

# Mini casa inteligente

Proyecto con Raspberry Pi 3

Kateryna Danilova



SHHH

# El proyecto consta de varias partes.

- La primera es **la idea** y cómo ensamblar todo lo necesario.
- La segunda es **el circuito de Proteus y Fritzing**.
- La tercera es **el circuito físico**.
- La cuarta es el **resultado**, con notas sobre **errores** y **aspectos interesantes**.

# Primera parte: La idea



□ La primera idea era algo similar a un monitor para bebés.

Detecta el ruido y, de ser así, enciende una bombilla y envía un mensaje a **Telegram**.

También se podría añadir un bot para que envíe una notificación a varios usuarios (por ejemplo, **un grupo de niños**) que diga:  
*«Mucho ruido, por favor, silencio».*

En este proyecto en concreto, creé una notificación específica para un solo usuario.

# Primera parte: La idea



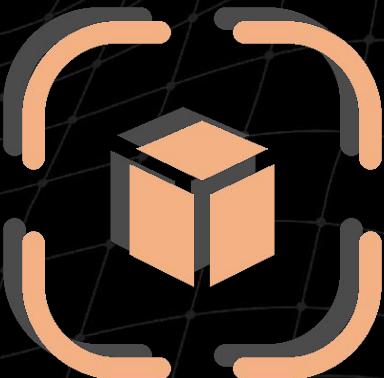
- La segunda idea es un despertador silencioso.

El usuario introduce **la hora deseada** y **una nota**. Cuando llega la hora, se envía una notificación a **Telegram** con la hora y la nota, junto con un LED parpadeante. **No hay sonido**, solo **la luz** y **la notificación de Telegram**.

También podríamos haber añadido varias luces y creado «un espectáculo de luces», pero decidí que, en este caso, un solo LED sería suficiente.

# Primera parte: La idea

- La tercera idea es la detección de objetos.

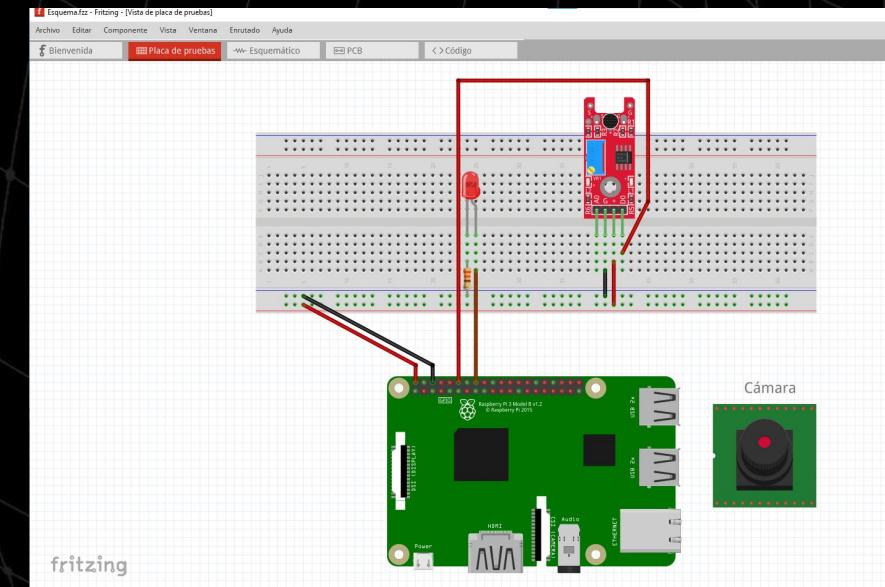
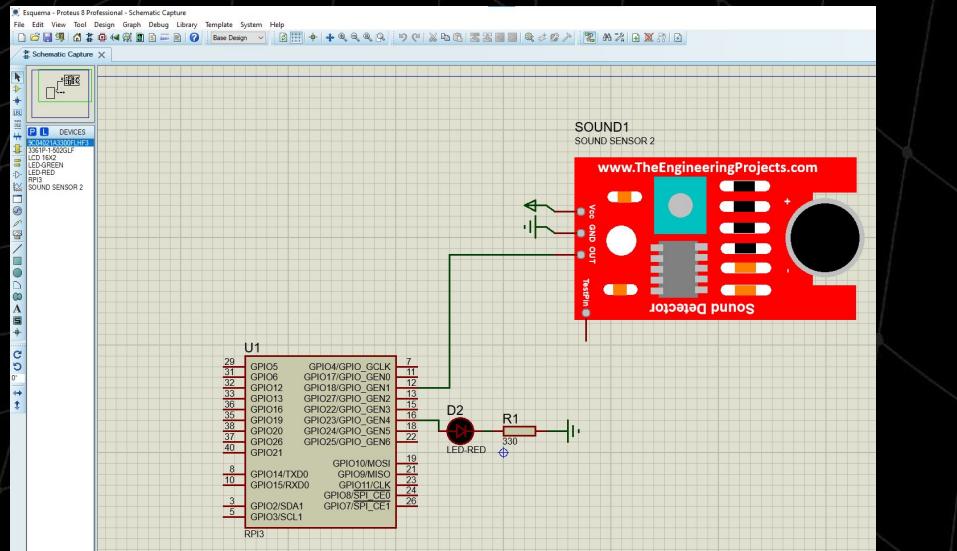


Esta es una idea útil; requiere pocos recursos, solo **una Raspberry Pi** y **una cámara**. La idea es instalar una cámara, ejecutar el código y, cuando la cámara detecte que un objeto ha desaparecido o reaparecido, se envía **una notificación a Telegram**.

También se pueden añadir registros que muestren, por ejemplo, **cuántas veces ha desaparecido y reaparecido** un objeto o **cuánto tiempo estuvo desaparecido**.

En mi caso, decidí simplemente añadir una notificación sencilla que indique la desaparición o reaparición del objeto en las notificaciones de Telegram.

# Segunda parte: Esquema en Proteus y Fritzing

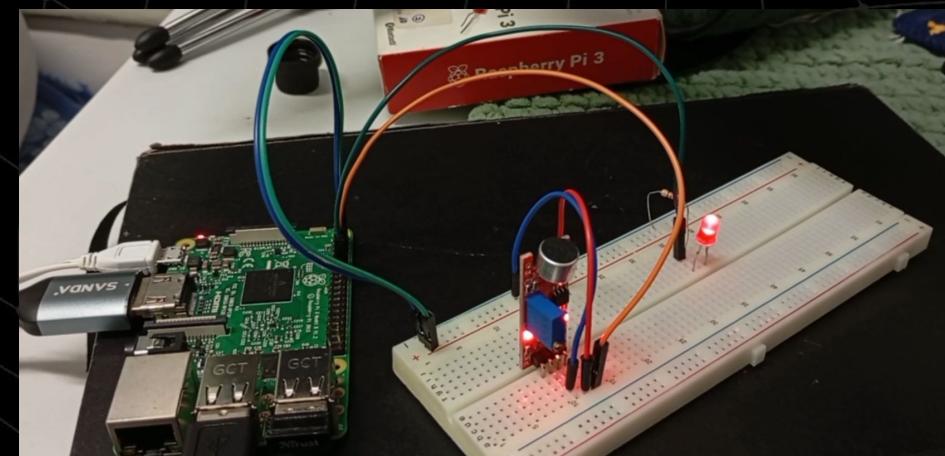


# Tercera parte: El circuito físico

- Necesitamos: un LED, un sensor de ruido, una Raspberry Pi 3 y algunos cables para la conexión.

Podríamos haber añadido una pantalla LCD si hubiéramos querido, pero decidí omitirla porque no me gustan los circuitos con LCD.

Disculpe, la pantalla LCD. 😞



# Cuarta parte: Resultados y notas

□ Hay cuatro archivos.

Uno es el principal, que es el que debemos ejecutar, y en él tendremos tres modos para elegir:



# Cuarta parte: Resultados y notas

□ Este es el enlace donde se encuentran los códigos:

[https://drive.google.com/drive/folders/1IjuybQH7uNSSro  
Sn4DQPdKNjvMWxwBwX?usp=sharing](https://drive.google.com/drive/folders/1IjuybQH7uNSSroSn4DQPdKNjvMWxwBwX?usp=sharing)

# Cuarta parte: Resultados y notas

- La idea original para la detección de objetos se basaba en los modelos "TensorFlow Lite" y "YOLO".

Sin embargo, surgió un problema: uno de los modelos no se entrenaba. El segundo modelo se entrenó correctamente y funcionó en un PC convencional, pero al intentar ejecutarlo en una Raspberry Pi, surgieron problemas con «PyTorch».

Profesor Joan y yo nos pusimos a trabajar en el submodelo "YOLOv4 Tiny". Funcionó, pero lamentablemente, no pude entrenarlo con mis propios objetos. Así que utilizamos la versión pública de los objetos.

# Cuarta parte: Resultados y notas

- En el código, la primera opción usa la biblioteca «OpenCV», mientras que la segunda usa «YOLOv4 Tiny».

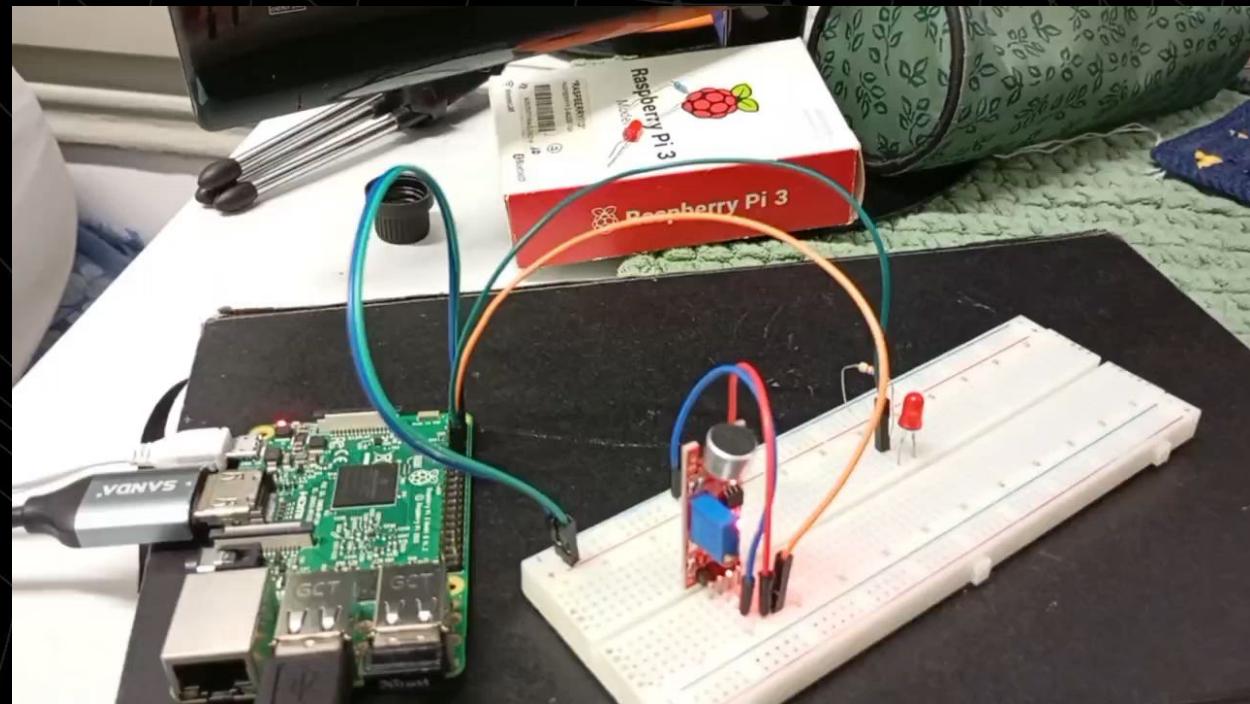
Ambas versiones funcionan y se pueden mejorar, pero en mi opinión, requeriría mucho tiempo y recursos.

¿Por qué? En resumen: tuve una buena idea, pero debido a mi **mala conexión** a internet en casa y pequeños problemas como que el programa de entrenamiento «TensorFlow Lite» no funcionara durante **más de seis meses**, surgieron muchos obstáculos.

En el futuro, planeo recrear este proyecto, pero con mejor funcionalidad, y quizás logre comprender mejor el entrenamiento de la IA con diferentes versiones del modelo. Incluso he llegado a pensar que, eventualmente, sería interesante abordar el tema de programar la IA desde cero.

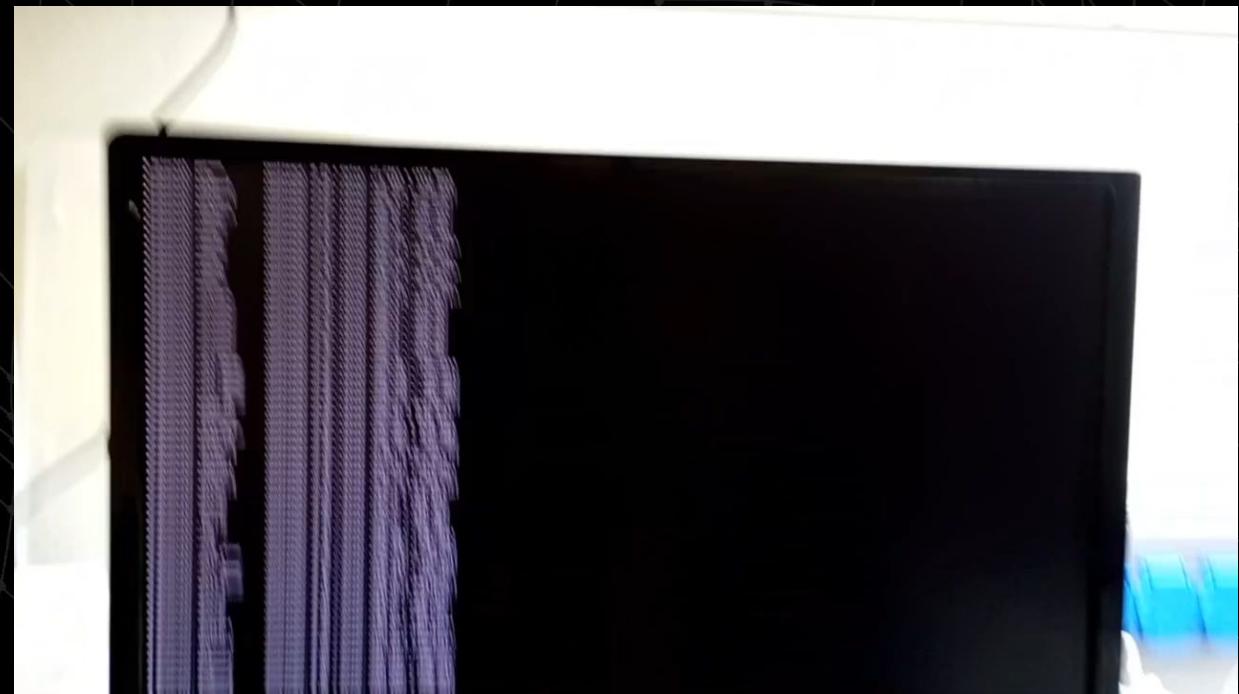
# Cuarta parte: Resultados y notas

<https://youtu.be/MRrDOtgfrdQ>



# Cuarta parte: Resultados y notas

<https://youtu.be/6XETQdRX2Go>



# Cuarta parte: Resultados y notas

- En general, el proyecto fue **interesante**, a pesar de los numerosos errores (con librerías, pines, código, etc.) y los largos tiempos de carga.  
Raspberry es una herramienta muy interesante con la que se pueden crear muchas cosas.

¡Gracias por su atención!

