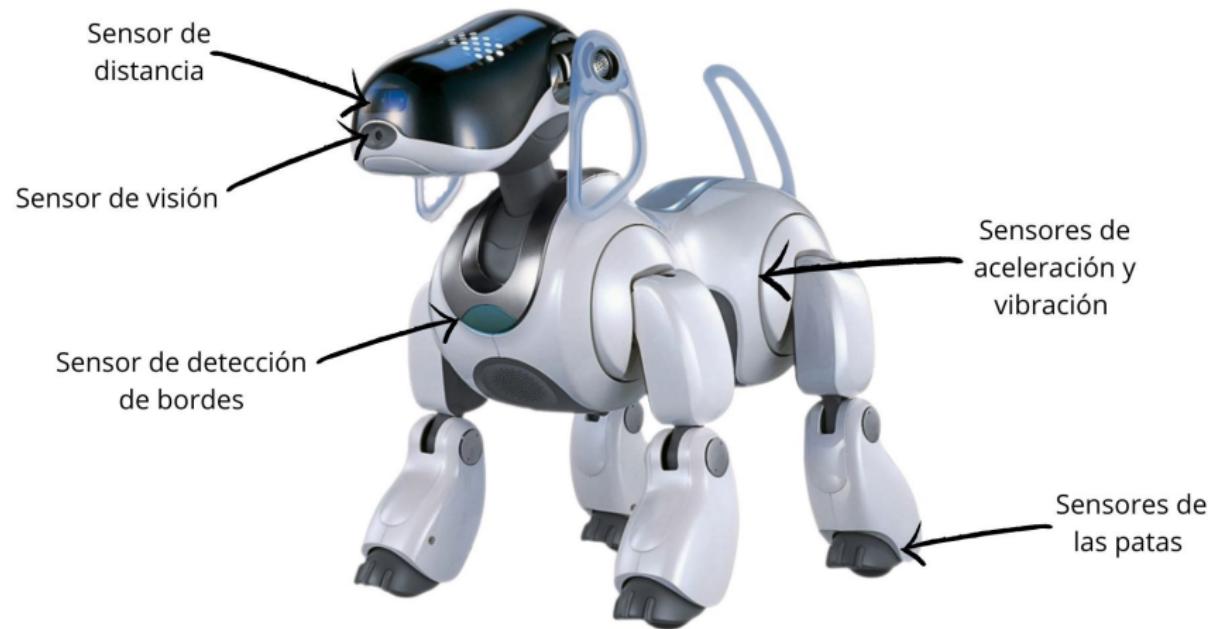
1. Existen distintos tipos de sensores que dotan de distintas cualidades. Entre ellos, uno de los más importantes es el sensor de visión, que requiere un proceso arduo para extraer información útil en tiempo real. Para el procesamiento de la información obtenida del sensor de visión, es necesario aplicar un algoritmo de IA.

# Inteligencia Artificial



1. La inteligencia artificial consiste en replicar los mecanismos del cerebro mediante algoritmos. Uno de los algoritmos más usados es el ML, preprogramados por un humano y que tratan de descubrir patrones que les hacen aprender. Los algoritmos más usados en visión artificial pertenecen a un subcampo del ML denominado DL, donde el autómata aprende por si mismo sin intervención humana, simulando el cerebro con unidades equivalentes a las neuronas.

# Sistemas multisensoriales



1. Además de utilizar sensores de visión, si a un robot se le incorporan diferentes tipos de sensores, este obtendrá mucha más información, por lo que podrá ser más preciso y simular a un humano mejor. Este trabajo se enmarca en el contexto de los sistemas multisensoriales, concretamente en los sistemas multisensoriales destinados al bienestar animal y al análisis de comportamiento.

# *Objetivos*

1. Pasemos ahora a comentar los objetivos que nos hemos marcado con este trabajo.

## Objetivos

- Recoger la lectura de los sensores en paralelo en un mismo fichero.
- Crear un servidor web para los sensores de las cámaras.
- Detectar los ratones mediante algoritmos de Deep Learning.

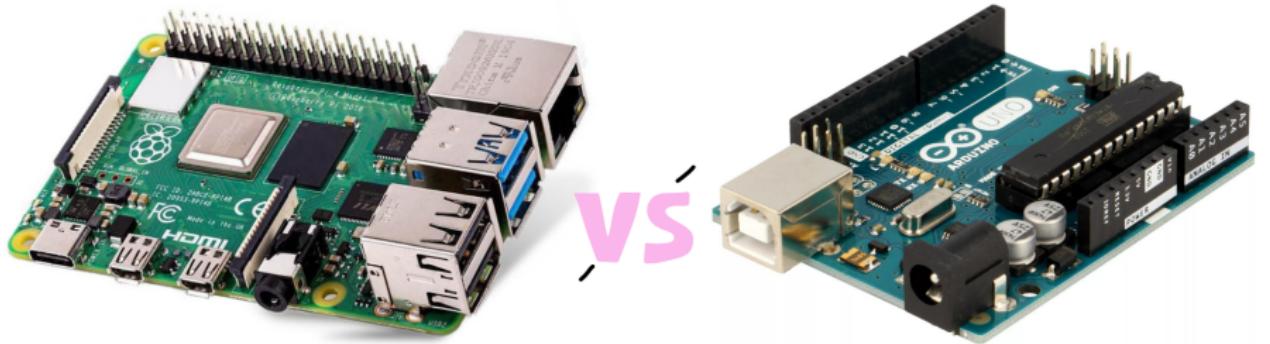
## Requisitos

- El sistema debe ser capaz de ejecutar en tiempo real sobre Raspberry.
- El lenguaje de programación debe ser Python.
- La interfaz de usuario se creará en Node-Red.

1. Una vez descritos los objetivos, veamos la infraestructura y plataformas utilizadas.

# *Diseño*

# Infraestructura hardware



Pines GPIO	Pines GPIO
Conexión de cámaras	No conexión de cámaras
Cuatro puertos USB	No tiene puertos USB
Dos puertos micro-USB	No tiene puertos micro-USB
Microordenador	Microprocesador
Sistema operativo completo	IDE

1. Existen distintos sistemas embebidos que se podrían haber utilizado para este trabajo. Los dos más conocidos son Raspberry y Arduino. Hay varias características favorables a Raspberry que han hecho que se seleccione esta placa frente a Arduino.



1. A través de los múltiples pines, se han conectado los sensores utilizados para el funcionamiento del sistema. El sensor DS18B20, arriba a la izquierda, permite medir la temperatura. Es resistente al agua, así que permite la medición en superficies húmedas o mojadas. Continuando a la derecha se encuentra el sensor BME680 permite medir cuatro parámetros: temperatura, humedad, presión y resistencia del aire. El sensor AMG8833 es un sensor térmico, que obtiene un mapa de 8x8 valores numéricos referentes a las temperaturas. El siguiente sensor, arriba a la derecha es un sensor que permite medir el nivel de agua. Empezando por la izquierda en la fila de abajo se encuentra la PiCam, una de las cámaras oficiales de raspberry que permite la visión del entorno. A su derecha está la Seek Thermal cam, que es una cámara térmica, originalmente creada para el uso en teléfonos móviles pero que permite su conexión a raspberry por el puerto USB con un adaptador. Finalmente, abajo a la derecha está en MQ-135, un sensor que permite detectar la concentración de ciertos gases en el aire, como el amoniaco.

## Infraestructura software



YOLOv5



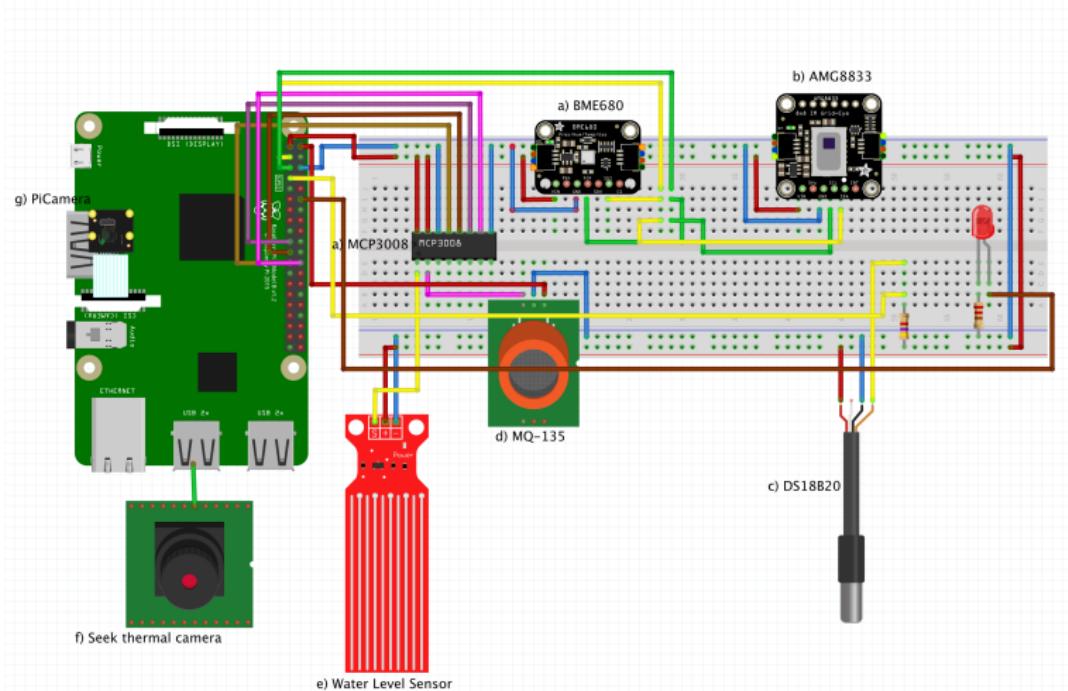
1. Como ya he presentado se utiliza para el desarrollo de este sistema la placa Raspberry. El sistema operativo que se ha usado en esta placa es Raspbian, basado en Debian, bajo la versión de 64 bits. El lenguaje seleccionado ha sido Python, debido a que está soportado de forma nativa en el SO, que es uno de los lenguajes más usados actualmente y que posee numerosas librerías útiles para el proyecto desarrollado. Una de ellas es OpenCV, que permite la manipulación de los vídeos. Se ha utilizado para realizar algunas modificaciones con la cámara, como añadir un timestamp así como para la detección de ratones. Para dicha detección, previamente ha sido necesario utilizar yolo para crear un modelo de detección de ratones. Por otro lado, para la creación de la IU se ha utilizado Node-Red, como previamente se estableció en los requisitos, y Flask para crear los servidores web.

## *Desarrollo del sistema*

1. A continuación se presenta el desarrollo que se ha llevado a cabo para la obtención del sistema planteado.

# Desarrollo hardware

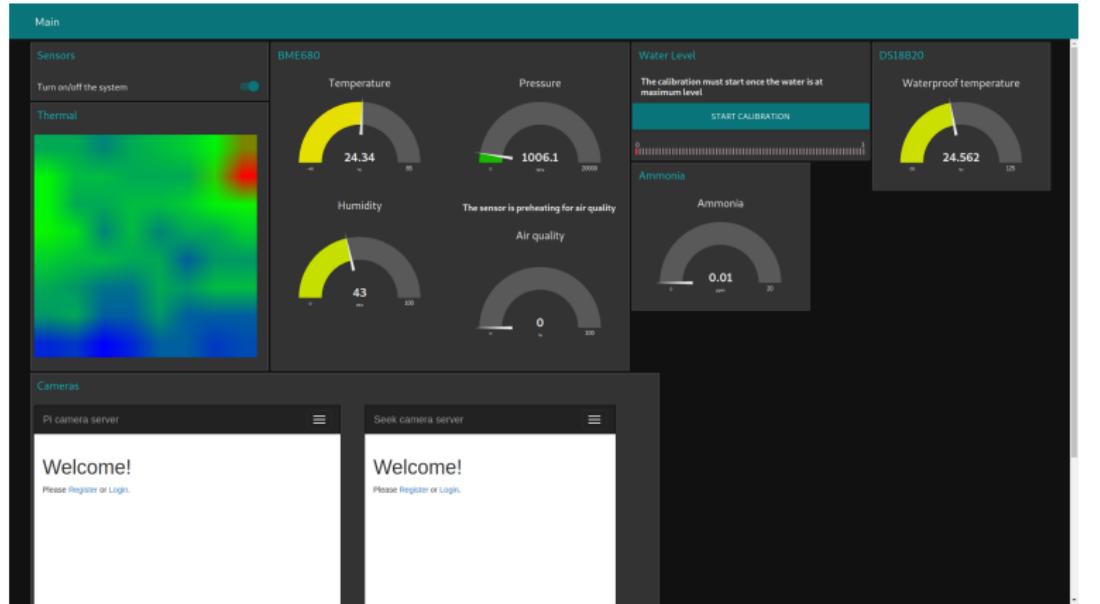
## Conexión de los sensores a la placa.



1. Se han conectado los sensores a la placa. Los sensores BME680 y AMG8833 requieren conexión con comunicación I2C, por lo que se han conectado a los pines pertinentes. Los sensores de nivel de agua y MQ-135 requieren de lecturas analógicas, por lo que se ha utilizado un CAD. Las cámaras se ha conectado en los puertos pertinentes y el sensor de temperatura resistente al agua junto con una resistencia necesaria para medidas estables.

# Desarrollo software

- ① Lectura sensorial con Python
- ② Creación de la interfaz de usuario
- ③ Integración de las cámaras en la IU



1. El primer paso, después de crear el diagrama de casos de uso así como el diagrama de clases del sistema, ha sido crear la lectura sensorial con Python. Con el uso de la librería Threads se ha conseguido leer de forma concurrente de todos los valores a la vez. Se han realizado algunos cambios en la lectura de los sensores, como en el caso del BME680 para que en lugar de mostrar la resistencia del aire se muestre la calidad de este en proporción a la humedad y resistencia del aire.

Una vez se ha obtenido este fichero, se ha procedido a la creación de la interfaz de usuario en Node-Red. Para ello, ha sido necesario combinar distintos tipos de nodos para la transformación de valores numéricos a la representación de ellos en widgets. En algunos casos, como en el del sensor AMG8833 (el sensor térmico) se han requerido más nodos para convertir el mapa en una imagen con una resolución mejor.

Con el uso de Flask, se han creado dos servidores: uno para cada cámara. De esta manera, se han podido incorporar a la interfaz de Flask a través del nodo template, que permite mostrar una url. Con Flask también se ha incorporado un botón que permite iniciar o parar la grabación de la PiCam. Además, se han añadido algunas mejoras con el uso de OpenCV a la visualización de las cámaras, como el timestamp.

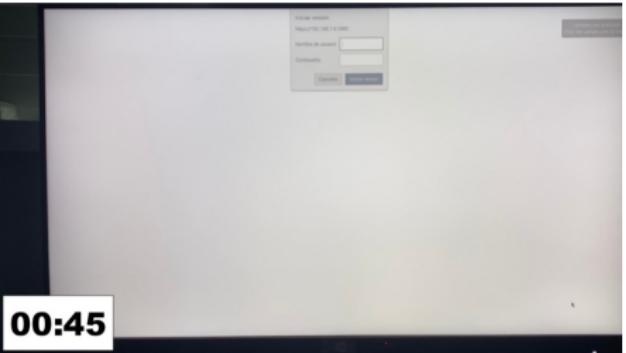
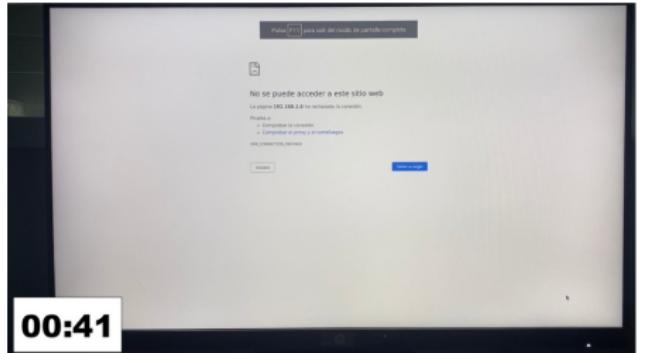
# Seguridad

- Servidores con login y 2FA
- Login en el flujo de Node-Red así como en la interfaz de usuario
- Cambio de HTTP a HTTPS

1. Debido a la existencia de servidores, así como el acceso al flujo de nodos de Node-Red y la interfaz de usuario, ha sido necesario dotar al sistema de más seguridad. En primer lugar, a los servidores se han añadido un login necesario basado en el factor de doble autenticación; con usuario, contraseña y un código de 30 segundos de vida.

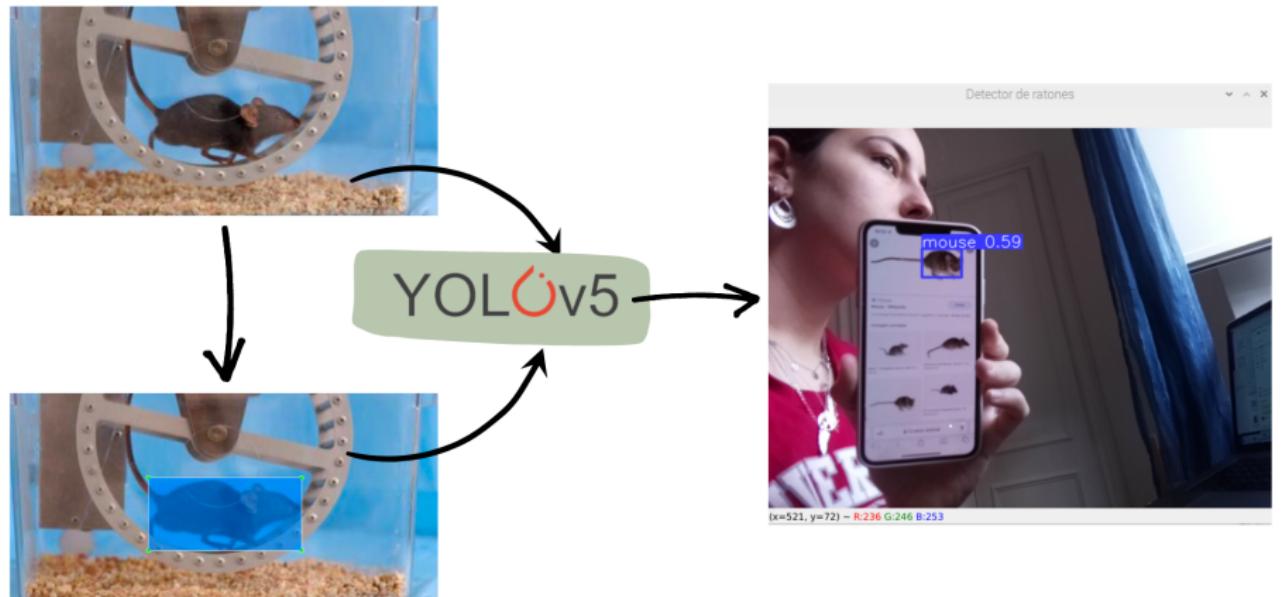
De la misma forma se ha añadido un login necesario al flujo de nodos así como a la interfaz de usuario. Es importante marcar que dichos login son diferentes, pues al primero solo podrá acceder el administrador, quien debe tener conocimiento para modificar cualquier nodo, y al segundo, cualquier usuario autorizado. Finalmente, tanto Node-Red como los servidores se encontraban bajo direcciones de HTTP. Esto se ha modificado a HTTPS bajo unos certificados autofirmados.

# Autoarranque



1. Por último, se ha adaptado el sistema de acuerdo a uso de un usuario ajeno a cualquier conocimiento de robótica. De esta manera, cuando se enciende la placa Raspberry el sistema se lanza automáticamente, de forma que la pantalla que recibe el usuario es la solicitud de datos para acceder a la interfaz.

# Detección de ratones mediante técnicas de Deep Learning



1. Cumpliendo el tercer objetivo comentado, se ha creado un modelo para la detección de ratones. Para ello, ha sido necesario crear un dataset, dado que no se ha encontrado ninguno cercano a los objetivos en Internet. El dataset creado consta de 354 imágenes: 275 para el proceso de entrenamiento y 79 para el de validación. Una vez se ha obtenido el dataset, se han etiquetado todas las imágenes en formato YOLO. Después se ha entrenado el modelo con el dataset y las etiquetas creadas. Para elegir el mejor modelo, se han probado distintos modelos de YOLO, desde el más rápido y menos preciso hasta el más lento y más preciso. Después de observar el comportamiento de los distintos modelos creados tanto en raspberry como en un ordenador normal, se ha decidido optar por el modelo YOLOv5s, uno de los más rápidos (pero también menos preciso) dado que el rendimiento de la CPU de raspberry es más limitado que un ordenador normal. Aun así, el flujo de las imágenes no es en tiempo real, y la detección no es del todo precisa.

# *Conclusiones*

1. Para acabar esta presentación, vamos a repasar lo hecho, unas breves conclusiones y las líneas futuras.

# Conclusiones

## Objetivos cumplidos

- Interfaz de usuario creado en Node-Red .
- Detección de ratones mediante algoritmo de Deep Learning.
- El sistema funciona en tiempo real en Raspberry: sistema *low-cost*.
- Interfaz para el usuario final.
- Accesible desde cualquier dispositivo de la misma red.

## Líneas futuras

- Adaptación de la IU a un servidor accesible desde cualquier lugar.
- Análisis del comportamiento de los animales con técnicas de DL.
- Creación de un Docker para la instalación por cualquier usuario.

1. La interfaz de usuario se ha creado en Node-Red a partir del fichero de lectura de sensores concurrentes y los servidores de las cámaras. También se ha conseguido la detección de ratones en la raspberry, a pesar de que no es en tiempo real. Además, se ha conseguido que sea un sistema low-cost al trabajar con Raspberry y sensores baratos. La interfaz es una interfaz intuitiva para cualquier perfil de usuario, y es accesible desde cualquier dispositivo conectado a la misma red que la raspberry.
2. Finalmente, algunas de las líneas futuras para finalizar este trabajo son:
  - Adaptar la IU a un servidor accesible desde cualquier lugar y no solo desde la misma red que raspberry.
  - Analizar el comportamiento de los ratones con el modelo de detección de ratones creado, para analizar cuánto tiempo pasan realizando las diferentes actividades así como en las distintas cubetas.
  - Y la creación de un Docker para permitir la instalación a cualquier usuario que no tiene por qué tener conocimiento sobre el tema.

# Sistema de monitorización para el control y estudio del bienestar de animales de laboratorio mediante una infraestructura de bajo coste

Isabel Cebollada Gracia

[i.cebollada.2018@alumnos.urjc.es](mailto:i.cebollada.2018@alumnos.urjc.es)



Trabajo fin de grado

27 de Julio de 2022

1. Y hasta aquí mi exposición.
2. Quedo a disposición del tribunal para cualquier duda que tenga.