

# Diseño e implementación de un dispensador remoto de alimento y agua para perros en la plataforma Arduino.

Design and implementation of a remote food and water dispenser for dogs on the Arduino platform.

Eric Steven Torres Stuar  
eric.torres00@usc.edu.co

Wilder Quintero Chocue  
wilder.quintero00@usc.edu.co

Ciro Antonio Dussán Clavijo, M.Sc  
ciro.dussan00@usc.edu.co

Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería de Sistemas

## **Resumen**

Este proyecto tiene como finalidad asegurar la alimentación de nuestros perros, mediante un dispositivo móvil que controla de forma remota un dispensador, suministrando agua y alimento regularmente de acuerdo con la necesidad. Tiene una capacidad de alimento correspondiente a 3kg, el agua la provee la vivienda mediante la conexión del dispositivo a un grifo y estará controlada por el usuario según la necesidad que requiera la mascota, estas especificaciones brindan una autonomía para 2 días de suministros alimenticios si el perro es de raza mediana. El proyecto se ejecuta bajo el modelo en cascada garantizando un enfoque ordenado y secuencial siguiendo rigurosamente cada una de sus etapas. El dispositivo ofrece total libertad y tranquilidad al dueño del canino permitiéndole la alimentación a distancia, garantizando una nutrición adecuada y el bienestar de la mascota con un producto de fácil utilización, que provee servicios y que es amigable con el medio ambiente por su composición.

*Palabras Clave:* Dispensador, Arduino, IoT, Automático, Modelo en Cascada, Alimentación para perros.

## **Abstract**

This project aims to ensure the feeding of our dogs, through a mobile device that remotely controls a dispenser, supplying water and food regularly according to the need. It has a food capacity corresponding to 3kg, the water is provided by the house by connecting the device to a tap and will be controlled by the user according to the need of the can, these specifications give us autonomy for 2 days of food supplies if the dog is of medium race. The project is executed under the cascade model, guaranteeing an orderly and sequential approach, rigorously following each of its stages. The device provides total freedom and tranquility to the owners of the dog allowing them to feed at a distance for several days, ensuring adequate nutrition and welfare of the pet with a product that is easy to use, provides services and is friendly to the environment his composition.

*Keywords:* Dispenser, Arduino, IoT, Automatic, Waterfall Model, Feeding dogs.

## **1. INTRODUCCIÓN**

En la sociedad actual, hay personas que comparten sus vidas con animales domésticos. En su mayoría la gente prefiere como mascotas a los perros. De acuerdo con un estudio realizado en 2015 por B&Optimos, empresa de estudios de mercado en Latinoamérica contratada por Fenalco, halló que, de un total de 1.500 personas encuestadas de diferentes ciudades de Colombia, 555 dijo tener mascota, es decir, un 37%. De este porcentaje el 70% afirmó que tiene como mascota un perro mientras que el 13% tiene gato (Arboleda Villa & Restrepo Perez, 2017). Asimismo, un estudio realizado por Euromonitor Internacional para el evento ExpoPet 2016, señaló que Colombia es el cuarto país con mayor mercado de consumo de productos para mascota. Al mismo tiempo, dicha investigación asevera que aproximadamente diez millones de perros se alimentan todos los días con concentrados (Arango Holguín, 2016).

Esta información permite inferir un crecimiento en el mercado de las mascotas, y además factores implícitos en la tenencia de estas, como los costos económicos asociados, el tiempo y cuidado que se debe invertir en ellas. Se recomienda dedicar a la mascota al menos 45 minutos por día (Estevez, 2005). No obstante, algunas personas dejan a sus mascotas solas durante gran parte del día debido a que no pueden abandonar sus compromisos diarios, además, tampoco todos los establecimientos abiertos al público permiten el ingreso de

animales, teniendo en cuenta el artículo 94 de la Ley 1801 del 2016, relacionado con la salubridad pública que afectan la actividad económica (Código Nacional de Policía y Convivencia, 2017), siendo esto, una dificultad cuando las personas desean salir con sus mascotas. Asimismo, en el artículo 03 de la Ley 1774 de 2016 (Congreso de la Republica, 2016), establece que el responsable o tenedor de animales deben asegurar como mínimo que no sufra de hambre ni sed.

Los perros deben de tener como mínimo dos porciones de alimento al día, ya que, si no se cumple esta rutina de alimentación, puede que el canino padezca de enfermedades. La mala nutrición puede generar enfermedades que afectan su metabolismo y el desarrollo de sus huesos y órganos a una temprana edad. Los perros demandan tiempo y atención, por lo tanto, el amo debe disponer adecuadamente sus recursos para dedicarle los cuidados correctos a su mascota sin desatender las responsabilidades de la vida cotidiana y laboral, dando importancia a una buena alimentación, que contribuya a la salud y calidad de vida del canino. En busca de ello, los autores de esta investigación, encontró diferentes dispositivos que resuelven este problema mediante una alimentación automatizada.

Zapata & Gil, [2017] en su trabajo de grado "Diseño e implementación de un prototipo de dispensador automático de comida para animales basado en Raspberry pi controlado mediante una aplicación móvil", tuvieron como objetivo la creación de un sistema automático de dosificación de alimento para perros de raza pequeña en el hogar. Desarrollándolo en 2 etapas, la estructura física y la parte de automatización, como resultado lograron la construcción del dispositivo que dispensa alimento cuando se le ordena desde la aplicación móvil ejecutando la dosificación para la mascota. Por otra parte, también existen dispositivos como el que plantean, Tuta, Rivera, Gutiérrez, & Roa, [2015] en su artículo científico "Diseño y construcción de un dosificador automático de alimento para perros de raza media", que tuvo como objetivo ayudar a las personas a alimentar a sus mascotas solucionando los problemas de tiempo y ocupación de los propietarios. Utilizando una metodología de diseño mecánico integrando diversos componentes electrónicos como Arduino y de comunicación GSM, en el que obtuvieron un prototipo que evaluaron de forma satisfactoria en resistencia estructural por su composición de madera, en cuanto a su dispensación y la forma de notificación por mensaje de texto al dueño del perro sobre la dosificación de comida. O como plantea Martínez, [2015] en su trabajo de pregrado "Dispensador de alimento para perros dosificable a través de dispositivos móviles Android" donde tuvo como objetivo proveer la cantidad de alimento suficiente para el perro con el fin de asegurar su alimentación y cuidado de forma automatizada. Fundamento su metodología en el modelo en cascada. Finalmente obtuvo el dispositivo que dispensa el alimento mediante la aplicación móvil en Android según horarios establecidos por el usuario.

Revisando las diferentes formas en las que se ha tratado de solucionar el problema por otros autores, se describe y contextualiza en este documento, el objetivo de diseñar e implementar un dispensador remoto de alimento y agua para perros en la plataforma Arduino, que realizará la actividad humana de proveer estas necesidades de forma remota sin la necesidad de que el amo este físicamente al lado de su mascota, ya que, de no ser tenida en cuenta, puede repercutir en graves problemas de salud. Lo anterior se logra, mediante la integración de diferentes dispositivos electrónicos que satisfacen los requerimientos del dispensador. Para el control y monitoreo se desarrolla una aplicación móvil en Android a la que se le realizarán pruebas de integración con el fin de valorar si los componentes individuales trabajan en conjunto y como se espera de ellos (Rodríguez, 2019).

El artículo está claramente dividido en 4 secciones donde se presenta nuestra investigación. En la sección 1 muestra la introducción, donde se contextualiza al lector basado en antecedentes que apoyan nuestra razón de construir un dispositivo que permita la alimentación del perro. Se identifica el problema que existe en la sociedad para aquellas personas que, por su vida laboral y cotidiana, no les es posible estar físicamente para abastecer de alimento y agua a su mascota. En la sección 2 se abordó el camino estratégico para resolver el problema mediante la clarificación del tipo de investigación que es el aplicado, alineado al desarrollo de sistemas informáticos con una estructura metodológica en cascada, adicionalmente se detallan las herramientas que se aplicaron en todas las fases del proyecto. En la sección 3 se encuentra la presentación de los resultados donde se resuelve el problema y así, cumpliendo con el objetivo de la investigación al obtener el prototipo funcional junto a su discusión que en gran medida es fundamental al identificar valor agregado. En la sección 4 se muestran las conclusiones enunciando algunas ideas que apoyan los resultados obtenidos.

## **2. METODOLOGÍA /MATERIALES Y MÉTODOS**

### **2.1 Tipo de Investigación y línea de investigación.**

El tipo de investigación implementada en el proyecto es el aplicado, donde se utiliza los conocimientos adquiridos en diferentes áreas, como la programación de sistemas embebidos, programación móvil, circuitos eléctricos, sensores, actuadores. Estos saberes, serán utilizados para el diseño e implementación de un prototipo dispensador de alimento para los perros, desarrollando un producto que innove y mejore las condiciones de vida de estas mascotas.

### **2.2 Línea de investigación**

La línea de investigación al que pertenece el proyecto es de desarrollo de sistemas informáticos. Se busca que, mediante una aplicación móvil desarrollada en Android controle la necesidad de alimentación y suministro de agua, donde actuadores y sensores integrados a un sistema embebido proporcionaran los datos a una arquitectura de software de microservicios ubicada en el Back-end, que de acuerdo a (Cherradi, El Bouziri, Boulmakoul, & Zeitouni, 2017) definen los microservicios como pequeños componentes de software, especializados en una tarea y que trabajan juntos para lograr una tarea de nivel superior. El objetivo de esta arquitectura es separar cada componente del negocio y que funcione de manera independiente, sin la necesidad de compartir información directa con algún otro

servicio.

## 2.3 Metodología

La aplicación móvil se elaboró implementando el modelo en cascada (Pressman, 2010), cada etapa de este enfoque representa una unidad de desarrollo con un pequeño descanso en el medio. Por tal motivo cada etapa que sigue comienza tan pronto como la anterior está terminada. Se opta por aplicar este método debido a que la solución que se propone en el sector de alimentos para mascotas, los objetivos son precisos desde el inicio y es apropiado para proyectos pequeños donde todos los aspectos del desarrollo son conocidos de antemano y no demandan más de 100 horas de desarrollo. Sobre todo, considerando que este enfoque permite a los negocios deshacer del papeleo innecesario, reuniones regulares que consumen mucho tiempo y retrasos en sus procesos de negocio (MegalPractical, 2019). La metodología define 5 fases llamadas análisis de requisitos, diseño, construcción, pruebas, y despliegue.

### 2.3.1 Análisis de requisitos

En esta fase se realizó la recolección de datos y requerimientos preliminares, de esta forma se pudo clarificar los objetivos del proyecto y resultados esperados. Se definieron las características del dominio de la aplicación, categorización de los requerimientos y la personalización del servicio. La obtención de requerimientos estableció una aplicación que permitió al usuario ejercer un control en los servicios ofrecidos por el prototipo dispensador, de manera remota. Además, la aplicación presentó una interfaz de usuario que ofreció facilidad en la interacción del usuario con los servicios. Se plantea que la aplicación móvil, controlará el prototipo dispensador de alimentos y debe cumplir con los siguientes requerimientos:

**Tabla 1. Requerimientos del Sistema Dispensador**

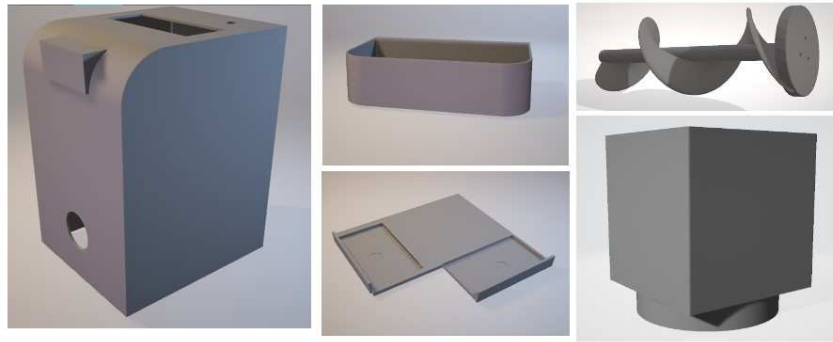
Categoría	Descripción del Requerimiento
Generales	El sistema debe almacenar en un medio persistente toda la información que la aplicación móvil permita registrar.
	El sistema debe distinguir entre diferentes usuarios del sistema dispensador y habilitar sus opciones según sea el rol de cada uno.
	El sistema debe cuantificar el alimento y agua al momento de dispensar a través de la aplicación móvil.
Gestión de Información	El sistema debe permitir el registro de usuarios con información personal (Nombre, Apellido, Identificación Usuario, Password, Email).
	El sistema debe validar a un usuario para el uso del prototipo dispensador mediante una base de datos no relacional.
	El sistema debe permitir al usuario, gestionar (actualizar, eliminar y guardar) un registro creado con anterioridad.
	El sistema debe permitir al administrador, gestionar (guardar, buscar, actualizar y eliminar) la información general de los usuarios registrados en la base de datos.
	El sistema debe permitir al usuario, registrar la información del can (nombre, raza, edad, peso, tamaño).
	Se requiere que el sistema dispense la cantidad exacta de alimento y agua pasado por parámetro mediante la App.
	El sistema debe informar a los usuarios, el nivel de alimento que se encuentra almacenado en la tolva. Dicha información será capturada por el sensor de distancia (a proximidad).
	El sistema debe informar al usuario la cantidad de alimento servido en gramos, en el plato. Dicha información será capturada por el sensor de fuerza.
Generar reportes estadísticos de la cantidad de alimento y agua que se le ha servido al canino, representados en gráficos.	

Fuente: Elaboración propia

### 2.3.2 Diseño

En esta etapa se precisó la estructura del hardware como del software aplicado en el proyecto. Se utiliza programas CAD como Autodesk 123D Design (Autodesk, 2019) para definir el dimensionamiento y la forma de los diferentes componentes que constituirán el prototipo dispensador.

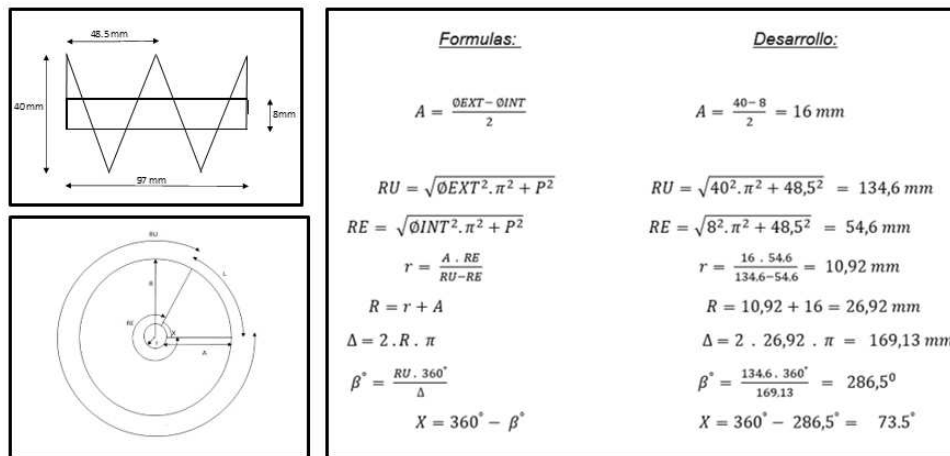
**Ilustración 1. Diseño de Componentes del Prototipo Dispensador**



Fuente: Elaboración propia

El mecanismo de dosificación para este proyecto fue el tornillo sin fin o de Arquímedes. De acuerdo con (León Quenguan & Rueda Almario, 2013) este mecanismo puede estar en funcionamiento en el tiempo de manera continua o intermitente. Este es el más usado por su implementación simple y porque se acomoda a la naturaleza de casi cualquier producto granulado, transportando cantidades pequeñas de alimento, del cual se requiere para aproximar la cantidad en gramos a dispensar mediante la aplicación móvil, cuando este se pase por parámetro. Se descarta otros mecanismos como: compuerta rotatoria por su construcción robusta y menor precisión, y de banda rodante porque implicaría controlar con un actuador como el servomotor, la apertura y cierre de una compuerta, y con otro, el movimiento de la banda trasportadora. En la Ilustración 2, se formula los cálculos del mecanismo del tornillo sin fin o de Arquímedes.

**Ilustración 2. Datos Entrada y Cálculos del Tornillo Sin Fin**



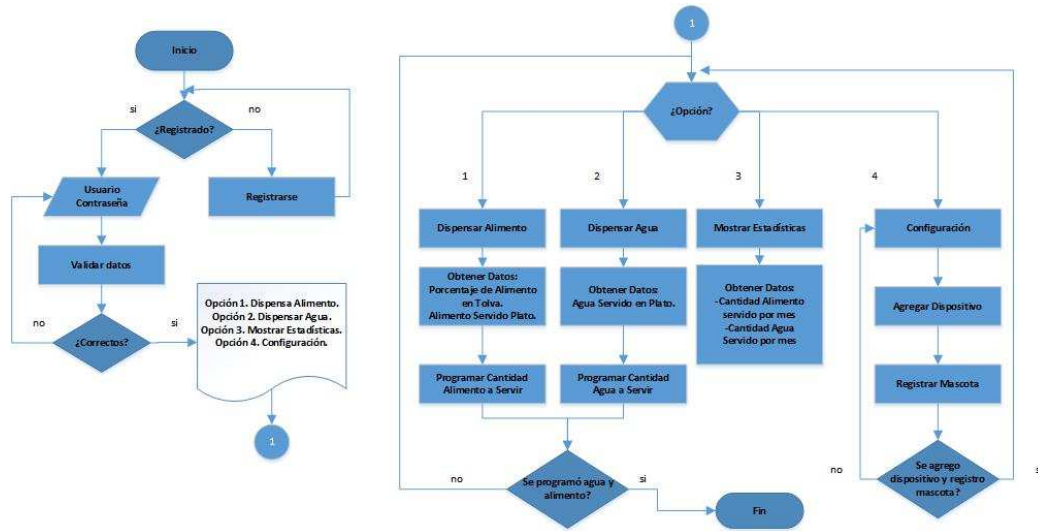
Fuente: Elaboración propia

Se representó la aplicación móvil mediante el Lenguaje Unificado de Modelado (Natsys, 2017) para sistemas de software, a través de diagrama de casos de usos, que soportaron el funcionamiento de las acciones de la aplicación. En la ilustración 3, se precisa el diagrama de casos de uso general que muestra las acciones que puede efectuar el usuario al interactuar con la aplicación móvil de la siguiente manera, así: el usuario ingresa a la aplicación con previa creación de un registro y validación de datos, representado por el caso CUS2 Registrar\_Usuario. Posterior valida el nuevo usuario mediante el caso CUS1. Validar\_Usuario. Después de acceder a la aplicación, el usuario puede optar por el caso CUS3. Dar\_Servicio, que permite ofrecer servicio como dispensar alimento o suministrar agua, identificados en los casos de usos CUS6. Dispensar\_Alimento y CUS7. Dispensar\_Agua, respectivamente. Una vez el usuario, haya interactuado con los dos casos mencionados, el sistema almacenara los datos de alimento servido y disponible en tolva para realizar reportes mediante el caso de uso CUS5. Mostrar\_Estadisticas. Por otro lado, mediante el caso CUS9. Configurar\_Dispositivo, el usuario puede hacer uso de los casos de uso CUS8. Agregar\_Dispositivo y caso de uso CUS4. Registrar\_Mascota, siendo el primero en asociar un dispositivo dispensador a un usuario registrado, y el último, el registro de los datos del perro.



app en Android. Con estas especificaciones de requerimientos y diseño, se procede a construir un diagrama de flujo de la APP que cumpla con estos requerimientos funcionales (ver Ilustración 5), y simultáneamente se realiza pruebas para encontrar errores en los servicios que ofrece el prototipo dispensador.

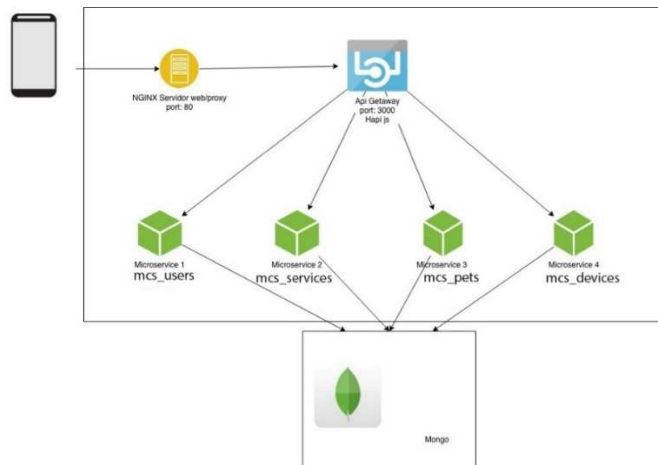
**Ilustración 5. Diagrama de Flujo del APP**



Fuente: Elaboración propia

Para poner en funcionamiento la arquitectura de microservicios se hará uso de Amazon Web Services, que es un proveedor de servicios de almacenamiento, bases de datos, servidores y aplicaciones a través de Internet en la nube bajo demanda (Sajee Mathew, 2017) y como base de datos se utiliza el sistema gestor MongoDB, “guarda la estructura de los datos en documentos tipo JSON con un esquema dinámico llamado BSON” (Martín, Chávez, Murazzo, & Rodríguez, 2015) que permite agilizar considerablemente la escalabilidad de las aplicaciones, debido a que no hay que preocuparse por la modificación de esquemas o agregar nuevas tablas cuando la aplicación móvil evoluciona (ver Ilustración 6).

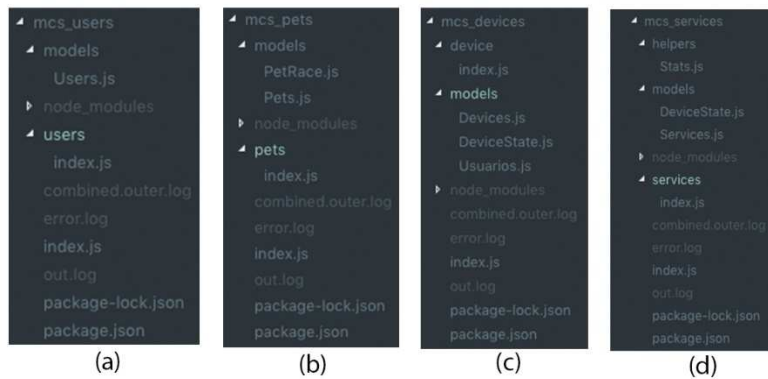
**Ilustración 6. Arquitectura de Microservicios**



Fuente: Elaboración propia

El contenido en los directorios de los microservicios (ver Ilustración 7), mantienen un mismo aspecto. El subdirectorio /models, incluye los modelos que se maneja con mongoose en la base de datos, donde cada archivo representa un documento (tabla). El subdirectorio/node\_modules, se encuentran las librerías que se utiliza para el desarrollo. Los ficheros de package-lock.json y package.json es donde se encuentra el mapa de configuración de los paquetes de Node.js y estos serán utilizados por npm (Node Package Manager). Los ficheros error.log y out.log son generados por pm2, que es la herramienta con que se ejecuta, y este a su vez, informa cuando deja de funcionar algún servicio.

**Ilustración 7. Estructura General Microservicios**



Fuente: Elaboración propia

Los cuatro (4) servicios se ubican en la parte del Back-End y se explican de la siguiente manera, así:

- `mcs_users`: Este servicio se encarga de administrar toda la información de los usuarios y gestionar todo lo relativo a la autenticación de estos.
- `mcs_pets`: Este servicio se encarga de administrar toda la información de la mascota, relativa a las acciones que ha registrado en el sistema.
- `mcs_devices`: Este servicio se encarga de la configuración de dispositivos dispensadores asociando a los usuarios registrados.
- `mcs_services`: Este servicio se encarga de ofrecer al usuario las funcionalidades de dispensar alimento y agua al can.

El api del proyecto se realiza en su totalidad en el lenguaje JavaScript usando Node.js como su entorno de ejecución. El framework usado es Hapi.js, el cual es un rico marco de trabajo que facilita la construcción de una arquitectura de microservicios (Nandaa, 2018) ya que es compatible con Chairo y Seneca (Lv & Wang, 2016), este último es un toolkit o herramienta para crear los pequeños servicios que permiten que cada microservicio se ejecute en un puerto específico en cada servidor. A parte de correr el servicio Seneca, se crea la interfaz con la que hapi.js se conectara para la ejecución de las funciones. Esta integración es permitida por Chairo, un plugin de hapi.js que permite de forma natural la conexión entre el api Gateway y el microservicio.

A nivel de seguridad, Hapi.js controla el plugin de jwt que provee después de validar el usuario, de una cadena (token) cifrada que permite la validación de cada petición para ser procesada o no. La estructura de la cadena es la siguiente.

**Ilustración 8. JWT Token**



Fuente: <https://codeburst.io/jwt-to-authenticate-servers-apis-c6e179aa8c4e>

Este token dentro de su configuración tiene un ttl que permite darle caducidad, obligando al usuario a realizar el logueo en cada determinado tiempo.

Todas las peticiones pasan primero por el api construido con hapi.js este es quien controla la concurrencia y el intercambio de

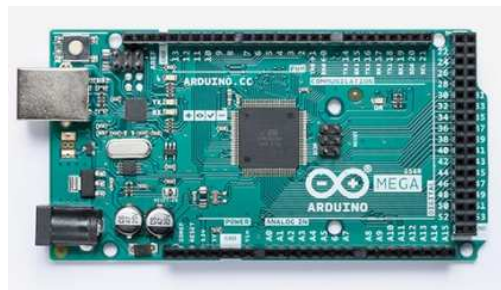
recurso de origen compartido (CORS), sin embargo, para estos puntos se pasa la responsabilidad al servidor de aplicaciones usado NGINX, este es configurado para controlar el número de peticiones y conexiones, que determina el puerto de entrada y el puerto a redireccionar funcionando como un proxy reverso. A nivel de base de datos, cada microservicio controla las conexiones y ejecuciones en distintas bases de datos con facilidad, siendo lo más destacado en esta arquitectura.

La estandarización es importante, porque permite tener los objetos claramente ubicados para la depuración de bugs y nuevas implementaciones, así mismo el nombramiento de variables y carpetas se realizan en inglés, así como el uso de lower camel case.

La selección de los componentes electrónicos empleados en el prototipo dispensador, se basaron en el cumplimiento de los requerimientos y teniendo en consideración el costo y vida útil de estos elementos.

- A. **Tarjeta de programación.** Se utiliza la placa Arduino mega por su gran capacidad de memoria, pines de conexión que brinda y además como manifiesta (Suárez Mora, Aparicio Gómez, Gallego Ibáñez, & Ramírez Salcedo, 2015) la placa tiene todo el desarrollo de hardware y software que facilita la integración con aplicaciones a bajo nivel, considerando que se trabaja con voltajes o corrientes provenientes de actuadores, pulsadores y sensores. En esta placa se logra integrar el componente módulo Nodemcu **ESP8266**, el cual este se obtiene a un bajo costo, en comparación a la placa Arduino uno, donde se debe comprar el chip Wi-Fi Shield a un elevado costo e integrarlo. Igualmente se considera en un alto costo la placa Arduino MKR1000, a pesar de que este dispositivo tiene integrado el Wi-Fi Shield y el chip Atmel ATSAMW25.

*Ilustración 9. Arduino Mega*



Fuente: Tomado de: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

- B. **Módulo WiFi para Arduino ESP8266.** Módulo Nodemcu. Es un chip integrado WIFI que soporta el Protocolo de Control de Transmisión/Protocolo de Internet (Transmission Control Protocol/Internet Protocol, TCP/IP), IP V4, UDP, HTTP y FTP. El objetivo de este chip es dar acceso a una red inalámbrica a un microcontrolador. Una de la característica que tiene este chip es su bajo consumo de energía en 200 miliamperios y bajo costo. Este se puede conectar a cualquier dispositivo electrónico, un ejemplo muy claro es en las instalaciones de domótica y control de maquinaria industrial.

*Ilustración 10. Módulo Nodemcu ESP8266*



Fuente: Tomado de: <https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/>

- C. **Servomotor.** Se emplea un Servo SG-5010 Tower Pro, los servomotores tienen como principal característica la baja velocidad y su posición es programable de acuerdo con los requerimientos. Además, posee un torque de 6.5Kg-cm que cumple con las exigencias de su función: mover 360° el tronillo sin fin o de Arquímedes hacia la izquierda, y transportar el alimento granulado al recipiente del perro.

**Ilustración 11. Servo SG-5010 Tower Pro**



Fuente: Tomado de <https://naylampmechatronics.com/servomotores/22-servo-sg-5010-6kg.html>

- D. **Sensor de Ultrasonido.** Este sensor es utilizado como detector de proximidad, con rangos de hasta algunos metros, mediante un sonido que el dispositivo emite, después procesa el tiempo que la señal tarda en regresar. Este sensor permitió conocer el porcentaje de alimento que se encontraba almacenado en la tolva.

**Ilustración 12. Sensor de Ultrasonido**



Fuente: Tomado de <https://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/>

- E. **Válvula Solenoide.** Para el proyecto implementaremos la dosificación de agua mediante una válvula solenoide de 12V normalmente cerrada, con una dimensión de 1/2 pulgada, y que se encargara de regular el flujo del líquido directamente desde la red de agua interna al plato de agua para el canino.

**Ilustración 13. Válvula Solenoide**



Fuente: Tomado de <https://www.mechatronicstore.cl/tienda-de-electronica/actuadores-y-motores/electrovalvula-12v-12-nc>

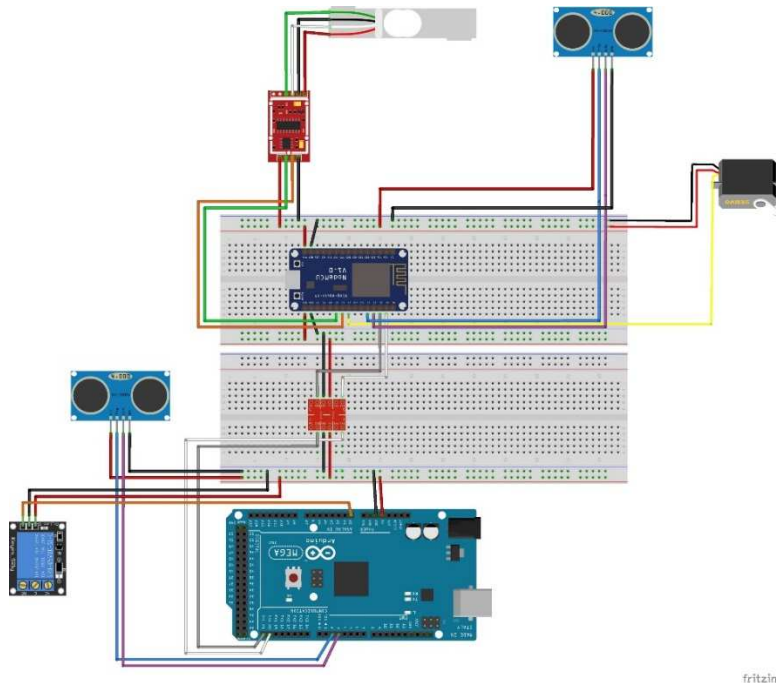
### 2.3.4 Pruebas

En esta etapa se evalúa la funcionalidad del prototipo dispensador mediante pruebas de integración, permitiendo conocer el cumplimiento de los objetivos propuestos, se realizó un seguimiento más riguroso a nivel técnico. Las pruebas del proyecto se verifican directamente desde la aplicación móvil conforme a la funcionalidad, interfaz del sistema y comportamiento ante ambientes de proceso ideal y proceso con datos erróneos con la manipulación de datos externos que la aplicación móvil solicita para su funcionamiento. Además, se plasma la evidencia de los resultados obtenidos mediante pantallazos que aparecieron durante la ejecución. “Documentar los resultados es de suma importancia, ya que si en un futuro surgen problemas que rondan a alguna de las pruebas ejecutadas, gracias a la evidencia se podrá demostrar que en el transcurso de las pruebas funcionaba correctamente” (Rodríguez, 2019).

## 3. RESULTADOS Y DISCUSIÓN

De acuerdo a los objetivos y los requerimientos levantados en la etapa de análisis de este proyecto, se puede evidenciar en la ilustración 15, el logro de la integración de los componentes electrónicos que permitió el adecuado funcionamiento de los servicios de alimento y agua, el cual fue controlado mediante la aplicación móvil desarrollada para el sistema operativo Android. Dicho esquema se realizó mediante el programa simulado de circuitos FRITZING (fritzing, 2019), y las conexiones fueron las siguientes: (a) Arduino Mega 2560, (b) Módulo Nodemcu ESP8266, (c) Servo SG-5010 Tower Pro, (d) Sensor de Ultrasonido, (e) Sensor de Fuerza, (f) Sensor Nivel de Agua, y (g) Válvula Solenoide.

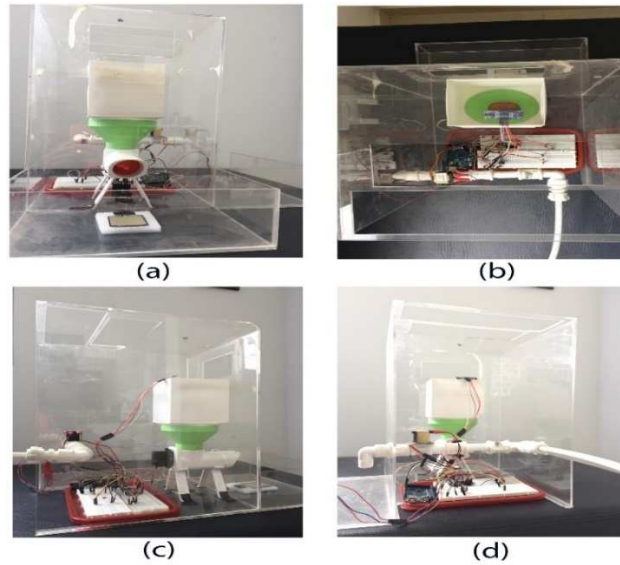
*Ilustración 14. Diagrama Esquemático*



Fuente: Elaboración propia

Es importante indicar que, al buscar la conexión a internet, se integró al Arduino, el Módulo Nodemcu ESP8266 v2, dio algunas ventajas como, acoplamiento a protoboard, reducción de costos económicos, poder de procesamiento al tener su propio controlador y facilidad en su programación al permitir cargar sus rutinas desde el IDE Arduino. De otro modo, se puede apreciar en la ilustración 16, la maqueta del prototipo dispensador que ofreció los servicios planteados en esta investigación. Se considera: a) parte frontal, b) parte superior, c) parte lateral y d) parte posterior.

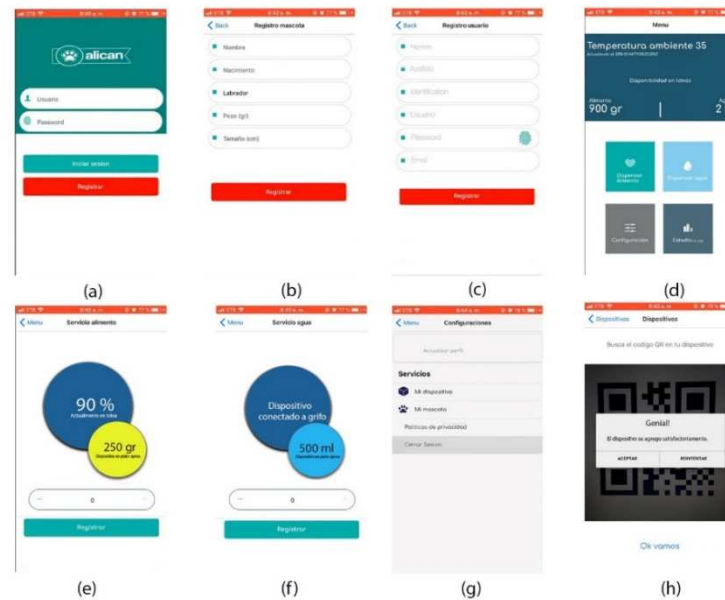
**Ilustración 15. Maqueta Prototipo Dispensador**



Fuente: Elaboración propia

En la Ilustración 17 se muestra el procedimiento de login (ver Ilustración 17-a), registro de usuario (ver Ilustración 17-b), registro de mascota (ver Ilustración 17-c) y el menú principal (ver Ilustración 17-d). Este último, ofreció los servicios principales de dispensar alimento y dispensar agua (ver Ilustración 17-e, 17, respectivamente). El registro de la mascota y la configuración del dispositivo se realiza mediante la opción Configuración. La aplicación móvil asoció mediante código QR un dispositivo dispensador a un usuario. Adicionalmente, se mostraron las opciones de estadísticas y configuración del dispositivo, en el reporte estadístico se observa la información del consumo de alimento y agua que ha ingerido el can por día, semana o mes.

**Ilustración 16. Funcionalidad de (a) Validar Usuario, (b) Registrar Usuario, (c) Registrar Mascota, (d) Ofrecer Servicios, (e) Servicio Agua, (f) Servicio Alimento, (g y h) Configurar Dispositivo**



Fuente: Elaboración propia

En la tabla 2 se muestran los componentes que se probaron, el caso de prueba, los resultados esperados de la ejecución de la aplicación móvil. Con el cumplimiento a este objetivo, se pudo evidenciar que obtuvimos un 100% de aceptación en cuanto a la aprobación de los 7 casos de pruebas que se verificaron para su correcto funcionamiento.

Tabla 2. Plan de Pruebas de Integración.

COMPONENTE	CASO DE PRUEBA	DESCRIPCION DE LA PRUEBA	PROCEDIMIENTO	RESULTADOS ESPERADOS	TIPO DE PRUEBA
USUARIO	Registro usuario	Comprobar el correcto funcionamiento de la aplicación móvil en el registro de datos para acceder por primera vez.	1. Ingresar al registró. 2. Ingresar datos entradas. Con respectivas validaciones 3. Registro (Guardar).	1. Crear el registro de usuario. 2. Mostrar en pantalla creación exitosa.	Positiva
	Validar usuario	Comprobar el correcto funcionamiento de la aplicación móvil con la base de datos para la validación de usuario e ingreso al menú servicios.	1. Digitar en el cuadro de texto el usuario. 2. Digitar en el cuadro de texto la contraseña. 3. Presionar el botón "Iniciar Sección".	1. La aplicación móvil reconozca los datos ingresados en la base de datos. 2. Ingresar a la pantalla de menú servicios.	Positiva
	Configurar Usuario	Comprobar el correcto funcionamiento de la aplicación móvil para la modificación del perfil de los usuarios.	1. Ingresar a la sección configuración de la aplicación móvil. 2. Presionar el botón "Actualizar Perfil" 3. Realizar modificaciones a los datos del perfil de usuario. 4. Presionar el botón "Edita cambios"	1. Modificar los datos del usuario ingresados con anterioridad. 2. Mostrar un mensaje de confirmación de exitosa la operación.	Positiva
CONFIGURAR DISPOSITIVOS	Agregar Dispositivo	Comprobar el correcto funcionamiento de la aplicación móvil, en asociar un dispensador prototipo a un usuario debidamente registrado, para el uso posterior de los servicios de Agua y Alimento.	1. El usuario se encuentra en la pantalla "Configuraciones". 2. Presiona el botón "Mi dispositivo". 2. El usuario enfoca la cámara hacia el código QR del prototipo dispensador. 3. Presiona el botón "OK VAMOS"	1. El sistema lee el código y muestra un mensaje de confirmación.	Positiva
REGITRO MASCOTA	Configurar Mascota	Comprobar el correcto funcionamiento de la aplicación móvil en el registro de la mascota (can).	1. El usuario se encuentra en pantalla "Configuraciones" 2. Presiona el botón "Mi Mascota". 3. Seleccionar la raza de la mascota. 4. Digitar en el cuadro de texto el Nombre. 5. Digitar en el cuadro de texto la Fecha de Nacimiento. 6. Digitar en el cuadro de texto el Peso. 7. Digitar en el cuadro de texto el Tamaño. 8. Presionar el botón "Registrar".	1. La aplicación móvil valide los datos ingresados 2. Mostrar mensaje de confirmación de registro éxito.	Positiva
SERVICIOS	Dispensar Alimento	Comprobar el correcto funcionamiento al momento de dispensar alimento, asignando la cantidad de alimento en gramos mediante la aplicación móvil.	1. Seleccionar el botón "+" para aumentar la cantidad en gramos a dispensar en múltiplos de 100. 2. Seleccionar el botón "-" para disminuir la cantidad en gramos a dispensar en múltiplos de 100. 3. Presionar el botón "Registrar" para enviar la petición al prototipo dispensador.	1. Enviar la petición de la cantidad de alimento al prototipo dispensador. 2. Dispensar la cantidad de alimento exacta en plato. 3. Visualizar en pantalla el nivel de porcentaje (%) de alimento en tolva y la cantidad en gramos que hay en plato. 4. Mostrar un mensaje de confirmación que se envió la petición.	Positiva
	Dispensar Agua	Comprobar el correcto funcionamiento al momento de dispensar agua, asignando la cantidad de agua en mililitros mediante la aplicación móvil.	1. Seleccionar el botón "+" para aumentar en múltiplos de 100 la cantidad agua en mililitros a dispensar. 2. Seleccionar el botón "-" para disminuir en múltiplos de 100 la cantidad agua a dispensar. 3. Presionar el botón "Registrar" para enviar la petición al prototipo dispensador.	1. Enviar la petición de la cantidad de agua al prototipo dispensador. 2. Dispensar la cantidad de agua exacta en plato. 3. Visualizar en pantalla la cantidad de agua que hay en plato. 4. Mostrar un mensaje de confirmación que se envió la petición.	Positiva

Fuente: Elaboración propia

A partir de los resultados obtenidos se afirma que logra resolver la hipótesis general, al asegurar una alimentación en proporciones recomendadas en los perros, de tal manera que los dueños de estos animales puedan cumplir con la responsabilidad de su alimentación sin la necesidad de estar presente en el hogar.

Estos resultados guardan relación con lo que plantean, Tuta, Rivera, Gutiérrez, & Roa, [2015] donde resaltan la gran importancia de los perros como animales domésticos en la sociedad, sustentando la necesidad de un dispositivo automatizado que brinde una dieta adecuada a los canes, permitiendo a sus dueños un poco más de libertad, lo cual es acorde con lo que sea resuelto. Pero en lo que no concuerda nuestra solución con lo planteado por los autores, es en la forma en la que ellos programan manualmente por medio de un teclado matricial los horarios y las cantidades de alimento a dispensar, por el contrario, nuestro prototipo da libertad al usuario de elegir los horarios desde la aplicación móvil en Android que le permite monitorear y controlar la cantidad de alimento y agua a dispensar de forma remota en el dispositivo.

Por otro lado Zapata & Gil, [2017] plantean que es posible la integración de dispositivos electrónicos como la Raspberry pi para la construcción de un prototipo de dispensador automático de comida para animales controlado mediante una aplicación móvil, esto concuerda con la concepción del dispositivo basado en componentes electrónicos como Arduino y NodeMCU ESP8266, sin embargo, se difiere en la implementación del servidor como lo es NGNIX, ya que ellos lo instalan directamente en la tarjeta y se someten a limitaciones de red para su configuración, por el contrario, este proyecto se basó en el paradigma de computación en la nube que obtuvo los servicios que ofrece AWS (Amazon Web Services), donde se instala la infraestructura bajo una arquitectura de Microservicios lo cual posibilita la expansión y el manejo de una muy grande cantidad de dispensadores pensando en un negocio a futuro.

#### 4. CONCLUSIONES

El desarrollo de un prototipo que logro dosificar alimento y agua para los canes de raza mediana, con autonomía de 2 días mediante una aplicación móvil, es lo que nos permite generar satisfacción al usuario. Con base en los resultados, se puede afirmar que los objetivos trazados se cumplieron de manera adecuada. La metodología empleada permitió que para proyectos software donde se brinda pocas

funcionalidades principales, se puede optar el modelo en Cascada, teniendo en cuenta que funcionalidades como registros de usuarios y validaciones, hoy en día son genéricas a todo producto informático. Componentes electrónicos como Arduino hicieron posible y asequible la construcción del dispensador, por la facilidad en el desarrollo, también por su gran capacidad de procesamiento.

Las herramientas CAD permitieron el diseño de las partes que finalmente fueron implementadas ensambladas y que dieron forma al dispensador. Los materiales y el diseño físico dieron resistencia, estética y son amigables con el medio ambiente. El sistema de dosificación de agua y alimento cumplió su propósito al brindar las porciones adecuadas mediante la aplicación móvil. La utilización de tecnologías open source, evitaron financiar recursos económicos en el pago de licencias.

Los conocimientos informáticos aplicados en las áreas de programación, control y telecomunicaciones, demostraron que un sistema embebido de plataforma independiente como Arduino puede ser utilizado para controlar, supervisar y adquirir datos, que permita administrar y obtener beneficios en un entorno IOT como el implementado.

Aunque existen trabajos relacionados con la temática de esta investigación, es evidente que con el desarrollo e implementación de este prototipo se brinda al usuario libertad, tranquilidad y sencillez para alimentar la mascota.

Dentro del proyecto se utiliza combinaciones en sistemas embebidos, programación en Android, sensores y mecánica, para lograr el correcto funcionamiento del prototipo, brindando a los amos de estas mascotas, la seguridad de que así no estén presentes en el hogar, podrán ofrecer dos aspectos básicos: la alimentación y el agua.

## REFERENCIAS

- Andriod Studio. (11 de 12 de 2018). *Android Studio*. Obtenido de Android Studio: <https://developer.android.com/studio/?hl=es-419>
- Arango Holguín, C. (19 de 10 de 2016). *El colombiano*. Recuperado el 11 de 05 de 2019, de El colombiano: <https://www.elcolombiano.com/cultura/mascotas/mascotas-mercado-crece-en-colombia-un-13-por-ciento-IX5202497>
- Arboleda Villa, M. P., & Restrepo Perez, C. (2017). Estudio de la oferta y la demanda de productos de mascotas relacionados con la alimentación y los accesorios en la ciudad de Medellín para determinar oportunidades de innovación y desarrollo de nuevos productos. *Universidad EAFIT*, 91. Obtenido de [https://repository.eafit.edu.co/bitstream/handle/10784/12723/MariaPaulina\\_ArboledaVilla\\_Camilo\\_RestrepoPerez\\_2017.pdf?sequence=2&isAllowed=y](https://repository.eafit.edu.co/bitstream/handle/10784/12723/MariaPaulina_ArboledaVilla_Camilo_RestrepoPerez_2017.pdf?sequence=2&isAllowed=y)
- Arduino. (14 de 12 de 2018). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/en/Main/Software>
- Arduino. (14 de 12 de 2018). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/en/Reference/Servo>
- Autodesk. (07 de 04 de 2019). *Autodesk 123D*. Obtenido de Autodesk 123D: <https://latinoamerica.autodesk.com/solutions/123d-apps>
- Bouraqaadi, N., & Mason, D. (2018). Test-driven development for generated portable Javascript apps. *Science of Computer Programming*, 2-17. doi:<https://doi.org/10.1016/j.scico.2018.02.003>
- Cherradi, G., El Bouziri, A., Boulmakoul, A., & Zeitouni, K. (2017). Real-Time HazMat Environmental Information System: A micro-service based architecture. *Procedia Computer Science*, 6. doi:<https://doi.org/10.1016/j.procs.2017.05.457>
- Código Nacional de Policía y Convivencia. (29 de 01 de 2017). Diario Oficial No. 49.949. Bogotá, Colombia: Congreso de la Republica. Recuperado el 11 de 05 de 2019, de [http://www.secretariassenado.gov.co/senado/basedoc/ley\\_1801\\_2016.html](http://www.secretariassenado.gov.co/senado/basedoc/ley_1801_2016.html)
- Estevez, P. (2005). *Manual basico para la educacion del cachorro*.
- fritzing. (2 de 4 de 2019). *fritzing*. Obtenido de fritzing: <http://fritzing.org>
- Genymotion. (12 de 12 de 2018). *Genymotion*. Obtenido de Genymotion: <https://www.genymotion.com/>

- He, G., Xu, B., Zhang, L., & Zhu, H. (2018). Mobile app identification for encrypted network flows by traffic correlation. *International Journal of Distributed Sensor Networks*. doi:10.1177/1550147718817292
- Infinite Red, Inc. (12 de 12 de 2018). *Infinite Red, Inc.* Obtenido de Infinite Red, Inc.: <https://infinite.red/>
- Javascript. (25 de 03 de 2019). *Javascript*. Obtenido de Javascript: <https://www.javascript.com/>
- León Quenguan, J. D., & Rueda Almario, D. (21 de 03 de 2013). *Universidad del Valle*. Recuperado el 11 de 05 de 2019, de Universidad del Valle: <http://bibliotecadigital.univalle.edu.co/bitstream/10893/9148/1/CB-0527751.pdf>
- Lv, H., & Wang, S. (2016). Design and Application of IoT Microservices Based on Seneca. *COMPUTER SCIENCE and ENGINEERING*, 4. Obtenido de <http://www.dpi-proceedings.com/index.php/dtce/article/view/4814/4446>
- Martín, A., Chávez, S., Murazzo, M., & Rodríguez, N. (8 de 5 de 2015). *MongoDB en ambiente Cloud Híbrido con OpenStack*. Obtenido de Repositorio Universidad Nacional de la Plata: <http://sedici.unlp.edu.ar/handle/10915/45569?show=full>
- Martínez, C. (2015). Dispensador de alimento para perros dosificable a través de dispositivos móviles Android. (*Tesis Pregrado*). Instituto Politécnico Nacional Escuela Superior de Computo, Ciudad de México, México. Obtenido de <http://tesis.ipn.mx/handle/123456789/22493>
- Mega Practical. (03 de 10 de 2018). *Mega Practical*. Obtenido de [https://cdn2.hubspot.net/hubfs/371274/Desarrollo de Software/metodologias de desarrollo de software.pdf](https://cdn2.hubspot.net/hubfs/371274/Desarrollo%20de%20Software/metodologias%20de%20desarrollo%20de%20software.pdf)
- Microsoft. (01 de 01 de 2019). *Microsoft*. Recuperado el 02 de 03 de 2019, de Microsoft: <https://code.visualstudio.com/docs>
- Nandaa, A. (2018). *Beginning API Development with Node.js: Build Highly Scalable, Developer-Friendly APIs for the Modern Web with JavaScript and Node.js*. Birmingham, Inglaterra: Packt Publishing.
- Natsys. (2017). *Introducción a UML: Lenguaje para modelar objetos*. Natsys. Obtenido de <https://books.google.com.co/books?id=vrvbDgAAQBAJ>
- Pressman, R. (2010). *Ingeniería de software un enfoque práctico*. Mexico: Mc Graw Hill.
- Rodríguez, N. G. (7 de 4 de 2019). Obtenido de [https://e-archivo.uc3m.es/bitstream/handle/10016/25775/PFC\\_Nuria\\_Gomez\\_Rodriguez.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/25775/PFC_Nuria_Gomez_Rodriguez.pdf)
- Rodríguez, N. G. (07 de 04 de 2019). *Las pruebas de integración como proceso de la calidad del software en el ámbito de las telecomunicaciones*. Obtenido de Las pruebas de integración como proceso de la calidad del software en el ámbito de las telecomunicaciones: [https://e-archivo.uc3m.es/bitstream/handle/10016/25775/PFC\\_Nuria\\_Gomez\\_Rodriguez.pdf](https://e-archivo.uc3m.es/bitstream/handle/10016/25775/PFC_Nuria_Gomez_Rodriguez.pdf)
- Sajee Mathew. (1 de Abril de 2017). *Amazon Web Services*. Recuperado el 4 de abril de 2019, de Amazon Web Services: Documento técnico de AWS: [https://d1.awsstatic.com/whitepapers/es\\_ES/aws-overview.pdf](https://d1.awsstatic.com/whitepapers/es_ES/aws-overview.pdf)
- Suárez Mora, D. R., Aparicio Gómez, A., Gallego Ibáñez, Y. K., & Ramírez Salcedo, J. C. (12 de 07 de 2015). Integración de los sistemas embebidos Raspberry Pi y Arduino para el manejo de un brazo robótico mediante una aplicación Android. *Revista Epsilon*, 69-96. doi:<http://usc.elogim.com:2048/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=118576958&lang=es&site=eds-live>
- Tiempo, E. (31 de Agosto de 2017). *Estilos de vida*. Obtenido de El Tiempo: <https://www.eltiempo.com/vida/ciencia>
- Tojeiro Calaza, G. (2015). *Taller de Arduino un enfoque práctico para principiantes*. Mexico D.F: Alfaomega.
- Tuta, G., Rivera, J., Gutiérrez, K., & Roa, S. (22 de Octubre de 2015). *Diseño y construcción de un dosificador automático de alimento para perros de raza media*. Artículo Científico, Universidad Autónoma de Bucaramanga, Facultad de Ingeniería, Envigado. Obtenido de <https://revistas.eia.edu.co/index.php/mem/article/view/822>

Visual Studio Code. (12 de 12 de 2018). *Visual Studio Code*. Obtenido de Visual Studio Code: <https://code.visualstudio.com/>

Zapata, J., & Gil, D. (2017). Diseño e implementación de un prototipo de dispensador automático de comida para animales basado en raspberry pi controlado mediante una aplicación móvil. (*Tesis de pregrado*). Universidad Tecnológica de Pereira, Pereira, Colombia. Obtenido de <http://repositorio.utp.edu.co/dspace/handle/11059/8054>

Zoocial, L. R. (25 de Agosto de 2017). *¿Por qué una mascota mejora su vida?* Obtenido de El Espectador: <https://www.elespectador.com/noticias/actualidad/por-que-una-mascota-mejora-su-vida-articulo-709872>