

Construcción de sistema domótico de bajo coste para la alimentación de animales domésticos

David Moreno Curós

Resumen– En este proyecto se expone como se ha llevado a cabo la construcción de un dispensador automático de comida para animales domésticos a partir de una placa de ordenador “single-board” y otros componentes electrónicos como sensores de peso y electroimanes. El propósito de este proyecto no es solo llevar a cabo el sistema mencionado sino hacerlo con el menor coste posible. Es por eso que, tanto la placa en sí (una Raspberry Pi Zero WH) como el resto de dispositivos han sido especialmente elegidos por su bajo precio y por su capacidad de integración al sistema. Las utilidades que se le pueden dar al dispensador no solo se basan en la propia alimentación de los animales sino también a la monitorización de esa alimentación y el control de dietas de estos: con el dispensador automatizado se puede controlar, desde una aplicación programada para Android, cuánto han comido los animales en los últimos días, u obtener una media mensual o semanal, que servirá para posteriormente extraer conclusiones como la de si un cambio de comida (p. ej.comprar un pienso A en vez de un pienso B) ha afectado realmente a la frecuencia con la que los animales comen o a la velocidad que lo hacen.

Palabras clave– Dispensador, automático, animales domésticos, alimentación, monitorización de dieta, microcontroladores, Raspberry, bajo coste

Abstract– This project shows how the construction of an automatic pet food dispenser has been carried out from a single-board computer and other electronic components such as weight sensors and electromagnets. The purpose of this project is not only to carry out the aforementioned system but to do it at the lowest possible cost. That is why both the board itself (a Raspberry Pi Zero WH) and the rest of the devices have been specially chosen for their low price and for their ability to integrate into the system. The utilities that can be given to the dispenser are not only based on the feeding of the animals but also on the monitoring of this feeding and the control of their diets: with the automated dispenser it can be controlled, from an application programmed in Android , how much the animals have eaten in the last days, or obtain a monthly or weekly average, which will serve to later draw conclusions such as whether a change of food (e.g. buying feed A instead of feed B) has really affected how often animals eat or how fast they do it.

Keywords– Dispenser, Automatic, Pets, Feeding, Diet Monitoring, Microcontrollers, Raspberry, Low Cost

1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

A DÍA de hoy la automatización de tareas se ha convertido en uno de los fines principales de la informática. El hecho de que un sistema informático pueda realizar algunos trabajos sin la necesidad de la supervisión humana ha dejado de ser algo anhelado y

futurista y ha pasado a ser la realidad.

No es de extrañar que cosas como pagar un peaje de autopista, o salir de un parquin de pago sin tener que hablar con alguien, se hayan vuelto cotidianas, y lo cierto es que cada vez aparecen más trabajos cuya interacción humana, que hasta el momento era considerada intrínseca, ha sido sustituida por un microcontrolador o microprocesador y algunos cuantos cables. Desde servir comida a través de máquinas expendedoras hasta salvar vidas con dispositivos

- E-mail de contacto: david.morenocu@e-campus.uab.cat
- Mencion realizada: Ingeniería de Computadores
- Trabajo tutorizado por: Diego Mostaccio (Departamento de Arquitectura de Computadores y Sistemas Operativos)
- Curso 2020/21

de socorro para personas mayores, entre otras muchas cosas, las placas computacionales pueden ser realmente útiles, y tal y como avanza la informatización global parece ser que estamos en uno de los primeros pasos de ese camino en la aplicación de microcontroladores y microprocesadores para multitud de fines que de seguro llegan para quedarse. La propuesta de uso de placas de bajo coste para la construcción de un proyecto domótico consiste en el uso de placas de ordenadores “single-board” como las Raspberry Pi [1] con un coste bajo y un tamaño pequeño para la realización de un proyecto domótico.

La finalidad de este trabajo es, pues, aplicar los conocimientos obtenidos a través de la formación en la mención de ingeniería de computadores, para acabar construyendo un sistema que logre automatizar algunos trabajos en el ámbito del hogar, documentando todo su desarrollo y valorando finalmente la viabilidad “adoptiva” de éstos sistemas en nuestras vidas.

El sistema elegido para este proyecto es un dispensador de comida para animales que se controla a través de una aplicación móvil y que, a diferencia de otros dispensadores automatizados del mercado, cuenta con un conjunto de sensores y otros dispositivos externos para, además de servir la comida, monitorizar la alimentación de los animales y mostrar toda esa información a través de la propia aplicación.

2 ESTADO DEL ARTE

Actualmente en el mercado existen dispensadores automatizados para la alimentación de animales domésticos. Muchos de ellos disponen de un sistema para controlar las raciones diarias de comida, y de hecho algunos de los más sofisticados incluyen también el control de éste a través de una aplicación para dispositivos móviles[2]. Si bien muchos de ellos tienen un acabado bonito y una estética discreta (algo muy buscado teniendo en cuenta que el dispensador pasa a ser un electrodoméstico más del hogar), el coste que tienen es muy elevado, y de hecho no es un dispositivo que esté al alcance de todos. Teniendo en cuenta que uno de cada cuatro hogares españoles tienen un perro como mascota y casi un 20 por ciento tienen un gato [3], no debería ser algo tan caro, pues las utilidades que se le pueden dar a estos sistemas no giran solo en torno a la comodidad del dueño, sino también en torno a la seguridad de la mascota, algo que debería ser siempre primordial.

Existen también algunos sistemas de bajo coste, aunque con funcionalidades más limitadas o fabricados de materiales menos duraderos. Los avances en la tecnología de hoy en día, han permitido hacer crecer los nuevos mercados de dispositivos domóticos, dando paso al surgimiento de opciones más económicas de esos mismos dispositivos. Eso sí, ¿hasta qué punto lo que no es caro puede llegar a ser bueno? ¿Es suficiente solo dar la comida de forma programada a los animales? ¿Qué nivel de confianza nos da el dispositivo como para poder dejar las mascotas solas durante unos días? Todas las opciones vistas en el mercado con un coste bajo tenían funciones muy simples y sin mecanismos de control [4], así que los objetivos, que a continuación detallaremos, se basaron en torno a cambiar esto.

3 OBJETIVOS

Para el planteamiento del proyecto, se establecieron los siguientes objetivos:

- Construir un sistema dispensador de comida automático, de bajo coste, y fiable, priorizando siempre esto último para evitar que dejar la casa sola unos días pueda suponer un riesgo para las mascotas.
- Desarrollar una aplicación móvil para poder controlar y monitorizar el funcionamiento del sistema.

4 COMPONENTES USADOS Y TABLA DE PRECIOS

- Placa Raspberry Pi Zero WH: el “cerebro” del sistema. Se eligió este modelo por ser la placa de la familia de las Raspberry con el segundo menor coste (después de la Raspberry Pi Pico), concretamente 10 euros aproximadamente, dependiendo del lugar de compra. Con una memoria RAM de 512 Mb y un puerto para insertar una tarjeta SD de hasta 256Gb [5], el modelo Zero WH se convierte en la placa ideal para un proyecto como éste en el que no necesitamos gran potencia de cómputo y priorizamos el bajo coste.
- Protoboard de 830 puntos: donde se realizó todo el conexionado.
- Cables “jumper”¹ hembra-hembra, hembra-macho, y macho-macho, para realizar las conexiones.
- Teclado Matriz Membrana 4x4: para dar seguridad al sistema, la apertura de la tapa del dispensador para el relleno de comida se realiza si y solo si se introduce una contraseña concreta a través del teclado [6].
- Electroimán de 5V con interfaz de 3 pines: Para precisamente permitir la apertura o no de la tapa, el electroimán se activará cuando la contraseña introducida sea la correcta [7].
- Pantalla LCD 16x2 retroiluminada: para mostrar información básica del sistema (contraseña introducida, mensaje de contraseña correcta o incorrecta, etc) [8].
- 2 potenciómetros de 10.000 ohmios: para regular la intensidad y el contraste del LCD.
- 1 servomotor 9g SG90: para accionar la compuerta del dispensador y servir la comida [9].
- Cámara focal ajustable con 60 grados de ángulo de visión [10].
- Celda de carga de 5kg: para medir el peso del plato de comida de forma analógica [11].

¹Los cables “jumper”, también conocidos como cables puente, son un tipo de cables con puntas rectangulares de plástico que tienen en su interior dos o más puntas metálicas con un espacio entre ellas de 0.2 mm hechos de fósforo-bronce, de una aleación de cobre-níquel, de estaño o de latón y con un color dorado o cromado, de tal manera que cuando se introducen y se empujan hacia los pines de un circuito, éstos cierran el circuito cubriendo completamente los pines, resultando en una conexión temporal.

TABLA 1: TABLA DE PRECIOS

Componente	Precio por unidad	Total
Raspberry Pi Zero WH	14,72	1x14,72
Protoboard de 830 puntos	2,40	1x2,40
Pack de cables jumper (20cm)	2,35	2x2,35
Pack de cables jumper (30cm)	3,19	1x3,19
Teclado Matriz Membrana	1,39	1x1,39
Electroimán de 5V	9,50	1x9,50
Pantalla LCD 16x2	2,59	1x2,59
Potenciómetro de 10k Ω	0,10	2x0,10
Servomotor 9g SG90	1,99	1x1,99
Cámara focal ajustable	10,15	1x10,15
Celda de carga de 5kg	4,76	1x4,76
Módulo conversor analógico-digital	2,54	1x2,54
Módulos infrarrojos para la cámara	1,90	1x1,90
Soporte acrílico para la cámara	3,12	1x3,12
Cable plano flexible	2,02	1x2,02
Policarbonato celular	21,5	1x21,50 ²
Total		84,67€

- Módulo conversor analógico-digital HX711: para traducir el valor leído por la celda de carga a un valor digital [12].
- Módulos infrarrojos para la cámara.
- Soporte acrílico para la cámara.
- Cable plano flexible (FFC) para el conexionado de la cámara a la placa.
- Policarbonato celular para la construcción de la carcasa [13].

Véase en la Figura 1 el diagrama de conexión:

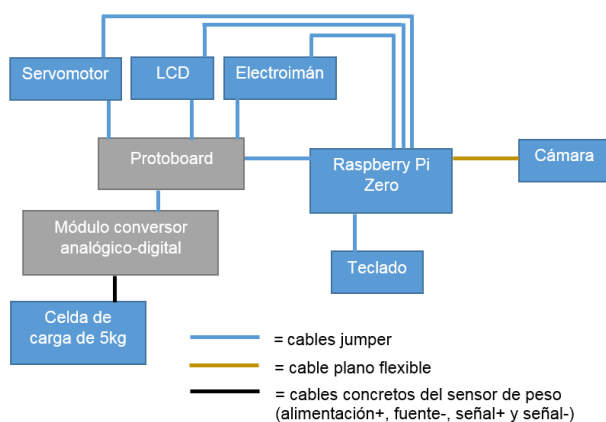


Fig. 1: Diagrama de conexión del sistema dispensador

De este diagrama podemos extraer 4 tipos distintos de conexiones (ver Tabla 1).

²Se compró más de lo necesario y sobró una parte de policarbonato, así que realmente el precio es orientativo.

TABLA 2: TABLA DE CONEXIONES

Conexión	Origen-destino	Realizada en
Conexión del cable de señal del dispositivo a uno de los pines GPIO de la placa.	Dispositivo-Placa	Servomotor, electroimán, LCD y Módulo conversor analógico-digital.
Conexión de los cables de alimentación y de tierra a las líneas de voltaje de 5V y 0V respectivamente	Dispositivo-Protoboard	Servomotor, electroimán, LCD y Módulo conversor analógico-digital.
Conexión desde el puerto de cable flexible de la cámara al puerto de cable flexible de la placa.	Dispositivo-Placa	Cámara.
Conexión de uno de los puntos de la protoboard a otro.	Protoboard-Protoboard	LCD

5 CONSTRUCCIÓN DE LA CARCASA

Para la construcción de la carcasa, se empezó realizando un modelo en 3d usando una herramienta de prototipado llamada Sketchup [14]. Sketchup es un programa con versión gratuita que ofrece gran cantidad de opciones para crear diseños en tres dimensiones de una forma fácil e intuitiva. Cuenta además con un "workshop" de modelos en el que los usuarios pueden compartir sus creaciones para que luego éstas puedan ser usadas por la comunidad. Véase en la Figura 2 el prototipo creado usando esta herramienta.

Después de modelar el sistema en tres dimensiones, se decidió construir de forma casera y no contratando algún servicio de impresión en 3d, que si bien es cierto hubiera sido la alternativa más fácil, no hubiera sido la más barata. Al fin y al cabo el proyecto está orientado a la construcción de un sistema domótico de bajo coste, y este bajo coste se consideró que no solo tenía que estar enfocado a las placas y los componentes electrónicos usados sino al proyecto en general. Así pues, se pensó en construir la carcasa usando policarbonato celular, comprado en forma de plancha y cortado y adaptado para que encajara con el modelo de prototipo que se tenía planeado.

Se decidió usar este material debido a sus múltiples propiedades ventajosas respecto a otros materiales, como por ejemplo la resistencia al impacto. El dispensador estará situado en el suelo y los animales se acercarán a él para comer, así que se tiene que asegurar un mínimo de robustez para evitar que las mascotas puedan romper la carcasa de un golpe al acercarse o por otros posibles motivos. La resistencia al impacto del policarbonato se sustenta precisamente en el hecho de que es un material relativamente blando y esa característica hace que pueda absorber sin problema

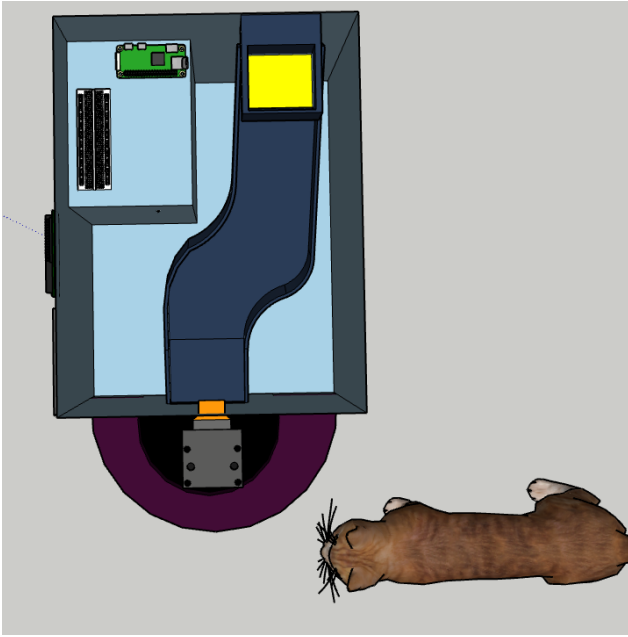


Fig. 2: Modelo en 3d de la carcasa

las fuerzas o golpes recibidos. Concretamente es un material que soporta los impactos 80 veces mejor que el cristal y 15 veces mejor que el metacrilato. Las cámaras alveolares que presentan las planchas son lo que dan esa resistencia, y aunque sean resistentes consiguen que el peso sea mínimo y mucho menor que el que presentan las placas de policarbonato compacto, por ejemplo. Para hacer a la idea, un cristal del mismo grosor que una pieza de policarbonato celular pesa 12 veces más.

Por lo tanto tenemos una robustez y un bajo peso que nos acaba dando una gran versatilidad a la hora de situar el dispensador por la casa, pero además es un material de bajo coste y con protección UV y resistencia al agua, así que incluso podríamos situar el dispensador fuera de casa sin preocuparnos de que el sol o la lluvia dañaran la carcasa. A continuación, en las Figuras 3 y 4 se muestra cómo es el material:



Fig. 3: Material utilizado

Así pues, se construyó una carcasa como la expuesta en las Figuras 5 y 6:

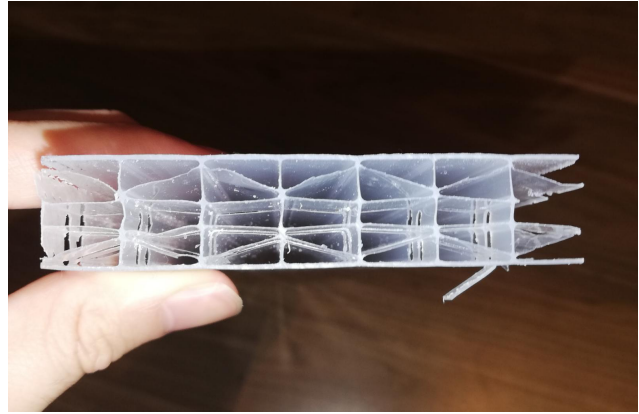


Fig. 4: Cámaras alveolares del material

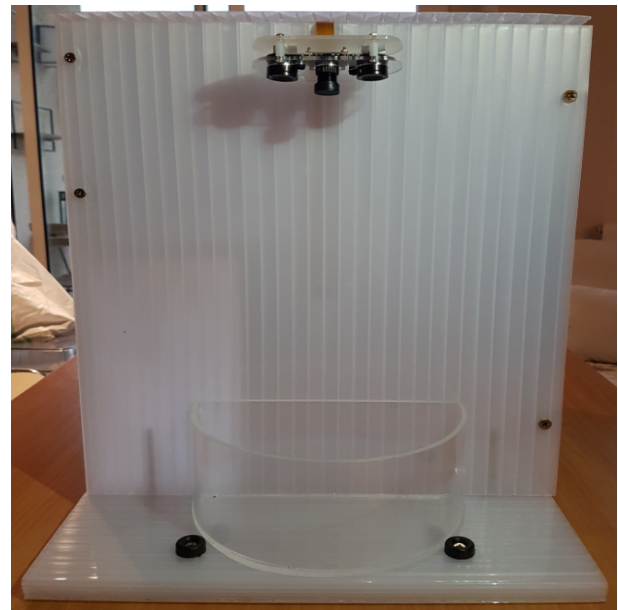


Fig. 5: Carcasa

Hay que mencionar que no todos los componentes de la carcasa fueron realizados con ese material. Por ejemplo, el recipiente donde comen los animales, la caja donde se almacena la comida, o la rampa que une los dos elementos anteriores, fueron realizados de metacrilato, y encargados a una empresa externa para poder dar la forma que se tenía pensada sin la necesidad de adquirir maquinaria compleja (véase Figura 7):

Podemos observar en la figura que tenemos un elemento más de los 3 que hemos comentado (una plancha simple situada en la parte derecha de la foto), y eso es debido a que inicialmente se tenía pensado que la caja donde se guarda la comida se abriera y cerrara usando una especie de raíl, a través del cual situando esa plancha por debajo de la caja y empujandola hacia delante la plancha actuaría como una tapa para evitar que la comida cayera, y cuando se empujara hacia atrás (sentido contrario del raíl) la plancha no taparía toda la parte inferior y por consiguiente caería la comida. No obstante, al final se pensó en una manera más simple de realizar esta tarea, y por eso se acabó descartando esa pequeña lámina. Con el nuevo método, simplemente se unieron a la caja que guarda la comida dos bisagras que sujetan una lámina como la previamente descartada, para que, de esta forma, colocando un motor debajo que hiciera



Fig. 6: Carcasa



Fig. 7: Partes hechas de metacrilato

fuerza, ya se pudiera aguantar el peso de la comida (que es de 400 gramos) y mantener la caja cerrada, y cuando se quisiera dar comida, con solo dejar de hacer tanta fuerza, la tapa bajaría ligeramente y caería comida hacia la rampa.

Se pensó en hacer el recipiente donde cae la comida de metacrilato y separado del sistema para que éste pueda ser limpiado con facilidad: al fin y al cabo los animales van a comer de ahí y es normal que ese recipiente se ensucie por muchos motivos. El hecho de no tenerlo unido a la carcasa nos permite sacarlo y limpiarlo de una forma cómoda y sin la necesidad de mover todo el dispensador. Se colocaron también unos topes de puerta en los extremos para impedir que los animales muevan el recipiente de sitio, cosa que podría hacer que al caer la comida ésta cayera fuera de sitio. Aún así, en el prototipo final que mostraremos más adelante, veremos que estos topes de puerta fueron sustituidos por unos tacos de plástico de 10 cm, puesto que no se pensó que debajo del recipiente de comida estaría el sensor de peso y eso implicaría que la altura a la que está el recipiente fuera mayor a la que se pensó inicialmente.

6 FUNCIONAMIENTO DEL DISPENSADOR

El dispensador lee constantemente el valor marcado por el sensor de peso. Este sensor de peso es básicamente la unión de una celda de carga con un módulo conversor analógico-digital, pues la placa usada en este proyecto no dispone de un conversor preintegrado. El peso es obtenido por torsión: cuando uno de los polos de la celda de carga recibe presión, al doblarse ligeramente es como se mide la masa que se puso encima, y el hecho de que se mida por torsión implica que el sensor ha tenido que ser construido usando un soporte acrílico en el cual se coloca la celda de carga en el interior y se sujeta cada polo de la celda con una lámina acrílica de forma circular y unos tornillos para fijarlo. A continuación se muestra el resultado:

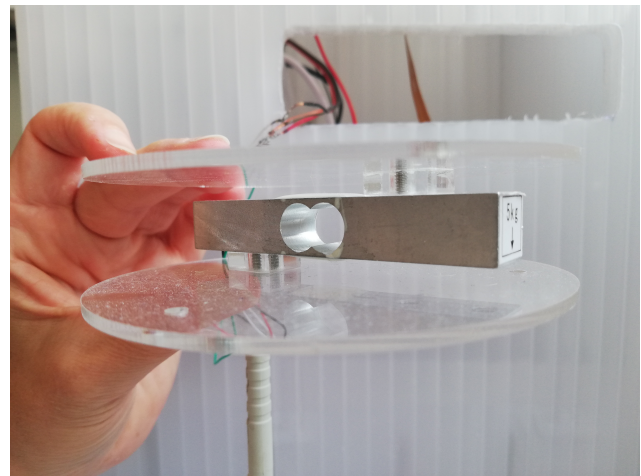


Fig. 8: Sensor de peso

El conexionado entre el módulo conversor se puede hacer directamente conectando el módulo ADC en la protoboard y luego hacer salir unos cables hembra-macho hacia la placa, pero la conexión entre el sensor de peso y el módulo conversor no se pudo hacer directamente, pues los extremos de los cables del sensor de peso eran simplemente cables pelados, así que se tuvo que recurrir a la soldadura.

6.1. Modo automático

Una vez se lee el valor del sensor de peso, se comprueba si éste está por debajo de 10 gramos. En ese caso, el programa escrito en Python acciona el servomotor y éste permite que la tapa de la caja donde se almacena la comida se abra ligeramente, permitiendo el paso de comida hacia la rampa y consecutivamente cayendo al recipiente donde los animales comen. Este proceso de abertura se realiza durante 4 segundos o hasta que el sensor de peso detecta más de 100 gramos, y una vez una de estas dos situaciones ocurren el servo vuelve a ejercer fuerza para volver a mover la tapa a su posición de cierre, impidiendo que se siga distribuyendo comida.

En el modo automático, cada vez que el sensor detecta un peso inferior a 10 gramos, además de dispensarse la comida, se incrementa un contador diario de servicios de comida. Ésto nos permite monitorizar la alimentación de las mascotas, puesto que podemos analizar cuánto han comido durante la semana o durante el mes haciendo una simple media

entre los valores diarios obtenidos en 7 días o 30 días, respectivamente, o simplemente ver lo que han comido en las últimas 24 horas. Con esto además podemos extraer algunas conclusiones interesantes, como por ejemplo si hay una comida que les gusta más o menos. Se puede cambiar la alimentación de los animales durante un periodo de tiempo y luego comparar los datos de alimentación obtenidos antes del cambio con los de ahora, así siempre podremos analizar lo que están comiendo las mascotas.

6.2. Modo manual

También, y de forma paralela, se puede establecer un modo manual para dar la comida, que consiste en usar la aplicación para Android desarrollada para accionar el motor cuando queramos. En el modo manual, al pulsar en el botón "Dar comida" de la aplicación, el servo se acciona y se repite el proceso previamente descrito. Además, y también desde la aplicación, si pulsamos en el botón de "Tomar foto", se realizará una fotografía mediante la cámara de la Raspberry y ésta será mostrada en la aplicación. Ésto último es lo que nos aporta fiabilidad: al fin y al cabo el sistema dispensador está pensado para usarse cuando los dueños de los animales no se encuentren en casa. El hecho de poder tomar una foto después de haber dado la comida para ver si realmente se ha dado comida es lo que aporta una seguridad al dueño de que los animales han comido.

7 DESARROLLO DE LA APLICACIÓN

Para la realización de la aplicación se usó Android Studio y se programó en Java. Android Studio es un software gratuito de desarrollo de aplicaciones para Android y tiene una gran versatilidad en cuanto a diseño de layout de aplicaciones e integración de ese layout en el código. Aunque la interfaz no sea uno de los fuertes, se realizó una aplicación funcional con la que comunicarse con la base de datos y a su vez con la Raspberry. Está compuesta por los dos botones previamente comentados de dar comida y tomar foto, y de una franja en la que se muestra la última foto capturada, que si el programa se encuentra en modo automático será la que se tomó hace 60 segundos o menos (en modo automático se toma una foto cada minuto y se muestra en la interfaz de la aplicación). Ver Figura 9.

8 INTEGRACIÓN DE LA APLICACIÓN EN EL SISTEMA Y BASE DE DATOS UTILIZADA

Para la base de datos se ha utilizado Firebase [15], una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014 que cuenta con un sistema de creación y gestión de bases de datos en tiempo real basadas en NoSQL y alojadas en la nube. El servicio es gratuito y dispone de muchas funcionalidades para para la comunicación con aplicaciones Android.

Para llevar a cabo la integración aplicación-DB-raspberry se realizó lo siguiente: en primer lugar, se instaló firebase-admin en la raspberry. Luego, se creó la base de datos en



Fig. 9: Aplicación Android

Firestore y se descargaron todas las credenciales autogeneradas por el servicio para poder acceder a la base de datos desde la aplicación y desde la placa. Finalmente se crearon las referencias a los datos almacenados en la base de datos para luego desde la aplicación y desde la raspberry poder leerlos y modificarlos. En las Figuras 10 y 11 se adjunta un diagrama descriptivo de cómo se realizó esta integración y los datos guardados en la base de datos, respectivamente:

El funcionamiento se basa en el uso de referencias a los datos almacenados: tanto en la aplicación como en el programa que ejecuta la raspberry se han creado unas variables vinculadas a datos de la base de datos, y a través de las lecturas y las escrituras a esos datos es como realmente podemos controlar todos los procesos del dispensador. Se adjunta a continuación un ejemplo de cómo se construyen esas referencias a los datos alojados en la base de datos de Firebase tanto desde la raspberry (Figura 12) como desde la aplicación (Figura 13).

Para el primer caso, lo que hacemos es, por orden de líneas:

1. Fijar las credenciales descargadas
2. Acceder a la base de datos

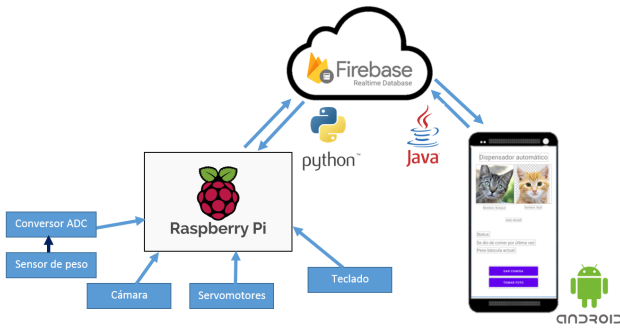


Fig. 10: Integración aplicación-DB-Raspberry

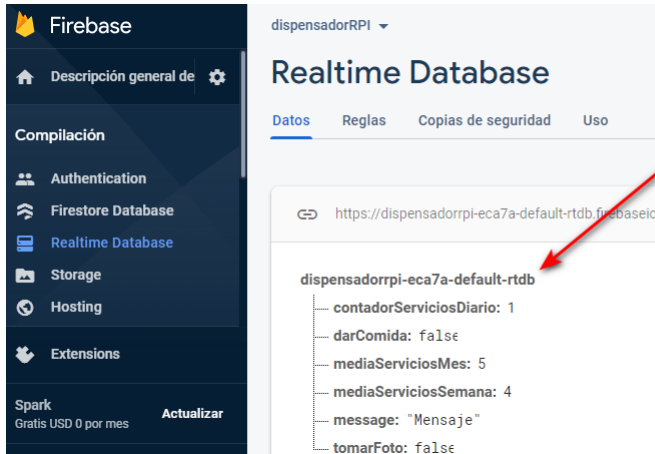


Fig. 11: Base de datos de Firebase

3. Crear la referencia a uno de los campos de la base de datos
4. Guardar el valor de esa referencia en una variable

En este caso el proceso es exactamente igual, solo que en este caso se utiliza Java para hacerlo y la escritura en el valor de la variable se hace dentro de un "setOnClickListener", pues lo que queremos es que **al pulsar en el botón de "Dar Comida"** nos ponga la variable de darComida a true durante unos segundos y la raspberry, que estará leyendo esa misma variable, vea ese valor y empiece a dispensar la comida.

9 FOTOS DEL DISPENSADOR ACABADO

A continuación se presenta el estado final del dispensador automatizado (véanse Figuras 14 15 16 17 18 19).

10 CONCLUSIONES

La automatización de tareas, y no solo en el ámbito doméstico, es un concepto que llega para quedarse. El hecho de poder realizar proyectos que realizan tareas rutinarias de forma automática para mejorar nuestras vidas y a un coste bajo era algo impensable hace unos años, pero gracias a los avances de la tecnología y a la aparición de competidores en los sectores tecnológicos ha hecho que éste deje de ser algo anhelado y monopolizado y pase a ser algo que todo el mundo con un mínimo nivel de informática pueda realizar. Este proyecto se inició con un objetivo real: poder estar

```

cred = credentials.Certificate('/home/pi/Desktop/dispensadorRPI/cred.json')
firebase_admin.initialize_app(cred, {
    'databaseURL': 'https://dispensadorrpi-eca7a-default-rtdb.firebaseio.com/'
})
ref = db.reference('darComida')
darComida = ref.get()
    
```

Fig. 12: Acceso a variables de la base de datos desde la Raspberry

```

FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("message");
DatabaseReference refDarComida = database.getReference("path" "darComida");
DatabaseReference refTomarFoto = database.getReference("path" "tomarFoto");

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    textoTitulo = findViewById(R.id.textoTitulo);
    botonDarComida = findViewById(R.id.botonDarComida);
    botonTomarFoto = findViewById(R.id.botonTomarFoto);
    textoStatus = findViewById(R.id.textoStatus);
    textoUltimaVezComida = findViewById(R.id.textoUltimaVezComida);
    textoPesoBascula = findViewById(R.id.textoPesoBascula);

    botonDarComida.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            textoUltimaVezComida.setText("Se dio de comer por última vez:" + LocalDateTime.now());
            refDarComida.setValue(true);
            refTomarFoto.setValue(true);
            textoTitulo.setText("Texto cambiado despues de pulsar boton");
            try {
                TimeUnit.SECONDS.sleep( timeout: 3);
            } catch (InterruptedException e) {
                System.out.println("got interrupted!");
            }
            refDarComida.setValue(false);
            refTomarFoto.setValue(false);
        }
    });
}
    
```

Fig. 13: Acceso a variables de la base de datos desde la aplicación

unos días fuera de casa sin tener que llamar a alguien para poner comida a los animales, y el hecho de haberlo materializado en un trabajo de final de grado no ha sido solo útil solo para mi, como estudiante, sino para todos los miembros convivientes de mi hogar.

El coste del proyecto de 80 euros, teniendo en cuenta que lo que se usó de material no fue siempre la totalidad (por ejemplo no se usaron todos los cables jumper ni toda la plancha de policarbonato celular), es un precio muy bajo, especialmente si lo comparamos con las alternativas existentes en el mercado de prestaciones similares a las del proyecto [2]. Así pues, se considera que la realización del proyecto ha sido un éxito, aunque por supuesto haya cosas por mejorar como la estética del dispensador o otros ámbitos de mejora en los aspectos de la interfaz de la aplicación, pero como popularmente se dice... el ingeniero construye y el diseñador pule.

AGRADECIMIENTOS

Me gustaría agradecer en primer lugar la ayuda y especialmente la paciencia brindada por Diego Mostaccio, mi tutor de proyecto, que me ha ido guiando durante el transcurso de éste. Respecto a lo primero, desde un primer momento tuve ideas muy claras sobre cómo llevar a cabo el proyecto, pero cuando empezaron a surgir los primeros imprevistos él fue quien me dio las ideas y los consejos para mejorar lo que estaba haciendo. Respecto a lo segundo,

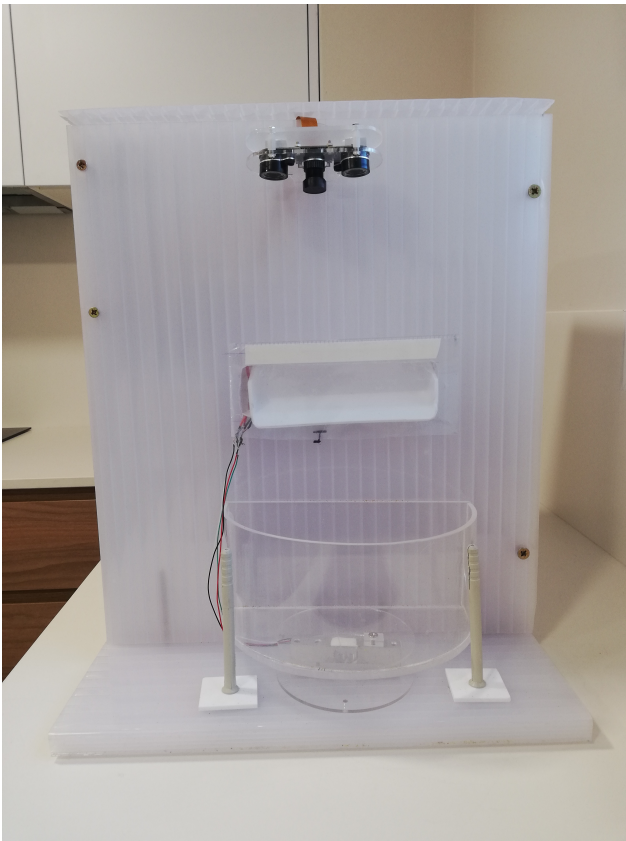


Fig. 14: Dispensador (parte delantera)

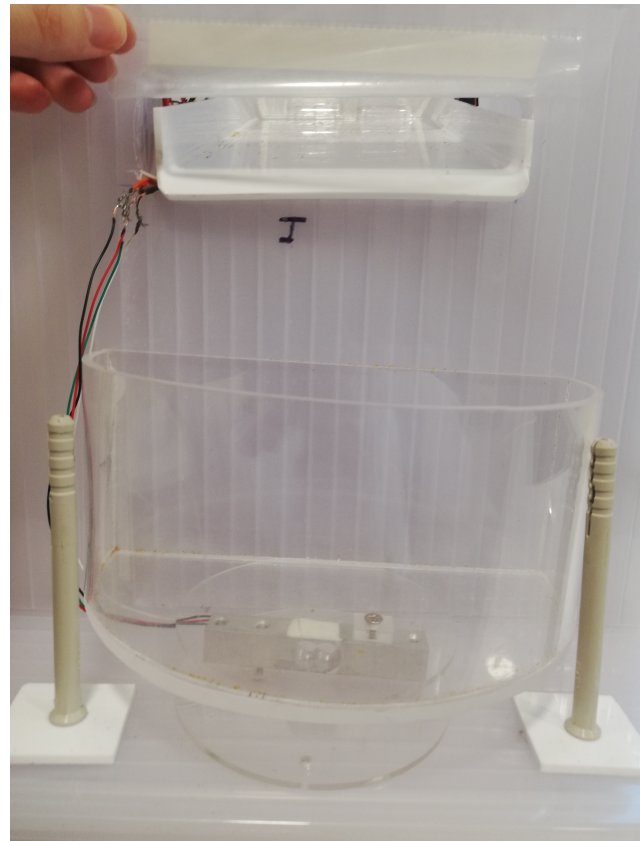


Fig. 15: Ranura por donde cae la comida y recipiente donde cae

dada mi situación de estudiante de último curso, en la que combiné trabajo con estudios, tuve que realizar un esfuerzo para poder compaginar las dos cosas y realizar las entregas en el plazo, así que agradezco su comprensión para permitirme hacer las entregas a poco tiempo del límite.

También me gustaría dar las gracias a mi familia que me apoyó de principio a fin durante el transcurso del desarrollo del proyecto.

REFERENCIAS

- [1] Documentación oficial de las Raspberry Pi (En línea). [Consultado: 1/2/21] <https://github.com/raspberrypi/documentation>
- [2] Dispensador de comida automático con aplicación (En línea). [Consultado: 19/6/21] <https://www.amazon.es/Comedero-Autom%C3%A1tico-Dispensador-Control-Nocturna/dp/B0863L6BF4>
- [3] Porcentaje de hogares españoles con algún animal doméstico (En línea). [Consultado: 19/6/21] <https://www.lavanguardia.com/vangdata/20150601/54431517328/porcentaje-hogares-mascotas.html>
- [4] Dispensador de comida automático con aplicación "low-cost" (En línea). [Consultado: 19/6/21] <https://www.amazon.es/DADYPET-Comedero-Automatico-Dispensador-Regularidad/dp/B08DJ5P4DQ/>
- [5] Capacidad de memoria de las SD en las Raspberry (En línea). [Consultado: 19/6/21] <https://www.raspberrypi.org/documentation/installation/sd-cards.md>
- [6] Datasheet del teclado de membrana 4x4 (En línea). [Consultado: 12/2/21] <http://www.electronicoscaldas.com/datasheet/Teclado-membrana-matricial-4x4.pdf>
- [7] Datasheet del electroimán (En línea). [Consultado: 2/6/21] <https://wiki.keyestudio.com/KS0320KeyestudioElectromagnet>
- [8] Datasheet de la pantalla LCD usada (En línea). [Consultado: 22/2/21] <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>
- [9] Datasheet del servomotor utilizado (En línea). [Consultado: 22/2/21] <https://datasheetpdf.com/pdf/791970/TowerPro/SG90/1>
- [10] Datasheet y descripción de producto de la cámara utilizada (En línea). [Consultado: 22/2/21] <https://es.aliexpress.com/item/32795660121.html>
- [11] Datasheet de la celda de carga de 5kg (En línea). [Consultado: 22/2/21] <https://solectroshop.com/es/sensores-de-presion/1072-celda-de-carga-5kg.html>
- [12] Datasheet del convertor ADC HX711 (En línea). [Consultado: 2/4/21] https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
- [13] Policarbonato celular (En línea). [Consultado: 19/5/21] <https://www.mwmaterialsworld.com/es/materiales/policarbonato/policarbonato-celular/plancha-de-policarbonato-celular-nido-de-abeja.html>

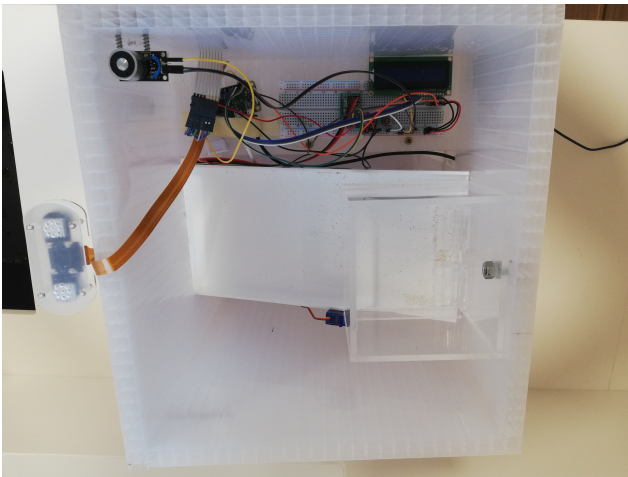


Fig. 16: Interior del dispensador (general)



Fig. 17: Interior del dispensador (lateral)

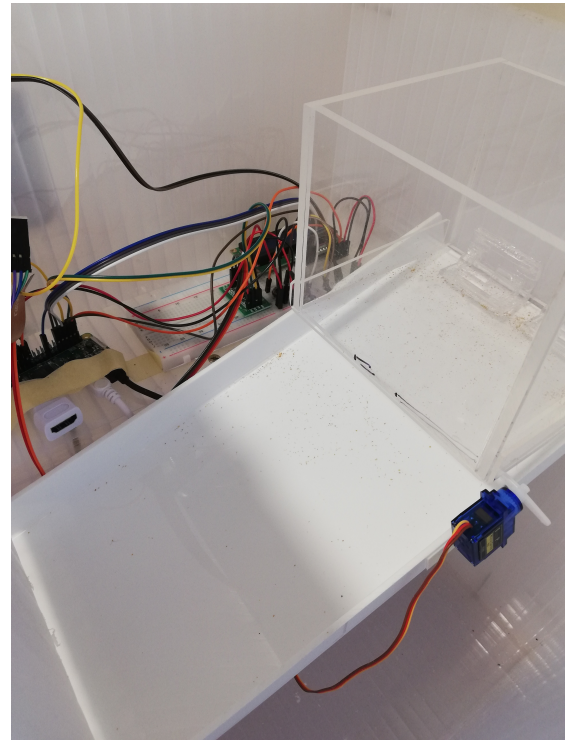


Fig. 18: Depósito de comida y rampa

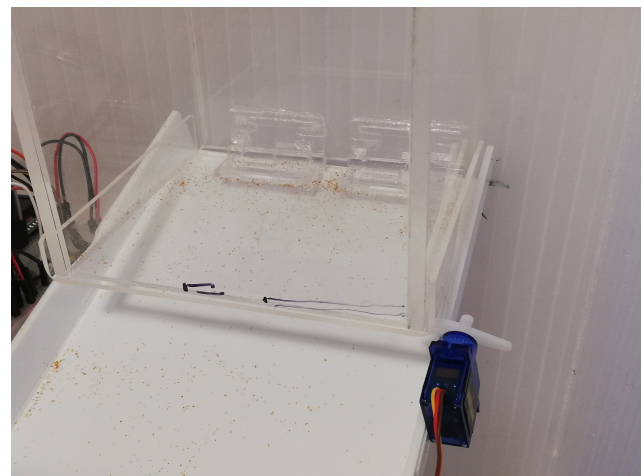


Fig. 19: Servo que mueve la tapa para abrirla y cerrarla

[14] Herramienta de modelaje 3d de Sketchup (En línea). [Consultado: 5/3/21] <https://www.sketchup.com/es/plans-and-pricing/sketchup-free>

[15] Documentación oficial de Firebase (En línea). [Consultado: 1/6/21] <https://firebase.google.com/docs?hl=es>