



UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
ESCUELA DE INGENIERÍA CIVIL EN INFORMÁTICA

“SmaFi”

Sistema informático de monitoreo, alimentación inteligente y estadísticas dietéticas de mascotas utilizando IoT.

Memoria para optar al título de Ingeniero Civil en Informática

Alumnos: Cristóbal Alejandro Sandoval Gutiérrez
Fernando Arturo Sepúlveda Palma

Profesor guía: Luis Daniel Gajardo Díaz

Chillán, Enero del 2019

Dedicatoria

“Queremos agradecer a todas aquellas personas que nos brindaron apoyo durante el desarrollo de nuestra tesis de grado, en especial a los profesores Luis Gajardo por guiarnos y orientarnos en el camino correcto, al profesor Juan Carlos Figueroa, por darnos excelentes ideas para implementar en nuestro proyecto, además de los conocimientos necesarios para llevarlas a cabo, y al profesor Fernando Santolaya, por la gestión de los elementos requeridos y por el conocimiento que nos ha brindado. También, queremos hacer un agradecimiento especial a nuestras familias por estar siempre con nosotros y dándonos apoyo durante este periodo. Finalmente, queremos agradecer a nuestras mascotas por colaborar con nuestro proyecto.”

Resumen

El proyecto que se presenta en el actual informe consiste en la creación de un sistema que permita solucionar una problemática específica, la cual se menciona en el siguiente párrafo. Dicho sistema consiste en una aplicación web y un producto destinado a los usuarios finales.

En la sociedad actual, gran parte de la población dispone de poco tiempo durante la semana laboral, por lo cual se ven obligados a dejar a sus mascotas solas durante prolongados periodos de tiempo. Esto se traduce en que los dueños no pueden brindarles una correcta alimentación a sus animales de compañía. Esta problemática es la principal motivación para el desarrollo del proyecto.

Para solucionar la problemática previamente mencionada, se proponen una aplicación web y un producto. El producto corresponde a un alimentador inteligente, el cual calcula de manera personalizada la porción de alimento de una mascota, según su edad, peso, tamaño y si está en alguna etapa especial de su vida, como la etapa de gestación y lactancia (cuando amamantan a sus cachorros), además de brindar agua fresca cada cierto período de tiempo. En tanto, la aplicación web permite a un usuario configurar los productos y gestionar sus mascotas, proveyendo estadísticas dietéticas. Cabe mencionar que, en un futuro cercano, el sistema incluirá un collar inteligente que monitoree la actividad de una mascota durante todo el día.

En cuanto a los resultados de las pruebas, estos indican que la aplicación web cumple con los requerimientos especificados en la Tabla 4. Las pruebas unitarias, de integración y funcionales realizadas por los desarrolladores se ejecutaron exitosamente, mientras que las pruebas de aceptación fueron realizadas por un usuario aleatorio, y de igual manera, el resultado fue exitoso. Si bien, no se especificaron pruebas de hardware, el alimentador inteligente implementa una interfaz de usuario para probar las diferentes funcionalidades especificadas en la Tabla 5.

Abstract

The project covered in the current report is about the creation of a system to solve a specific problem, which is mentioned in the following paragraph. This system includes a web application and a product intended to end users.

In today's society, a significant number of people don't have or have a slight amount of time during working days, so they are forced to let their pets alone during long periods of time. This means that the owners cannot give their pets an adequate alimentation. This problem is the main motivation for developing the current project.

To solve the problem mentioned above, a web application and a product are proposed. The proposed product is a smart feeder, which calculates a portion of food according to the pet's age, weight, size and according to different moments of their lives, like pregnancy or lactancy (when a pet nurse their puppies or kittens), as well as giving fresh water periodically. On the other hand, the web application allows a user to configure its products and to manage their pets, providing dietary statistics. It is necessary to mention that in a near future, the system will include a smart a pet collar, which tracks the pet activity during all day.

As for the test results, they indicate that the web application met the specified requirements present in the Table 4. The unit, the integration and the functional tests were executed successfully, whereas the acceptance test where executed by a random user and its results were also successful. Although the hardware tests were not specified, the smart feeder has a user interface to test the different characteristics specified in the Table 5.

Índice

Contenido

CAPÍTULO 1	INTRODUCCIÓN	9
1.1.	Introducción	10
CAPÍTULO 2	ANTECEDENTES GENERALES.....	12
2.1.	Motivación	13
2.2.	Cuidados de las mascotas	13
2.3.	Descripción de la problemática	14
2.4.	Descripción de la solución	16
CAPÍTULO 3	DEFINICIÓN DEL PROYECTO.....	18
3.1.	Objetivos del proyecto	19
3.1.1.	Objetivo general.....	19
3.1.2.	Objetivos específicos	19
3.2.	Ambiente de ingeniería de software	19
3.2.1.	Metodología	19
3.2.2.	Frameworks	20
3.2.3.	Herramientas	20
3.2.4.	Lenguajes	21
3.2.5.	Software	22
CAPÍTULO 4	ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE Y HARDWARE	24
4.1.	Alcances y limitaciones.....	25
4.2.	Objetivos del software.....	26
4.2.1.	Objetivo general.....	26
4.3.	Descripción global del producto final.....	26
4.3.1.	Interfaz de usuario	26
4.3.2.	Interfaces de hardware	26
4.3.3.	Interfaces de software.....	27
4.3.4.	Interfaces de comunicación	27
4.4.	Requisitos específicos	28
4.4.1.	Requisitos funcionales del sistema	28
4.4.2.	Atributos del producto	30

CAPÍTULO 5 ANÁLISIS DE FACTIBILIDAD	31
5.1. Análisis de factibilidad técnica	32
5.1.1. Desarrollo de aplicaciones	32
5.1.2. Hardware	32
5.1.3. Conclusión del análisis de factibilidad técnica:.....	32
5.2. Análisis de factibilidad operativa	33
5.2.1. Conclusión del análisis de factibilidad operativa:	33
5.3. Análisis de factibilidad económica.....	33
5.3.1. Conclusión del análisis de factibilidad económica:.....	39
5.4. Conclusión del análisis factibilidad	39
CAPÍTULO 6 ANÁLISIS	40
6.1. Casos de uso.....	41
6.1.1. Identificación de actores	41
6.1.2. Diagrama de casos de uso	41
6.1.3. Especificación de los casos de uso.....	43
6.2. Diagrama Entidad-Relación	52
6.3. Especificación de las entidades y relaciones	54
6.3.1. Entidades:	54
6.3.2. Relaciones:	55
CAPÍTULO 7 DISEÑO	56
7.1. Diseño físico de la base de datos	57
7.2. Diseño del hardware	59
7.2.1. Diseño de circuitos.....	59
7.2.2. Diseños 2D	62
7.2.3. Diseños 3D	65
7.3. Diseño de interfaz y navegación	66
7.3.1. Landing page, registrarse e iniciar sesión	66
7.3.2. Vistas de usuario	68
7.3.3. Vistas de administrador	69
CAPÍTULO 8 PRUEBAS	70
8.1. Casos de prueba	71
8.2. Especificación de las pruebas.....	72
8.3. Resultados de las pruebas.....	73
CONCLUSIONES	75

FUTURO DEL PROYECTO	76
BIBLIOGRAFÍA.....	77
ANEXOS.....	80
Interfaces de software	81
Diseños 3D de componentes del alimentador.....	83
Construcción del prototipo del alimentador	84
Prototipo finalizado del alimentador.....	87

Índice de tablas

Tabla 1 - Interfaces de software	27
Tabla 2 - Interfaces de comunicación.....	27
Tabla 3 - Requisitos funcionales de software	29
Tabla 4 - Requisitos funcionales de hardware.....	29
Tabla 5 - Atributos del producto	30
Tabla 6 - Inversión en el alimentador	35
Tabla 7 - Inversión en el collar	36
Tabla 8 - Inversión en requisitos de la aplicación web.....	36
Tabla 9 - Inversión en RRHH e inversión total.....	36
Tabla 10 - Cálculo del VAN	38
Tabla 11 - Especificación de caso de uso 1	43
Tabla 12 - Especificación de caso de uso 2	44
Tabla 13 - Especificación de caso de uso 3	45
Tabla 14 - Especificación de caso de uso 4	45
Tabla 15 - Especificación de caso de uso 5	46
Tabla 16 - Especificación de caso de uso 6	46
Tabla 17 - Especificación de caso de uso 7	47
Tabla 18 - Especificación de caso de uso 8	47
Tabla 19 - Especificación de caso de uso 9	48
Tabla 20 - Especificación de caso de uso 10	48
Tabla 21 - Especificación de caso de uso 11	49
Tabla 22 - Especificación de caso de uso 12	49
Tabla 23 - Especificación de caso de uso 13	50
Tabla 24 - Especificación de caso de uso 14	50
Tabla 25 - Especificación de caso de uso 15	51
Tabla 26 - Especificación de caso de uso 16	51
Tabla 27 - Especificación de pruebas	73
Tabla 28 - Resultados de pruebas	74

Índice de imágenes

Ilustración 1 - Diagrama de casos de uso	42
Ilustración 2 - Diagrama Entidad-Relación	53
Ilustración 3 - Diseño físico de la base de datos	58
Ilustración 4 - Diseño del circuito del alimentador	60
Ilustración 5 - Diseño del circuito del collar	61
Ilustración 6 - Diseño la estructura del alimentador pt. 1	62
Ilustración 7 - Diseño la estructura del alimentador pt. 2	63
Ilustración 8 - Diseño la estructura del alimentador pt. 3	64
Ilustración 9 - Diseño conceptual del alimentador	65
Ilustración 10 - Diseño de la vista “landing page”	66
Ilustración 11 - Diseño de la vista “registro”	67
Ilustración 12 - Diseño de la vista “inicio de sesión”	67
Ilustración 13 - Diseño de la vista “mis mascotas”	68
Ilustración 14 - Diseño de la vista “mi alimentador”	68
Ilustración 15 - Diseño de la vista de administrador “usuarios”	69
Ilustración 16 - Diseño de la vista de administrador “regiones”	69

CAPÍTULO 1

INTRODUCCIÓN

1.1. Introducción

Antes de adentrarse en lo que respecta al proyecto, es necesario poner en contexto al lector, refiriéndose de manera sencilla lo que éste abarca.

En primer lugar, se referirá al internet de las cosas, más conocido como IoT (Internet Of Things, por sus siglas en inglés). De acuerdo con la empresa especializada en el área de domótica e IoT Domodesk, la IoT es un concepto que hace alusión a la interconexión digital de objetos diarios con la internet. Este concepto toma cada vez más fuerza en la vida diaria de las personas, logrando encontrarse con dispositivos que antes no eran capaces de interactuar con el entorno y las personas. Dicha interacción permite el control e interacción de dichos dispositivos de manera remota, como el manejar un televisor a la distancia con un Smartphone, el controlar las luces de la casa con una aplicación móvil o que una alarma avise al teléfono que algo ha ocurrido en el hogar (Domodesk, 2014). Dicho lo anterior, se puede afirmar que el principal objetivo de la IoT es hacer más cómoda la vida de las personas, así como proporcionar una mayor seguridad en diversos ámbitos.

Luego, es necesario referirse a la sociedad actual en relación con los animales de compañía o mascotas. La periodista Carmen Aniorte del periódico ABC de España, indica que “En la actualidad, las mascotas corresponden a una parte fundamental de la sociedad, y ellas traen consigo diferentes tipos de beneficios, como los son los beneficios emocionales, psicológicos y físicos. Dichos beneficios tienen múltiples evidencias científicas” (Aniorte, 2016). Por otro lado, el mismo periódico indica que “el incremento de la población en las zonas urbanas, las extensas jornadas de trabajo, la crisis económica y sobre todo la falta de reconocimiento del papel positivo de los animales de compañía sobre las personas, constituyen importantes barreras para la tenencia de mascotas en el futuro, ya que resulta difícil combinar sus cuidados y atenciones con las exigencias de la vida moderna” (Aniorte, Mascotas, una parte fundamental de la sociedad, 2017).

Además, en la actualidad en Chile está vigente una ley (ley número 21.020) que vela por la tenencia responsable de mascotas. Esta ley abarca un conjunto de obligaciones que contrae una persona cuando decide aceptar y mantener una mascota, y que consisten -entre otras- proporcionarle alimento, albergue y buen trato, brindarle los cuidados veterinarios y no someterlo en sufrimientos, además de respetar las normas de salud y seguridad pública.

Como se puede notar en el párrafo anterior, las mascotas forman un pilar fundamental dentro de la sociedad, sin embargo, no siempre es posible otorgarles el mejor de los cuidados, debido a la forma de vida de la sociedad actual.

Situado ya el contexto que abarca el proyecto, se puede proceder con su descripción. Este proyecto consiste en un sistema de IoT que busca brindarle una mejor calidad de vida a las mascotas, mediante el uso de un alimentador que calcula una dieta acorde a las características de cada mascota, además, de incluir un collar que permite medir el nivel de actividad de estas durante el día. Además, la ventaja de que este proyecto pertenezca a la creciente área de la IoT, es que los dueños pueden configurar el alimentador de manera remota, y pueden llevar control acerca de los eventos que ocurren con el alimentador, como lo son el que el alimento fue servido y el que se hizo reposición del agua.

El presente proyecto se organiza de la siguiente manera, en el capítulo 1 se dará una breve introducción a dicho proyecto, en el capítulo 2 se darán a conocer los antecedentes generales que motivaron la realización de este proyecto, en el capítulo 3 se definirán los objetivos que se desean alcanzar con la realización del proyecto y el ambiente en el cual se desarrollará, en el capítulo 4 se especifican los alcances, objetivos del software y los requisitos de este, en el capítulo 5 se muestran los estudios de factibilidad realizados y una conclusión general de estos, en el capítulo 6 se analizan los modelos del sistema, en el capítulo 7 se ilustran los diseños del software y hardware, en capítulo 8 se especificaran las pruebas realizadas al sistema y los resultados obtenidos, en capítulo 9 se darán a conocer las conclusiones del proyecto, en el capítulo 10 se indicaran las actualizaciones a futuro del proyecto, finalmente en el capítulo 11 se indicara la bibliografía del presente proyecto.

CAPÍTULO 2

ANTECEDENTES GENERALES

2.1. Motivación

La principal motivación que originó la idea de realizar este proyecto, es el amor que los dueños sienten por sus mascotas (incluyendo a los autores de este documento), y el querer brindarles a sus dueños una solución para que estos sean capaces de otorgar una alimentación correcta y acorde a los rasgos físicos de la mascota (como el peso y la estatura).

2.2. Cuidados de las mascotas

En la actualidad las mascotas se han vuelto un agente importante en la vida de las personas, brindando compañía, alegrías y amor, llegando a ser consideradas “un miembro más de la familia”, por ende, es imprescindible brindarle los cuidados que necesitan.

A continuación, se mencionan algunos de los cuidados generales que se deben tener con las mascotas, acorde con el artículo de divulgación de Elias Arcas (Arcas, s.f.).

Actividad: Actividad física es igual a salud. Los perros necesitan salir a pasear. Mientras más grande sea el perro más tiempo se le debe dedicar a la actividad física. Por supuesto en casa también se le debe dedicar tiempo a jugar con él, con lo que se consigue que este realice actividad física.

Dieta: Los perros y gatos con sobrepeso son algo ya habitual. No es cuestión de matarlos de hambre, pero tampoco de llevarlos al sobrepeso.

Es importante asesorarse por veterinarios de qué pueden y no pueden comer. Hay que vigilar el nivel de proteínas, ya que al igual que les ocurre a las personas, un exceso de proteínas sobre carga los riñones y acidifica el organismo.

Hidratación: Los animales tienden a beber menos agua y la hidratación es precisamente uno de los factores más importantes para una buena salud.

Es importante vigilar los niveles de agua de sus bebederos para controlar que estén bebiendo lo adecuado.

Vitaminas y antioxidantes: Cada vez hay más personas que en lugar de darles como premio una golosina a su mascota le da un comprimido multivitamínico o antioxidante masticable, los cuales son muy beneficiosos para la salud de estas.

Amor y afecto: Este puede que sea el cuidado más importante de las mascotas, aunque muchos piensan que, simplemente, dándoles de comer y sacándoles a pasear ya hacen más que suficiente, pero están equivocados, porque los animales agradecen mucho las muestras físicas de afecto. La prueba de esto es que muchas mascotas que estaban en perreras, aunque estuvieran bien alimentados se suelen encontrar tristes, pero cuando alguien los adopta y disfruta con ellos, recuperan el entusiasmo y les cambia hasta el brillo del pelo.

Medicación: Las mascotas siguen muy de cerca los pasos de las personas en temas de salud y así cada vez van más a la consulta del veterinario, ya que van padeciendo de más dolencias. Eso implica también el tomar medicamentos químicos con sus efectos adversos incluidos.

2.3. Descripción de la problemática

La problemática consta de una breve lista de problemas identificados. Uno de los principales problemas, es que muchos dueños de mascotas se ven en la obligación de dejarlas solas en el hogar por motivos de trabajo. Además del problema previamente mencionado, se identificaron los siguientes:

- a) No llegar a la hora estimada al hogar, retrasando con ello la hora de alimentación de las mascotas.
- b) Tener un horario de trabajo completo (ej. De 8:00 hrs. a 18:00 hrs.) con un pequeño tiempo para almorzar, por lo cual muchas personas no alcanzan a viajar a su hogar para alimenta a su mascota.
- c) Salir por algunos días fuera de su hogar (ej. fin de semana), sin la posibilidad de dejarlas con alguien por aquellos días o de poder llevarlas consigo.
- d) No saber cuánto alimento darles a sus mascotas, acorde a su tamaño y su peso.
- e) No considerar que el ejercicio que realiza una mascota influye en su dieta.
- f) No almacenar el alimento de manera apropiada.
- g) No brindarles agua fresca periódicamente.

Además, se identificó un problema bastante grave, que deriva de las personas que poseen una agenda sumamente ocupada y que tienden a dejar solas a sus mascotas durante el día, cometiendo el grave error de servirles abundante alimento y agua a sus mascotas, creyendo que con eso éstas saciaran su hambre y sed durante el tiempo que no estén en el hogar. Sin embargo, esta es una práctica peligrosa y no recomendada, dado que la mayoría de las mascotas, en especial los canes, tratarán de comer la mayor cantidad de alimento posible cada vez que vean que su plato con alimento, lo cual puede traer serias consecuencias para su salud. Dicho lo anterior, Anna Burke del American Kennel Club (AKC) recomienda que los perros coman entre 3 y 4 raciones al día cuando son cachorros y 1 o 2 raciones al día cuando son adultos (Burke, 2018). Además, el AKC indica que las hembras en etapa de gestación deben comer 3 raciones al día, mientras que, durante la etapa de lactancia se debe brindar tanto alimento y agua como la perra requiera. Estos aumentos en la alimentación se deben al mayor requerimiento nutricional que se presentan durante estas etapas (AKC, 2013).

2.4. Descripción de la solución

La solución abarca dos de los cuidados más importantes de las mascotas, los cuales fueron mencionados en el punto 2.3. Estos corresponden a la “dieta” e “hidratación”. Sin embargo, esta estará disponible solo para perros de manera inicial.

A grandes rasgos, la solución que se quiere otorgar es proporcionar un sistema que sea capaz de brindar una alimentación e hidratación acorde a los especificado por el AKC en el punto 2.3, además de llevar un control de la nutrición y actividad física de la mascota.

La solución consiste en un sistema de IoT con el cual se busca mejorar la calidad de vidas de las mascotas. Para ello se creó un alimentador inteligente que calcula una dieta personalizada para la mascota, es decir, se calcula una dieta acorde a sus características, considerando los factores de peso, peso ideal, estatura, edad, nivel de actividad¹, cantidad de alimento requerido según edad y tamaño de raza y si se está en etapa de gestación o lactancia en el caso de las hembras. El dispositivo posee no solo la capacidad de brindar alimento, sino que también puede brindar agua fresca cada cierto periodo de tiempo, y en caso de que el agua se acabe antes de que se cumpla dicho período, este servirá inmediatamente, asegurándose así de que haya siempre agua disponible. Además, se consideró el que los sonidos que produce el alimentador pueden molestar durante la noche, por lo cual se implementó un modo automático, que evita que se sirva y bombee agua durante estas horas. Este dispositivo al requerir que sea conectado a internet, da la posibilidad de acceder a este por un único usuario a través del sitio web oficial, permitiendo así que sea posible de configurar desde cualquier lugar en que se encuentre el propietario. Cabe mencionar que una vez configurado el alimentador, este servirá comida y agua a la mascota automáticamente en los horarios establecidos.

También se estudia la opción de incluir un collar inteligente, el cual permitiría medir el nivel de actividad de la mascota durante todo el día y enviarlo al servidor.

¹ Este factor está incluido en el algoritmo de cálculo, pero su uso está sujeto a la inclusión del collar en el sistema.

Estos dispositivos se conectan con un sitio web, el cual permite al usuario configurarlos luego de ser agregados al perfil del propietario. El sitio web también permite agregar, modificar y eliminar mascotas de la plataforma, añadir, configurar y eliminar alimentadores de la plataforma y ver la información nutricional y actividad² de sus mascotas.

² Este ítem depende de la inclusión del collar en el sistema.

CAPÍTULO 3

DEFINICIÓN DEL PROYECTO

3.1. Objetivos del proyecto

3.1.1. Objetivo general

Implementar un sistema que proporcione a los propietarios de mascotas un sistema domótico de alimentación e hidratación, velando por dietas acorde a las características de cada mascota.

3.1.2. Objetivos específicos

- A. Entregar un mejor cuidado a las mascotas, respecto de la correcta alimentación e hidratación que estas deben tener, calculando su dieta a diario.
- B. Facilitar el cuidado de mascotas por medio de la tecnología (IoT).
- C. Proporcionar un registro de eventos, respecto al estado del alimentador, como el que se sirvió alimento y que la mascota se comió todo el alimento.
- D. Proporcionar un sistema de alimentación que sea capaz de calcular la dieta de la mascota en base a diferentes variables.

3.2. Ambiente de ingeniería de software

3.2.1. Metodología

- **Scrumban:** Es una metodología ágil, la cual nace de la combinación entre Scrum y Kanban, la cual reúne la naturaleza preceptiva de Scrum y la capacidad de mejora del proceso de Kanban. Esta es la metodología que utiliza para el desarrollo del proyecto (Pahuja, s.f.).

3.2.2. Frameworks

- **Bootstrap:** es un framework que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Se utiliza en el proyecto para brindar parte del diseño la interfaz de la aplicación web, además de dotar de responsividad a dicha aplicación (Bootstrap, s.f.).
- **Laravel:** Es un framework de código abierto simple, sencillo y potente para desarrollar aplicaciones y servicios en lenguaje PHP. Se utiliza en el proyecto para programar la aplicación web en PHP (Laravel, s.f.).

3.2.3. Herramientas

- **Composer:** Es un gestor de paquetes a nivel de aplicación para el lenguaje de programación PHP. Proporciona un formato estándar para gestionar las dependencias del software PHP y las bibliotecas requeridas. Se utiliza en el proyecto para gestionar las dependencias y bibliotecas que se utilizan para el desarrollo de la aplicación web en PHP (Composer, s.f.).
- **GitKraken:** Es un sistema para el control de versiones para git multiplataforma. Se utiliza en el proyecto para almacenar el proyecto en el servidor de GitLab y llevar un control de versiones (Gitkraken, s.f.).
- **MySQL Workbench:** Es una herramienta que permite crear y gestionar bases de datos, además de modelar diagramas de Entidad-Relación para bases de datos MySQL. Se utiliza en el proyecto para generar el modelo físico de la base de datos (MySQL, s.f.).
- **Trello:** Es una herramienta para la organización de tareas basada en la metodología Kanban. Se utiliza en el proyecto para gestionar las tareas del proyecto (Trello, s.f.).

- **PhpMyAdmin:** PhpMyAdmin es una herramienta que se utiliza para la administración de bases de datos MySQL o MariaDB a través de páginas web utilizando Internet. Se utiliza en el proyecto para simplificar la administración de bases de datos, mediante la creación, edición, borrado, exportación e importación de tablas y bases de datos (PhpMyAdmin, s.f.).

3.2.4. Lenguajes

- **Arduino Language:** Es el lenguaje con el cual se programan las placas Arduino y similares. Este lenguaje está basado en C/C++. Se utiliza en el proyecto para programar las placas NodeMCU y Wemos D1 Mini (Arduino, s.f.).
- **CSS:** Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Se utiliza en el proyecto para dar el estilo y diseño a la aplicación web (W3schools, s.f.).
- **HTML:** Es un lenguaje de marcas de hipertexto, hace referencia al lenguaje de marcado para la elaboración de páginas web. Se utiliza en el proyecto para elaborar la página web (W3Schools, s.f.).
- **JavaScript:** Es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas (jQuery por ejemplo). Se utiliza en el proyecto para agregar funciones complejas a la aplicación web (W3schools, s.f.).
- **PHP:** Es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Se utiliza en el proyecto para programar el Backend de la aplicación web (PHP, s.f.).

- **SQL:** Es un lenguaje específico del dominio utilizado, diseñado para administrar sistemas de gestión de bases de datos relacionales. Se utiliza en el proyecto para realizar consultas a la base de datos (W3schools, s.f.).

3.2.5. Software

- **Arduino IDE:** Es un entorno de desarrollo de código abierto multiplataforma, con el cual se pueden programar microcontroladores como Arduino y placas basadas en ESP8266. Se utiliza en el proyecto para programar las placas NodeMCU y Wemos D1 Mini con el lenguaje de Arduino (Arduino, s.f.).
- **Atom:** Es un editor de texto de código multiplataforma desarrollado por GitHub. Atom cuenta con soporte para plug-ins escritos en Node.js y control de versiones Git integrado. Se utiliza en el proyecto para escribir las páginas web en HTML, las hojas de estilo CSS y para programar el Backend en PHP y JavaScript (Atom, s.f.).
- **Adobe Lightroom:** Es un programa de fotografía digital, diseñado para ayudar a fotógrafos profesionales y aficionados en el tratamiento de imágenes digitales y trabajos de post-producción. Se utiliza en el proyecto para tratar las imágenes utilizadas en la aplicación web (Adobe, s.f.).
- **Adobe Photoshop:** Es un editor de gráficos rasterizados, utilizado principalmente para el retoque de fotografías y gráficos. Se utiliza en el proyecto para el diseño y retoque de las imágenes a utilizadas en la aplicación web (Adobe, s.f.).
- **Adobe XD:** Es una aplicación de software de diseño de experiencia de usuario, compatible con el diseño vectorial y el cableado de sitios web, y la creación de prototipos interactivos de simple clic. Se utiliza en el proyecto para el prototipado de la aplicación web (Adobe, s.f.).
- **Autodesk Tinkercad:** Es una aplicación web de modelado en 3D que se caracteriza por su sencillez. Se utiliza en el proyecto para modelar el diseño del producto y con este generar un archivo capaz de ser impreso por una impresora 3D (Tinkercad, s.f.).

- **Blender:** Es un programa de modelado en 3D de código abierto y multiplataforma apoyado por varias herramientas. Se utiliza en el proyecto para modelar el diseño del producto y con este generar un archivo capaz de ser impreso por una impresora 3D (Blender, s.f.).
- **Draw.io:** Es una aplicación web que nos permite crear diagramas de todo tipo desde nuestro navegador, de manera cómoda y gratuita. Se utiliza en el proyecto para el modelado de diversos diagramas, como lo son los casos de uso y el modelo Entidad-Relación (Draw.io, s.f.).
- **Fritzing:** Es una aplicación de diseño de circuitos electrónicos de código abierto. Se utiliza en el proyecto para modelar los circuitos de los productos (Fritzing, s.f.).

CAPÍTULO 4
ESPECIFICACIÓN DE REQUERIMIENTOS DEL SOFTWARE Y
HARDWARE

4.1. Alcances y limitaciones

Dada la problemática descrita en el punto 2.2, este proyecto es una iniciativa que calza bastante bien con la sociedad actual, por lo cual destacan sus prestaciones. Dichas prestaciones y limitaciones corresponden a:

- A. Ser un alimentador inteligente y automatizado.
- B. Tener la capacidad de no solo brindar alimento, sino que también agua fresca.
- C. Brindar una dieta acorde a las características de la mascota, como el peso y la altura.
- D. Brindar una interfaz de usuario simple y atractiva en la aplicación web.
- E. Permitir configurar los horarios del alimentador, lo cual abarca raciones y horas.
- F. Calcular el nivel de actividad, gracias al uso del collar.
- G. Calcular estadísticas dietéticas de la mascota.
- H. Registrar eventos relacionados con el alimentador, como el que se sirvió alimento y que el plato ya se vació.
- I. Asignar una mascota al alimentador.
- J. Disponer de un modo “no interrumpir” que se activa automáticamente en la noche (No se realizan cambios de agua en la noche).
- K. El alimentador no puede calcular el peso de la mascota para llevar control de sus variaciones de peso.
- L. El alimentador no puede ser asignado a más de una mascota a la vez.
- M. El alimentador no puede ser utilizado con baterías.
- N. El alimentador debe utilizado con un voltaje de 12 voltios.
- O. El sistema está dirigido únicamente a perros.

4.2. Objetivos del software

4.2.1. Objetivo general

El sistema considera el manejo de la información relacionada con la dieta de las mascotas, lo cual permite brindar una alimentación correcta y personalizada a estas.

4.3. Descripción global del producto final

4.3.1. Interfaz de usuario

Se indican las siguientes características en lo que respecta a la interfaz de usuario.

- Interfaz simple y atractiva.
- Interfaz responsiva.
- Paleta de colores en tonalidades celestes y anaranjadas.

Respecto a los colores seleccionados para la interfaz de usuario, se escogieron estas tonalidades dado que son las más atractivas y llamativas a juicio de los desarrolladores (Sandoval y Sepúlveda) dentro de una lista de 10 pares.

4.3.2. Interfaces de hardware

A pesar de que el producto cuenta con diseño e implementación de hardware en los dispositivos, estos no cuentan con interfaces de hardware, las cuales permiten a un usuario interactuar directamente con la máquina mediante algún tipo de control, ya sean botones, perillas, teclas, reguladores, entre otros. Esto se debe a que los productos son manejados vía web, a través de la aplicación o a través del servidor web que los productos levantan cuando es necesario.

4.3.3. Interfaces de software

En la Tabla 1 se indican las interfaces de software utilizadas por el sistema, indicando el nombre de la interfaz y su fuente. Cabe mencionar que dichas interfaces son utilizadas únicamente por el alimentador inteligente.

Abreviación	Nombre	Versión	Fuente
NTP Pool Project	NTP Pool Project	No aplica	https://www.ntppool.org/en/
No-IP	No-IP	No aplica	https://www.noip.com/

Tabla 1 - Interfaces de software

4.3.4. Interfaces de comunicación

En la Tabla 2 se indican las diferentes interfaces de comunicación utilizadas por el sistema, indicando el protocolo utilizado, el dispositivo que lo utiliza y una breve descripción del uso que se le da a la interfaz.

Protocolo	Aplicación / Dispositivo	Descripción
HTTP	Alimentador	Utiliza el protocolo HTTP para enviar y recibir la información desde la aplicación web hacia el alimentador.
	Collar	Utiliza el protocolo HTTP para enviar y recibir la información desde la aplicación web hacia el collar.
HTTPS	Aplicación web	Utiliza el protocolo HTTPS para establecer una conexión entre el cliente y el servidor.

Tabla 2 - Interfaces de comunicación

4.4. Requisitos específicos

4.4.1. Requisitos funcionales del sistema

En las Tablas 3 y 4 se indican los requisitos funcionales tanto del software como del hardware, indicando el criterio (si es software of hardware), una ID, el nombre y descripción para cada requerimiento.

Criterio	ID	Nombre	Descripción
SW	RF_1	Registrarse	El usuario debe ser capaz de registrarse en la aplicación web.
	RF_2	Iniciar sesión	El usuario debe ser capaz de iniciar sesión en la aplicación web.
	RF_3	Cerrar sesión	El usuario debe ser capaz de cerrar sesión en la aplicación web.
	RF_4	Recuperar contraseña	El usuario debe ser capaz de solicitar una recuperación de contraseña.
	RF_5	Agregar mascota	El usuario debe ser capaz de agregar una mascota en la aplicación web.
	RF_6	Editar mascota	El usuario debe ser capaz de modificar los datos de una mascota en la aplicación web.
	RF_7	Eliminar mascota	El usuario debe ser capaz de eliminar una mascota en la aplicación web.
	RF_8	Listar mascotas	El usuario debe ser capaz de ver todas sus mascotas en la aplicación web.
	RF_9	Mostrar estadísticas dietéticas de una mascota	El usuario debe ser capaz de ver las estadísticas dietéticas de sus mascotas en la aplicación web.
	RF_10	Agregar horario	El usuario debe ser capaz de agregar un horario en la aplicación web.
	RF_11	Editar horario	El usuario debe ser capaz de editar un horario en la aplicación web.
	RF_12	Eliminar horario	El usuario debe ser capaz de eliminar un horario en la aplicación web.
	RF_13	Configurar el alimentador	El usuario debe ser capaz de configurar el alimentador a través de la aplicación web.

	RF_14	Asignar mascota al alimentador	El usuario debe ser capaz de asignar una mascota alimentador desde la configuración y a través de la aplicación web.
	RF_15	Configurar el collar	El usuario debe ser capaz de configurar el collar a través de una aplicación web exclusiva para el este.
	RF_16	Asignar una mascota al collar	El usuario debe ser capaz de asignar una mascota al collar a través de una aplicación web exclusiva para este.

Tabla 3 - Requisitos funcionales de software

HW	RF_17	Liberar alimento en un horario definido	Una porción de alimento será liberado cuando se llegue a la hora indicada.
	RF_18	Liberar agua luego de un periodo determinado	El agua será servida cuanto transcurra una determinada cantidad de tiempo y una vez que el agua anterior haya sido bombeada.
	RF_19	Pesar alimento	El alimento será pesado mientras se sirve, hasta cuando este alcance la porción indicada.
	RF_20	Reponer agua cuando se esté por acabar	Cuando el sensor detecta que el agua está llegando a niveles por debajo del establecido, se llenará con agua nuevamente.
	RF_21	Bombear agua a un estanque con agua no fresca	Cuando se requiere realizar un cambio de agua tras pasar cierta cantidad de tiempo, esta será bombeada a un estanque.
	RF_22	Almacenar datos en la EEPROM	Al momento de configurar el alimentador, la información indicada en la aplicación web será enviada al alimentador.
	RF_23	Leer datos de desde la EEPROM	Cuando se inicia el alimentador, este carga la información almacenada desde la EEPROM.
	RF_24	Enviar datos al servidor	Los datos capturados con el alimentador serán enviados al servidor.

Tabla 4 - Requisitos funcionales de hardware

4.4.2. Atributos del producto

En la Tablas 5 se indican los atributos que aseguran la calidad del producto, indicando además el factor y una breve descripción de lo que el usuario y el sistema son capaces.

Atributo	Factor	Descripción
Funcionalidad	Exactitud	El sistema debe proporcionar información y estadísticas aproximadas al entero más cercano en el caso de números y una aproximación a la centésima más cercana en el caso de porcentajes.
	Seguridad	El sistema debe tener un control de acceso a través de una vista de login, y toda la información enviada al servidor debe ser enviada de manera segura.
Usabilidad	Aprendizaje	El usuario debe ser capaz de aprender la utilización del sistema en un tiempo inferior a 30 minutos.
	Comprensión	El usuario debe ser capaz de comprender el funcionamiento del sistema en un tiempo inferior a 2 horas.
	Operabilidad	El sistema debe facilitar su operación al usuario.
Eficiencia	Tiempo de respuesta	El sistema debe garantizar tiempos de respuesta inferiores a 1 segundo cuando se comunica con el servidor, el alimentador.
Mantenibilidad	Estabilidad	El sistema debe ser robusto, evitando así el mal funcionamiento de sí mismo debido a condiciones inesperadas.
Portabilidad	Adaptabilidad	El sistema debe ser capaz de adaptarse a diferentes entornos, sin requerir de configuración adicional.
	Instalación	La instalación y configuración del sistema debe tardar un tiempo inferior a 1 hora.

Tabla 5 - Atributos del producto

CAPÍTULO 5

ANÁLISIS DE FACTIBILIDAD

5.1. Análisis de factibilidad técnica

5.1.1. Desarrollo de aplicaciones

Para el desarrollo de la aplicación web se opta por el lenguaje PHP, utilizando el framework Laravel. Para la parte FrontEnd se utilizan los lenguajes HTML, CSS y JavaScript. Además, se cuenta con una base de datos implementada utilizando MySQL. Lo anterior es programado utilizando un editor de texto, en este caso Atom, y para el manejo de la base de datos, se utiliza PhpMyadmin. Cabe mencionar que la aplicación web es hospedada en la nube.

En cuanto a la aplicación que está contenida en las placas de desarrollo, se utiliza el lenguaje Arduino y HTML, lo cual se logra utilizando la IDE de Arduino.

5.1.2. Hardware

Para la construcción del dispositivo de domótica, se opta por el uso del de la placa de desarrollo NodeMCU V3 (basado en ESP8266), la cual proporciona todas las funcionalidades necesarias para la conexión a internet y lectura de datos de los sensores. En cuanto al collar, se opta por utilizar la placa Wemos D1 Mini, la cual igualmente está basada en el chip ESP8266 y posee las mismas funcionalidades que la anterior, sin embargo, esta es de menor tamaño, lo cual lo hace ideal para espacios reducidos. Cabe mencionar que todos los sensores y actuadores considerados en el proyecto son compatibles con ambas placas.

5.1.3. Conclusión del análisis de factibilidad técnica:

Dado que las herramientas que se utilizan para el desarrollo de la aplicación y para la programación de las placas son gratuitas, se puede asegurar que técnicamente el desarrollo de la aplicación es factible.

5.2. Análisis de factibilidad operativa

La aplicación y el dispositivo de alimentación de mascotas en conjunto con el collar están orientados a cualquier persona que posea mascotas y que cuente con conexión Wi-Fi. Idealmente, se recomienda que se cuente con un dispositivo móvil Android o iOS (dado que están dotados de pantallas de mayor resolución), para sacar el máximo provecho del conjunto. Además, se cuenta con un manual de usuario digital junto a la aplicación web, en el cual se explica en sencillos pasos cómo conectar los dispositivos a la red Wi-Fi del hogar del usuario y los pasos requeridos para que este dispositivo sea manipulado desde fuera de la propiedad. Por ende, no se espera un mayor obstáculo en la incorporación y uso de este servicio en los hogares de los clientes.

5.2.1. Conclusión del análisis de factibilidad operativa:

Desde el punto de vista operativo, la aplicación en conjunto con el dispositivo de alimentación de mascotas y el collar es totalmente factible, ya que no se requiere de grandes conocimientos para su configuración y uso. Además, se cuenta con un manual de usuario digital que indica los pasos a seguir para configurar el alimentador.

5.3. Análisis de factibilidad económica

En las Tablas 6, 7, 8 y 9 se indican las inversiones realizadas en el sistema bajo distintas categorías.

Alimentador			
Inversión en elementos de construcción			
Elemento	Cantidad	Precio	
Panel de madera ranurada MDF (plancha)	1	\$12.990	
Cola fría de madera	1	\$1.590	
Tornillos (pack)	1	\$2.890	
Botella plástica de 5 litros	1	\$600	

Lija para madera	1	\$130	
Estaño fino para soldar circuitos (1m)	1	\$150	
Adhesivo instantáneo líquido	1	\$1.900	
Adhesivo para PVC	1	\$650	
Tubo de PVC de 1/2"	1	\$260	
Salida de estanque de PVC de 1/2"	1	\$890	
Uniones de PVC de 1/2"	2	\$520	
Cinta aislante	1	\$610	
Tornillos para módulo HX711	4	\$200	
Tuercas para módulo HX711	8	\$400	
Precio total		\$23.780	
Inversión en hardware			
Elemento	Cantidad	Precio	
NodeMCU + Shield de expansión (pack)	1	\$8.990	
Diodo led bicolor de ánodo común	2	\$300	
Bomba de agua de [12v]	1	\$8.990	
Válvula solenoide de 1/2" [12v]	1	\$8.990	
Mini protoboard	1	\$2.500	
Resistencias de 220 Ω	2	\$80	
Diodo rectificador 1N4148	3	\$120	
Servomotor de rotación continua [5v]	1	\$9.590	
Sensor de peso + Transmisor HX711 [5v] (pack)	1	\$5.990	
Sensor de nivel de agua + Transmisor XKC-Y25 [5-24v] (pack)	1	\$10.990	
Cables Jumper (pack)	2	\$4.000	
Relé de 4 canales	1	\$3.800	
Jack DC macho + Jack DC hembra (pack)	1	\$500	
Cable cordón [1m]	1	\$490	
Precio total		\$65.330	
Inversión en impresiones 3D			

Elemento	Cantidad	Precio
Tornillo sin fin	1	\$4.966
Tubería	1	\$7.653
Platos	2	\$27.407
Logo	1	\$2.295
Precio total		\$42.320
Precio final del alimentador		\$131.430

Tabla 6 - Inversión en el alimentador

Collar		
Inversión en hardware		
Elemento	Cantidad	Precio
Giroscopio MPU6050	1	\$3.900
Mini reloj RTC	1	\$3.600
Wemos D1 Mini	1	\$6.890
Resistencia 220 kΩ	1	\$50
Batería LiPo de 3.7v y 600mAh	1	\$4.500
Módulo cargador de baterías lipo TP4056	1	\$1.500
Precio total		\$20.440
Inversión en impresiones 3D y collar		
Elemento	Cantidad	Precio
Cuerpo del montaje del collar (mitad)	2	\$4.872
Collar para perro	1	\$6.500
Precio total		\$11.372
Precio final del collar		\$31.812

Tabla 7 - Inversión en el collar

Requisitos de la aplicación web		
Inversión		
Elemento	Cantidad	Precio
Dominio (Anual)	1	\$9.950
Servidor (Anual)	1	\$20.000
Precio final de requisitos de la aplicación web		\$29.950

Tabla 8 - Inversión en requisitos de la aplicación web

Recursos humanos		
Inversión en RRHH		
Precio del analista y programador (Hora):	\$4.500	
Cantidad de programadores:	2	
Elemento	Horas	Precio
HH en análisis y diseño de la solución	168	\$1.512.000
HH en diseño y modelado 2D y 3D	72	\$648.000
HH en desarrollo y construcción	420	\$3.780.000
HH en instalación	1	\$9.000
Precio final de la aplicación web		\$5.949.000
Inversión total		\$6.142.192

Tabla 9 - Inversión en RRHH e inversión total

Para calcular el precio de la impresión en 3D, se utilizó un algoritmo que considera diferentes variables, para así obtener un precio de mercado realista.

- **PI:** Precio de la impresora 3D.
- **VU:** Vida útil de la impresora 3D.
- **TI:** Tiempo de impresión.
- **PF:** Precio del filamento.
- **GF:** Gramos de filamento total.
- **FU:** Gramos de filamento utilizado.
- **PKW:** Precio de kilovatio por hora.
- **KWH:** Kilovatio por hora que consume la impresora.

Gracias a estas variables, es posible determinar las fórmulas para calcular el precio de impresión, donde se realizarán además algunos supuestos.

- **Supuesto 1:** Dado que calcular el precio de los artículos extra es bastante complejo, se asumirá que el monto equivale a \$100, lo cual cubre los gastos.
- **Supuesto 2:** Se asumirá que la impresora tiene una vida útil de 5 años (valor comúnmente utilizado para realizar estimaciones), considerando que la impresora se utiliza 2 horas diarias. Esto brinda un total de 3650 horas de vida útil.
- **Uso de material** = $PF * FU / GF$.
- **Depreciación** = $PI * TI / VU$.
- **Consumo eléctrico** = $KWH * PKW$.
- **Precio de impresión** = Uso de material + Depreciación + Consumo Eléctrico + Gastos extra.

En la Tabla 10 se indica el cálculo del VAN utilizando los criterios definidos en el siguiente párrafo.

Para el cálculo del VAN se estimó una venta creciente a lo largo de 5 años, en la cual se estima que en el primer año se vendan 6 dispositivos, para los siguientes años se estima que las ventas crezcan, llegando a las 10 ventas en el segundo año, 15 ventas en el tercer año, 20 ventas en el cuarto año y 24 para el quinto año. Cabe destacar que el precio de un dispositivo \$310.990 (incluye costo de la aplicación), mientras que el costo de construcción de un dispositivo es de \$163.242.

	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Ventas		6	10	15	20	24
Ingresos anuales		\$1.865.940	\$3.109.900	\$4.664.850	\$6.219.800	\$7.463.760
Costos anuales		\$1.009.401	\$1.662.368	\$2.478.577	\$3.294.786	\$3.947.754
Total	\$0	\$856.539	\$1.447.532	\$2.186.273	\$2.925.014	\$3.516.006
Inversión						
Dominio y servidor	\$29.950					
Desarrollo de la aplicación y producto	\$5.949.000					
Total	-\$5.978.950	\$856.539	\$1.447.532	\$2.186.273	\$2.925.014	\$3.516.006
Tasa de descuento anual	0,15					
VAN	\$718.381					

Tabla 10 - Cálculo del VAN

5.3.1. Conclusión del análisis de factibilidad económica:

Debido a que se cuenta con el computador y los implementos para la construcción, los costos anuales de la aplicación y el dispositivo para la alimentación de mascotas se centran en la construcción de cada dispositivo, servidor y mantenimiento de la aplicación. Los costos anteriormente nombrados van incluidos en el ingreso total por cada unidad vendida más un margen de utilidad. Además, el VAN indica que el proyecto es económicamente factible, ya que el cálculo del éste arrojó un valor mayor a 0.

5.4. Conclusión del análisis factibilidad

Tanto los estudios de factibilidad técnica, económica y operativa demuestran que el proyecto es completamente factible. El costo de la aplicación (desarrollo y mantenimiento) se encuentra incluido en cada unidad que se venda, lo que hace que se recupere con mayor facilidad el costo de inversión de ésta.

CAPÍTULO 6

ANÁLISIS

6.1. Casos de uso

Un caso de uso es un artefacto que define una secuencia de acciones mediante la interacción entre uno o más actores y varios eventos, lo cual produce un resultado observable. Los casos de uso proveen una estructura para expresar los requerimientos funcionales dentro del contexto de los procesos de negocio y de sistema. Los casos de uso pueden ser representados como un elemento gráfico en un diagrama y como una especificación de caso de uso en forma de texto (IBM, s.f.).

6.1.1. Identificación de actores

- **Usuario:** Corresponde a aquel actor que hará uso del sistema.

6.1.2. Diagrama de casos de uso

En la Ilustración 1 se puede apreciar el diagrama de casos de uso, donde se representan los requerimientos funcionales del sistema (ilustración en la siguiente página).

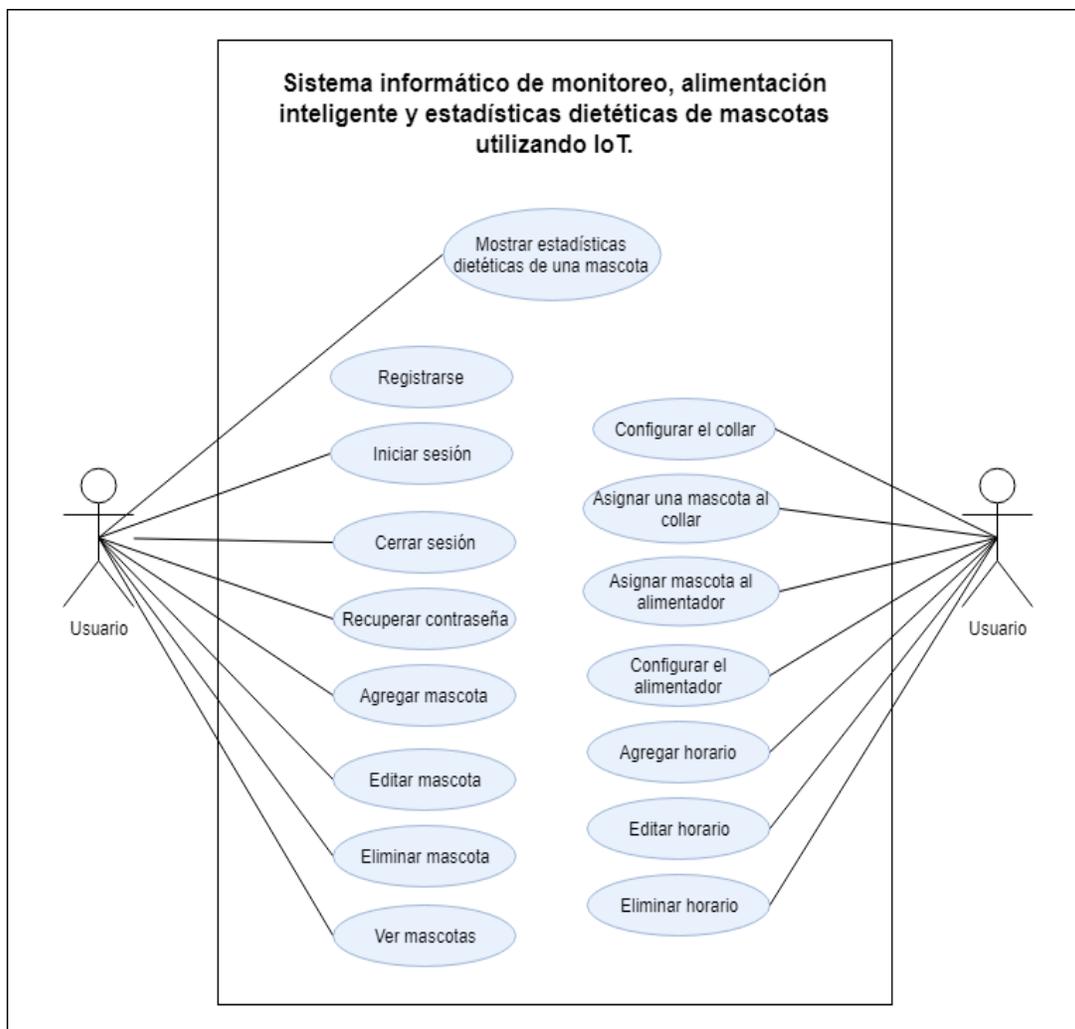


Ilustración 1 - Diagrama de casos de uso

* Notar que los requerimientos funcionales relacionados con el collar se encuentran incluidos en diagrama de casos de uso para garantizar el futuro funcionamiento de este dispositivo.

6.1.3. Especificación de los casos de uso

Desde las tablas 11 a la 26 se especifican los de casos de uso, donde se indica cómo debe comportarse el sistema en relación con un determinado usuario y evento.

Caso de uso:	CU-01	Nombre:	Registrarse
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere registrarse.		
Precondiciones:	El usuario debe estar en la pantalla de registro.		
Postcondiciones:	Registro del usuario en el sistema.		
Flujo principal:	1.- El usuario ingresa sus datos y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos. 4.- El sistema redirige al usuario a la página "mis mascotas".		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	-		

Tabla 11 - Especificación de caso de uso 1

Caso de uso:	CU-02	Nombre:	Iniciar sesión
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere iniciar sesión.		
Precondiciones:	El usuario debe estar en la pantalla de inicio de sesión.		
Postcondiciones:	Se abre la sesión del usuario.		
Flujo principal:	1.- El usuario ingresa sus datos y hace clic en aceptar. 2.- El sistema verifica los datos 3.- El sistema inicia la sesión del usuario. 4.- El sistema redirige al usuario a la página “mis mascotas”.		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	-		

Tabla 12 - Especificación de caso de uso 2

Caso de uso:	CU-03	Nombre:	Cerrar sesión
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere cerrar sesión.		
Precondiciones:	El usuario debe tener la sesión iniciada.		
Postcondiciones:	Se cierra la sesión del usuario.		
Flujo principal:	<ol style="list-style-type: none"> 1.- El usuario hace clic en su imagen de perfil ubicada en la esquina superior derecha. 2.- El sistema despliega un menú. 3.- El usuario hace clic en cerrar sesión. 4.- El sistema cierra la sesión del usuario. 		
Flujo secundario:	-		
Excepciones:	Cierre de sesión por inactividad		

Tabla 13 - Especificación de caso de uso 3

Caso de uso:	CU-04	Nombre:	Recuperar contraseña
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario solicita recuperar su contraseña.		
Precondiciones:	El usuario debe estar en la pantalla de inicio de sesión.		
Postcondiciones:	El sistema envía un correo al usuario para que pueda cambiar su contraseña.		
Flujo principal:	<ol style="list-style-type: none"> 1.- El usuario ingresa su email y hace clic en aceptar. 2.- El sistema verifica el correo. 3.- El sistema envía un correo al usuario con el link para cambiar la contraseña. 4.- El usuario ingresa al link e ingresa su nueva contraseña. 5.- El sistema actualiza en la base de datos la contraseña del usuario 		
Flujo secundario:	<ol style="list-style-type: none"> 2.1.- El sistema muestra una alerta con el error al usuario. 4.1.- El sistema muestra una alerta con el o los errores al usuario. 		
Excepciones:	Cierre de sesión por inactividad		

Tabla 14 - Especificación de caso de uso 4

Caso de uso:	CU-05	Nombre:	Agregar mascota
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere agregar una mascota.		
Precondiciones:	El usuario debe estar en la pantalla de agregar mascota.		
Postcondiciones:	Se registra la mascota del usuario.		
Flujo principal:	1.- El usuario ingresa los datos de su mascota y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos. 4.- El sistema redirige al usuario a la página “mis mascotas”.		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 15 - Especificación de caso de uso 5

Caso de uso:	CU-06	Nombre:	Editar mascota
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere editar una mascota.		
Precondiciones:	El usuario debe estar en la pantalla de editar una mascota y tener al menos una mascota registrada.		
Postcondiciones:	Se edita la información de la mascota.		
Flujo principal:	1.- El usuario edita los datos de su mascota y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos actualizados en la base de datos. 4.- El sistema redirige al usuario a la página de la mascota.		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 16 - Especificación de caso de uso 6

Caso de uso:	CU-07	Nombre:	Eliminar mascota
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere agregar eliminar una mascota.		
Precondiciones:	El usuario debe estar en la pantalla de información de la mascota.		
Postcondiciones:	Se elimina la mascota del usuario.		
Flujo principal:	1.- El usuario hace clic en eliminar. 2.- El sistema verifica la petición. 3.- El sistema oculta los datos de la mascota. 4.- El sistema redirige al usuario a la página “mis mascotas” y notifica que la mascota fue eliminada con éxito.		
Flujo secundario:	-		
Excepciones:	Cierre de sesión por inactividad		

Tabla 17 - Especificación de caso de uso 7

Caso de uso:	CU-08	Nombre:	Ver mascotas
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere ver sus mascotas.		
Precondiciones:	El usuario debe estar en la pantalla de mis mascotas.		
Postcondiciones:	Se muestran todas las mascotas del usuario.		
Flujo principal:	1.- El sistema busca todas las mascotas vigentes del usuario. 2.- El sistema carga todas las mascotas del usuario (recuadro con el nombre e imagen).		
Flujo secundario:	2.1.- Si el usuario no tiene mascotas registradas, se muestra un mensaje.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 18 - Especificación de caso de uso 8

Caso de uso:	CU-09	Nombre:	Mostrar estadísticas dietéticas de una mascota
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere visualizar las estadísticas dietéticas de una mascota.		
Precondiciones:	El usuario debe estar en la pantalla de estadísticas y debe tener al menos una mascota registrada y vigente en el sistema.		
Postcondiciones:	Se muestran gráficos con información de la mascota		
Flujo principal:	1.- El sistema carga las estadísticas de la mascota seleccionada. 2.- El sistema muestra las estadísticas de la mascota seleccionada.		
Flujo secundario:	-		
Excepciones:	Cierre de sesión por inactividad		

Tabla 19 - Especificación de caso de uso 9

Caso de uso:	CU-10	Nombre:	Asignar mascota al alimentador
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere asignar una mascota al alimentador.		
Precondiciones:	El usuario debe estar en la pantalla del alimentador a configurar y tener al menos una mascota registrada en el sistema.		
Postcondiciones:	Se registra la asignación de una mascota del usuario al alimentador.		
Flujo principal:	1.- El usuario elige una de sus mascotas, el alimento que le da y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos y los envía al alimentador. 4.- El sistema redirige al usuario a la página del alimentador.		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 20 - Especificación de caso de uso 10

Caso de uso:	CU-11	Nombre:	Asignar mascota al collar
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere asignar una mascota al collar.		
Precondiciones:	El usuario debe estar en la pantalla del collar a configurar y tener al menos una mascota registrada en el sistema.		
Postcondiciones:	Se registra la asignación de una mascota del usuario al collar.		
Flujo principal:	<ol style="list-style-type: none"> 1.- El usuario elige una de sus mascotas y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos y los envía al collar. 4.- El sistema redirige al usuario a la página del collar. 		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 21 - Especificación de caso de uso 11

Caso de uso:	CU-12	Nombre:	Configurar alimentador
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere configurar el alimentador.		
Precondiciones:	El usuario debe estar en la página de configuración del alimentador.		
Postcondiciones:	Se guarda la configuración del alimentador.		
Flujo principal:	<ol style="list-style-type: none"> 1.- El usuario ingresa los datos correspondientes y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos y los envía al alimentador. 4.- El sistema redirige al usuario a la página del alimentador. 		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 22 - Especificación de caso de uso 12

Caso de uso:	CU-13	Nombre:	Configurar el collar
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere configurar el collar.		
Precondiciones:	El usuario debe estar en la página de configuración del collar.		
Postcondiciones:	Se guarda la configuración del collar.		
Flujo principal:	1.- El usuario ingresa los datos correspondientes y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos y los envía al collar. 4.- El sistema redirige al usuario a la página del collar.		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 23 - Especificación de caso de uso 13

Caso de uso:	CU-14	Nombre:	Agregar horario
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere agregar un nuevo horario.		
Precondiciones:	El usuario debe estar en la página de información del alimentador al cual desea agregar un nuevo horario de alimentación.		
Postcondiciones:	Se registra el horario de alimentación en el sistema y alimentador..		
Flujo principal:	1.- El usuario ingresa los datos del nuevo horario y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos y envía el nuevo horario al alimentador. 4.- El sistema redirige al usuario a la página del alimentador.		
Flujo secundario:	2.1.- El sistema muestra una alerta con el o los errores al usuario.		
Excepciones:	Cierre de sesión por inactividad		

Tabla 24 - Especificación de caso de uso 14

Caso de uso:	CU-15	Nombre:	Editar horario
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere editar un horario.		
Precondiciones:	El usuario debe estar en la página de información del alimentador al cual desea editar un horario de alimentación y tener al menos un horario registrado.		
Postcondiciones:	Se edita y guarda el horario de alimentación en el sistema y alimentador.		
Flujo principal:	<ol style="list-style-type: none"> 1.- El usuario ingresa los datos del horario a editar y hace clic en aceptar. 2.- El sistema verifica los datos. 3.- El sistema registra los datos en la base de datos y envía el horario editado al alimentador. 4.- El sistema redirige al usuario a la página del alimentador. 		
Flujo secundario:	-		
Excepciones:	Cierre de sesión por inactividad		

Tabla 25 - Especificación de caso de uso 15

Caso de uso:	CU-16	Nombre:	Eliminar horario
Actores:	Usuario.		
Descripción:	El sistema debe comportarse de la siguiente manera cuando un usuario requiere eliminar un horario.		
Precondiciones:	El usuario debe estar en la página de información del alimentador al cual desea eliminar un horario de alimentación y tener al menos un horario registrado.		
Postcondiciones:	Se elimina el horario del sistema y del alimentador.		
Flujo principal:	<ol style="list-style-type: none"> 1.- El usuario hace clic en eliminar. 2.- El sistema verifica la petición. 3.- El sistema elimina los datos de la mascota y notifica al alimentador. 4.- El sistema redirige al usuario a la página a la página del alimentador y notifica que el horario fue eliminado con éxito. 		
Flujo secundario:	-		
Excepciones:	Cierre de sesión por inactividad		

Tabla 26 - Especificación de caso de uso 16

6.2. Diagrama Entidad-Relación

El Diagrama Entidad-Relación (también conocido como Modelo Entidad-Relación) es una herramienta de modelado de datos que permite realizar representaciones gráficas de las entidades de un sistema de información. Este modelo es generalmente utilizado para la creación de una base de datos relacional (Rouse, 2018).

En la Ilustración 2 representa el Diagrama Entidad-Relación del sistema, diferenciando los diferentes elementos de este con colores, donde el color amarillo representa las relaciones, el color celeste representa las entidades, el color púrpura representa la clave primaria y el color blanco representa los demás atributos (Ilustración en la siguiente página).

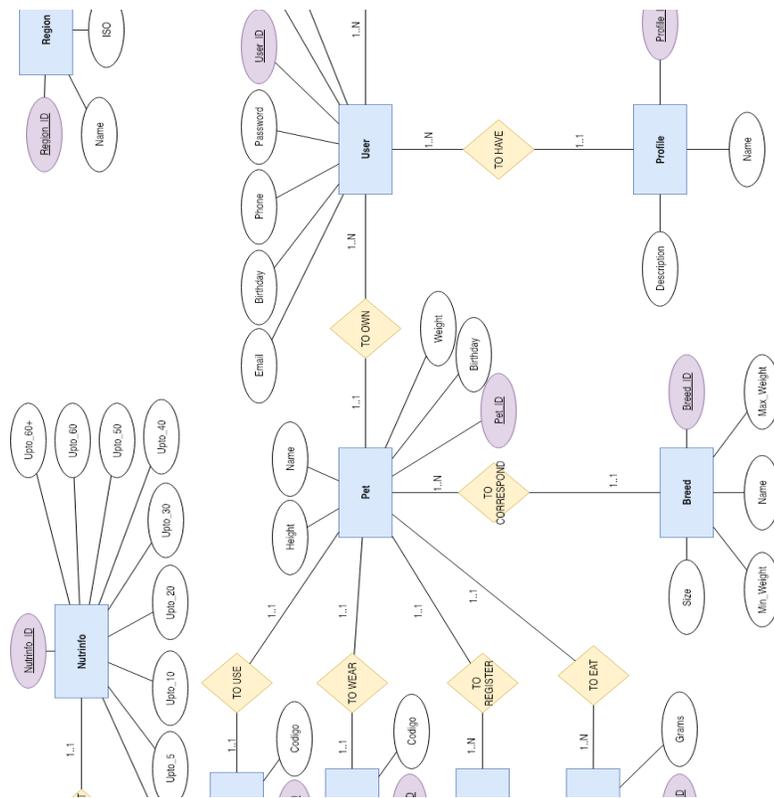


Ilustración 2 - Diagrama Entidad-Relación

6.3. Especificación de las entidades y relaciones

A continuación, se describen brevemente las entidades y las relaciones del modelo anterior.

6.3.1. Entidades:

- **Activity:** Representa la actividad física realizada por una mascota.
- **Breed:** Representa las distintas razas de perros que existen.
- **Food:** Representa las diferentes comidas de mascotas que se pueden encontrar en el mercado.
- **Nutrinfo:** Representa la información nutricional de las distintas comidas de mascotas.
- **User:** Representa a los usuarios del sistema.
- **Pet:** Representa las mascotas de los usuarios.
- **Feeder:** Representa los alimentadores de los usuarios.
- **Schedule:** Representa los horarios de comida programados en el alimentador.
- **Collar:** Representa los collares de los usuarios.
- **Portion:** Representa cada comida suministrada a la mascota.
- **Profile:** Representa el perfil de sistema que posee el usuario.
- **Region:** Representa las regiones del país.
- **Province:** Representa las provincias del país;
- **City:** Representa las ciudades del país.

6.3.2. Relaciones:

- **To Correspond:** Representa la relación entre las entidades Pet y Breed.
- **To Eat:** Representa la relación entre las entidades Pet y Portion.
- **To Have:** Representa la relación entre las entidades User y Profile.
- **To Use:** Representa la relación entre las entidades Pet y Feeder.
- **To Wear:** Representa la relación entre las entidades Pet y Collar.
- **To Register:** Representa la relación entre las entidades Pet y Activity.
- **To Suggest:** Representa la relación entre las entidades Food y Nutrinfo.
- **To Program:** Representa la relación entre las entidades Feeder y Schedule.
- **To Containt:** Representa la relación entre las entidades Feeder y Food.
- **To Own:** Representa la relación entre las entidades Pet y User.
- **To Live:** Representa la relación entre las entidades User y City
- **To Belong (I):** Representa la relación entre las entidades Region y Province.
- **To Belong (II):** Representa la relación entre las entidades Province y City.

CAPÍTULO 7

DISEÑO

7.1. Diseño físico de la base de datos

El Modelo Físico de la base de datos se genera a partir del modelo lógico de esta. En este modelo (modelo físico) se especifican las estructuras de almacenamiento internas, los índices a utilizar, la traducción de entidades al modelo físico (que nombre utilizar), los caminos de acceso específicos y que vistas definir en las tablas (IBM, s.f.).

En la Ilustración 3 se puede apreciar el Modelo Físico de la base de datos, la cual representa exactamente como es la base de datos que es utilizada por el sistema (Ilustración en la siguiente página).

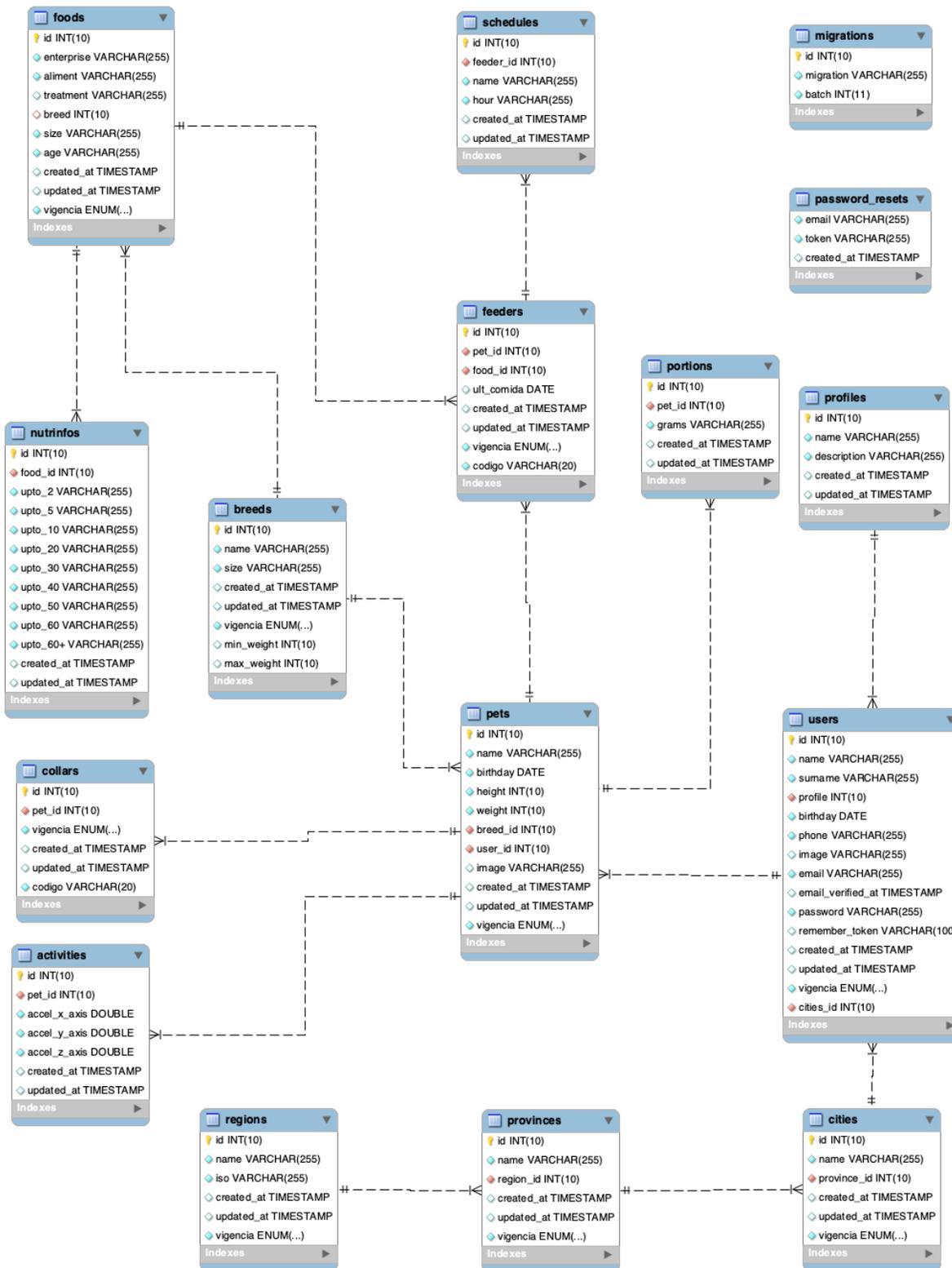


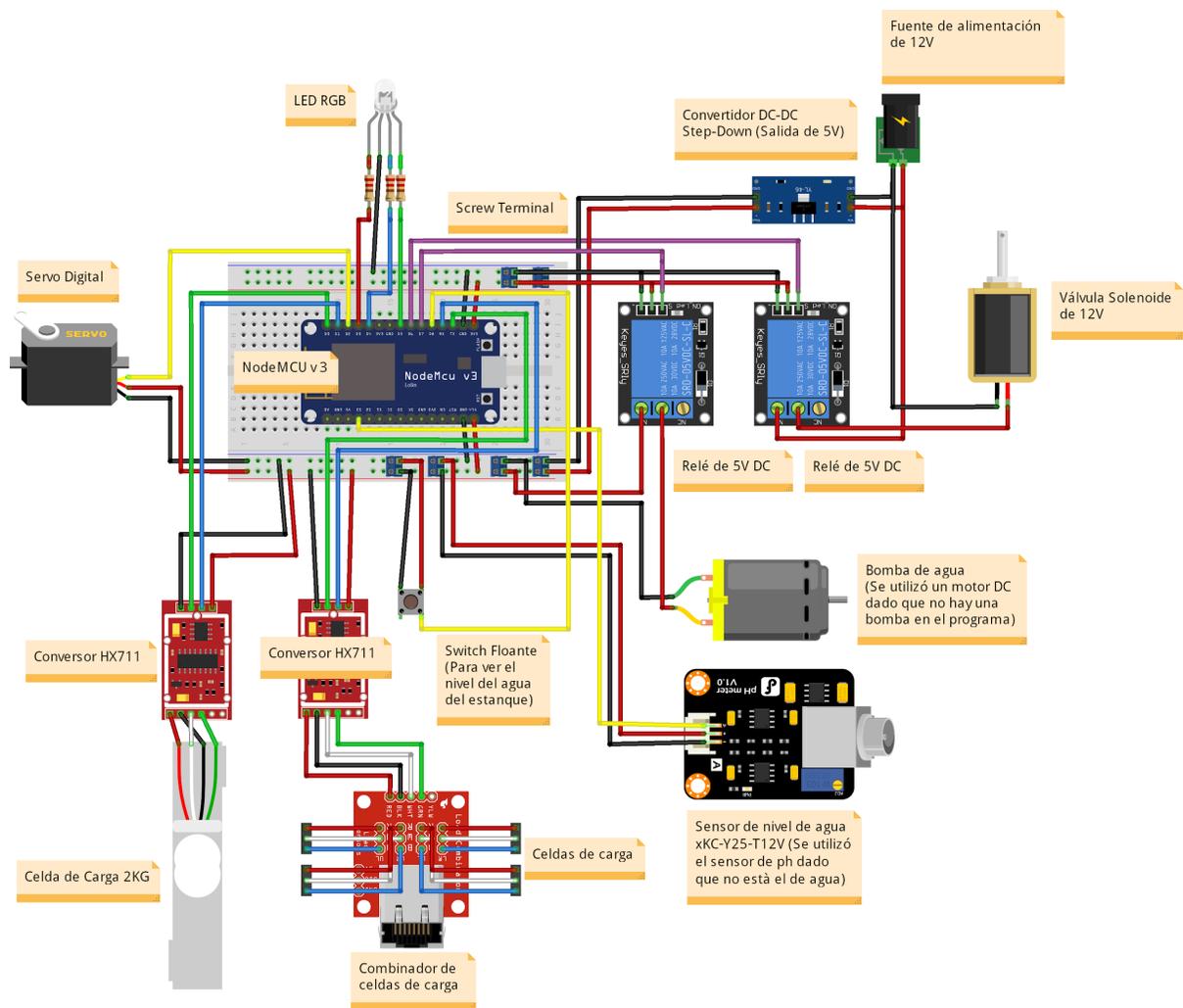
Ilustración 3 - Diseño físico de la base de datos

7.2. Diseño del hardware

El diseño del hardware incluye tanto circuitos como planos en 2D y 3D, sin embargo, dada la extensión de archivos que se posee, solo se incluye el diseño 2D y el diseño final en 3D, además del circuito correspondiente. Cabe mencionar que los circuitos no corresponden al diseño final del circuito, sino a una guía para montar el hardware, ya que los requerimientos eléctricos de algunos componentes cambian durante el desarrollo.

7.2.1. Diseño de circuitos

En la Ilustración 4 (Sandoval & Sepúlveda, Diseño de circuitos) se puede apreciar el diseño inicial del circuito electrónico del alimentador inteligente, donde además de mostrar la estructura del circuito, muestra la explicación de cada componente (Ilustración en la siguiente página).



fritzing

Ilustración 4 - Diseño del circuito del alimentador

En la Ilustración 4 (Sandoval & Sepúlveda, Diseño de circuitos) se puede apreciar el diseño inicial del circuito electrónico del alimentador inteligente, donde además de mostrar la estructura del circuito, muestra la explicación de cada componente (Ilustración en la siguiente página).

En la Ilustración 4 (Sandoval & Sepúlveda, Diseño de circuitos) se puede apreciar el diseño inicial del circuito electrónico del alimentador inteligente, donde además de mostrar la estructura del circuito, muestra la explicación de cada componente (Ilustración en la siguiente página).

En la Ilustración 5 (Sandoval & Sepúlveda, Diseño de circuitos) se puede apreciar el diseño inicial del circuito electrónico del collar inteligente, donde además de mostrar la estructura del circuito, muestra la explicación de cada componente.

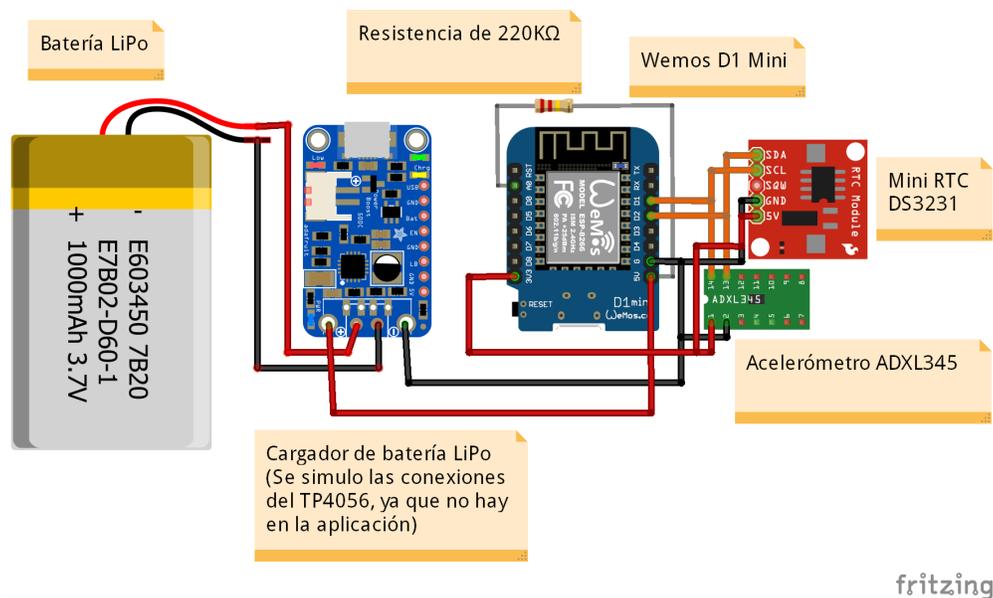


Ilustración 5 - Diseño del circuito del collar

7.2.2. Diseños 2D

En la Ilustraciones 6, 7 y 8 (Sandoval & Sepúlveda, Diseño de planos) se pueden apreciar los planos a escala real del alimentador, donde se muestra en cada una de las imágenes una parte diferente del alimentador.

Soporte para platos del alimentador

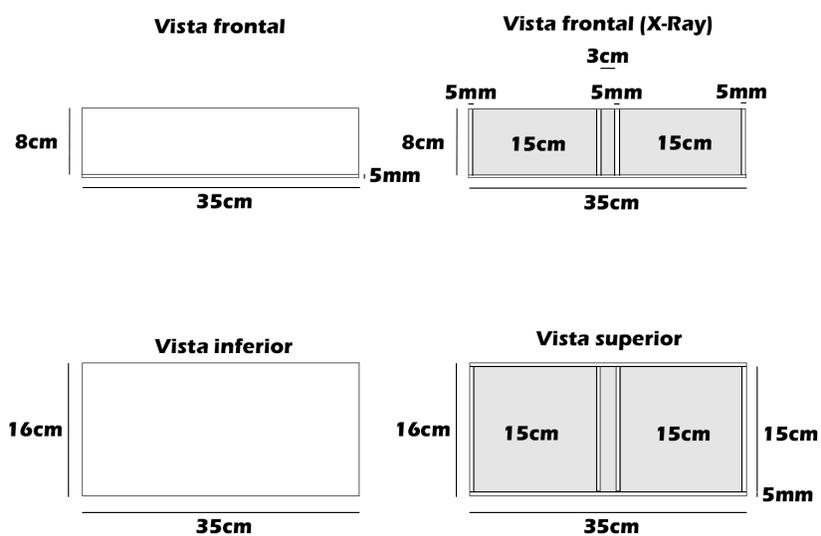


Ilustración 6 - Diseño la estructura del alimentador pt. 1

Base del alimentador

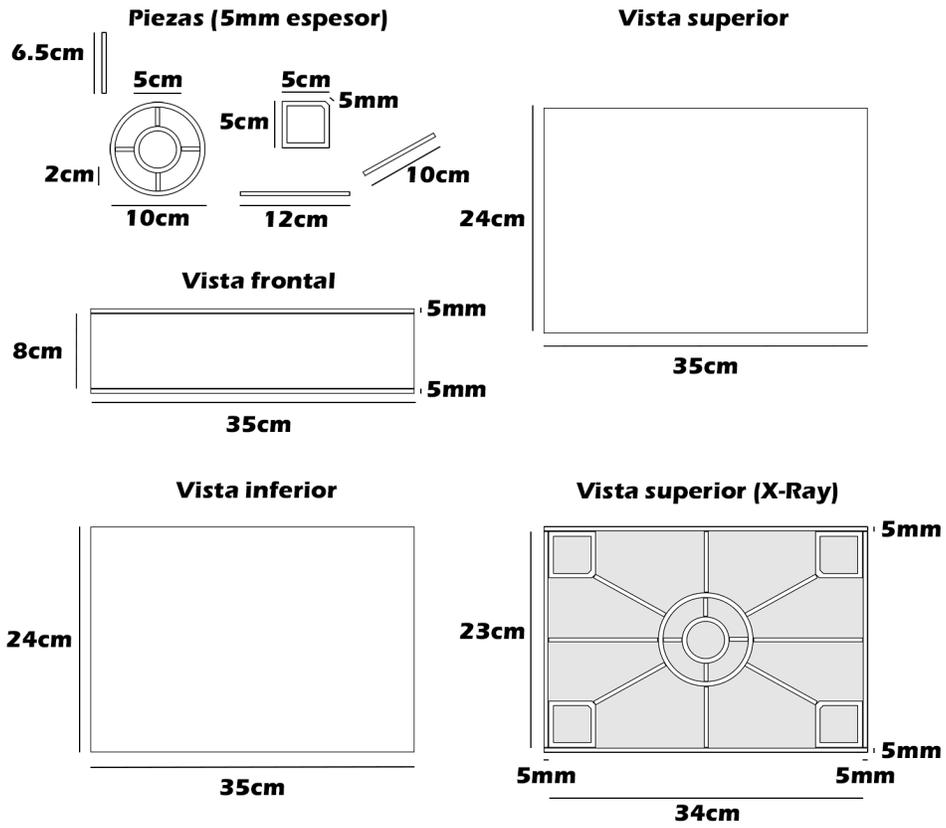


Ilustración 7 - Diseño la estructura del alimentador pt. 2

Cuerpo del alimentador

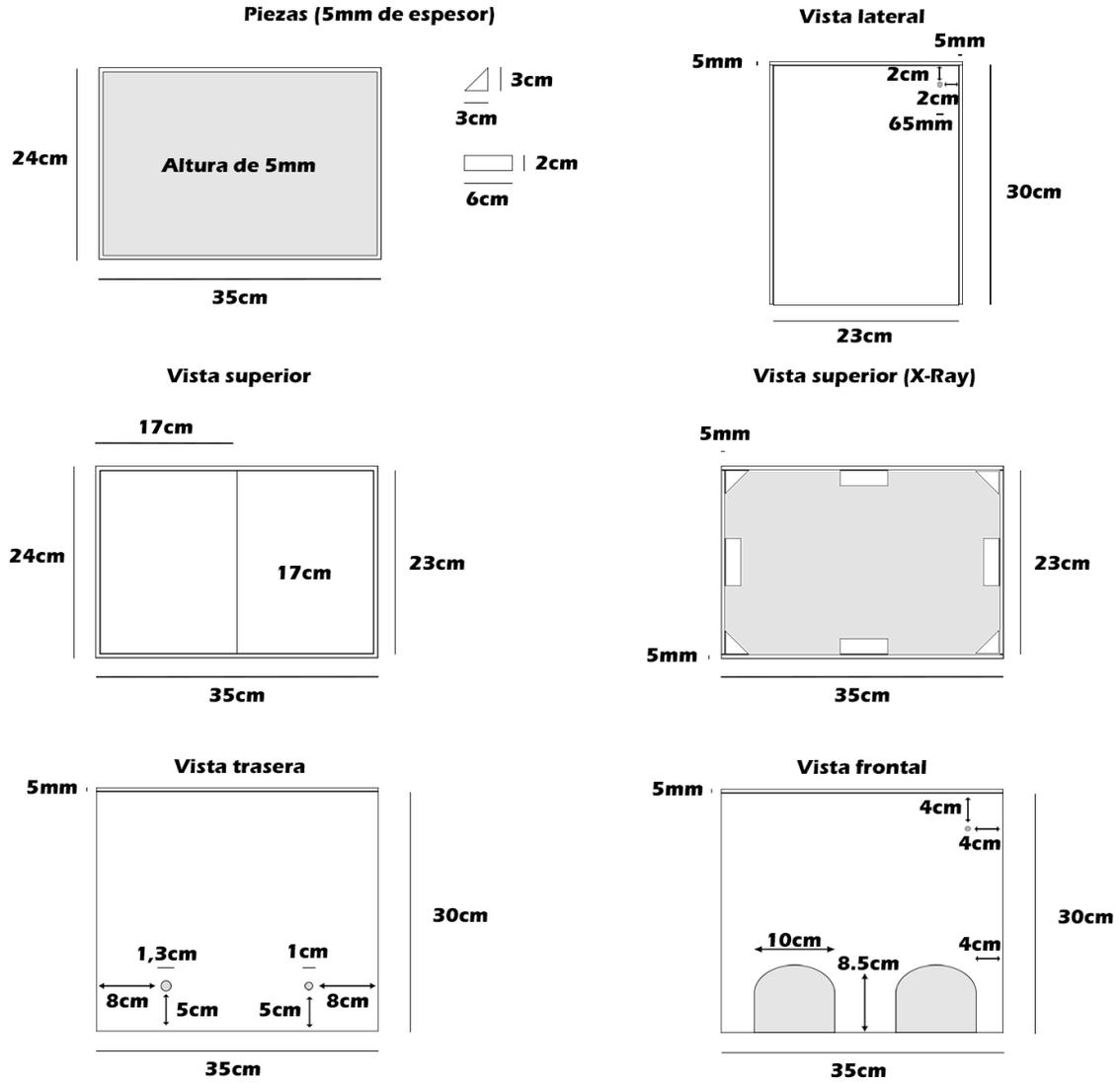


Ilustración 8 - Diseño la estructura del alimentador pt. 3

7.2.3. Diseños 3D

En la Ilustración 9 (Sandoval & Sepúlveda, Diseños 3D) se puede apreciar el diseño conceptual en 3D del alimentador.

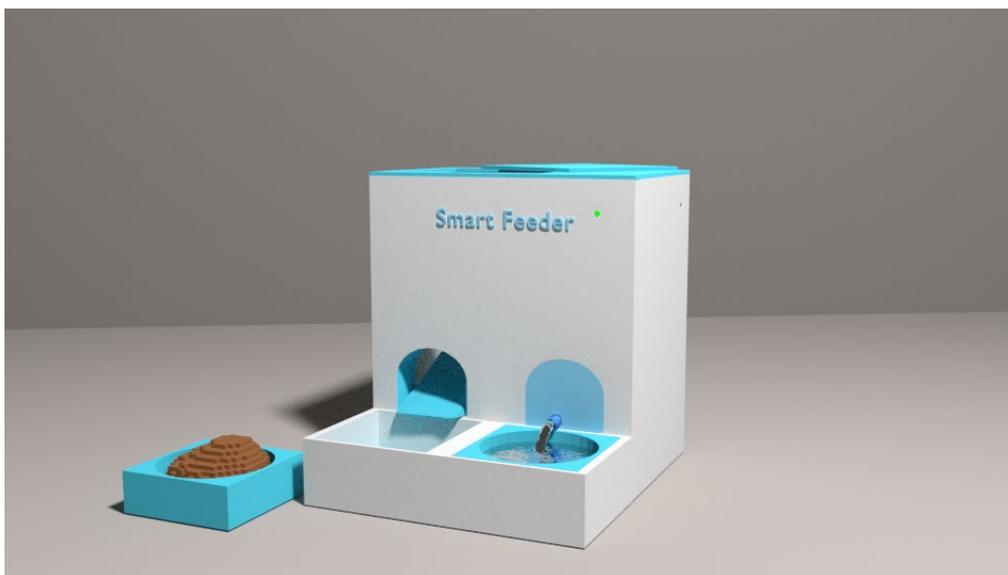


Ilustración 9 - Diseño conceptual del alimentador

7.3. Diseño de interfaz y navegación

El diseño de la interfaz y navegación corresponde al diseño inicial de la aplicación, ya que, a lo largo del desarrollo, las interfaces suelen cambiar drásticamente con la finalidad de mejorar la experiencia de usuario. Cabe mencionar que no se incluyen todos los diseños, debido a la cantidad de diseños generados.

7.3.1. Landing page, registrarse e iniciar sesión

En las Ilustraciones 10, 11, 12, 14 y 14 (Sandoval & Sepúlveda, Diseños de aplicación web) se pueden apreciar los diseños iniciales de las vistas de usuario de la aplicación web.

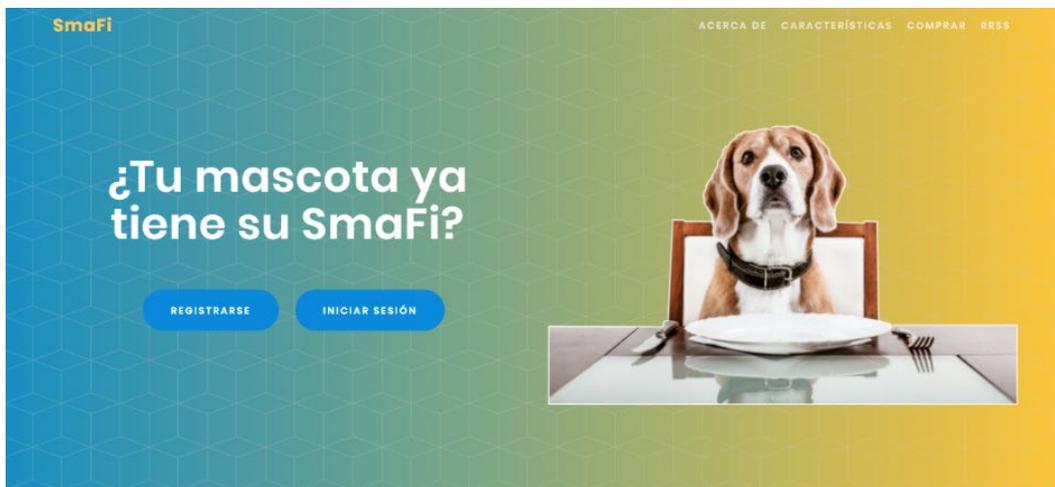


Ilustración 10 - Diseño de la vista "landing page"

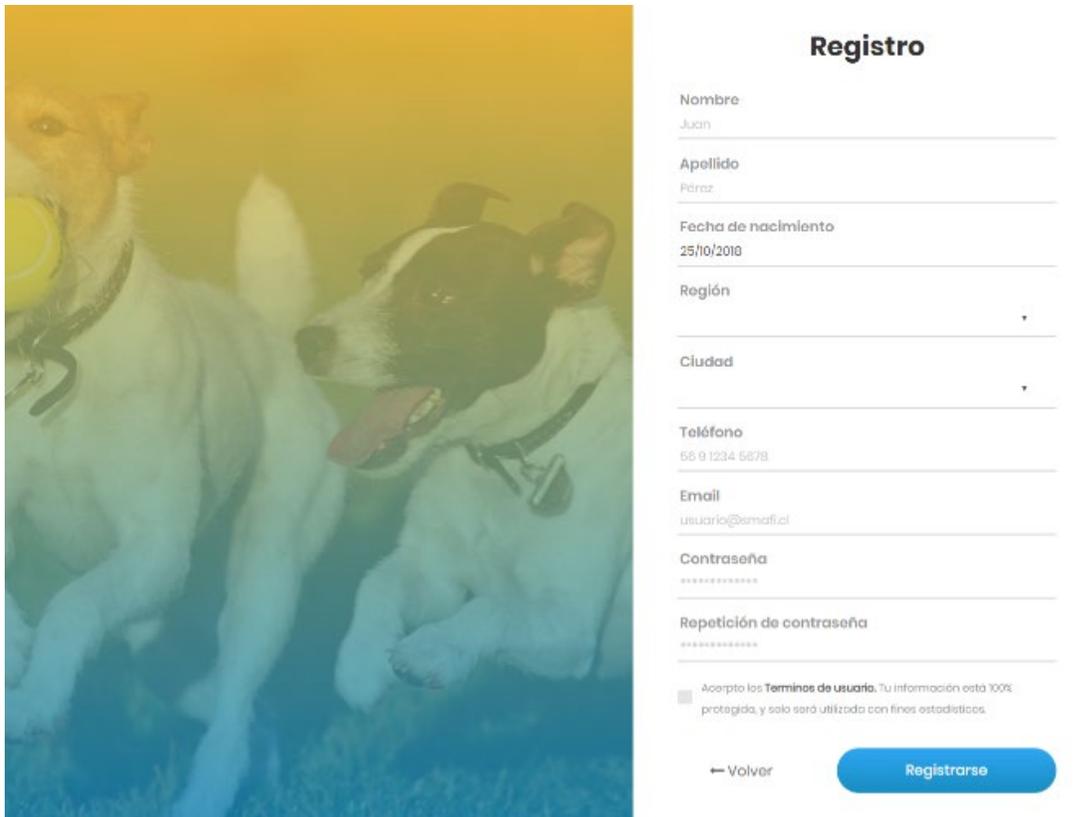


Ilustración 11 - Diseño de la vista “registro”

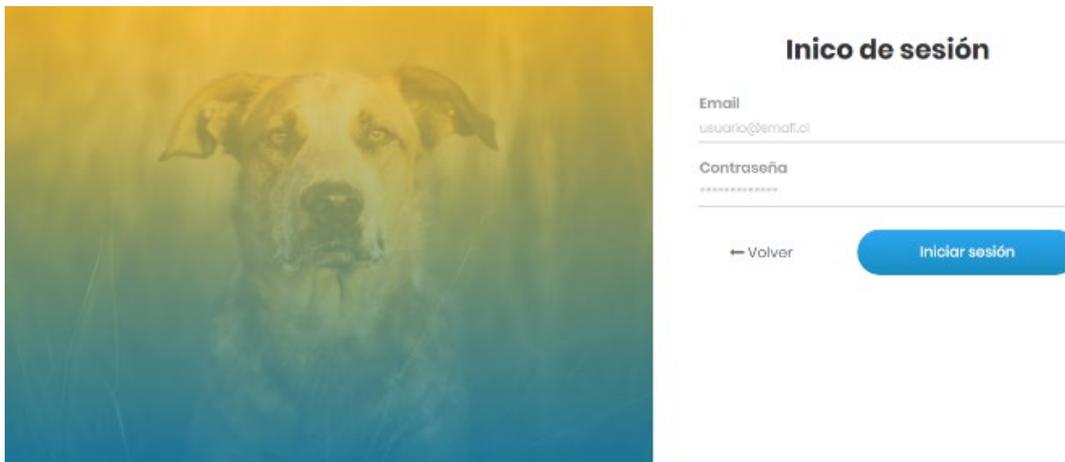


Ilustración 12 - Diseño de la vista “inicio de sesión”

7.3.2. Vistas de usuario

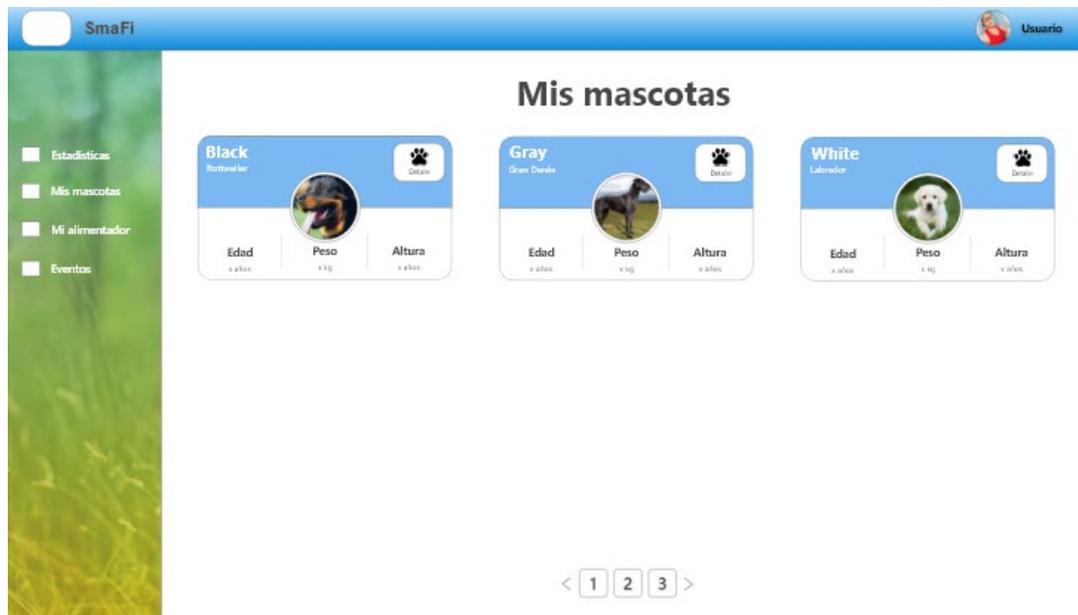


Ilustración 13 - Diseño de la vista “mis mascotas”



Ilustración 14 - Diseño de la vista “mi alimentador”

7.3.3. Vistas de administrador

En las Ilustraciones 15 y 16 (Sandoval & Sepúlveda, Diseños de aplicación web) se pueden apreciar los diseños iniciales de las vistas de administrador de la aplicación web.

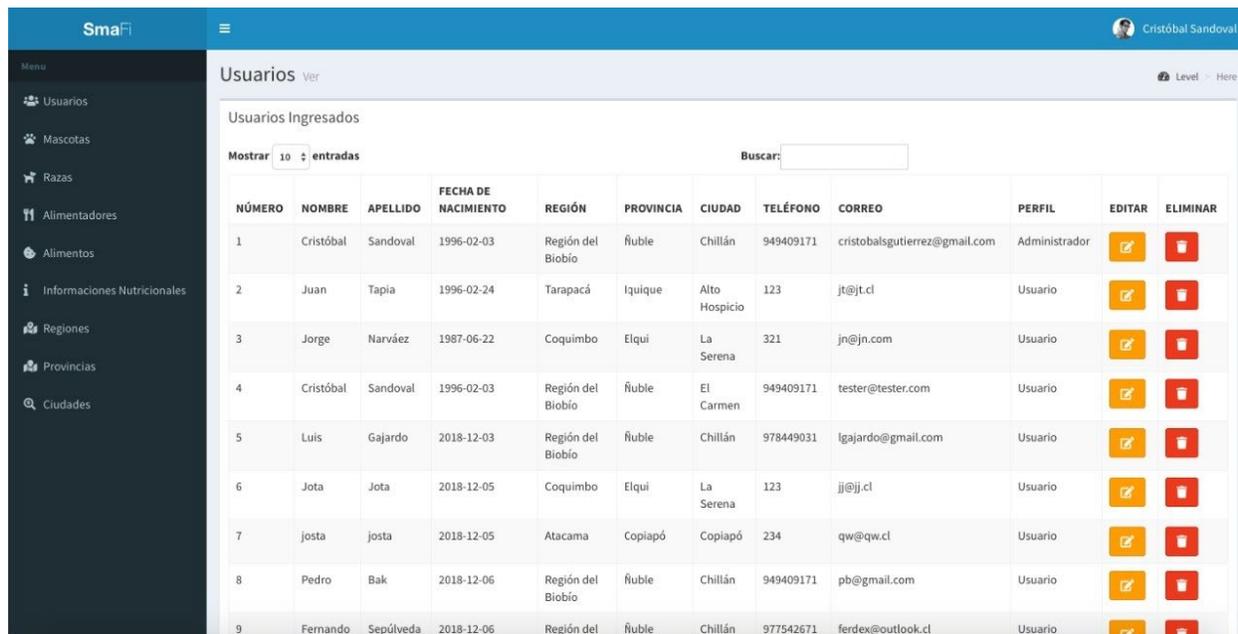


Ilustración 15 - Diseño de la vista de administrador “usuarios”

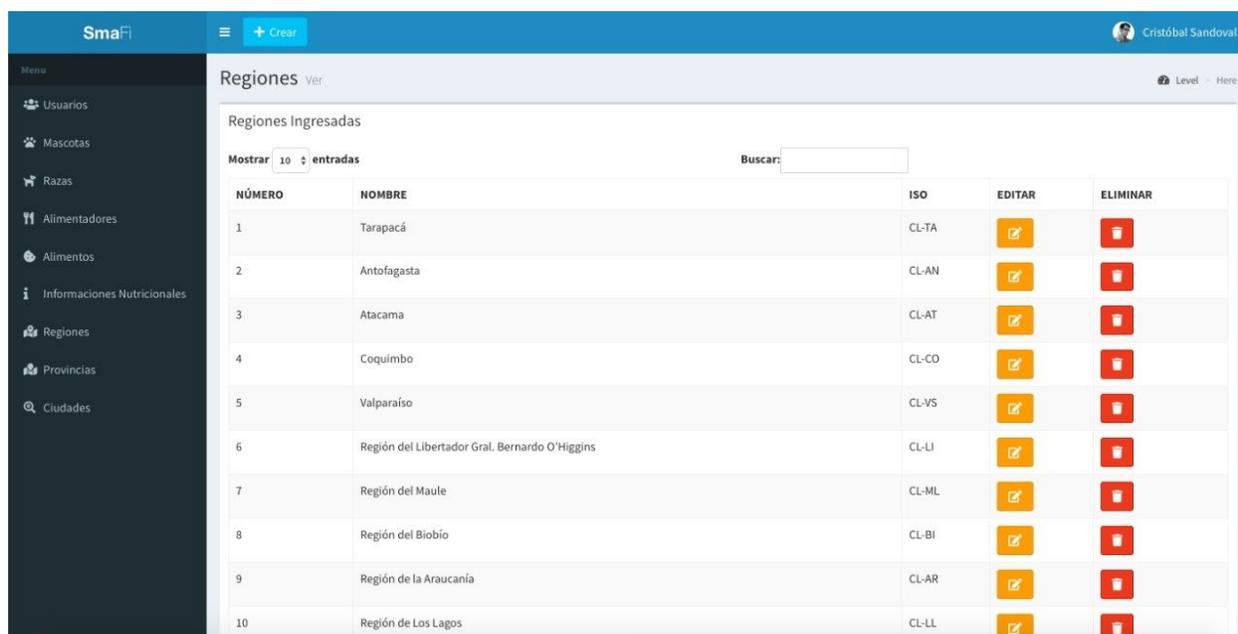


Ilustración 16 - Diseño de la vista de administrador “regiones”

CAPÍTULO 8

PRUEBAS

8.1. Casos de prueba

Los casos de prueba contemplados abarcan distintos tipos de pruebas de software, donde cada tipo de prueba vela por verificar el comportamiento del sistema en un ámbito específico. En cuanto las pruebas de hardware, estas no se contemplarán en el informe, dado a que no hay un formato específico de cómo realizar dichas pruebas.

Los tipos de pruebas a realizar son:

- 1) **Pruebas unitarias:** Prueban los métodos de una clase determinada, para así asegurar el correcto funcionamiento de estos de forma aislada (García, 2014).
- 2) **Pruebas de integración:** Corresponde a la prueba de la mezcla de dos o más componentes de la aplicación, para así asegurar el correcto funcionamiento en conjunto (García, 2014).
- 3) **Pruebas funcionales:** Estas pruebas comprueban que el software cumple con las funcionalidades esperadas (García, 2014).
- 4) **Pruebas de aceptación:** Son pruebas que verifican si el software cumple realmente con lo que el cliente pidió, o en este caso, si soluciona la problemática identificada (García, 2014).

8.2. Especificación de las pruebas

En la Tabla 27 se especifican las pruebas necesarias, indicando el id, el rol del usuario, la funcionalidad que busca probar y el criterio de aceptación necesario para aprobar el test.

ID	Rol	Funcionalidad	Criterio de aceptación
P1	Usuario	Seguridad (autenticación)	Dado un usuario que aún no se ha identificado en el sistema, cuando éste intenta acceder a alguna funcionalidad de la parte privada, entonces se le redirige automáticamente a la página de login para que pueda identificarse
P2	Usuario	Inicio de sesión	Dado un usuario registrado en el sistema, cuando éste intenta acceder a su cuenta con datos correctos, entonces el sistema accede a su cuenta y muestra sus mascotas
P3	Usuario	Registrar mascota	Dado un usuario registrado en el sistema, cuando éste se dirige a la pantalla de registro de mascota e ingresa los datos solicitados, entonces el sistema registra la mascota, se lo notifica al usuario y redirige al usuario a la página de la nueva mascota.
P4	Usuario	Ver mascota	Dado un usuario y una mascota (perteneciente a él) registrados en el sistema, cuando éste hace clic en ver más información de la mascota, entonces el sistema muestra información más detallada de ésta.
P5	Usuario	Editar mascota	Dado un usuario y una mascota (perteneciente a él) registrados en el sistema, cuando éste se dirige a la pantalla de editar información de la mascota (el sistema muestra la pantalla de edición con la información actual de mascota cargada), entonces el sistema edita y registra la información de la mascota, luego redirige al usuario a la pantalla de información de la mascota (con la nueva información).
P6	Usuario	Agregar alimentador	Dado un usuario y una mascota (perteneciente a él) registrados en el sistema, cuando éste va a la pantalla de agregar alimentador e ingresa los datos solicitados, entonces el sistema registra los datos del alimentador, se lo notifica al usuario y redirige al usuario a la pantalla del nuevo alimentador.
P7	Usuario	Editar alimentador	Dado un usuario y una mascota (perteneciente a él) registrados en el sistema, cuando éste se dirige a la pantalla de editar información del alimentador (el sistema muestra la pantalla de edición con la información actual del alimentador cargada), entonces el sistema edita y registra la información del alimentador, luego redirige al usuario a la pantalla de información del alimentador (con la nueva información).
P8	Usuario	Cargar mascotas en formularios de agregar y editar alimentador	Dado un usuario registrado en el sistema, cuando este se dirige a cualquiera de los formularios (agregar o editar) del alimentador, entonces el sistema carga solo a las mascotas del usuario y que no tienen un alimentador asignado en una lista desplegable, en la cual el usuario puede seleccionar una.
P9	Usuario	Registrar nuevo usuario	Dado un nuevo usuario no registrado en el sistema, cuando éste se dirige a la pantalla de registro, ingresa todos los datos solicitados y acepta los términos y condiciones, entonces el sistema lo registra en el sistema e inicia su sesión por primera vez.

P10	Usuario	Cargar regiones, provincias y ciudades	Dado un nuevo usuario, cuando éste se dirige a la pantalla de registro de usuario, entonces el sistema carga las regiones, carga dinámicamente las provincias de la región seleccionada y carga dinámicamente las ciudades pertenecientes a la provincia seleccionada.
-----	---------	--	--

Tabla 27 - Especificación de pruebas

8.3. Resultados de las pruebas

En la Tabla 28 se muestran los resultados a partir de las pruebas.

ID	Entrada	Resultado
P1	Un usuario no registrado intenta ingresar a alguna de las funcionalidades del sistema.	El sistema redirige al usuario a la pantalla de login.
P1	Un usuario registrado, sin iniciar sesión, intenta ingresar a alguna de las funcionalidades del sistema.	El sistema redirige al usuario a la pantalla de login.
P2	Un usuario no registrado intenta iniciar sesión con datos falsos.	El sistema recarga la pantalla de login y notifica al usuario que las credenciales no se encuentran registradas en el sistema.
P2	Un usuario registrado ingresa un dato incorrecto al momento de iniciar sesión.	El sistema recarga la pantalla de login y notifica al usuario que las credenciales no se encuentran registradas en el sistema.
P2	Un usuario registrado intenta iniciar sesión con datos correctos.	El sistema ingresa a la sesión del usuario (muestra las paginas registradas en el sistema).
P3	Un usuario registrado ingresa un dato incorrecto o vacío al momento de registrar una nueva mascota.	El sistema recarga la página de registro de mascota y notifica los errores al usuario (la página conserva los datos ingresados al momento de actualizar).
P3	Un usuario registrado ingresa todos los datos solicitados correctamente.	El sistema registra la mascota, notifica el registro exitoso de la mascota y lo redirige a la página de la nueva mascota registrada.
P4	Un usuario registrado hace clic en la opción ver más información de una mascota.	El sistema redirecciona al usuario a la página de la mascota, en la cual éste puede ver toda la información de la mascota.
P5	Un usuario registrado ingresa datos incorrectos en el formulario de edición de la mascota.	El sistema recarga la página de edición de la mascota y notifica los errores al usuario (la página conserva los datos ingresados al momento de actualizar).
P5	Un usuario registrado ingresa datos correctos en el formulario de edición de la mascota.	El sistema registra la información actualizada de la mascota, notifica la edición exitosa de la mascota y lo redirige a la página de la mascota editada.
P6	Un usuario registrado ingresa un dato incorrecto o vacío al momento de registrar un nuevo alimentador.	El sistema recarga la página de registro de alimentador y notifica los errores al usuario (la página conserva los datos ingresados al momento de actualizar).
P6	Un usuario registrado ingresa todos los datos solicitados correctamente.	El sistema registra el alimentador, notifica el registro exitoso del alimentador y lo redirige a la página del nuevo alimentador registrado.

P6	Un usuario registrado intenta registrar un alimentador sin tener una mascota registrada	El sistema registra el alimentador (sin mascota).
P7	Un usuario registrado ingresa datos incorrectos en el formulario de edición del alimentador.	El sistema recarga la página de edición del alimentador y notifica los errores al usuario (la página conserva los datos ingresados al momento de actualizar).
P7	Un usuario registrado ingresa datos correctos en el formulario de edición del alimentador.	El sistema registra la información actualizada del alimentador, notifica la edición exitosa del alimentador y lo redirige a la página del alimentador editado.
P8	Un usuario registrado ingresa a uno de los formularios del alimentador (agregar o editar).	El sistema carga el select (lista desplegable) con las mascotas del usuario que no tengan registrado un alimentador.
P9	Un nuevo usuario ingresa datos incorrectos y/o vacíos al momento de registrarse en el sistema.	El sistema recarga la página de registro de usuario y notifica los errores al usuario (la página conserva los datos ingresados al momento de actualizar).
P9	Un nuevo usuario ingresa un correo que ya existe en el sistema.	El sistema recarga la página de registro de usuario y notifica el error al usuario (la página conserva los datos ingresados al momento de actualizar).
P9	Un nuevo usuario ingresa los datos correctos en el formulario de registro de usuario.	El sistema registra el nuevo usuario, inicia su sesión por primera vez rediriéndolo a la página de todas las mascotas.
P10	Un nuevo usuario ingresa al formulario de registro de usuarios.	El sistema carga todas las regiones, después el sistema carga las provincias pertenecientes a la región seleccionada, luego el sistema carga las ciudades pertenecientes a la provincia seleccionada, si el usuario cambia uno de estos datos, el sistema cambiará el resto de datos que dependan de ese (exceptuando la ciudad, porque nada depende de ella).

Tabla 28 - Resultados de pruebas

CONCLUSIONES

- Un proyecto de IOT puede ser implementado en un sinnúmero de áreas y no tan solo en la informática, sin embargo, estos proyectos tienen algo en común, y es que integran el uso de diversas tecnologías (tanto de software como de hardware), con la lógica de negocio del área donde está siendo implementado dicho proyecto.
- Respecto a la metodología de trabajo utilizada (Scrumban), esta brindó excelentes resultados en cuanto al desarrollo y planificación de tareas. Esto se debe a que el tablero permite ir recordando las tareas que se deben realizar, quien la debe realizar y el plazo asignado a cada tarea. Además, gracias al trabajo de flujo continuo, las pruebas y las reuniones frecuentes, se puede obtener feedback (realimentación) para mejorar los puntos débiles proyecto instantáneamente.
- En cuanto a las porciones de alimento, para asegurar que se brinda la máxima fidelidad, se cuenta con un algoritmo que considera la información suficiente para poder brindar planes personalizados.
- Finalmente, el desarrollo de este proyecto ha finalizado de manera exitosa, brindando un prototipo funcional totalmente diseñado por los autores de este proyecto (Sandoval & Sepúlveda). Además, se ha cumplido con los objetivos propuestos y solucionando la problemática identificada.

FUTURO DEL PROYECTO

El futuro de este proyecto es rentable, sin embargo, está orientado a un sector específico de personas. Por el contrario, la meta es llegar aún a más personas, haciendo que el sistema pueda alimentar no solo a perros, sino que, a otro tipo de mascotas, además de hacer más accesible el precio de los productos construidos. Para ello, es necesario cambiar de proveedor a uno mayorista e idealmente internacional, los cuales poseen los mismos y aún más productos de los que se pueden encontrar en el mercado nacional, pero a un precio excesivamente más bajo.

Dados los planes de expansión de ventas, es necesario aumentar la seguridad en el producto y las capacidades del producto, por lo cual se requiere de un nuevo hardware, como lo es una placa más potente como Raspberry Pi o Banana Pi. Esto permitiría encriptar todas las peticiones enviadas y recibidas por parte del producto, evitando además los ataques a los que es susceptible la placa actualmente utilizada. Cabe mencionar que ningún sistema jamás estará exento de ataques, por más seguro que este sea. En cuanto al aumento de capacidades del producto, se desea hacer aún más inteligente y amigable con el usuario el producto.

BIBLIOGRAFÍA

- Adobe. (s.f.). *Dream bigger. Creative Cloud*. Obtenido de Adobe:
<https://www.adobe.com/creativecloud.html?promoid=CD922NLK&mv=other>
- AKC. (13 de Julio de 2013). *The Care and Feeding of the Breeding Bitch – Part Three*.
 Obtenido de AKC: <https://www.akc.org/expert-advice/dog-breeding/the-care-and-feeding-of-the-breeding-bitch-part-three/>
- Aniorte, C. (17 de Febrero de 2016). *Mascotas: Mucho cuidado con la alimentación*. Obtenido de ABC: https://www.abc.es/sociedad/abci-mascotas-mucho-cuidado-alimentacion-201602172319_noticia.html
- Aniorte, C. (29 de Agosto de 2017). *Mascotas, una parte fundamental de la sociedad*. Obtenido de ABC: https://www.abc.es/sociedad/abci-mascotas-parte-fundamental-sociedad-201603022058_noticia.html
- Arcas, E. (s.f.). *Cuidado de nuestras mascotas*. Obtenido de En Buenas Manos:
<https://www.enbuenasmanos.com/cuidado-de-nuestras-mascotas>
- Arduino. (s.f.). *Frequently Asked Questions*. Obtenido de Arduino:
<https://www.arduino.cc/en/Main/FAQ#toc2>
- Atom. (s.f.). *Why Atom?* Obtenido de Atom: <https://flight-manual.atom.io/getting-started/sections/why-atom/>
- Blender. (s.f.). *About*. Obtenido de Blender: <https://www.blender.org/about/>
- Bootstrap. (s.f.). *Introduction*. Obtenido de Bootstrap: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- Burke, A. (15 de Agosto de 2018). *How Many Times a Day Should a Dog Eat?* Obtenido de AKC: <https://www.akc.org/expert-advice/nutrition/how-many-times-a-day-should-a-dog-eat/>
- Composer. (s.f.). *Introduction*. Obtenido de Composer: <https://getcomposer.org/doc/00-intro.md>
- Domodesk. (2014). *A fondo: ¿Qué es IOT (El Internet de Las Cosas)?* Obtenido de Domodesk:
<https://www.domodesk.com/221-a-fondo-que-es-iot-el-internet-de-las-cosas.html>
- Draw.io. (s.f.). *About us*. Obtenido de Draw.io: <https://about.draw.io/about-us/>
- Fritzing. (s.f.). *About*. Obtenido de Fritzing: <http://fritzing.org/about/context/>
- García, A. (Julio de 2014). *¿Pruebas de integración, funcionales, de carga...? ¡Qué jaleo! ¿Qué diferencias hay?* Obtenido de Javiergarzas: <http://www.javiergarzas.com/2014/07/tipos-de-pruebas-10-min.html>
- Gitkraken. (s.f.). *About*. Obtenido de Gitkraken: <https://www.gitkraken.com/about>

- IBM. (s.f.). *Help - IBM Rational*. Obtenido de IBM Rational: https://jazz.net/help-dev/clm/index.jsp?re=1&topic=/com.ibm.rational.clm.doc/helpindex_clm.html&scope=null
- IBM. (s.f.). *Physical database design*. Obtenido de IBM: https://www.ibm.com/support/knowledgecenter/en/SSEPEK_10.0.0/intro/src/tpc/db2z_physicaldatabasedesign.html
- Laravel. (s.f.). *Introduction*. Obtenido de Laravel: <https://laravel.com/docs/4.2/introduction>
- MySQL. (s.f.). *MySQL Workbench*. Obtenido de MySQL: <https://www.mysql.com/products/workbench/>
- No-IP. (s.f.). *Free Dynamic DNS - Managed DNS - Managed Email - Domain Registration - No-IP*. Obtenido de No-IP: <https://www.noip.com/>
- NTP Pool Project. (s.f.). *Chile — cl.pool.ntp.org*. Obtenido de NTP Pool Project: <https://www.pool.ntp.org/zone/cl>
- Pahuja, S. (s.f.). *What is Scrumban?* Obtenido de Kanban Tool: <https://kanbantool.com/kanban-library/scrumban/what-is-scrumban>
- PHP. (s.f.). *General Information*. Obtenido de PHP: <http://php.net/manual/en/faq.general.php>
- PhpMyAdmin. (s.f.). *Introduction*. Obtenido de PhpMyAdmin: <https://docs.phpmyadmin.net/en/latest/intro.html>
- Rouse, M. (Marzo de 2018). *Entity relationship diagram (ERD)*. Obtenido de Search Data Management: <https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD>
- SAG. (2017). *Reglamento de alimentos para animales*. Obtenido de SAG: https://www.sag.gob.cl/sites/default/files/d_4-2017_regl_alimentos_pdf-difusion_tapa.pdf
- Sandoval, C., & Sepúlveda, F. (s.f.). Construcción del prototipo. *Construcción del prototipo*. Universidad del Bío-Bío, Chillán.
- Sandoval, C., & Sepúlveda, F. (s.f.). Diseño de circuitos. *Diseños del alimentador y collar*. Universidad del Bío-Bío, Chillán.
- Sandoval, C., & Sepúlveda, F. (s.f.). Diseño de planos. *Diseños del alimentador y collar*. Universidad del Bío-Bío, Chillán.
- Sandoval, C., & Sepúlveda, F. (s.f.). Diseños 3D. *Diseños del alimentador y collar*. Universidad del Bío-Bío, Chillán.
- Sandoval, C., & Sepúlveda, F. (s.f.). Diseños de aplicación web. *Diseños del alimentador y collar*. Universidad del Bío-Bío, Chillán.
- Tinkercad. (s.f.). *Tinkercad features*. Obtenido de Tinkercad: <https://www.tinkercad.com/about/features>

Trello. (s.f.). *About*. Obtenido de Trello: <https://trello.com/about>

W3schools. (s.f.). *CSS Introduction*. Obtenido de w3schools: CSS Introduction

W3Schools. (s.f.). *HTML Introduction*. Obtenido de w3Schools:

https://www.w3schools.com/HTML/html_intro.asp

W3schools. (s.f.). *Introduction to SQL*. Obtenido de w3schools:

https://www.w3schools.com/sql/sql_intro.asp

W3schools. (s.f.). *JavaScript Introduction*. Obtenido de w3schools:

https://www.w3schools.com/js/js_intro.asp

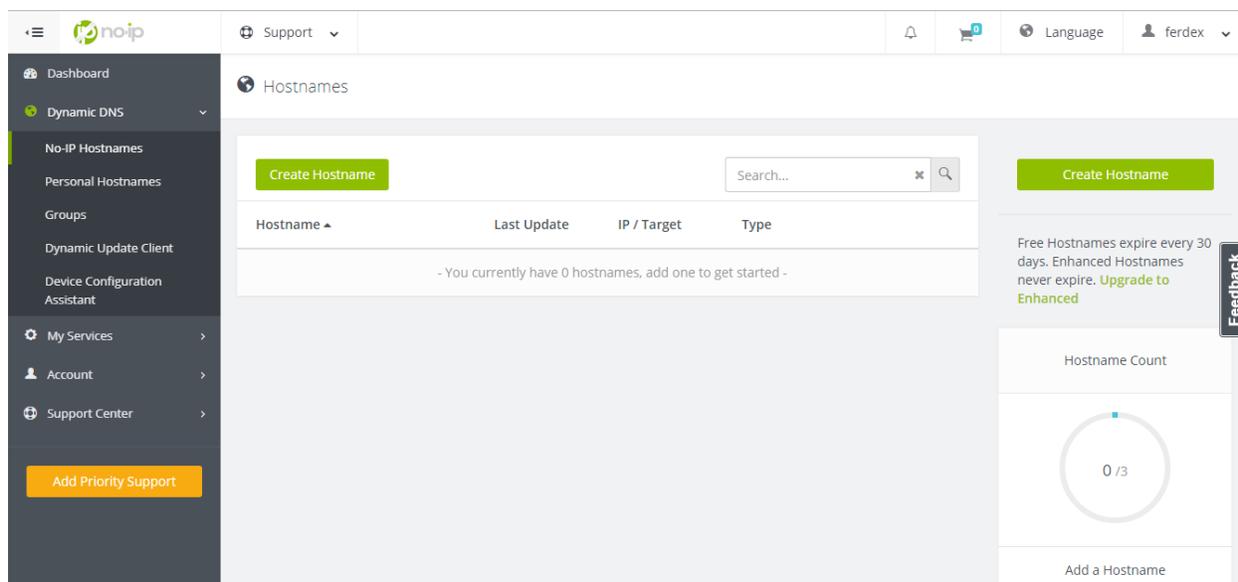
ANEXOS

Interfaces de software

En las Ilustraciones 17 (No-IP), 18 (No-IP) y 19 (NTP Pool Project), se pueden apreciar las interfaces de software utilizadas.



Anexo: Ilustración 17 - No-IP, Landing Page



Anexo: Ilustración 18 - No-IP, Servicios DDNS



- JOIN THE POOL
- USE THE POOL
- MANAGE SERVERS

Chile — cl.pool.ntp.org

We need more servers in this country. If you have a server with a static IP, please consider [joining the pool!](#)

There are not enough servers in this zone, so we recommend you use the South America zone ([south-america.pool.ntp.org](#)):

```
server 0.south-america.pool.ntp.org
server 1.south-america.pool.ntp.org
server 2.south-america.pool.ntp.org
server 3.south-america.pool.ntp.org
```

In most cases it's best to use [pool.ntp.org](#) to find an NTP server (or [0.pool.ntp.org](#), [1.pool.ntp.org](#), etc if you need multiple server names). The system will try finding the closest available servers for you. If you distribute software or equipment that uses NTP, please see our [information for vendors](#).

- News
- [How do I use pool.ntp.org?](#)
- [How do I join pool.ntp.org?](#)
- [Information for vendors](#)
- [The mailing lists](#)
- [Additional links](#)

Translations

- [Dansk](#)
- [Deutsch](#)
- [English](#)
- [Español](#)
- [Suomi](#)
- [Français](#)
- [हिन्दी](#)
- [Italiano](#)

IPv4

There are 3 active servers in this zone.

- 2 (+1) active 1 day ago
- 2 (+1) active 7 days ago
- 0 (+3) active 14 days ago
- 0 (+3) active 60 days ago
- 7 (-4) active 180 days ago
- 5 (-2) active 1 year ago
- 7 (-4) active 3 years ago
- 5 (-2) active 6 years ago

IPv6

There is 1 active server in this zone.

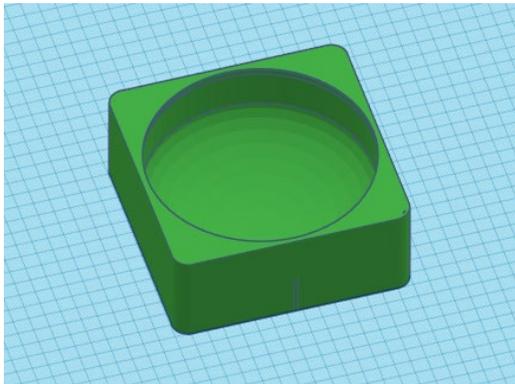
- 1 active 1 day ago
- 2 (-1) active 7 days ago
- 2 (-1) active 14 days ago
- 0 (+1) active 60 days ago
- 4 (-3) active 180 days ago
- 3 (-2) active 1 year ago
- 3 (-2) active 3 years ago
- 1 active 6 years ago

[See all zones in South America.](#)

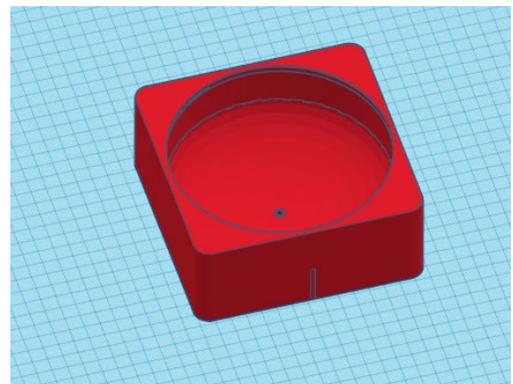
Anexo: Ilustración 19 - NTP Pool Project, Servidores NTP para Chile

Diseños 3D de componentes del alimentador

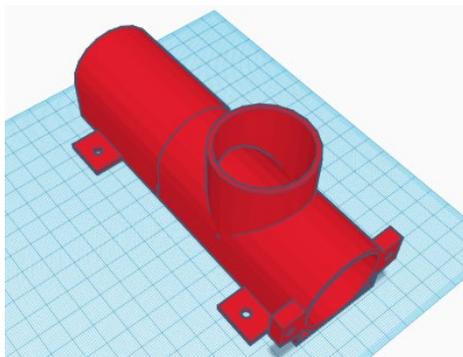
Desde la ilustración 20 hasta la 23, se pueden apreciar los diferentes elementos diseñados en 3D (Sandoval & Sepúlveda, Diseños 3D) que se utilizaron en la construcción del alimentador.



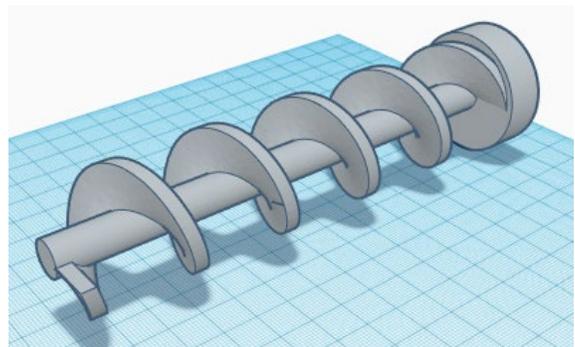
Anexo: Ilustración 23 - Plato para alimento en 3D



Anexo: Ilustración 22 - Plato para agua en 3D



Anexo: Ilustración 20 - Contenedor para tornillo sin fin en 3D



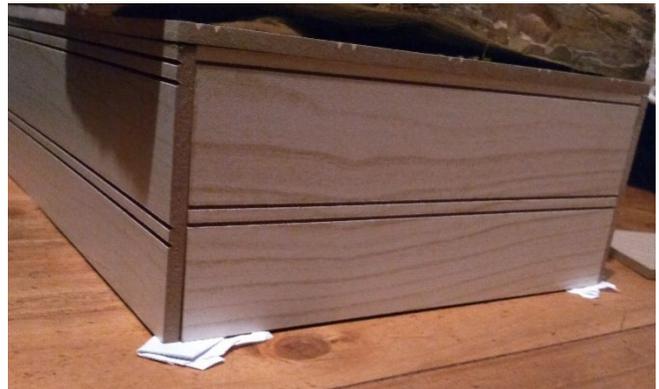
Anexo: Ilustración 21 - Tornillo sin fin en 3D

Construcción del prototipo del alimentador

Desde la ilustración 24 hasta la 33 (Sandoval & Sepúlveda, Construcción del prototipo), se pueden apreciar las diferentes etapas durante la construcción del alimentador.



Anexo: Ilustración 25 - Construcción del prototipo (Parte 1)



Anexo: Ilustración 24 - Construcción del prototipo (Parte 2)



Anexo: Ilustración 27 - Construcción del prototipo (Parte 3)



Anexo: Ilustración 26 - Construcción del prototipo (Parte 4)



Anexo: Ilustración 31 - Construcción del prototipo (Parte 5)



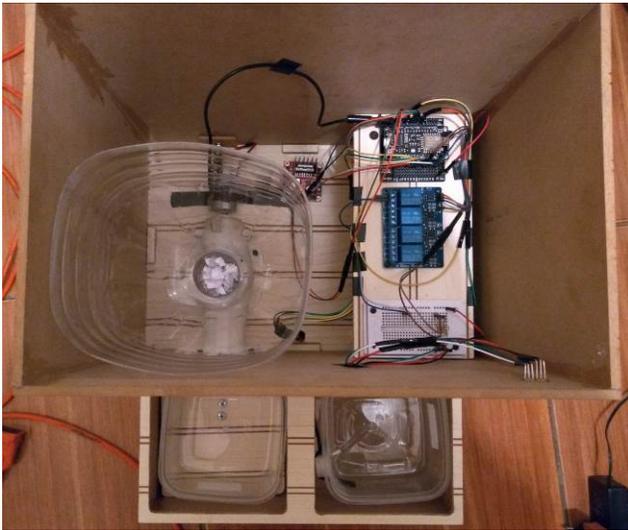
Anexo: Ilustración 30 - Construcción del prototipo (Parte 6)



Anexo: Ilustración 29 - Construcción del prototipo (Parte 7)



Anexo: Ilustración 28 - Construcción del prototipo (Parte 8)



Anexo: Ilustración 33 - Construcción del prototipo
(Parte 9)



Anexo: Ilustración 32 - Construcción del prototipo
(Parte 10)

Prototipo finalizado del alimentador

Desde la ilustración 34 hasta la 38, se pueden apreciar las diferentes etapas durante la construcción del alimentador.



*Anexo: Ilustración 35 – Prototipo finalizado
(Desarmado 1)*



*Anexo: Ilustración 34 - Prototipo finalizado
(Desarmado 2)*



Anexo: Ilustración 37 - Prototipo finalizado (Vista trasera)



Anexo: Ilustración 36 - Prototipo finalizado (Vista superior sin tapa)



Anexo: Ilustración 38 - Prototipo finalizado (Vista frontal)