

Diseño e implementación de un prototipo de dispensador automático de comida para animales basado en raspberry pi controlado mediante una aplicación móvil.

Jorge Iván Zapata Valencia
Daniel Alejandro Gil Agudelo

Proyecto de grado para optar al título de
Ingeniero Electrónico

Universidad Tecnológica de Pereira
Facultad de Ingenierías
Ingeniería Electrónica
Pereira, Colombia
Enero 2017

Diseño e implementación de un prototipo de dispensador automático de comida para animales basado en raspberry pi controlado mediante una aplicación móvil.

Por:

Jorge Iván Zapata Valencia
Cód: 1087998966
Daniel Alejandro Gil Agudelo
Cód: 75158899

Proyecto de grado para optar al título de
Ingeniero Electrónico

Director:

M.Sc. José Andrés Chaves Osorio
Profesor del Programa Ingeniería Electrónica
Ingeniero Electricista

Universidad Tecnológica de Pereira
Facultad de Ingenierías
Ingeniería Electrónica
Pereira, Colombia
Enero de 2017

Nota de aceptación:

Director:

Jurado:

Enero de 2017

Agradecimientos

He recorrido un arduo camino para obtener este importante logro en mi vida, pero no he llegado solo hasta aquí, cada paso, cada aprendizaje y cada resurgir, ha estado acompañado del apoyo incondicional de mi familia, que ha sido mi guía y mi fortaleza, por quienes recorrería mil caminos más si fuera necesario; agradezco además por las sabias enseñanzas de mis abuelos y por cada esfuerzo suyo, por los nobles consejos y la inmensa paciencia de mi novia que ha sido partícipe de esta evolución, por los aportes del ingeniero Alexander Ríos gracias a los cuales se materializó un sueño, al ingeniero José Andrés Chaves por creer en nosotros y motivarnos a ser mejores cada día, al ingeniero Edwin Andrés Quintero por inspirarnos, a amigos y futuros colegas por hacer el camino más interesante y sobretodo a Dios por permitirme vivir estas experiencias y adquirir valiosos aprendizajes de ellas, por darme la oportunidad de construir un futuro y por poner a cada uno de los nombrados en mi camino.

Jorge Iván Zapata Valencia

Para obtener un gran logro toca vivir un camino de triunfos, derrotas, tristezas y alegrías, por eso se le debe agradecer a DIOS por la fortaleza, por la salud, por mi familia y por darme la oportunidad de lograr el sueño de ser Ingeniero Electrónico. La obtención de este reconocimiento no es solo mío, detrás está mi familia, que me encaminaron a ser un profesional, que me han dado un sin número de valores que me han ayudado a obtener tantos éxitos, por el amor, la paciencia, el respeto, por estar ahí cuando los necesitaba y por todos los días impulsarme a ser una mejor persona. También agradecer a la vida por las experiencias vividas, por las grandes amistades que forje durante mi ciclo por la universidad, amistades que te muestran puntos de vista diferentes de la vida y que estaban para darte la mano cuando no te las sabías todas, amistades que son para toda la vida. Inmensa gratitud para la Universidad Tecnológica de Pereira por la calidad de educación que ofrecen, para sus docentes que impulsan, orientan y brindan sus conocimientos.

Daniel Alejandro Gil Agudelo

Índice general

1. DESCRIPCIÓN DEL PROYECTO	11
1.1. DEFINICIÓN DEL PROBLEMA	11
1.2. JUSTIFICACIÓN	12
1.3. OBJETIVOS	13
1.3.1. OBJETIVO GENERAL	13
1.3.2. OBJETIVOS ESPECÍFICOS	13
2. MARCO REFERENCIAL	14
2.1. MARCO TEÓRICO	14
2.1.1. Leyes, Estadísticas y Vínculos Afectivos	14
2.1.2. Mercado de Alimentos para Perros	17
2.1.3. Cómo Alimentar Correctamente a la Mascota	18
3. DESARROLLO DEL PROYECTO	24
3.1. Diseño Comedero de Mascotas en Solidworks	24
3.1.1. ¿Por qué la utilización de solidworks para el desarrollo del diseño?	25
3.1.2. Piezas que conforman el sólido final.	25
3.2. Diseño de Programa de Control, Interfaz Gráfica y Tarjeta de Potencia.	35
3.2.1. ¿Qué es Raspberry Pi?	35
3.2.2. Pines GPIO de la Raspberry Pi	36
3.2.3. Lenguaje PYTHON	37
3.2.4. Servidor Web Apache	38
3.2.5. Lenguaje HTML	39
3.2.6. Lenguaje PHP	42
3.2.7. Tarjeta de Potencia	42
3.3. Diseño e Instalación de la Aplicación Móvil.	45
3.3.1. Aplicación Móvil, Tecnología 4G e Internet.	45
3.3.2. Sistema Operativo Android	46
3.3.3. Creación Aplicación Móvil	47
3.3.3.1. Java Development Kit (JDK):	47
3.3.3.2. Android Studio:	48
3.3.3.3. Componentes de la Aplicación	50
3.3.4. Descripción Final de la Aplicación y Conceptos Claves para su Creación.	61
3.3.5. Instalación de la Aplicación Móvil.	61
3.3.5.1. Generación de Certificado y Correspondiente APK	61
4. PRUEBAS Y RESULTADOS	65

5. CONCLUSIONES Y FUTUROS TRABAJOS	76
5.1. Conclusiones	76
5.2. Futuros Trabajos	77

Índice de figuras

3.1. Pieza Superior	25
3.2. Rampas de Caída	26
3.3. Recipiente Base	26
3.4. Cavidad Circular	27
3.5. Unión de Piezas	27
3.6. Tapa Exterior	28
3.7. Tapa Superior Derecha	28
3.8. Tapa Frontal	29
3.9. Tapa Izquierda	29
3.10. Tapa Trasera	30
3.11. Tapa Techo	30
3.12. Cavidad Circular	31
3.13. Tornillo Sin Fin	31
3.14. Cálculos Tornillo Sin Fin	32
3.15. Pieza Tuerca - Tornillo	33
3.16. Proceso Selección Documento	33
3.17. Comedero Parcialmente Ensamblado	34
3.18. Comedero Totalmente Ensamblado	34
3.19. Tarjeta Raspberry Pi 3	36
3.20. Pines de la Raspberry Pi	36
3.21. Estructura de Red Implementada en el Proyecto	39
3.22. Interfaz Gráfica	41
3.23. Motoreductor	43
3.24. Circuito de Potencia	44
3.25. Suscriptores	45
3.26. Abonados Telefonía Móvil	46
3.27. Aplicación JAVA	48
3.28. IDE Android Studio	48
3.29. Sistemas Operativos Android	49
3.30. Interfaz Inicial Android Studio	49
3.31. Primer Activity	50
3.32. Segunda Activity	51
3.33. Tercera Activity	51
3.34. Cuarta Activity	52
3.35. Quinta Activity	52
3.36. RelativeLayout	52
3.37. Tipos de Layouts	53
3.38. Ejemplo RelativeLayout	54
3.39. MainActivity.java	56

3.40. Main2Activity.java Tabla 1	57
3.41. Main2Activity.java Tabla 2	57
3.42. Main4Activity.java	58
3.43. Main3Activity.java	59
3.44. Main5Activity.java Tabla 1	60
3.45. Main5Activity.java Tabla 2	60
3.46. Generación APK	61
3.47. Selección del Módulo	62
3.48. Asignación de Credenciales	62
3.49. Formulario de Datos Básicos	62
3.50. Exportación de Fichero	63
3.51. Configuración Permisos Smartphone	63
3.52. Advertencia de Seguridad	64
3.53. Instalación APK en Smartphone	64
4.1. Piezas Corel Finas	66
4.2. Piezas Corel Cortes	66
4.3. Primeros Cortes en Cortadora Láser	67
4.4. Prototipo Armado en Cartón	67
4.5. Piezas Cortadas en Acrílico	68
4.6. Piezas Finas Cortadas en Acrílico	68
4.7. Ensamble de Piezas	69
4.8. Tornillo y Base Impresos	69
4.9. Compartimento para Tornillo	70
4.10. Acople Tornillo en el Compartimiento	70
4.11. Piezas para Adherir los Componentes	71
4.12. Comedero Ensamblado Totalmente	71
4.13. Vista Superior Comedero Ensamblado	72
4.14. Vista Interna Comedero	72
4.15. Instalación Aplicación en Smartphone	73
4.16. Interfaz Inicial Aplicación Comedero	73
4.17. Interfaz Final de la Aplicación	74
4.18. Prueba Funcionamiento Comedero	74

Índice de cuadros

2.1. Necesidades Diferentes Nutrientes	19
2.2. Proteína y Grasa	20
2.3. Minerales	20
2.4. Vitaminas	21
2.5. Ración Diaria	22
3.1. Especificaciones Técnicas Motor	43
3.2. Porcentaje de Ventas Smartphones	47
4.1. Costos del Prototipo	75

Capítulo 1

DESCRIPCIÓN DEL PROYECTO

1.1. DEFINICIÓN DEL PROBLEMA

El crecimiento poblacional conlleva a la aparición de necesidades que deben ser satisfechas con el apoyo de la tecnología, se debe plantear un desarrollo obligado en todos sus campos para buscar soluciones que traigan consigo innovación y agilización; se está viviendo la era donde el ser humano desea realizar una gran cantidad de tareas de una manera rápida y eficiente, por ello las soluciones deben ser prácticas, sencillas, eficientes y oportunas.

Las mascotas hacen parte de este crecimiento. La estrecha relación ser humano mascota se hace cada día más notoria, pasando de ser la mascota de la casa, a ser parte fundamental del núcleo familiar, por tal razón, sus dueños buscan la manera de brindar soluciones tecnológicas a sus necesidades. A menudo las personas dueñas de mascotas sienten una enorme preocupación al salir de sus hogares y no poder llevarlas consigo (por factores ajenos a ellos), debido a esta situación, habitualmente intentan realizar sus diligencias en el menor tiempo posible para así poder regresar pronto a sus hogares y no sentir que abandonan a sus mascotas.

Un factor decisivo es su alimentación, ya que su comida en ocasiones debe ser suministrada constantemente y en ciertas porciones o debe ser servida en horarios específicos, en muchas ocasiones influyen circunstancias tales como el olvido o en otros casos el buscar ahorrar tiempo; para evitar todas estas preocupaciones y darles solución se propone construir un dispensador controlado por una aplicación móvil, el cual le permita a las personas alimentar sus mascotas sin la necesidad de estar presente en los hogares, bastaría solamente acceso a internet y unos cuantos comandos utilizando un smartphone. Actualmente en el mercado se pueden encontrar dispensadores automáticos para mascotas como Petnet, el cual cuenta con una aplicación para la plataforma de smartphones iOS, lastimosamente su precio es bastante elevado. Bistro es otro dispensador automático, este, basa su funcionamiento en una cámara que reconoce la mascota y puede medir su tamaño, sin embargo, aún se encuentra en fase de desarrollo. Trabajos de grado similares en la Universidad Tecnológica de Pereira no han desarrollado todavía prototipos físicos y son controlados mediante bluetooth o simplemente programables.

El prototipo contará con una aplicación móvil para la plataforma Android inicialmente y tendrá como principal componente la tarjeta Raspberry Pi, esto permite disminuir el costo del dispensador, los costos para el usuario final e incrementar la cobertura en el mercado ya que los usuarios que utilizan dispositivos con plataforma Android representan la mayoría de los smartphones usados en el mundo [1].

1.2. JUSTIFICACIÓN

El desarrollo de este trabajo atiende parte de la problemática actual que viven miles de mascotas en el país y millones en el mundo a causa del poco tiempo que pasan sus dueños en el lugar donde reside la mascota y de los problemas de salud que les acarrea la mala dosificación del alimento.

Diversos estudios indican que los animales domésticos que consumen alimentos sin un control horario o raciones específicas disminuyen aproximadamente en un 15 % su expectativa de vida y son más propensos a múltiples padecimientos de salud debido a la obesidad [2]. Lo anterior soportado en el Estatuto Nacional de Protección Animal y su actual reforma, la ley 1774 de 2016, la cual demanda en el artículo 3, punto b1, en cuanto al principio de bienestar animal, que los animales no deben sufrir hambre, ni sed. Para dar solución a la problemática se propone un mecanismo de alimentación a distancia el cual permite darle a la mascota el concentrado y las cantidades a las horas más adecuadas, proporcionadas por su amo sin importar dónde este se encuentre.

Con los comederos automatizados, la dificultosa tarea de alimentar adecuadamente a la mascota por distintos factores, queda supeditada al simple llenado de la tolva una vez cada dos o tres días, dependiendo la dosificación programada para cada ración de concentrado en función de la edad y la raza. También se debe recalcar que con el tiempo se ha desarrollado una estrecha conexión entre el amo y su mascota y en muchas ocasiones esta pasa a convertirse en un miembro importantes del núcleo familiar y no ser solo visto como un animal sino como un "ser sintiente" que necesitan y merecen protección [3].

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Diseñar e implementar un prototipo de dispensador automático de comida para (perros de raza pequeña) animales basado en Raspberry Pi controlado mediante una aplicación móvil.

1.3.2. OBJETIVOS ESPECÍFICOS

- Realizar el diseño de la estructura física del prototipo de acuerdo a los parámetros para la creación de un producto.
- Diseñar e implementar el circuito electrónico que se acople con la tarjeta Raspberry Pi y que permita el control del sistema de acuerdo a los requerimientos del diseño físico.
- Desarrollar una aplicación móvil que permita la comunicación entre la tarjeta Raspberry Pi y un Smartphone.
- Comprobar el funcionamiento del prototipo en un ambiente real.

Capítulo 2

MARCO REFERENCIAL

2.1. MARCO TEÓRICO

2.1.1. Leyes, Estadísticas y Vínculos Afectivos

Para dar solución al problema mencionado anteriormente se debe tener en cuenta lo siguiente:

- **Leyes que protegen los animales:** Toda persona que tenga a cargo una mascota debe cumplir con ciertos aspectos que son primordiales y elementales según el Estado de Nacional de Protección Animal Ley 84 de 1989 – Colombia [4], entre uno de ellos se destacan en dicha ley en el Capítulo II de los deberes para con los animales en el Artículo 5 en uno de sus párrafos se expresa claramente que se debe suministrar bebida, alimento, en cantidad y calidad suficientes, así como medicinas y cuidados necesarios para asegurar su salud, bienestar y para evitar daño, enfermedad o muerte y anexa en el Parágrafo: Cuando se trata de animales domésticos o domesticados, en cautividad o confinamiento, las condiciones descritas en el presente Artículo deberán ser especialmente rigurosas, de manera tal que los riesgos de daño, lesión, enfermedad o muerte sean mínimos.

Por ende cada persona que tiene a cargo una mascota asume una responsabilidad con la cual debe cumplir. El pasado noviembre 11 de 2015 [5], por decisión unánime la Comisión Primera del Senado, aprobó en tercer debate el Proyecto de Ley 172 de 2014, que busca endurecer las penas y sanciones a los que violen los derechos de los animales.

- **Cantidad de personas que tienen a cargo una mascota:** Si se toma como punto de partida la ciudad de Pereira, según un reconocido diario en una de sus publicaciones menciona que la firma GFK (es la compañía de investigación de mercados más grande de Alemania y la quinta más grande del mundo). en el año 2012 realizó una encuesta tirando números increíbles sobre el crecimiento acelerado de mascotas en los hogares [6], según lo suministrado por dicha firma el 29 % de los hogares colombianos tiene una mascota. En Pereira, el registro de la Oficina de Zoonosis de la Secretaría de Salud, estima que hay 44 mil animales de compañía, (perros y gatos), de los cuales el 98 % tiene un doliente, es decir, aunque vivan en la calle, alguna persona se encarga de darles comida y estar pendiente de ellos.

En el año 2006 se cree había un estimado de 22 mil animales de compañía donde en este número la mayoría fue abarcada por perros y gatos, demostrado así el gran crecimiento de mascotas en los hogares pereiranos en los últimos años, actualmente no se encontró un nuevo registro o alguna firma que haya realizado una nueva encuesta, pero si se toma la curva de crecimiento que se viene planteando actualmente en la ciudad de Pereira debe existir un estimado “más o menos” de 55 a 60 mil, destacando variables como el crecimiento poblacional para la realización de un nuevo estudio. El estudio

realizado no nombró la cantidad en este caso de canes en las calles, pero según datos de la Secretaría de Salud de Pereira en el 2012 había 8394 perros callejeros [7], porque se debe tener en cuenta estos números, porque actualmente la ciudad cuenta con albergues y fundación de protección y adopción, incrementando con estos números la posibilidad de aumento en el registro de posibles personas que pueden llegar a tener en su hogar una mascota.

Si se extiende el estudio a nivel nacional y directamente enfocado a los caninos, la firma GFK estimo a cinco países de América Latina que hicieron parte de su estudio, donde el perro lidera con un 83 % de preferencia entre la gente que suele tener animales en el hogar [8]. En segundo lugar se encuentra el gato, con un 22 %, seguido de los pájaros, con 11 %, los peces registran 5 % y en último lugar, algún otro tipo de mascotas con 4 %.

En México el 57 % de la población cuenta con una mascota, en Ecuador el 48 %, en Panamá el 47 %, en Venezuela el 38 % y en Colombia el 29 % y de éstos, el 80 % de los mexicanos tiene a un perro como mascota en su casa, 87 % los venezolanos, 80 % los colombianos, 82 % los ecuatorianos y 93 % de los panameños, tiene uno.

“El mejor amigo del hombre”, es también el preferido en las más grandes ciudades de Colombia: Medellín y Barranquilla, en donde el 86 % de quienes tienen mascota, mencionan tener uno, en tanto que en Bogotá el registro fue del 80 % y en Cali del 74 %. De acuerdo con la encuesta realizada por GFK en el país los perros son “reyes”, seguidos por los gatos.

- **Vínculo afectivo con los animales de compañía:** ¿Será que en la actualidad una mascota está a la par con un ser humano? Para muchas personas así puede ser, porque ven en su mascota un amigo, un integrante más de su familia, al cual se le brinda cariño, respeto y se le deben suplir sus respectivas necesidades y porque no llenarlo de lujos y comodidades, al fin y al cabo quién no quiere ver a la persona que te brinda un cariño incondicional bien, en este caso “La mascota”.

Algo muy visto desde siempre es el estrecho vínculo persona-mascota, pero este vínculo sigue siendo aún más fuerte con el pasar de los días. La mascota es la ayuda para aquella persona que tiene alguna discapacidad, pues brinda apoyo y representa una nueva opción de terapia alternativa para el tratamiento de diferentes padecimientos tanto físicos como emocionales, posicionándolo así como una ayuda medicinal [9].

Actividades Asistidas por Animales (AAA)

Consiste en actividades que involucran animales que visitan a personas. Estas actividades pueden ser llevadas a cabo con todo tipo de personas. La AAA provee oportunidades para beneficiarse de manera motivacional, educacional, recreativa y aumentando la calidad de vida. Estas actividades no tienen una meta específica, sólo representan sesiones de interacción en las cuales el animal visita a los pacientes para permitir el contacto.

Cuando el animal terapéutico está presente, las personas se sienten más seguras y cómodas para iniciar una conversación con otra persona aunque esta sea un extraño, ya que el animal provee de un tema neutro de conversación.

En la AAA el animal actúa como un “lubricante social” favoreciendo la conversación y facilitando establecer un vínculo con los participantes, lo cual resulta útil a futuro para otros procedimientos terapéuticos [10].

Terapia Asistida con Animales (AAT)

Se define a la AAT como una actividad que involucra a un profesional de la salud o de servicio

social que usa animales como parte de su trabajo. Las metas específicas para cada paciente han sido previamente determinadas por el terapeuta, todo el avance del programa es medido y documentado. La AAT representa una intervención dirigida a cumplir objetivos, en donde el animal y el facilitador cumplen con criterios particulares como parte integral del tratamiento. La AAT está diseñada para promover el mejoramiento en las diferentes áreas de desarrollo de la persona.

Por lo general en la AAT se han usado perros, gatos y peces; sin embargo, en la actualidad se usan también conejos, serpientes y hámster. Las aves no son permitidas para esta terapia debido a que son portadoras de diferentes enfermedades infecciosas como la fiebre aviar la cual puede ser transmitida por medio de las plumas [10].

Beneficios de la Terapia Asistida con Animales

Estudios del vínculo animal - persona han revelado beneficios tanto físicos como psicológicos. En donde el acariciar a una mascota puede proveer relajación, baja de la presión arterial, y mejoras en la temperatura corporal.

La mayoría de las personas que adquieren mascotas lo hacen con un fin social. Poseer una mascota disminuye el sentimiento de soledad y aumenta el bienestar físico y psicológico, y también sirve para disminuir la ansiedad.

Las personas presentan una relación especial con sus mascotas que no encuentran con otras personas, "Los animales tienen una forma de aceptación sin juicios, no condicionan su amor, perdonan fácilmente". Es por esto que las relaciones con los animales resultan menos amenazantes que las relaciones con otras personas.

En un mundo en donde las relaciones con las personas pueden ser demandantes, estresantes, llenas de conflicto, los animales proveen al ser humano un descanso al solo buscar la compañía de su dueño.

Por ejemplo, en niños el tener mascotas ayuda a desarrollar el sentimiento de responsabilidad, competencia, autoestima, confianza, y sentimientos de empatía. La mascota provee al niño no solo de un amigo de juegos, sino de un confidente y alguien con quien practicar las habilidades de socialización, al mismo tiempo, provee una de las primeras experiencias de la muerte y la pérdida.

Este vínculo se puede ver reflejado como un cariño mutuo, que con el pasar del tiempo se hace aún mayor, es tanto así que según Investigadores de la Universidad de Azabu (Japón) creen que dicho vínculo afectivo entre el perro y su dueño es recíproco y se establece a través de la mirada [11].

Dicha universidad ha descubierto que la oxitocina, sustancia química conocida como la «hormona del amor» ya que se relaciona con el apego materno infantil, también puede generar vínculos afectivos entre miembros de especies diferentes y explicar así la unión emocional que muchos humanos mantienen con sus perros. Además, han visto que el aumento de dicha hormona se desencadena con una simple mirada. La oxitocina actúa como neurotransmisor en el cerebro de los seres vivos porque funciona en el núcleo accumbens, al que se le atribuye una función importante en el placer. Al crecer el vínculo, con ello crece el interés por estudiar más afondo esta conexión y tratar de demostrar a que se debe esta concordancia en la relación entre dueños y mascotas. Se han realizado varios experimentos con perros hembras y machos de diferentes razas y edades y sus dueños, para de este modo determinar como perros y humanos pueden reaccionar y reforzar sus vínculos biológicos en un circuito neuronal de retroalimentación impulsado por la oxitocina, tal y como ocurre entre padres e hijos, gracias a una simple mirada mutua. "Cuando perro y dueño se miran, ambos muestran un aumento de la oxitocina", según Takefumi Kikusui, investigador en el departamento de Ciencia Animal y Biotecnología de la Universidad Azabu (Japón) [11]. Para tomar como partida dichos estudios se debieron realizar expe-

rimento, estableciendo parámetros de ciertos de interacción mutua entre el perros y su dueño durante un determinado tiempo y registrar cada acción al hablar, tocar y mirar entre las dos especies.

2.1.2. Mercado de Alimentos para Perros

El diario El Tiempo en su página web en su columna de economía publicó el 9 de septiembre de 2014 [12] una nota acerca de la producción de alimentos para mascotas, la cual título “Mascotas tienen a tope producción de alimentos balanceados” y resalta con una frase “Las principales empresas del sector están invirtiendo en ensanches”, resaltando la importancia y el auge que ha tomado para las personas el tener su mascota y alimentarla bien y como las grandes empresas ven con buenos ojos esto y ven una fiable y sostenible posibilidad de inversión.

En dicha publicación dicen que cerca de 4,3 millones de perros y 1,4 millones de gatos, que son respectivamente las poblaciones de estos animales en Colombia, según la firma Euromonitor International [13]. El crecimiento del mercado de las mascotas tiene a los productores y comercializadores de alimentos balanceados ampliando afanosamente sus plantas. Se estima que cada año el mercado de alimentos para mascotas sigue creciendo en el país, recibiendo grandes ganancias y recaudos para los inversionistas, que ven este como un negocio seguro y en constante crecimiento.

Además no solo las inversiones van sujetos a la alimentación de la mascota también hay que tener en cuenta que este es un mercado demasiado extenso y que necesita de innovación, muchos inversionistas dan manejo de cifras, promueven páginas de internet para brindar asesoría sobre las distintas problemáticas o beneficios para la mascota, se ofrecen servicios de paseos diarios, al negocio de las mascotas se les pueden nombrar de distintas maneras ya sean vistos como negocios “perrunos” o más bien un asunto público que compete a muchos.

En internet hay numerosas sitios web que hacen publicaciones, balances generales y demuestra con cifras y estadísticas la importancia de hoy por hoy de las mascotas y como influyen en cambios económicos, culturales y porque no sociales en todo un país. El mercado de mascotas es extenso y sigue creciendo enormemente creando día a día necesidades a sus dueños, los cuales no escatiman en gastos para darles solución.

Un ejemplo claro que muestra la positiva dinámica del negocio de comida para perros [14] es el portal de Internet Ciudad de Mascotas. La página que comercializa productos para perros y gatos nació como un emprendimiento en enero de 2013 y a la fecha tiene más de 40.000 personas registradas y más de 6.000 clientes recurrentes.

Dicho portal en agosto del 2015 logro captar la atención de un fondo de inversión de gran rango en nuestro país y proveedores de servicios en distintas ramas, dicho fondo estaba compuesto por EPM, Sura y el Banco Interamericano de Desarrollo (BID), el cual ha invertido unos \$900 millones.

Como se ha señalado varias veces al ser un mercado enorme, también genera enormes ingresos, ingresos que facturan fácilmente millones cada año, siendo cifras que fácilmente pueden ser duplicadas.

Actualmente en el mercado existen innumerables marcas de productos para mascotas, al igual que marcas codificadas que fabrican todo tipo de alimentos. Los fabricantes promueven sus vitaminas, proteínas y demás componentes y beneficios los cuales son plasmados en tablas alimenticias expuestas en sus empaques, ya establecer qué tipo de alimento es mejor para la mascota está determinado dependiendo del tamaño, masa, raza, recomendaciones del veterinario o cual puede suplir más sus necesidades, como se sabe estos pueden varían según su calidad y claro de su precio en el mercado actual, entre los alimentos concentrados para mascotas se destacan los siguientes:

- **Premium:** Contienen Carne Fresca, sub productos animales, incluyen maíz y otros Granos (en menos proporción), la proteína es de muy buena calidad y mucho más digerible y aprovechable, la mayoría de la proteína proviene de una mezcla de granos con algo de carne (poca).
- **Superpremium:** holísticas naturales, libres de granos, vitaminas E, C y MOS, Luteína y Taurina. Nutrientes que ayudan a prevenir la aparición de esta placa dental, etc.

En el mercado, hay una gran cantidad de marcas que están compitiendo con las tradicionales, brindando productos que se acomoden más que todo a la comodidad del dueño de mascota, como hay productos de excelente calidad, también hay productos que no lo son tanto como lo son los concentrados regulares que en sus tablas de alimentación están algo por debajo de las marcas Premium y Superpremium, pero que suplen la necesidad de alimentación de la mascota.

Los empresarios promedian los gastos que realiza el dueño de mascota y chequean el consumo mensual que conlleva este, en los cuales incluyen juguetes y accesorios, pero se estimula que dicho promedio en su gran parte va determinado para la compra del alimento. Los gastos van determinados muchas veces según el tamaño del canino. Una raza pequeña come 4 kilos; una mediana 12 kilos; y si es una de las grandes se puede comer 15 kilos al mes aproximadamente.

2.1.3. Cómo Alimentar Correctamente a la Mascota

Existen varios métodos de alimentación para mascotas que se consideran los más acertados, hay instituciones dedicadas a realizar estudios especializados sobre cómo debe ser la forma correcta y de cómo se debe brindar una alimentación balanceada y libre de bacterias u otras enfermedades que no generen problemas en el sano vivir tanto de la mascota como del amo.

AAFCO (*Association of American Feed Control Officials*) es una de las asociaciones más grandes a nivel mundial, quien ha realizado estudios muy completos sobre cómo deben ser las regulaciones en la creación de alimentos para mascotas, los estándares a seguir, los controles alimenticios a tener en cuenta, entre muchas otras cosas [15].

Si se tiene una mascota, se es absolutamente responsable de su salud y alimentación. La mascota no puede decidir la base de su dieta y el amo es el responsable, no solamente en elegir el tipo de comida que aporte los nutrientes necesarios al amigo, sino también se encargará de determinar la cantidad justa de las raciones.

Como se reseñó anteriormente la mayoría de las personas tienen en su casa como mascota un gato o un perro. Conocer las pautas de una alimentación sana para el felino es muy importante y tener como punto de partida sus características genéticas. Se sabe que este es un animal de tipo carnívoro con necesidades energéticas específicas, que con el tiempo se ha adaptado a cierta dieta especializada. Es importante ofrecerle un concentrado adecuado e impedir que coma sobras, al conocer sus necesidades y sus hábitos, se le debe entonces proporcionar un alimento que le brinde sobre todo compuesto a base de carne y que contengan un aminoácido importante para el funcionamiento de su metabolismo, la taurina debe estar en sus componentes alimenticios para de estar forma potencializar su condición física.

Como todos hemos comprobado, el gato se muestra mucho más exigente que el perro al elegir sus alimentos pudiendo ayunar largo tiempo si una comida no es de su agrado. En gran medida este aspecto está influenciado por los hábitos alimenticios en sus primeros meses de vida ya que tiende a comer y aceptar aquellos alimentos a los que se habituó en este periodo.

Con respecto a la frecuencia de alimentación, el gato es selectivo, prefiere echar a un lado alimentos apetitosos y tomar muchas comidas ligeras, entre 10 y 20, en lugar de dos o tres comidas abundantes [16].

Los felinos prefieren que la temperatura de su comida se encuentre de forma ideal entre 38° C y 40° C, ya que sus presas suelen tener esta temperatura, de esta forma se estimula su instinto de cazar. Con respecto al agua la prefieren fresca e idealmente en movimiento, la cantidad que estos consumen es variable e indeterminada. Situaciones de estrés influyen enormemente en su deseo de comer, el cambio de residencia de forma habitual o una hospitalización pueden suponer que un gato no coma durante varios días.

Necesidades Energéticas del Gato Adulto

La mayoría de los gatos caseros, no realizan esfuerzo físico regularmente ni se ven sometidos a condiciones ambientales extremas. El gato en su transición de cachorro a su etapa adulta difiere muy poco en tamaño y conformación corporal, oscilando la media entre un peso de 3 kg y 6 kg. En estas condiciones, el gato necesita energía para mantener su temperatura corporal constante (la temperatura normal en el gato es 38,5° C) y mantener su organismo en estado de renovación de tejidos y órganos, asegurando todas las funciones esenciales como son digestión, respiración, circulación, etc. Por tanto se ha calculado que la mayoría de los gatos necesitan unas 60 kcal (kilocaloría) de energía metabolizable (EM) por kg de masa al día mientras que en el caso de gatos más activos, que puedan salir al exterior o estén en condiciones de estrés, sus necesidades oscilan entre 70 kcal y 90 kcal de EM por kg de masa y día. (Cuadro 2.1).

Con respecto a las necesidades específicas de cada nutriente, comentar que las necesidades en proteínas de los gatos son mucho mayores que las necesidades del perro. El pienso compuesto o concentrado, es un alimento elaborado para animales que según la normativa legal europea, es cualquier sustancia o producto, incluido los aditivos, destinado a la alimentación por vía oral de los animales.

Proteínas	Gatos Adultos	Mínimo de 20 % de la EM
Arginina	Gatitos	Mínimo de 30 % de la EM
Taurina	Cualquier Fase	1000 mg/kg de materia seca piensos 2500 mg/kg de materia seca latas
Calcio	Gatitos	0,8 % - 1.5 %
Fósforo	Gatitos	0,7 % - 1.2 %
Ac. Linoleico	Cualquier fase	6 % de la EM
Ac. Araquidónico	Cualquier fase	0,2 % de la EM
Magnesio	Cualquier fase	0,04 % a 0,1 %

Cuadro 2.1: Necesidades Diferentes Nutrientes

Necesidades de Diferentes Nutrientes en los Alimentos Formulados para Gatos.

AAFCO (*Association of American Feed Control Officials*) revisó los estándares que existían y creó unas tablas de necesidades básicas en cachorros y reproducción y en perros adultos, expresadas en materia seca (MS) [16].

Como en los piensos las cantidades de nutrientes van expresadas en términos de ración, para comparar ambos datos hay que tener en cuenta el contenido de humedad. Por ejemplo, si un pienso tiene un 10 % de humedad (90 % de materia seca) y un 20 % de proteína, se divide 20 entre 0,90 lo que dará un 22 % de proteína, con lo cual el pienso cubriría los mínimos. Y así tendríamos que hacer con todos los nutrientes. A continuación se presentan las tablas con los resultados:

NUTRIENTE	Unidad MS	Crecimiento y Reproducción	Mantenimiento Adultos	Máximo
Proteína	%	22,0	18,0	-
Arginina	%	0,62	0,51	-
Histidina	%	0,22	0,18	-
Isoleucina	%	0,45	0,37	-
Leucina	%	0,72	0,59	-
Lisina	%	0,77	0,63	-
Metionina-cisteína	%	0,53	0,43	-
Fenilalanina-tirosina	%	0,89	0,73	-
Treonina	%	0,58	0,48	-
Triptófano	%	0,20	0,16	-
Valina	%	0,48	0,39	-
Grasa	%	8,0	5,0	-
Ácido Linoléico	%	1,0	1,0	-

Cuadro 2.2: Proteína y Grasa

MINERALES				
NUTRIENTE	Unidad MS	Crecimiento y Reproducción	Mantenimiento Adultos	Máximo
Calcio	%	1,0	0,6	2,5
Fósforo	%	0,8	0,5	1,6
Relación Ca:P		1:1	1:1	2:1
Potasio	%	0,6	0,6	-
Sodio	%	0,3	0,06	-
Cloro	%	0,45	0,09	-
Magnesio	mg/kg	0,04	0,04	0,3
Hierro	mg/kg	80,0	80,0	3000,0
Cobre	mg/kg	7,3	7,3	250,0
Manganeso	mg/kg	5,0	5,0	-
Zinc	mg/kg	120,0	120,0	1000,0
Yodo	mg/kg	1,5	1,5	50,0
Selenio	mg/kg	0,11	0,11	2,0

Cuadro 2.3: Minerales

VITAMINAS				
NUTRIENTE	Unidad MS	Crecimiento y Reproducción	Mantenimiento Adultos	Máximo
Vitamina A	IU/kg	5000,0	5000,0	25000,0
Vitamina D	IU/kg	500,0	500,0	5000,0
Vitamina E	IU/kg	50,0	50,0	1000,0
Vitamina B1 (tiamina)	mg/kg	1,0	1,0	-
Vitamina B2 (riboflavina)	mg/kg	2,2	2,2	-
Vitamina B3 (niacina)	mg/kg	11,4	11,4	-
Vitamina B5 (ácido pantoténico)	mg/kg	10,0	10,0	-
Vitamina B6 (piridoxina)	mg/kg	1,0	1,0	-
Vitamina B9 (ácido fólico)	mg/kg	0,18	0,18	-
Vitamina B12 (cianocobalamina)	ug/kg	22,0	22,0	-
Colina	g/kg	1,2	1,2	-

Cuadro 2.4: Vitaminas

Sin embargo es muy importante aprender a conocer los distintos tipos de comida que existen en el mercado para poderles dar el uso más correcto [17].

En la etiqueta deben ser bien visibles los componentes del producto, entre los que se establecen:

- Los alimentos simples (son de origen animal o vegetal).
- Los alimentos compuestos (son mezclas de productos de origen animal o vegetal).
- Los alimentos integrados (son comidas simples o compuestas a las que se les añade integradores, ya sean vitaminas y/o minerales).

Otra manera de conocer las comidas es el grado de humedad que contienen:

Las comidas secas tienen una humedad inferior o igual al 14 %, y están compuestas por mezclas de cereales, harinas de pescado y carne, leche en polvo, soja, integradores vitamínicos minerales y grasas. Son menos apetitosas que las semi húmedas enlatadas, pero se conservan por más tiempo, son más económicas y fáciles de utilizar.

Las comidas húmedas tienen un grado de humedad superior al 34 %, y vienen enlatadas o ultracongeladas. Son bastante caras pero muy apetitosas. Sus componentes son residuos de carnicerías, cereales, verduras y soja.

Cantidad de Comidas Diarias

- Hasta los 3 meses: 4 comidas.
- De los 3 meses a los 6 meses: 3 comidas.
- De los 6 meses a los 17 meses: 2 comidas.
- Después de los 18 meses se podrá optar por 1 comida (se recomienda siempre 2 ingestas para una mejor asimilación).

Alimentar a un perro es algo que hay que aprender, no es un secreto que de la alimentación depende gran parte la salud del animal, también se debe aprender a no sobrealimentarlo; especialmente en las primeras etapas de la vida. Una deficiente alimentación puede ser la sentencia para poseer un perro débil y enfermo.

Es primordial contar con una excelente alimentación en los canes; en la actualidad, una gran demanda de alimento para mascotas ha impulsado a las grandes marcas a ofertar concentrados de diferentes tipos, algunos llamados balanceados que en ocasiones confunden al propietario del perro, por ello la importancia de saber qué se está comprando, pues muchas veces el precio no determina la calidad del producto, ni saciará totalmente las necesidades del can. Los veterinarios siempre recomiendan el alimento concentrado en vez del casero, ya que el alimento para mascotas contiene todos los nutrientes que el perro necesita de forma balanceada, sin condimentos o excesos que podrían dañar su salud. A continuación se suministra información sobre los componentes nutritivos que deben tener los alimentos, en la siguiente tabla se proporciona información sobre las posibles porciones diarias dependiendo de la masa de la mascota, en este caso para el perro. (Cuadro 2.5)

Ración Diaria (Tazas/Gramos)		
MASA (kg)	GRAMOS (g)	CANTIDAD (taza)
1 - 3	50 - 90	0,5 a 1
3 - 5	90 - 120	1 a 1,25
5 - 10	120 - 190	1,25 a 1,75
10 - 15	190 - 260	1,75 a 2,5
15 - 20	260 - 310	2,5 a 3
20 - 30	310 - 410	3 a 4
30 - 40	410 - 500	4 a 4,75
40 - 50	500 - 590	4,75 a 5,5
50 - 60	590 - 660	5,5 a 6,25
60 - 70	660 - 740	6,25 a 7
70 - 80	740 - 800	7 a 7,5

Cuadro 2.5: Ración Diaria

Buenos Consejos para la Alimentación [18].

- Los perros deben ser alimentados a la misma hora todos los días.
- Alimentar a tu perro con mismo tipo y marca de comida todos los días. A diferencia de los seres humanos, el sistema digestivo de un perro no puede manejar los cambios en la alimentación, lo que puede causar malestar estomacal y diarrea.
- Cuando se cambia a un nuevo alimento, haga una transición gradual a la nueva comida mezclando porciones de ambos alimentos.
- Si estás mezclando agua con el alimento seco, se debe mezclar 4 partes de alimento seco por 1 parte de agua.
- Mantenga agua potable fresca disponible en todo momento. Cambie el agua al menos una vez al día, más para los perros que babean.
- Mantenga los cuencos de alimentos y agua limpios.
- No sobrealimentar a tu perro.
- Siga las instrucciones en el paquete de comida para perros para las cantidades de alimento recomendadas.
- Vigila el nivel de peso y la actividad de tu perro, y haga ajustes de alimentación, según sea necesario.

Datos Curiosos

- Los perros tienen cerca de 1.700 papilas gustativas. Los humanos tenemos alrededor de 9.000 papilas gustativas, y los gatos cerca de 470.
- Los perros tienen papilas gustativas para el agua, algo que los humanos no tenemos.
- Los perros no anhelan la sal, como los humanos.

Capítulo 3

DESARROLLO DEL PROYECTO

3.1. Diseño Comedero de Mascotas en Solidworks

El diseño del comedero en solidworks basa su construcción en el requerimiento de satisfacer la necesidad de la mascota de alimentarse y hacer de este proceso lo más sencillo posible, sin generar dificultades por la no presencia de su amo en el momento de suministrar una ración de alimento, ya que dicho proceso se realizara de forma remota. El diseño se enfoca en dar comodidad y eficiencia, elaborando un sólido con las medidas requeridas para una alimentación de forma segura; con diversos parámetros como son el tipo de material, fabricación, sobredimensión del modelo, tiempo y diseño del molde, tiempo de solidificación, planos de las piezas finales, entre otros aspectos. La pieza final consta de 63 piezas en acrílico que es un material resistente, que puede permanecer largo tiempo en la intemperie, sin sufrir daño alguno. El hecho de que es un tipo de plástico más flexible de lo normal lo hace aún más fácil de trabajar y de esta forma poder cumplir con los requerimientos de ensamble del diseño. Dentro de sus características técnicas, se puede señalar, que el acrílico soporta largas horas a la exposición de los rayos ultravioletas, sin dañar su estructura y los colores del mismo, también es bastante utilizado en el embazado de alimentos, por su alta capacidad de conservación. Por otra parte, el acrílico es muchísimo más resistente que el vidrio. Con lo cual, no es fácil que se rompa y de hacerlo no se astilla. Con lo cual, no se corre riesgos de lesiones, debido a cortes producidos por su quebradura, además es un material ecológico, el acrílico es sumamente útil, ya que puede ser reciclado en un 100 %. La lámina de acrílico puede ser trabajada para darle una gran variedad de formas valiéndose de distintos procesos industriales, artesanales y artísticos [19].

La densidad del PMMA (polimetilmetacrilato) es del orden de 1190 kg/m^3 , es decir $1,19 \text{ gm/cm}^3$. Esto es algo menos de la mitad de la densidad del vidrio, la cual cae dentro del rango de 2400 a 2800 kg/m^3 . La resistencia al impacto del acrílico estándar es del orden de 15 veces mayor que la del vidrio no templado [19].

Otra ventaja de la utilización del acrílico en la elaboración final del diseño es su precio, si se compara con los altos costos que conlleva la impresión completa de una pieza en una impresora 3D. Con el diseño se busca dar un valor agregado, que es hacer de él lo más liviano posible, que conserve en buen estado el alimento y que sea amigable con el medio ambiente; con el acrílico se cumple en gran parte con estos parámetros, pero el diseño no está construido en su totalidad en este material puesto que hay ensambles que necesitaban piezas más precisas y de mejor elaboración, por ello la necesidad de la impresión de algunas piezas en una impresora 3D.

3.1.1. ¿Por qué la utilización de solidworks para el desarrollo del diseño?

Para elaborar el diseño se necesita un programa que abarque varios procesos de desarrollo, el software de diseño se hace sencillo por la gran cantidad de documentación que se encuentra en la red, además de la información suministrada por el propio solidworks, esto hace que se haga más sencillo su aprendizaje y se pueda convertir una idea en realidad, haciendo de una pieza un módulo escalable que puede sufrir modificaciones dependiendo de las necesidades futuras. Al contar SOLIDWORKS con un entorno de diseño en 3D integrado, que permite abarcar todos los aspectos de desarrollo del ensamble total y ayudar a maximizar la productividad del diseño y la producción [20].

Otro aspecto sobresaliente de solidworks es que utiliza un entorno gráfico basado en Microsoft Windows, que permite que sea intuitivo y de fácil manejo. Además dicho software posibilita plasmar ideas de forma rápida sin necesidad de realizar operaciones complejas y lentas, haciéndola una herramienta versátil y precisa, con capacidad asociativa, variacional y paramétrica de forma bidireccional con todas sus aplicaciones.

Al contar el diseño del comedero para mascotas con una gran cantidad de piezas que van asociadas, se necesitaba de un gestor de diseño (FeatureManager) que facilitara enormemente la modificación rápida de operaciones tridimensionales y de croquis de operación sin tener que rehacer los diseños ya plasmados en el resto de las piezas asociadas. Si se plantea a futuro realizar cambios significativos al diseño, solidworks no solo proporciona herramientas de diseño de Pieza, Ensamblajes y Dibujo, sino que también incluye herramientas de productividad, de gestión de proyectos, de presentación y de análisis y simulación que permite hacer un diseño competitivo con estándares de calidad [21].

3.1.2. Piezas que conforman el sólido final.

El sólido final está conformado por 63 piezas en acrílico que está conformado por:

Pieza Superior: Posee 24 componentes. Tolva donde se arroja y se almacena el alimento.

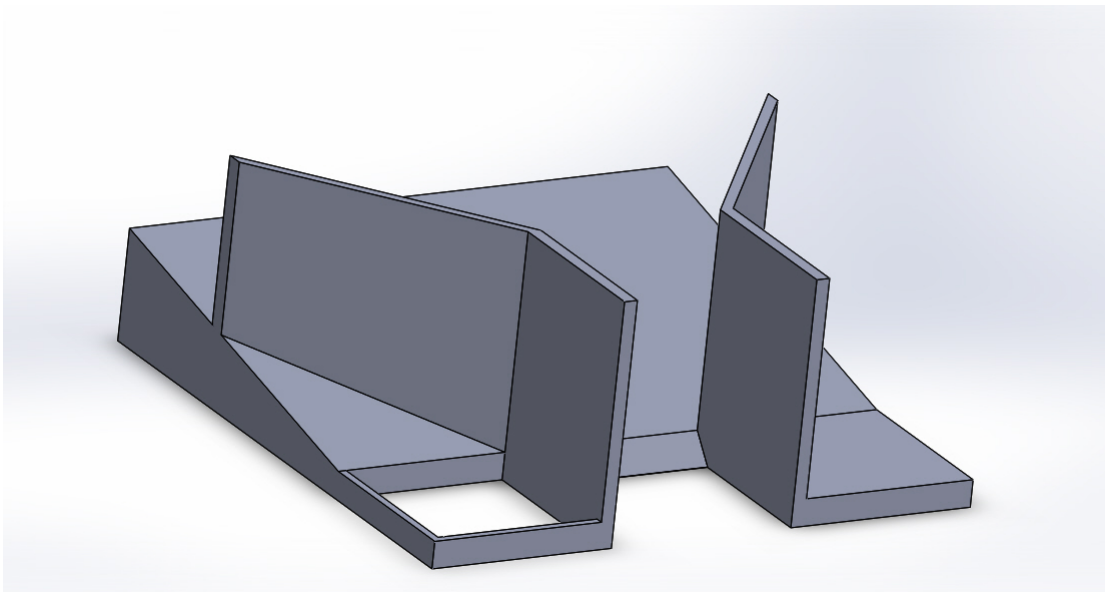


Figura 3.1: Pieza Superior

Rampas de Caída: Posee 11 componentes. Base del motor y rampas por donde cae el alimento, para luego llegar al recipiente final.

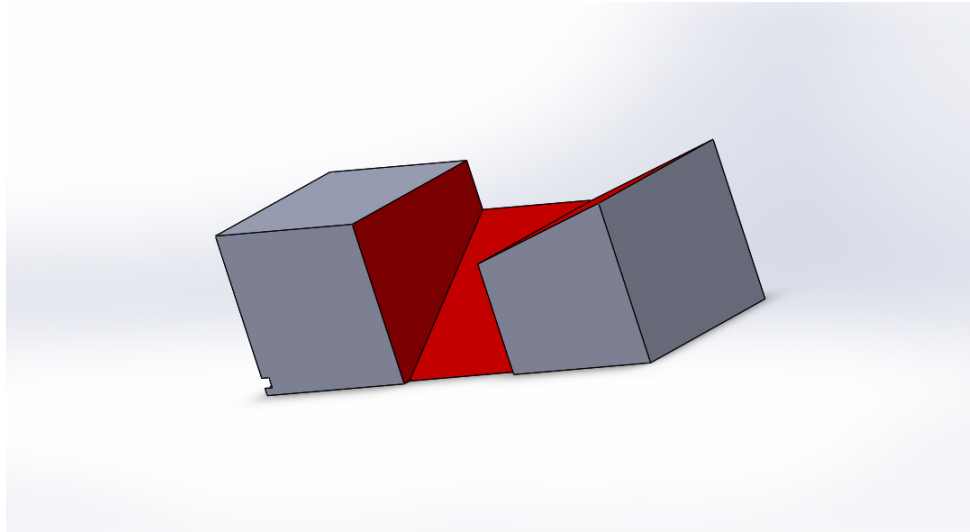


Figura 3.2: Rampas de Caída

Recipiente Base: Posee 17 componentes. Recipiente donde lleva el alimento para ser consumido por la mascota está compuesto por 6 piezas.

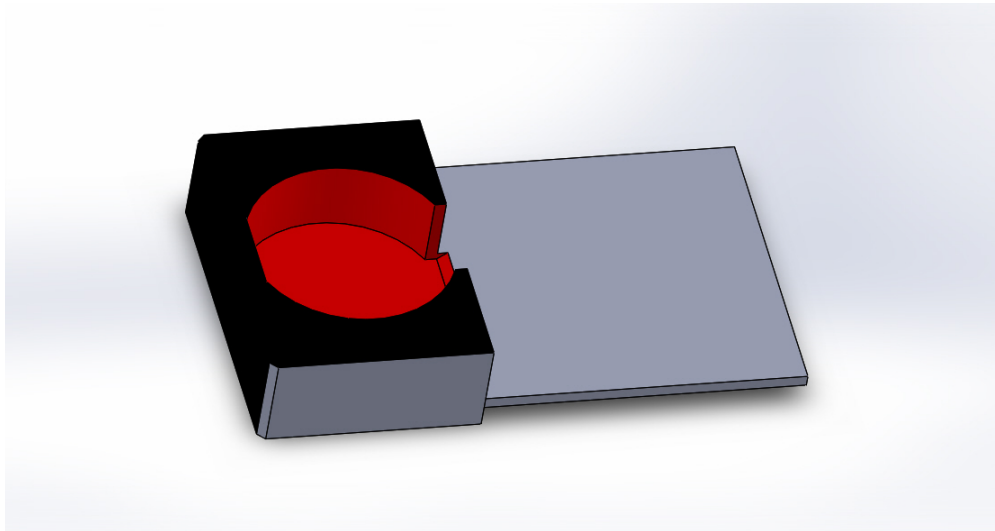


Figura 3.3: Recipiente Base

Los 11 piezas restantes en acrílico lo conforman 8 tapas y 3 adornos. Las tapas dan firmeza entre piezas, permitiendo el ensamble entre cada uno de los sólidos y en el caso de los adornos dar un toque llamativo y atractivo a la vista. En dos de los adornos se crean pequeñas rejillas que permiten el paso de aire hacia el interior donde se encuentra el circuito electrónico, estas rejillas toman la forma de huesos.

Tapa Cavity Circular

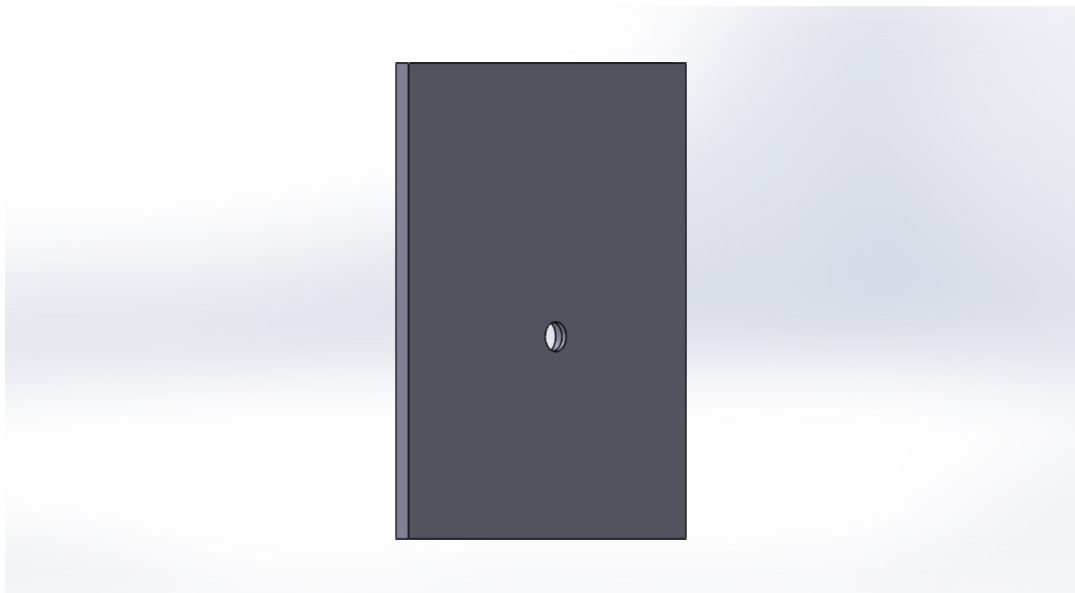


Figura 3.4: Cavity Circular

Tapa Unión de Piezas: Unión cavity circular y rampas de caída.



Figura 3.5: Unión de Piezas

Tapa Exterior: Tapa que cubre el circuito electrónico, con sistema de entrada de aire y orificios para entrada de tornillo para dar sujeción.

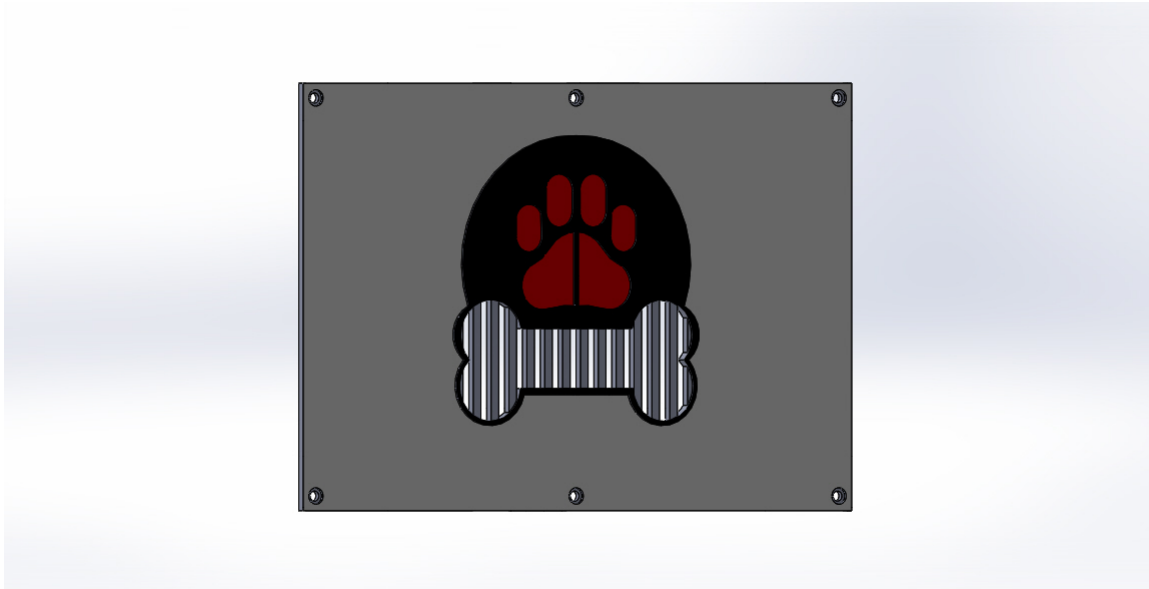


Figura 3.6: Tapa Exterior

Tapa Pieza Superior Derecha

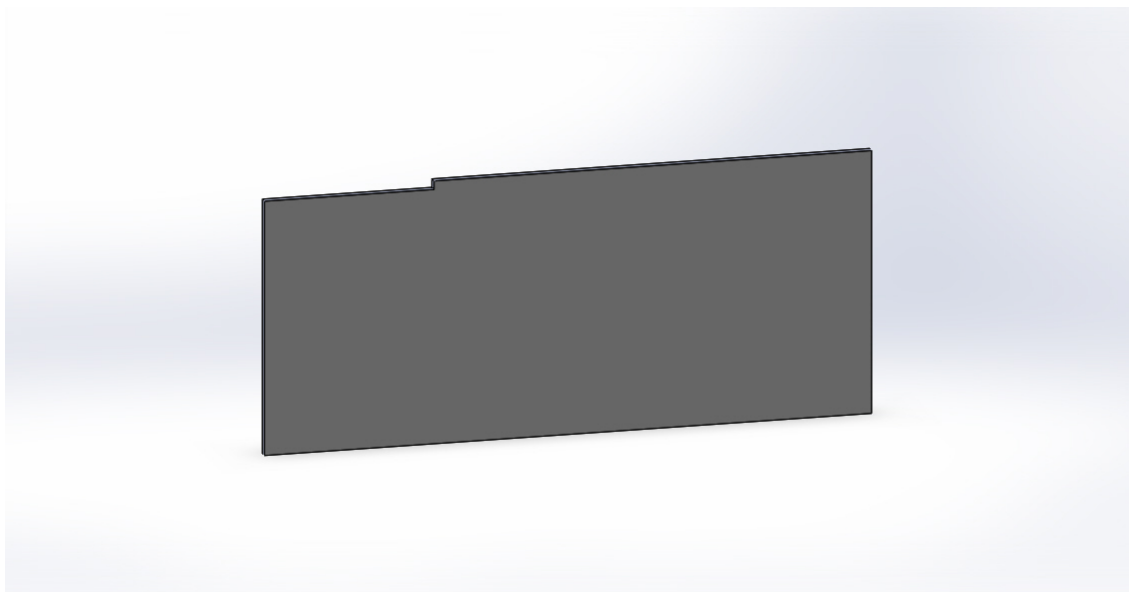


Figura 3.7: Tapa Superior Derecha

Tapa Frontal: Posee abertura por donde sale el alimento hacia el recipiente final.

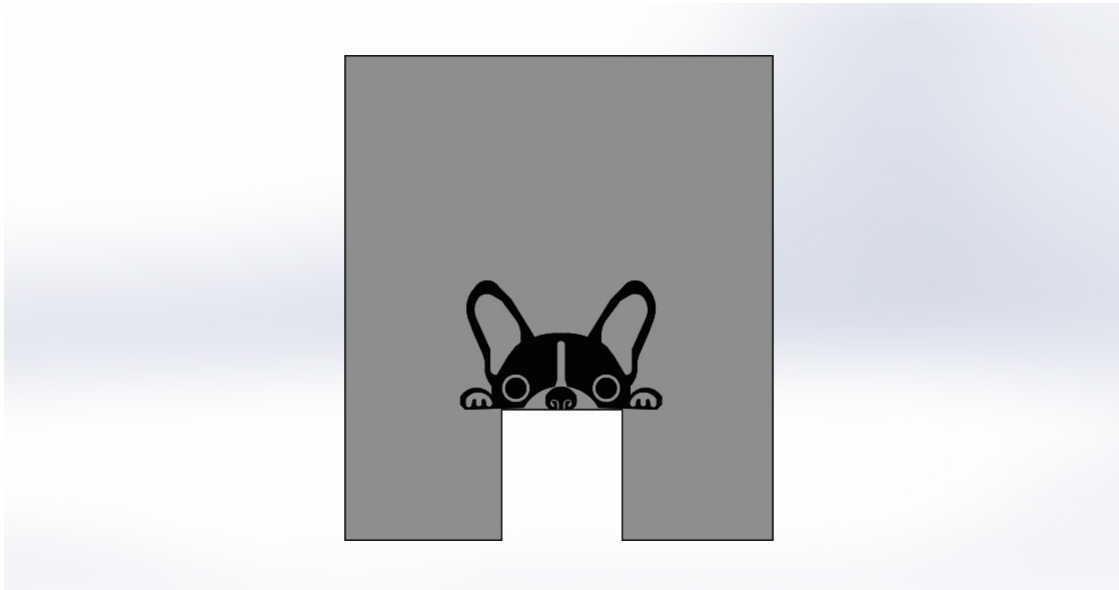


Figura 3.8: Tapa Frontal

Tapa Izquierda: Cubre el sector izquierdo del diseño dando ensamble a la parte inferior con la superior.

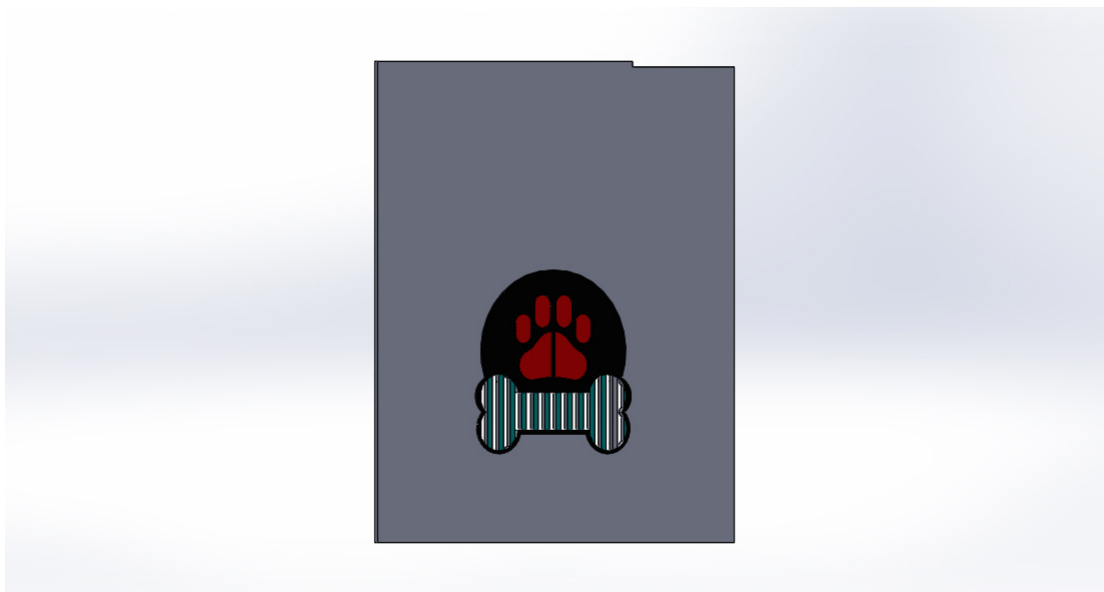


Figura 3.9: Tapa Izquierda

Tapa Trasera: Cubre en su totalidad la parte trasera del diseño y posee aberturas para dar entrada a cables de potencia y de Ethernet.

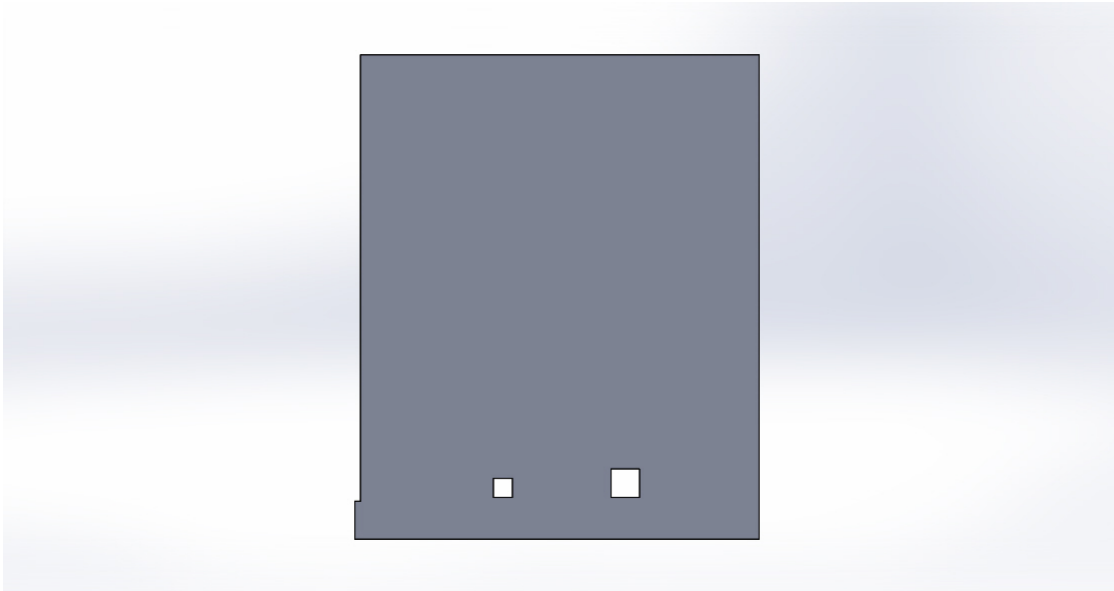


Figura 3.10: Tapa Trasera

Tapa Techo: Como su nombre lo indica cubre la parte superior.

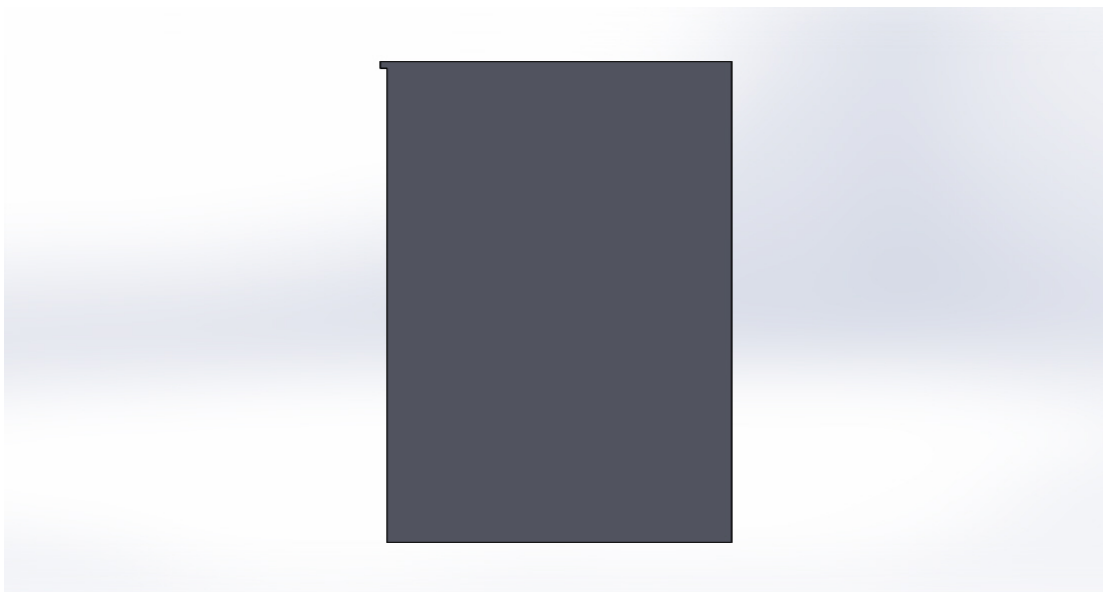


Figura 3.11: Tapa Techo

A continuación se realiza una descripción de las piezas detalladas (impresión 3D)

Cavidad Circular: Sección donde se incrusta y se fija el tornillo sin fin.

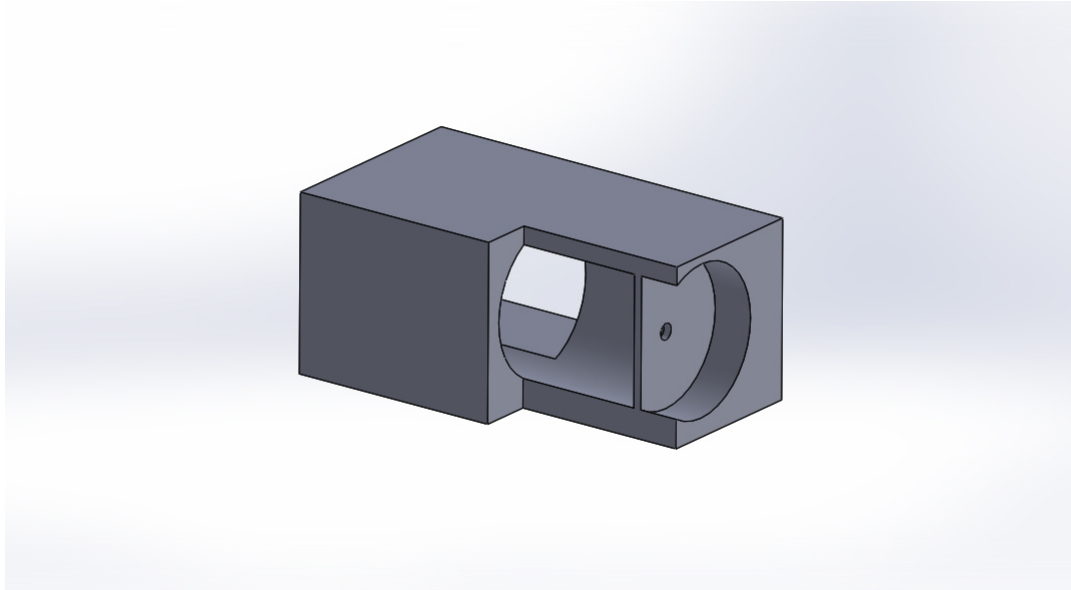


Figura 3.12: Cavidad Circular

Tornillo Sin Fin: Transporta el alimento que cae de la tolva a las rampas.

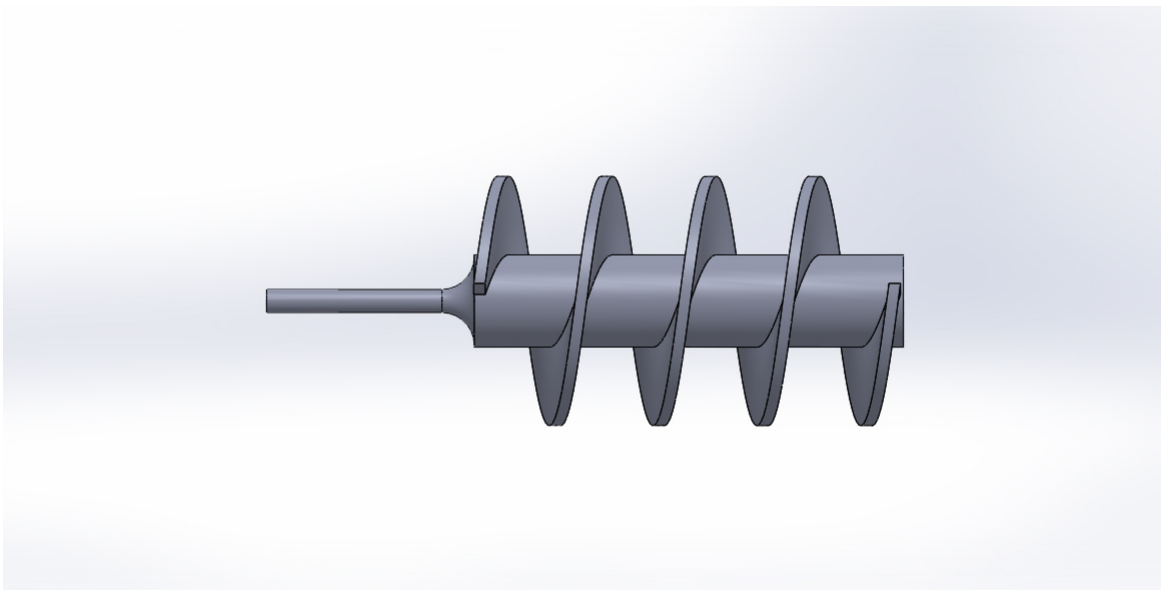
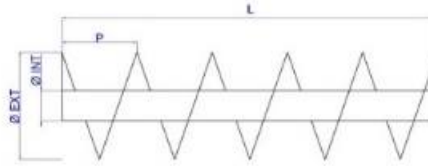


Figura 3.13: Tornillo Sin Fin

Gusano helicoidal

DATOS:
 $\varnothing_{EXT} = 54 \text{ mm}$
 $\varnothing_{INT} = 20 \text{ mm}$
 $P = 22.5 \text{ mm}$
 $L = 95 \text{ mm}$



FORMULAS:

$$A = \frac{\varnothing_{EXT} - \varnothing_{INT}}{2}$$

$$RU = \sqrt{(\varnothing_{EXT}^2 \times \pi^2 + PASO^2)}$$

$$RE = \sqrt{(\varnothing_{INT}^2 \times \pi^2 + PASO^2)}$$

$$r = \frac{A \times E}{U - E}$$

$$R = r + A$$

$$\Delta = 2 \times R \times \pi$$

$$\beta = \frac{RU \times 360^\circ}{\Delta}$$

$$X = 360^\circ - \beta$$

DESARROLLO:

$$A = \frac{54 - 20}{2} = 17$$

$$RU = \sqrt{(54^2 \times 3.14^2 + 22.5^2)} = 171.13 \approx 171$$

$$RE = \sqrt{(20^2 \times 3.14^2 + 22.5^2)} = 66.7 \approx 67$$

$$r = \frac{17 \times 67}{171 - 67} = 10.95 \approx 11$$

$$R = 11 + 17 = 28$$

$$\Delta = 2 \times 28 \times 3.14 = 175.9 \approx 176$$

$$\beta = \frac{171 \times 360^\circ}{176} = 349.7^\circ \approx 350^\circ$$

$$X = 360^\circ - 350^\circ = 10^\circ$$

r y R con un error $\pm 1 \text{ mm}$

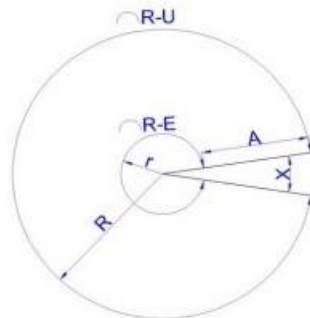


Figura 3.14: Cálculos Tornillo Sin Fin

Pieza Tuerca – Tornillo: Piezas para permitir el ajuste de la tapa exterior del circuito electrónico.

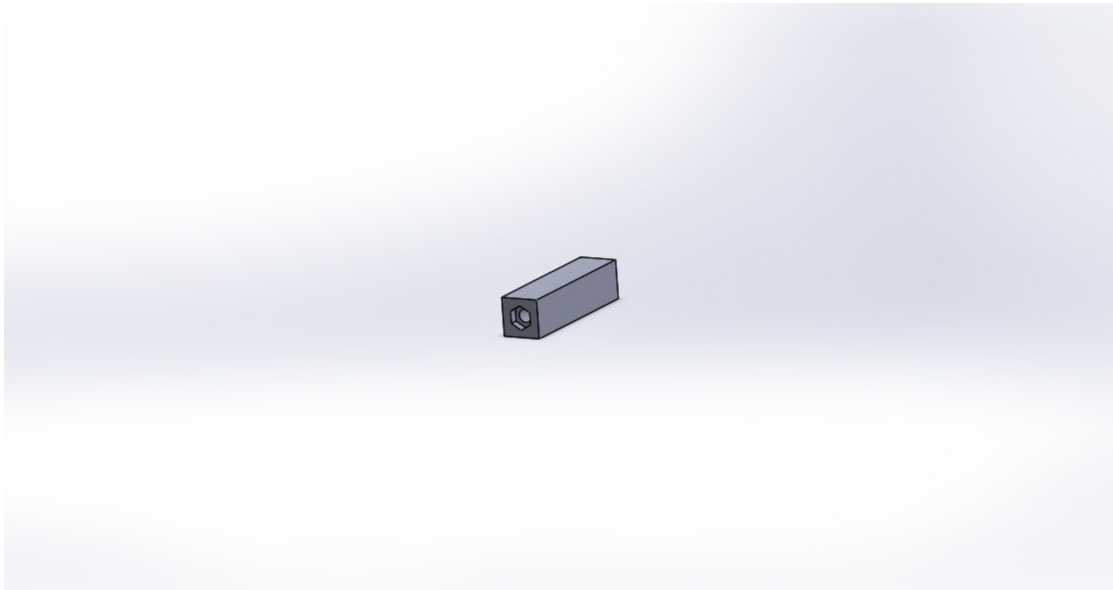


Figura 3.15: Pieza Tuerca - Tornillo

Una de las características que hacen de solidworks un programa muy completo es el poder asociar cada una de las piezas en un ensamble único (Asociatividad).

SolidWorks contiene tres módulos: Pieza, Ensamblaje y Dibujo. La creación de un documento en cada uno de ellos genera un fichero con distinta extensión. Los documentos, aunque no pueda observarse, están asociados y vinculados entre ellos [21].

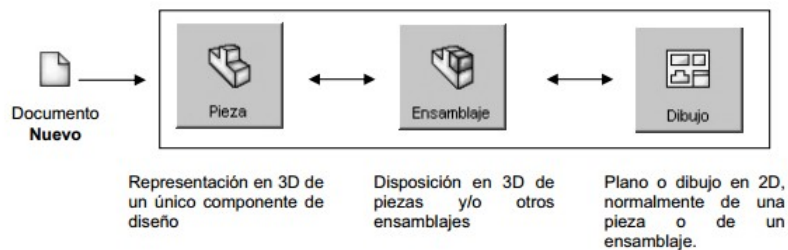


Figura 3.16: Proceso Selección Documento

Para realizar un conjunto o ensamblaje debe diseñar cada una de las piezas que lo conforman y guardar como ficheros de pieza distintos (cada uno con un nombre). El módulo de ensamblaje permite insertar cada una de las piezas y asignar relaciones geométricas de posición para definir tridimensionalmente el ensamblaje. Finalmente, puede obtener los planos las piezas o del propio ensamblaje de forma automática. Cuando se dice que SolidWorks es asociativo quiere decir que todos los documentos (Pieza, Ensamblaje o Plano) están vinculados y que la modificación de un fichero de pieza modifica el ensamblaje y los planos asociados de forma automática, sin la participación del usuario. Los ficheros se actualizan aunque se encuentren cerrados [21].

Al tener las piezas totales se pasa al ensamble final y la agrupación de cada una de ellas, a continuación el diseño final sin tapa exterior, mostrando la sección donde va el circuito electrónico y el motor.

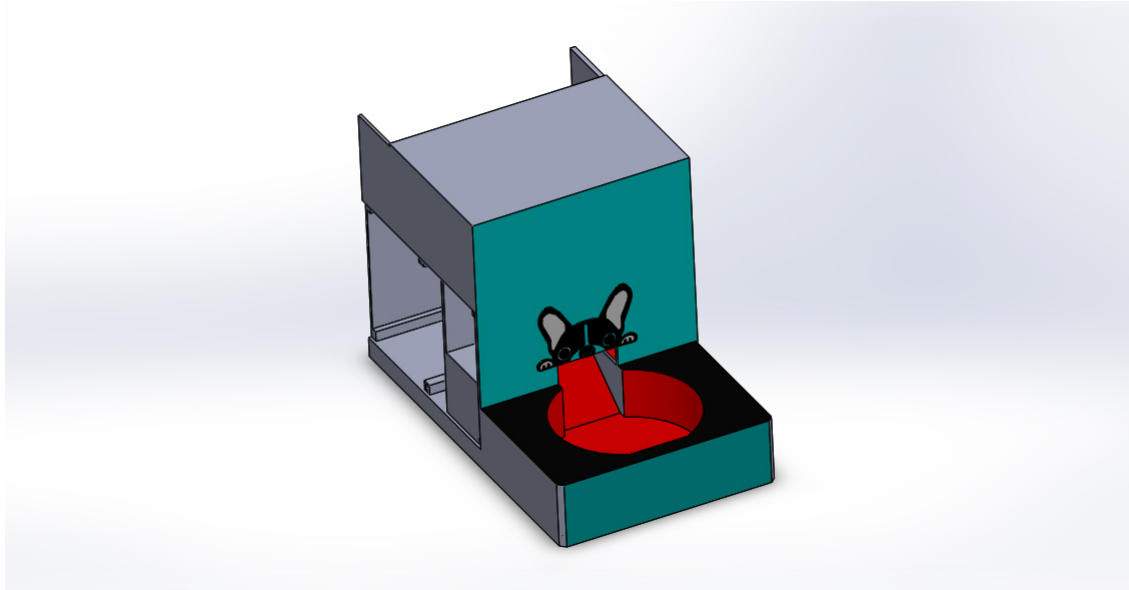


Figura 3.17: Comedero Parcialmente Ensamblado

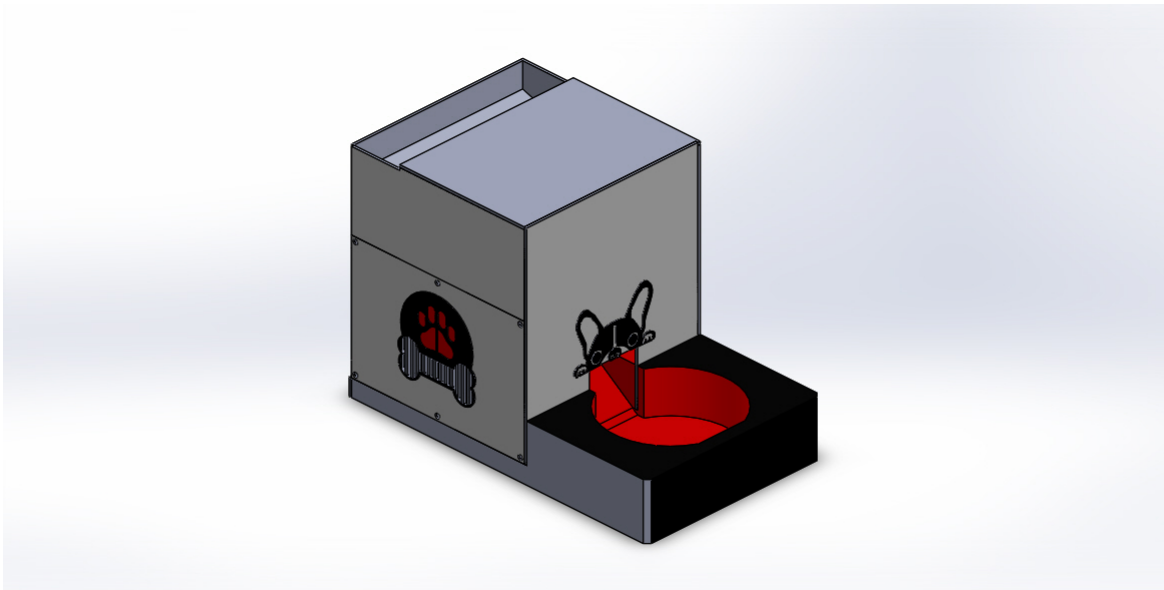


Figura 3.18: Comedero Totalmente Ensamblado

3.2. Diseño de Programa de Control, Interfaz Gráfica y Tarjeta de Potencia.

3.2.1. ¿Qué es Raspberry Pi?

La Raspberry Pi es una tarjeta de un tamaño pequeño que se puede utilizar como un computador de placa reducida. Esto le permite tener un bajo coste y adicionalmente tiene una excelente capacidad de procesamiento de datos. Esta tarjeta fue creada por la Raspberry Pi Foundation, con el objetivo de enseñar de manera simple temas relacionados con la programación en las escuelas del Reino Unido [22].

La Raspberry Pi es una tarjeta que como se nombró anteriormente, es usada para enseñar a los niños nociones básicas de programación; adicionalmente, el bajo costo de la tarjeta, permite a niños de pocos recursos el acceder a plataformas para practicar programación de lenguajes como python, nativos en esta tarjeta [23].

El software utilizado como sistema operativo en esta tarjeta es por lo general uno basado en el núcleo de linux; en este proyecto es utilizado el raspbian jessie, el cual es una distribución de Debian optimizada para la tarjeta raspberry pi, la cual es muy intuitiva de usar y fácil de configurar.

La tarjeta utilizada en este proyecto, es la raspberry pi 3; la cual cuenta con una gran ventaja (en comparación con sus tarjetas predecesoras) y es que tiene integrada una tarjeta de red inalámbrica; lo que le permite a la pi 3, poder conectarse a internet mediante una red wifi; algunas características de la pi 3 son: [24]

- Procesador Quad-Core ARMv8 de 1.2 Ghz de 64-bit
- Conexión Wireless LAN 802.11n
- Bluetooth 4.1
- 1 Gb de memoria RAM
- 4 Puertos USB
- 40 Pines GPIO (General Purpose Input/Output, Entrada/Salida de Propósito General)
- Puerto HDMI
- Puerto Ethernet
- Socket para memoria SD

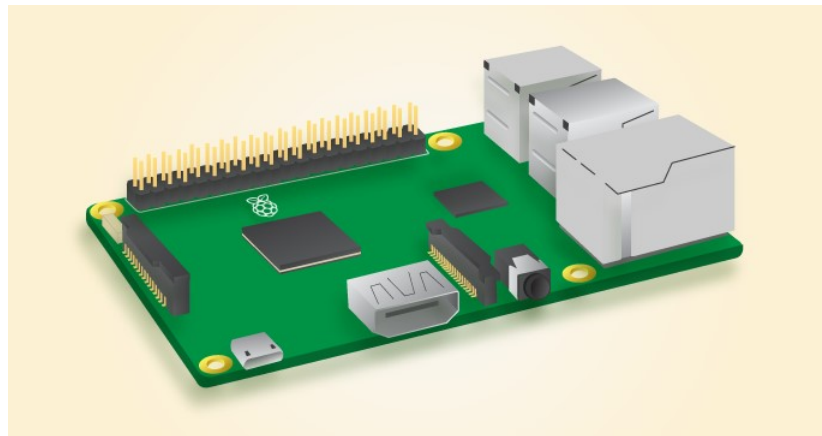


Figura 3.19: Tarjeta Raspberry Pi 3

3.2.2. Pines GPIO de la Raspberry Pi

Una de las mayores ventajas que ofrece la tarjeta raspberry pi 3, es que dispone de 40 pines GPIO, estos pines permiten la conexión física entre la raspberry pi y diferentes módulos con los cuales se pueden intercambiar datos [25]. El uso que se le dará en este proyecto es el de generar la señal de control que dará inicio al giro de un motor.

Alternate Function	Pin	Pin	Alternate Function
	3.3V PWR	1	2 5V PWR
I2C1 SDA	GPIO 2	3	4 5V PWR
I2C1 SCL	GPIO 3	5	6 GND
	GPIO 4	7	8 UART TX
	GND	9	10 UART RX
	GPIO 17	11	12 GPIO 18
	GPIO 27	13	14 GND
	GPIO 22	15	16 GPIO 23
	3.3V PWR	17	18 GPIO 24
SPI0 MOSI	GPIO 10	19	20 GND
SPI0 MISO	GPIO 9	21	22 GPIO 25
SPI0 SCLK	GPIO 11	23	24 GPIO 8
	GND	25	26 GPIO 7
	Reserved	27	28 Reserved
	GPIO 5	29	30 GND
	GPIO 6	31	32 GPIO 12
	GPIO 13	33	34 GND
SPI1 MISO	GPIO 19	35	36 GPIO 16
	GPIO 26	37	38 GPIO 20
	GND	39	40 GPIO 21
			SPI0 CS0
			SPI0 CS1
			SPI1 CS0
			SPI1 CS1
			SPI1 MOSI
			SPI1 SCLK

Figura 3.20: Pines de la Raspberry Pi

La Raspberry pi cuenta con una serie de pines especiales; estos pines especiales son utilizados para interfaces UART; es decir, cuenta con un pin RX y un pin TX con los cuales se pueden implementar

comunicaciones Seriales, lo cual es muy útil para conectar con otras tarjetas como por ejemplo un Arduino. Estos pines especiales también son usados para implementar protocolos de comunicación I2C, ya que se cuenta con pines SCL y SDA. La raspberry pi cuenta también con protocolo de comunicación SPI, cuenta con pines MOSI, MISO SCLK CE0 Y CE1 [26].

El GPIO escogido para controlar el motor en este proyecto es el número 11; el cual puede generar una señal de alto de 3.3 Volt; esta señal funciona como señal de control para el módulo de potencia que controla el motor que hace girar el tornillo sin fin (descrito en el capítulo anterior).

El lenguaje de programación en el cual se programan el uso y las funciones de los GPIO de la raspberry pi es por defecto python; a continuación se realiza una breve descripción de este lenguaje que está incluido en el Sistema Operativo de la tarjeta.

3.2.3. Lenguaje PYTHON

Guido Van Rossum, creador de este lenguaje lo define de la siguiente manera: «Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas» [27].

Para poder interactuar con los pines GPIO de la raspberry, se debe descargar e instalar la librería RPi.GPIO, esta librería nos permite controlar los pines de una manera muy sencilla, a través de unos simples comandos, esta librería se puede descargar de la página oficial de python [28].

Cuando se haya descargado, procedemos a ingresar a la consola de PYTHON, seguido a esto ejecutamos la siguiente línea de comando:

```
import RPi.GPIO
```

También podemos descargarla mediante la consola usando el siguiente comando:

```
wget 'http://downloads.sourceforge.net/project/raspberry-gpio-python/RPi.GPIO-0.5.4.tar.gz'
```

Después de descargar la librería descomprimos el archivo con el siguiente comando:

```
tar zxvf RPi.GPIO-0.5.4.tar.gz
```

Cuando descomprimos el archivo se crea una carpeta donde están alojados los archivos, se debe ingresar a esta carpeta y ejecutar el siguiente comando para realizar la instalación:

```
sudo python setup.py install
```

Una vez tengamos instalada la librería procederemos a realizar el programa en python que ejecute las acciones que deseamos, en este caso se pondrá en alto en pin 11 por un tiempo de 2 segundos y luego se apagará.

Algoritmo 3.1 Código en Python

```

#CODIGO PYTHON COMEDERO
#Código que activa el pin que controla el motor
#importamos la librería y cambiamos su nombre por "GPIO"
import RPi.GPIO as GPIO
#necesario para los delays
import time
#establecemos el sistema de numeracion que queramos, en este caso BCM
GPIO.setmode(GPIO.BCM)
#configuramos el pin GPIO11 como una salida
GPIO.setup(11, GPIO.OUT)
#colocamos en alto la salida
GPIO.output(11, GPIO.HIGH)
#esperamos 2 segundos
time.sleep(2)
#colocamos en bajo la salida
GPIO.output(11, GPIO.LOW)
#devuelve los pines a su estado inicial
GPIO.cleanup()

```

El proyecto del comedero tiene 2 aspectos que son fácilmente diferenciables:

1. Aplicación Web; esta es la sección donde se incluyen la comunicación entre el cliente y el servidor a través de los protocolos HTTP/HTML; la cual permite el envío y recepción de datos y señales de control.
2. Control Físico del sistema; esta es la sección donde se reciben las señales provenientes del servidor y se transforman en señales de control para la manipulación del motor que controla el tornillo sin fin.

Para poder completar la sección correspondiente a la aplicación web, primero se debe instalar un servidor web apache en la tarjeta.

3.2.4. Servidor Web Apache

Apache es actualmente el servidor web más ampliamente utilizado en el mundo, es de código libre. Apache es el encargado de recibir las peticiones para páginas que son emitidas por clientes o visitantes para acceder a un determinado sitio web y habilitar o no su acceso a dicho sitio [29].

Algunas de las características de Apache son:

- Atención eficiente a los visitantes (maneja un alto número de peticiones HTTP).
- Restricción de acceso a los archivos de acuerdo a los permisos configurados por el administrador.
- Evaluar e informar los errores encontrados en las páginas web.
- Transmitir de manera adecuada la información entre el usuario y el servidor.
- Creación de un registro sobre las peticiones, errores, consultas e información general, con el fin de que esta pueda ser analizada para generar estadísticas del sitio al que se está visitando.

Instalación Apache en la Raspberry Pi

Cuando deseamos instalar algún programa en la tarjeta, lo primero que debemos hacer es actualizar los repositorios y el software de la tarjeta mediante el siguiente comando:

```
sudo apt-get update && sudo apt-get upgrade
```

Una vez actualizados los repositorios y el software de la tarjeta, procedemos a instalar el servidor web apache en la tarjeta raspberry pi, solo es necesario ejecutar el siguiente comando:

```
sudo apt-get install apache2
```

Una vez hemos instalado el servidor web apache en nuestra raspberry; ya podemos realizar peticiones a sitios web; sin embargo, cómo comunico esas solicitudes con el código en python? Para poder realizar este puente de comunicación, se utilizará el lenguaje de programación php; el cual es un lenguaje utilizado específicamente para aplicaciones web y que permite una fácil integración con HTML [30].

A continuación se muestra el diseño del servidor apache que será implementado en la tarjeta.

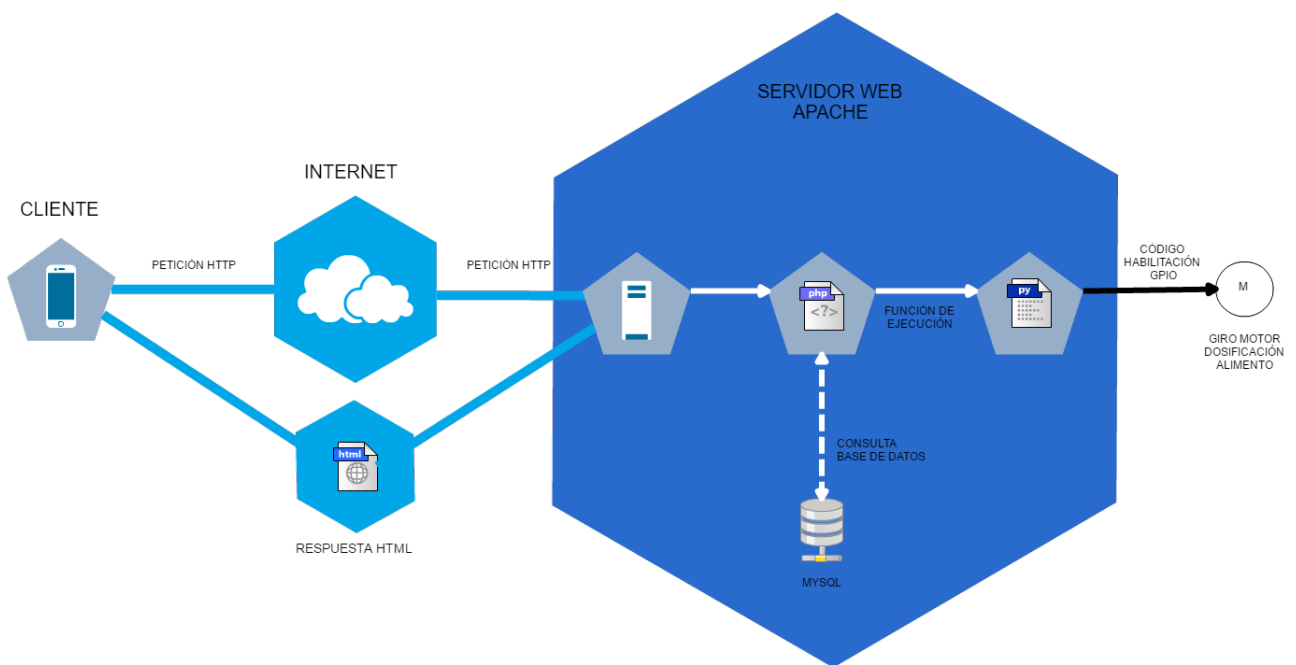


Figura 3.21: Estructura de Red Implementada en el Proyecto

3.2.5. Lenguaje HTML

Una de las definiciones más apropiadas y simples sobre el lenguaje de programación HTML es la siguiente: “HTML es el lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc.” [31].

Una vez entendido el significado del lenguaje HTML, se procede a crear el código que se cargará en la tarjeta, el cual funcionará como interfaz gráfica del controlador. Para crear este código no es necesario tener una plataforma de programación web; basta con crear el código en un archivo de texto plano (como el bloc de notas de Microsoft), colocarle el nombre que se desee y agregarle la extensión .html; en nuestro caso el nombre que le hemos dado al archivo ha sido “consumo.html”

La estructura de organización de los códigos en html tienen la siguiente estructura:

1. Etiqueta de “<!DOCTYPE>”, la cual le indica al navegador que se iniciará la versión estándar de html.
2. Etiqueta de <html lang=”en”> </html>; esta es la que envuelve el contenido del código principal; indica el inicio y el fin del código. La mayoría de las etiquetas en html funcionan en pares, y se distingue la etiqueta de apertura de la etiqueta de cierre porque esta última tiene una barra invertida antes del nombre.
3. Etiqueta de <head> </head>; esta es la sección de cabecera, uno de los 2 grandes segmentos de los códigos escritos en lenguajes html. En esta sección se definen cosas como los títulos, fuentes, imágenes e incluso los archivos externos de estilo y archivos Javascript. La gran parte del código escrito en esta sección no es visible por el usuario.
4. Etiqueta <body> </body>; esta es la sección del cuerpo; la otra gran sección de los códigos en html; aquí es donde se escribe todo el código, donde se debe colocar el contenido de la página [32].

El código empleado en este proyecto es el siguiente:

Algoritmo 3.2 Código HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>SISTEMA DE CONSUMO</title>
</head>
<body>
<div class="logo">
<a href="#"><h1> SISTEMA DE CONSUMO ANIMAL</h1>
</a>
</div>
<div class="contenedor">
<div class="contenedor1">
<!--GPIO17-->
<form action="" method="POST">
<h3> Proporción 1</h3> <input type="submit" name="comer1" value="Activar" class=" button">
</form>
<br><br>
<form action="" method="POST">
<h3> Proporción 2</h3> <input type="submit" name="comer2" value="Activar" class="button">
>
</form> <br><br>
<form action="" method="POST">
<h3> Proporción 3</h3> <input type="submit" name="comer3" value="Activar" class="button">
>
</form> <br><br>
</body>
</html>

```



Figura 3.22: Interfaz Gráfica

3.2.6. Lenguaje PHP

Es un lenguaje de programación “Open Source” disponible para múltiples plataformas; su principal función o enfoque es ser orientado hacia la parte del servidor, generando contenido dinámico para los sitios web. Estar enfocado del lado del servidor significa que toda la ejecución de su código se realiza en el servidor web; esto se realiza con antelación al envío de la página web al cliente; es decir, crear las bases de datos, las conexiones de red y otras tareas que sean necesarias para crear la página web. Esta página web resultante en HTML es el resultado de toda la ejecución del código PHP [33].

Cómo se ha comentado con antelación, se utilizará PHP para enlazar el código HTML con el código realizado en PYTHON; para esto es necesario ejecutar en la raspberry el siguiente comando:

```
sudo apt-get install php5 libapache2-mod-php5
```

Al igual que HTML, PHP también debe estar contenido entre las etiquetas `<?php.....; ?>`

Una vez instalado PHP y HTML, procedemos a realizar los códigos respectivos que nos permitirán controlar nuestra tarjeta.

Algoritmo 3.3 Código PHP

```
<?php
////////////////////////////////////// Funciones PHP del pin GPIO 17
$state1 = $_POST['comer1'];
$state2 = $_POST['comer2'];
$state3 = $_POST['comer3'];
if($state1)
{ exec('sudo python /var/www/html/gpio/comedero.py');
}
if($state2)
{ exec('sudo python /var/www/html/gpio/comedero2.py');
}
if($state3)
{ exec('sudo python /var/www/html/gpio/comedero3.py');
}
////////////////////////////////////// Fin de las funciones del pin GPIO 17
?>
```

3.2.7. Tarjeta de Potencia

Como se comento al inicio de la sección, la tarjeta raspberry pi cuenta con un serie de pines GPIO, estos pines se pueden colocar en estados lógicos de HIGH y LOW, estos estados lógicos tienen un valor de 3.3 volt y 0 volt respectivamente, sin embargo este valor lógico no es suficiente para hacer girar el motor, por lo tanto, se ha realizado una tarjeta electrónica que controlará el suministro de corriente de una fuente que provee una mayor potencia para que el motor tenga un funcionamiento óptimo. El motor utilizado en este proyecto es un motoreductor de 12V; a continuación se relacionan sus características técnicas:



Figura 3.23: Motoreductor

Característica	Detalle
Dimensión	37 mm X 25 mm
Diámetro del Eje	6 mm
Tensión del Motor	12 Volt
Torque	6 kg/cm
RPM	32/min
Corriente Requerida	430 mA (para 12 V)

Cuadro 3.1: Especificaciones Técnicas Motor

El diseño de control del motor está basado en un transistor TIP41 en modo interruptor, de esta manera se permite el paso de corriente a través del circuito, haciendo girar el motor y accionando de esta manera el mecanismo del prototipo que distribuye el alimento al recipiente ubicado en la base del comedero.

Para calcular la resistencia de base del transistor, se utilizó la siguiente fórmula:

$$R_{base} = \frac{(Volt - 1.5)}{\frac{Cte}{HFE}}$$

- Volt, es el voltaje aplicado a la base del transistor, en este caso corresponde a los 3.3 Volt que genera el pin de la Raspberry Pi.
- El valor de 1.5, corresponde al voltaje base-emisor, caída de tensión típica en este tipo de transistores; este es un valor obtenido de la hoja de datos del elemento proporcionada por el fabricante.
- Cte, es la corriente que pasará a través del transistor desde su colector a su emisor.
- HFE, corresponde al valor de la ganancia de corriente, comúnmente conocido como el Beta del transistor.

El valor calculado en este caso no corresponde al valor de una resistencia comercial, por tal motivo se utilizará el valor de resistencia más próximo.

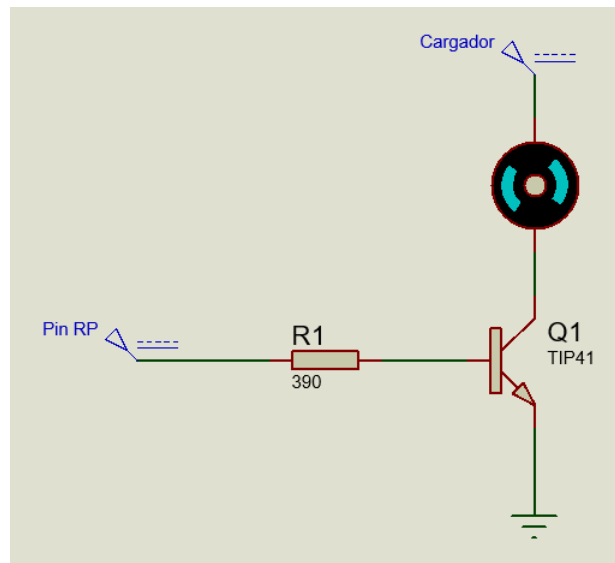


Figura 3.24: Circuito de Potencia

3.3. Diseño e Instalación de la Aplicación Móvil.

3.3.1. Aplicación Móvil, Tecnología 4G e Internet.

La fase de encendido remoto de funciones del comedero para alimentar mascotas se fundamenta en el desarrollo de una aplicación móvil que realiza ciertas tareas de forma rápida y eficiente.

La función primordial de la aplicación es entablar comunicación vía datos o wifi desde cualquier punto con el comedero, dando así acceso al servidor web y otorgando un control pleno sobre el funcionamiento de la Raspberry pi y las instrucciones establecidas dentro de esta.

Las aplicaciones móviles basan su funcionamiento en la efectividad y la buena ejecución de ciertas tareas creadas por el programador, pero el programador también necesita de una excelente comunicación entre la aplicación, el teléfono móvil y el elemento a controlar, por eso la necesidad de una buena conexión de internet ya sea móvil o fija.

En los últimos años el crecimiento tecnológico ha sido constante y el sector comunicativo específicamente en el área del mercado de teléfonos móviles (celulares) no es indiferente a este crecimiento. El manejo de datos e información se realiza en cuestión de segundos, esto se ve reflejado en la implementación de la tecnología 4G en gran parte de Colombia, teniendo actualmente casi un cubrimiento total.

Colombia para comienzo del segundo semestre de 2016 se convertía en el primer país en Latinoamérica en tener mayor cantidad de competidores en el mercado 4G, con la tecnología 4G se gana calidad, mejores precios y velocidad de procesamiento de datos. [34]

Al cierre del primer trimestre de 2016, el número total de suscriptores a Internet estaba compuesto principalmente por accesos móviles a Internet con 8.041.314 suscripciones y una participación del 59 %, mientras que los suscriptores fijos a Internet alcanzaron los 5.665.837 suscriptores y una participación del 41 % [35].

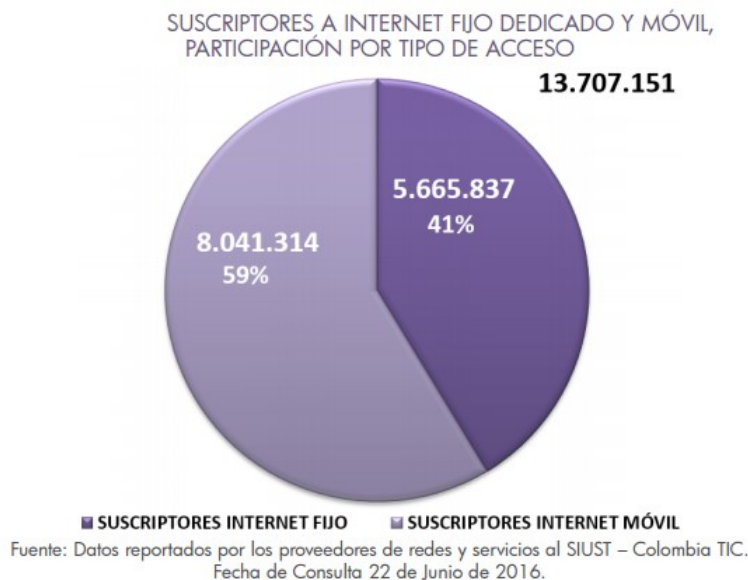


Figura 3.25: Suscriptores

Para el primer trimestre de 2016 la variación porcentual de suscriptores a Internet fijo dedicado fue del 8,6 % respecto al primer trimestre del 2015, representado en una variación absoluta de 448.033 suscriptores. Por su parte, los suscriptores a Internet móvil presentaron una variación porcentual del 29,1 % respecto al primer trimestre de 2015 [35].

Los datos suministrados muestran un crecimiento que sigue en aumento y son más las personas que día a día se hacen con los servicios de internet en los hogares o con servicios de internet en los equipos móviles. Al término del primer trimestre de 2016, el número de abonados en servicio de telefonía móvil en Colombia alcanzó un total de 57.292.621, y un índice de penetración del 117,5 %, presentando un aumento de 1 punto porcentual con relación al índice de penetración del mismo trimestre del año anterior, el cual se ubicó en el 116,5 % [35].

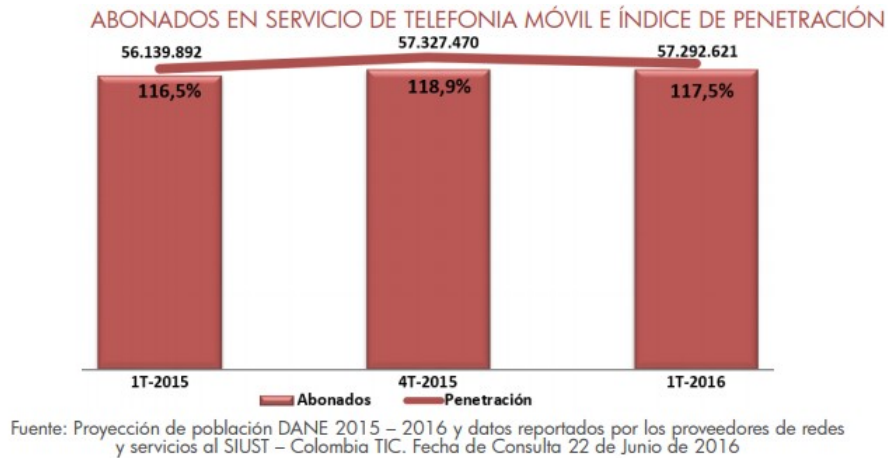


Figura 3.26: Abonados Telefonía Móvil

El crecimiento de las ventas de servicios de telefonía móvil se refleja en las altas ventas de equipos que proporcionan servicios de datos (internet móvil).

3.3.2. Sistema Operativo Android

El sistema operativo Android basa su creación en el kernel de Linux y amparado por la licencia apache, la cual es una licencia de software libre. Android en la actualidad es el sistema operativo para teléfonos móviles más vendido en el mundo, no solo por estar instalado en los equipos proporcionados por las grandes marcas fabricantes como Samsung, Lg, Huawei, Motorola, Sony, entre otros; sino también por las facilidades de descarga de aplicaciones por medio de su tienda “play store”. Android además permite la instalación de aplicaciones creadas por los mismos usuarios desde distintas plataformas e instaladas en el sistema como aplicaciones de “orígenes desconocidos”. Estos factores, hacen del sistema operativo de propiedad de Google junto a iOS (Apple) los más apetecidos para los usuarios en todo el mundo. Android a diferencia de IOS posee equipos inteligentes de varias gamas y precios lo cual hace que sea más asequible para cualquier tipo de comprador [36].

País	Sistema Operativo	Abril 2015	Abril 2016	% Cambio
Alemania	Android	74,4	76,8	2,4
	iOS	17,0	15,3	- 1,7
	Windows	7,5	5,9	- 1,6
	Otro	1,1	2	0,9
Inglaterra	Android	54,4	58,6	4,1
	iOS	34,7	35,1	0,4
	Windows	9,6	5,8	- 3,8
	Otro	1,3	0,6	- 0,7
Francia	Android	68,6	74,7	6,1
	iOS	17,9	21,5	3,6
	Windows	12	3,6	- 8,4
	Otro	1,4	0,3	- 1,1
Estados Unidos	Android	62,4	67,6	5,2
	iOS	33,2	30,7	- 2,5
	Windows	3,8	1,3	- 2,5
	Otro	0,7	0,4	- 0,3
China	Android	74,0	78,8	4,8
	iOS	24,5	20,1	- 4,4
	Windows	1,0	0,7	- 0,3
	Otro	0,6	0,4	- 0,2
España	Android	88,6	93,9	5,3
	iOS	8,8	5,5	- 3,3
	Windows	2,5	0,6	- 1,9
	Otro	0,1	0	- 0,1
Japón	Android	51,6	55,7	4,1
	iOS	45,6	43,2	- 2,4
	Windows	0,2	0,1	- 0,1
	Otro	2,5	1	- 1,5
Italia	Android	69,4	79,6	10,2
	iOS	15,8	13,3	- 2,5
	Windows	13,3	6,4	-6,9
	Otro	1,5	0,7	- 0,8

Cuadro 3.2: Porcentaje de Ventas Smartphones

3.3.3. Creación Aplicación Móvil

Para realizar el diseño de la aplicación móvil del sistema remoto del comedero para mascotas, se debe tener instalado en el equipo de trabajo las siguientes herramientas:

3.3.3.1. Java Development Kit (JDK):

Es un software que provee herramientas de desarrollo para la creación de programas en lenguaje Java. Se puede descargar de manera gratuita desde la página del fabricante [37]. Java es compatible tanto en Windows como Linux.



Figura 3.27: Aplicación JAVA

3.3.3.2. Android Studio:

Es el entorno oficial de desarrollo para aplicaciones Android. Se adquiere de manera gratuita del sitio web del fabricante [38] y se puede utilizar en Windows y Linux.

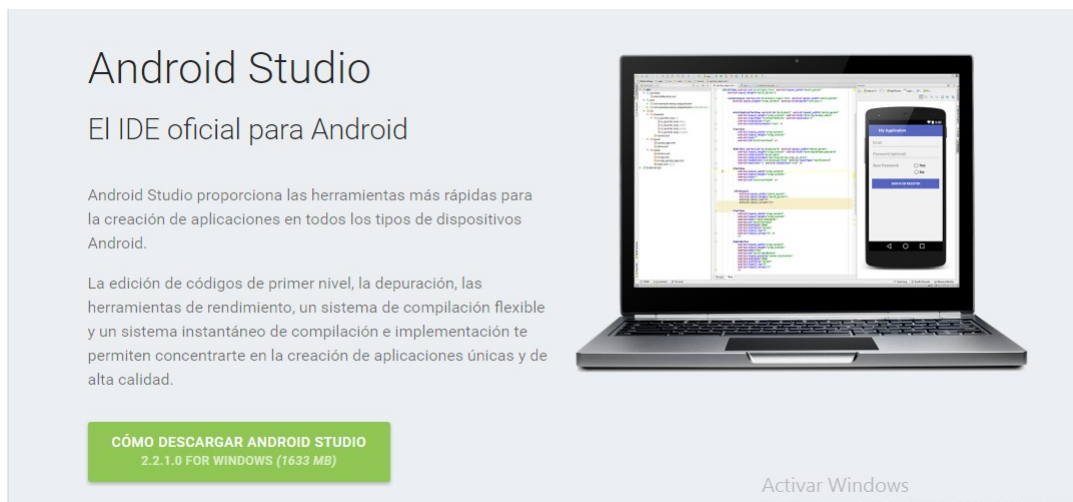


Figura 3.28: IDE Android Studio

Luego de tener instalado el jdk y el Android Studio se procede a la creación de la aplicación móvil que se acomode a nuestra necesidad, teniendo como prioridad el ser de fácil manejo para nuestro usuario final y cumpliendo plenamente con las funciones para la cual fue creada. Cuando se crea el proyecto en Android Studio se debe seleccionar el sistema mínimo que soportaría la app (aplicación móvil) y la plataforma sobre la cual se quiere trabajar, teniendo esto en cuenta se debe saber que para

el segundo semestre del año 2016 las versiones que ha desarrollado android desde sus inicios son las siguientes:

Informe Android enero 2016: Lollipop crece pero Marshmallow no despeg

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.7%
4.1.x	Jelly Bean	16	9.0%
4.2.x		17	12.2%
4.3		18	3.5%
4.4	KitKat	19	36.1%
5.0	Lollipop	21	16.9%
5.1		22	15.7%
6.0	Marshmallow	23	0.7%

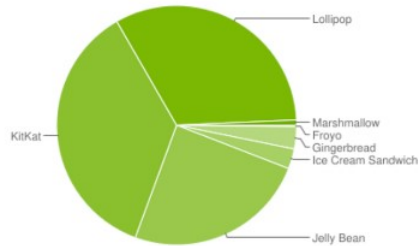


Figura 3.29: Sistemas Operativos Android

Para la aplicación se selecciona como versión mínima la 2.3 Gingerbread (android a todas sus versiones les ha asignado nombres de postres o dulces) es decir, que en sistemas por debajo de este no funcionaría, esta decisión se toma porque dispositivos con versiones por debajo de 2.3 son relativamente pocos y con ello tratar de abarcar el número más grande de dispositivos android a los cuales se les pueda instalar la app sin inconvenientes. La plataforma seleccionada para trabajar fue la 5.0 Lollipop, esto puede variar dependiendo de la decisión de cada programador.

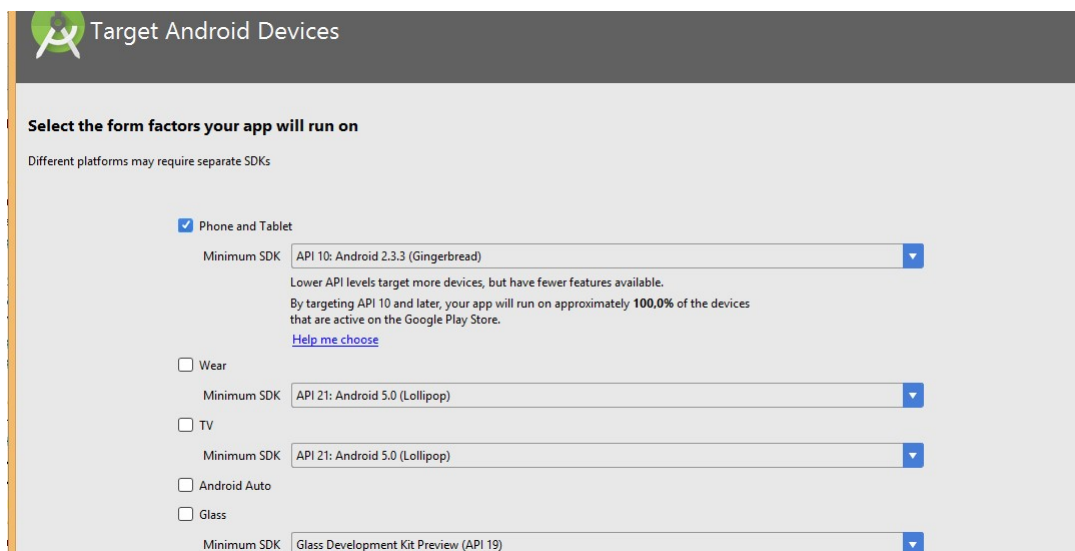


Figura 3.30: Interfaz Inicial Android Studio

3.3.3.3. Componentes de la Aplicación

La aplicación creada está conformada por 5 activities, donde cada una de ellas posee una Relative-Layout y en el interior de esta contiene botones, imágenes y textos que son agregados según el criterio de diseño o la utilidad que se busca con la creación.

ACTIVITY

Corresponde a una ventana o a un cuadro de dialogo en una aplicación de escritorio. Un Activity es una clase donde mostraremos Views (Vistas) para generar la interfaz de usuario y responderemos a eventos que se realicen sobre ella.

Son entidades independientes, capaces de llamarse entre ellas, pasándose parámetros y recibiendo respuestas, de modo que su funcionamiento sea un conjunto. Cada vez que una Activity llama a otra, la actividad que crea la llamada se introduce en una pila siguiendo el orden LIFO (Last Input First Output, último en entrar primero en salir), así el usuario pulsando el botón de “volver atrás” del teléfono, podrá recuperar la actividad anterior. Esto es una característica muy común de las actividades, a pesar de que el programador puede anular esa acción eliminando de la lista las actividades que menos interese. A cada Activity se le asigna una ventana sobre la cual se puede dibujar una interfaz de usuario. En esta ventana el programador define qué elementos visuales y en qué lugar se van a mostrar. Este contenido se indica de un modo jerárquico mediante vistas, que son objetos que implementan la clase View (Vista). Esta clase es el nexo de unión entre las Activity y el usuario, ya que se encargan de recibir los eventos realizados y saber cómo va a responder cada uno de los elementos gráficos que derivan de la clase View [39].

A continuación se muestran las activities_main desarrolladas para la creación de la aplicación móvil del comedero para mascotas, al lado izquierdo la parte gráfica y al lado derecho los elementos que contiene (botones, imágenes, texto, etc).

La primera Activity, es la bienvenida a la aplicación:



Figura 3.31: Primer Activity

La segunda Activity, es la de inicio (home) y opciones:

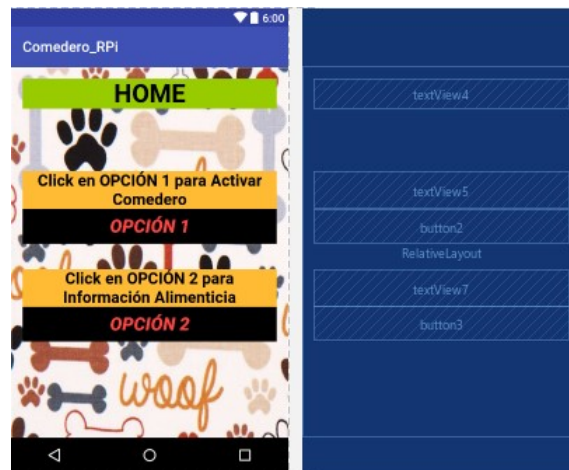


Figura 3.32: Segunda Activity

La tercera Activity, es la de suministro de alimento:

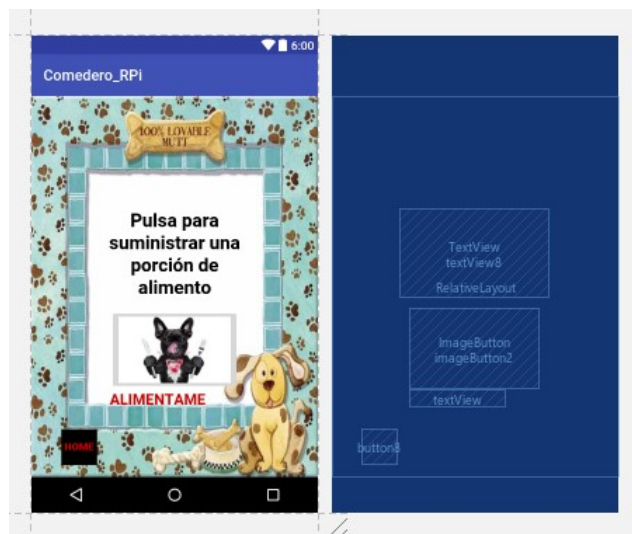


Figura 3.33: Tercera Activity

La cuarta Activity, información extra:

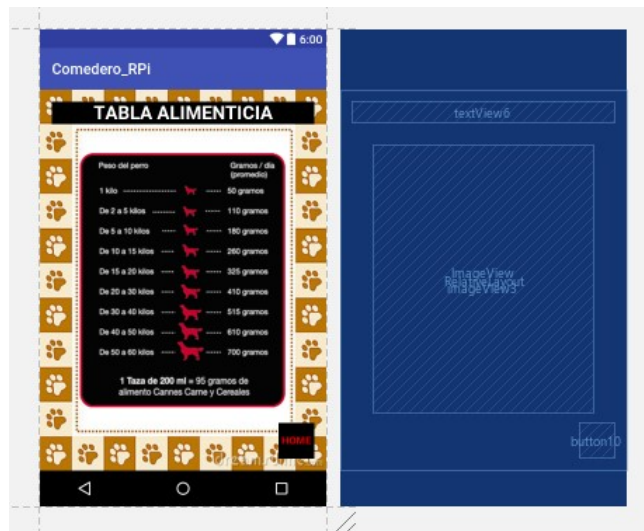


Figura 3.34: Cuarta Activity

La quinta Activity, WebView:

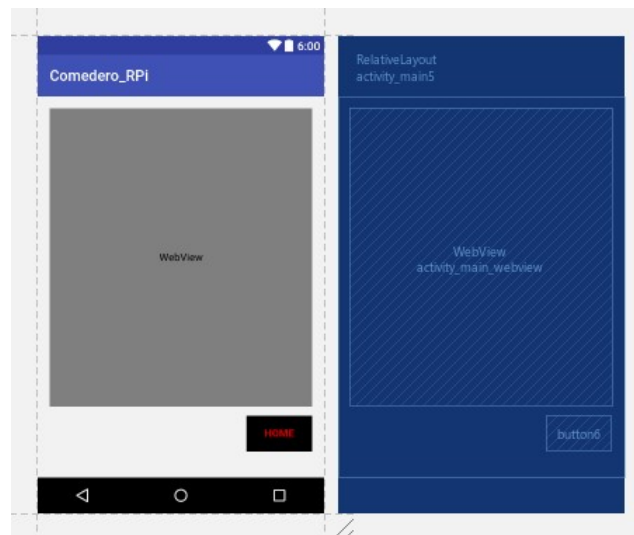


Figura 3.35: Quinta Activity

La versión utilizada de android studio es la 2.2, la cual al crear un nuevo proyecto o agregar una nueva activity trae por defecto en cada una de ellas la layout de tipo relative, si se requiere utilizar una layout de algún otro tipo se modifica por medio de código en la activity_main.xml en la pestaña text.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

Figura 3.36: RelaviteLayout

LAYOUTS

Son elementos no visuales destinados a controlar la distribución, posición y dimensiones de los controles que se insertan en su interior. Estos componentes extienden a la clase base `ViewGroup`, como muchos otros componentes contenedores, es decir, capaces de contener a otros controles. Igual que la vistas, pueden ser definidos en código, aunque la forma más habitual y sencilla de definirlos es utilizando código XML [39].

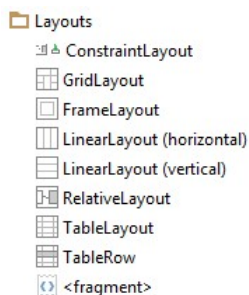


Figura 3.37: Tipos de Layouts

FrameLayout: Permite el cambio dinámico de los elementos que contiene. Esta característica se puede usar en actividades pero con la aparición de los fragmentos tiene más sentido. `FrameLayout` posiciona las vistas usando todo el contenedor, sin distribuirlos espacialmente. Se comporta como una pila de elementos, según se van añadiendo elementos a él, se van colocando uno encima de otro, quedando ocultos los que están en la parte inferior.

Linear Layout: En una disposición lineal, como su nombre indica, todos los elementos se muestran en una forma lineal, ya sea horizontalmente o verticalmente.

Relative Layout: En una disposición relativa de cada elemento, se arregla en relación con otros elementos o un elemento padre.

Table Layout: Son disposiciones de una tabla en Android, la cual funciona de la misma manera que las disposiciones de la tabla de trabajo HTML. Puede dividir sus diseños en filas y columnas.

Después de tener definida la actividad con su respectiva layout se pasa a introducir las vistas. Una vista es un objeto que se puede dibujar y utilizar como un elemento en el diseño de la interfaz de usuario (un botón, una imagen, un texto...). Cada uno de estos elementos se define como una subclase de la clase `View`; por ejemplo, la subclase para representar un texto es un `TextView`, para una imagen es una `ImageView` y para un botón es un `Button`. Esto puede variar en cada una de las actividad que se anexen, pues se debe tener en cuenta las vistas que se agreguen en cada una de ellas y su propósito final.

TextView: Muestra un texto y opcionalmente permite su edición. Otros parámetros para su visualización son el tamaño de letra, la posición, la distancia con otros objetos y el tamaño del `TextView` en sí. En este caso mostramos un texto que encontraremos en una clase Java de vectores para mostrar por orden los textos en la lista.

ImageView: Muestra una imagen arbitraria. Esta se encuentra en una clase Java creada desde el código principal para invocar la llamada y mostrar por orden en un vector, las imágenes de la lista del menú de navegación desplegable.

Button: Representa un botón que puede ser pulsado. Deberá tener escuchadores que indiquen una acción determinada cada vez que el botón se presione.

Tener presente que la layout (relative, linear, frame, table) y las vistas van incorporadas en la activity y que esta puede visualizarse en forma gráfica (design) o en forma de texto de código (text), por medio de dos pestañas en la parte inferior izquierda de la activity_main.xml. Si se agrega otra activity está por defecto asigna un número secuencial activity_main2.xml y así sucesivamente dependiendo del número de activity que se agreguen para la creación de la app, esto no quiere decir que no se le pueda asignar un nombre, pero para la app del comedero se respetó la secuencia. Cada una de las activity_main.xml se guarda en una carpeta que lleva el nombre de Layout. Esto se puede observar en la siguiente imagen al igual que el código utilizado en la activity_main.xml.

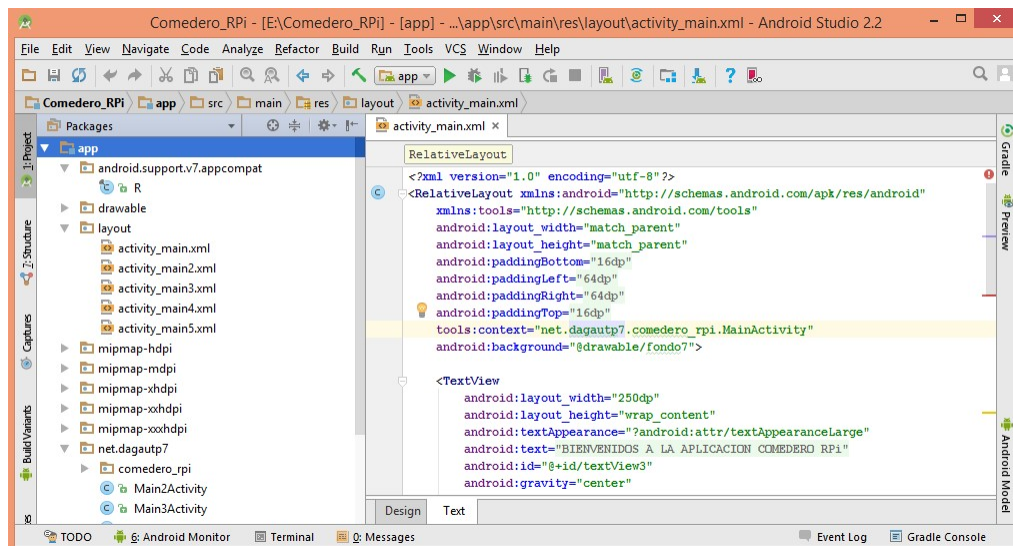


Figura 3.38: Ejemplo RelativeLayout

En resumen las activity cumplen con un sistema de visualización que muestra como sería observada la app por el usuario en forma de presentación y en forma de código.

Luego de realizado el diseño, se pasa a dar las funciones explícitas a cada elemento que están dentro de la activity_main.xml, quien permite que las tareas sean ejecutadas son las MainActivity.java (allí por medio del lenguaje de programación java se le asignan acciones a los botones, a las imágenes, etc).

Por ende las activities también están conformadas por dos partes: la parte lógica y la parte gráfica. La parte lógica es un archivo .java que es la clase que se crea para poder manipular, interactuar y colocar el código de esa actividad.

MainActivity.java: Es un método que utiliza Android para iniciar una actividad, invocando los métodos específicos que corresponden a las etapas del ciclo de vida. Cuando se crea un nuevo proyecto este viene con su activity_main.xml que a su vez trae consigo el MainActivity.java [39].

El MainActivity.java trae consigo un código con las siguientes líneas de comando:

Algoritmo 3.4 Algoritmo MainActivity.java

```
Package "nombre" //Esta línea únicamente indica el nombre del paquete en donde se encuentra
nuestra clase//
import android.os.Bundle;
import android.app.Activity; //Los dos imports, son la forma de decir que necesitamos esos ar-
chivos para trabajar dentro de la clase, los que ya nos da Android para no tener que escribir las cosas
desde cero y donde nos aparecen las variables después de ser declaradas//
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
}
```

En esta última sección de código se crea una clase que se llama "MainActivity" y se extiende de activity, esto es el concepto de herencia de la famosa programación orientada a objetos, estamos diciendo que "mainactivity" es una clase que hereda las cosas de la clase Activity que ya tiene Android definida. Todas las activities deben llevar por lo menos un método, el método "oncreate", que es en donde se crea la actividad o podemos decir que es donde se le da vida.

Del método "onCreate" lo más importante es la línea de código `SetContentView(R.Layout.activity_main)`, que es la que hace el trabajo de enlazar la parte lógica con la parte gráfica. El archivo XML que va a mostrarse cuando se mande a llamar la clase "MainActivity" es el archivo XML llamado "activity_main".

Para cerrar la explicación, si yo creo una actividad nueva y la llamo "Comedero", debo hacer que herede de activity si quiero que funcione como actividad y para decirle que el archivo XML que va a mostrar sea el "comedero.xml". La línea que dice "SetContentView" debe llevar dentro algo parecido a esto: `"setContentView"(R.layout.comedero)` [39].

En la aplicación creada, como se nombró anteriormente, se le dio por nombre a la activity la asignada por android studio al crear el nuevo proyecto. El código java utilizado en la MainActivity.java para la realización de la app del comedero de mascotas es el siguiente:

```

package net.dagautp7.comedero_rpi;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;

import net.dagautp7.Main2Activity;
import net.dagautp7.Main3Activity;

public class MainActivity extends AppCompatActivity {
    ImageButton siguiente;

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

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        siguiente = (ImageButton) findViewById(R.id.ibutton1);
        siguiente.setOnClickListener((v) -> {

            Intent siguiente = new Intent(MainActivity.this, Main2Activity.class);
            startActivity(siguiente);

        });
    }
}

```

Figura 3.39: MainActivity.java

Las Main2Activity.java, Main3Activity.java y Main4Activity.java tienen cambios mínimos con respecto a la MainActivity.java, como son los id asignados a los botones o a las imágenes que deben variar de una actividad a otra, también el cambio de botones por imágenes que cumplirían con la función de crear saltos entre las actividades. Por ejemplo en la MainActivity.java solo se agrega una imagen que cumple la función de dar un salto a la Activity 2, en este caso la imagen realiza la misma función que puede realizar un botón, dicha imagen se ve visualizada con el rostro de un bulldog francés pero que para java en la MainActivity.java tiene como id ibutton1. Si se compara con la Main2Activity.java donde se utilizan dos botones que visualmente aparecen como OPCION 1 y OPCION 2 pero que para java en la Main2Activity.java su id es button2 y button3, que también cumplen la opción de realizar saltos.

```

Main2Activity.java x
package net.dagautp7;

import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;

import net.dagautp7.comedero_rpi.R;

public class Main2Activity extends AppCompatActivity {

    Button siguiente2;
    Button siguiente3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        siguiente2 = (Button) findViewById(R.id.button2);
        siguiente2.setOnClickListener((v) -> {

```

Figura 3.40: Main2Activity.java Tabla 1

```

        Intent siguiente2 = new Intent(Main2Activity.this, Main3Activity.class);
        startActivity(siguiente2);
    });

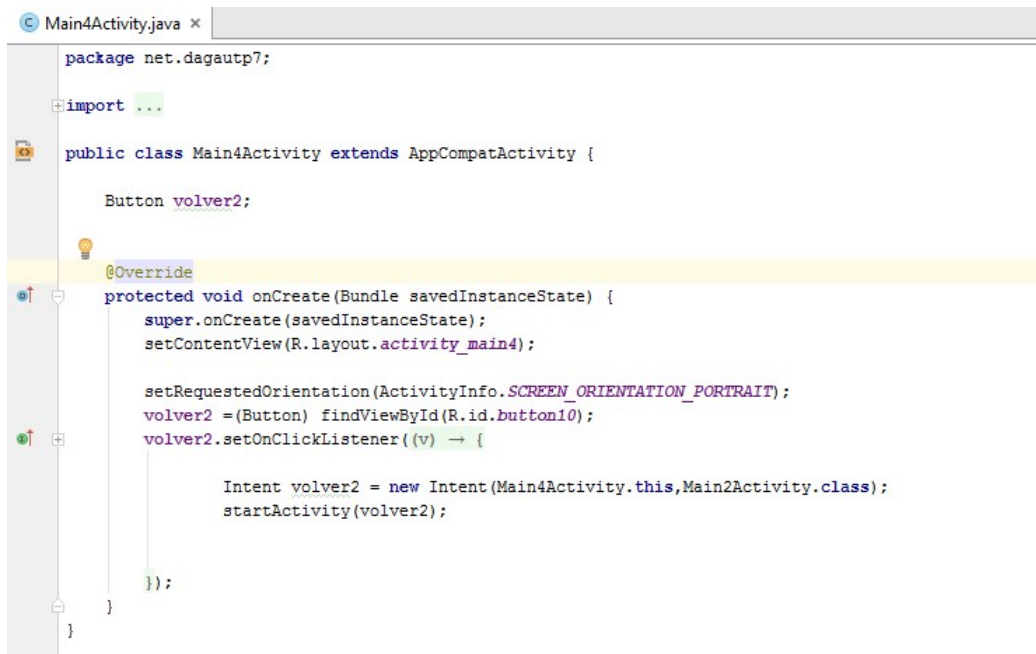
    siguiente3 = (Button) findViewById(R.id.button3);
    siguiente3.setOnClickListener((v) -> {

        Intent siguiente3 = new Intent(Main2Activity.this, Main4Activity.class);
        startActivity(siguiente3);
    });
}
}

```

Figura 3.41: Main2Activity.java Tabla 2

Un punto en común entre las `Main3Activity.java`, `Main4Activity.java` y `Main5Activity.java` es que contienen un botón llamado HOME (HOME es como se visualiza, pero para java en cada una de las `MainActivity.java` esta agregado con un id diferente `Button6`, `Button8` y `Button10`) que permite al usuario volver a la Activity 2 (inicio o home) donde están contenidas las dos opciones fundamentales de la aplicación. La OPCION 2 es el salto a la Activity 4 (información extra) la cual contiene la imagen de una tabla que da al usuario la opción de saber más sobre las porciones por gramos que debe recibir cada perro dependiendo de su masa en kilogramos.



```
package net.dagautp7;

import ...

public class Main4Activity extends AppCompatActivity {

    Button volver2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main4);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        volver2 = (Button) findViewById(R.id.button10);
        volver2.setOnClickListener((v) -> {

            Intent volver2 = new Intent(Main4Activity.this, Main2Activity.class);
            startActivity(volver2);

        });
    }
}
```

Figura 3.42: Main4Activity.java

La OPCION 1 es el salto a la Activity 3 (suministro de alimento) que aparece visualmente representada con un bulldog francés con cubiertos en sus manos (id en java imageButton2), que al dar click sobre esta imagen nos permite ejecutar la WebView (Activity 5), la cual a su vez nos permite obtener la conexión con nuestro servidor y estando allí dar la orden de activación al comedero (permite seleccionemos una de las tres opciones de porciones que se crea la correcta para alimentar nuestra mascota).

```

package net.dagautp7;

import ...

public class Main3Activity extends AppCompatActivity {
    Button volver;
    ImageButton siguiente5;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        volver = (Button) findViewById(R.id.button8);
        volver.setOnClickListener((v) -> {

            Intent volver = new Intent(Main3Activity.this, Main2Activity.class);
            startActivity(volver);

        });

        siguiente5 = (ImageButton) findViewById(R.id.imageButton2);
        siguiente5.setOnClickListener((v) -> {

            Intent siguiente5 = new Intent(Main3Activity.this, Main5Activity.class);
            startActivity(siguiente5);

        });
    }
}

```

Figura 3.43: Main3Activity.java

La Main5Activity.java a diferencia de las anteriores contiene un Containers del tipo WebView la cual permite la comunicación de la app directamente con el servidor (y también contiene el ya llamado botón HOME nombrado anteriormente).

WebView nos va a permitir utilizar el motor interno de Android, WebKit, para mostrar diferentes páginas web dentro de la misma aplicación (ya sea en la misma u en otra actividad). De esta manera, todo se ejecutará en la misma aplicación y evitaremos tener que abrir otras aplicaciones [39].

Código java utilizado en la Main5Activity.java:

```

package net.dagautp7;

import ...

public class Main5Activity extends AppCompatActivity {
    Button volver4;
    // INI AGREGADO
    private WebView mWebView;
    // FIN AGREGADO

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main5);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
        volver4 = (Button) findViewById(R.id.button6);
        volver4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent volver4 = new Intent(Main5Activity.this, Main2Activity.class);
                startActivity(volver4);
            }
        });
    }
}

```

Figura 3.44: Main5Activity.java Tabla 1

```

    // INI AGREGADO
    mWebView = (WebView) findViewById(R.id.activity_main_webview);
    // Activamos javascript

    WebSettings webSettings = mWebView.getSettings();
    webSettings.setJavaScriptEnabled(true);

    // Url que carga la app (webview)
    mWebView.loadUrl("http://190.128.118.56");
    // Forzamos el webview para que abra los enlaces internos dentro de la APP
    mWebView.setWebViewClient(new WebViewClient());
    // Forzamos el webview para que abra los enlaces externos en el navegador
    mWebView.setWebViewClient(new MyAppWebViewClient());
    // FIN AGREGADO
}

// INI AGREGADO
@Override
// Detectar cuando se presiona el botón de retroceso
public void onBackPressed() {
    if(mWebView.canGoBack()) {
        mWebView.goBack();
    } else {
        super.onBackPressed();
    }
}
// FIN AGREGADO
}

```

Figura 3.45: Main5Activity.java Tabla 2

En el código se observa un http con un ip este se puede variar dependiendo de la necesidad de conexión que se requiera, de igual manera se puede agregar un url en vez de un ip.

3.3.4. Descripción Final de la Aplicación y Conceptos Claves para su Creación.

La aplicabilidad del sistema va diseñada esencialmente para ejecutar una interface de conexión entre el móvil y el servidor. Para la creación de la app se cumplieron ciertos requerimientos. En primer lugar la aplicación debía ser portable y de fácil acceso para el usuario, además operar en un dispositivo Android con conexión a Internet o datos para fines prácticos de funcionamiento.

La aplicación debía ser sencilla y ser creada para efectuar una tarea concreta que es la de implementar un protocolo TCP para la comunicación de datos entre un servidor y un usuario valiéndose de la implementación de sockets y un navegador web, además brindar una gran versatilidad en cuanto a su utilización y aplicación práctica.

En palabras más simples, su función era establecer conexión desde cualquier punto con la red doméstica con la cual se encuentra enlazada la raspberry pi por medio de un servidor y por tanto con el comedero para mascotas, acto seguido al realizar conexión con el servidor mostrar en la pantalla del dispositivo el servidor web con el que se realiza la conexión, que se puede observar cargándose en el módulo como si se tratase de una página web cualquiera, mediante la cual se harían las labores de control correspondientes para la ejecución de la tarea final que es proporcionar una ración de comida para la mascota.

Ya al tener el producto final con los conceptos de funcionamiento estipulados anteriormente, se pasó a la instalación y ensayo de la aplicación móvil en varios teléfonos celulares, buscando ver que la app no sufriera alteraciones dependiendo del tamaño de pantalla, resolución de imagen y gama de colores que se maneja de un equipo a otro según las reseñas de fabricación de cada marca comercializadora.

3.3.5. Instalación de la Aplicación Móvil.

Una vez que se ha terminado la arquitectura de la aplicación se procede a su instalación en el dispositivo móvil. La forma en que se realiza la instalación es generando un archivo de extensión apk desde el mismo android studio, la cual se anexa a una memoria externa micro sd o a la memoria interna del dispositivo y desde allí se instala como una aplicación de origen desconocida.

3.3.5.1. Generación de Certificado y Correspondiente APK

Seleccionar en el Menú de Android Studio Build > Generated Signed APK

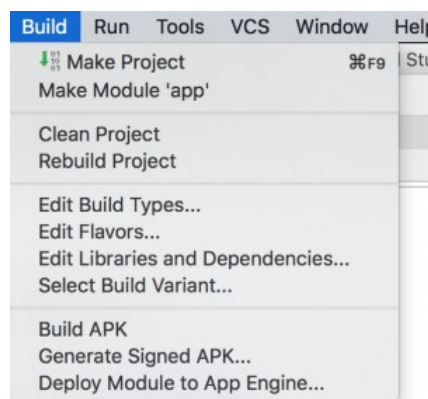


Figura 3.46: Generación APK

Se selecciona el módulo que se va a exportar



Figura 3.47: Selección del Módulo

A continuación, se debe crear el certificado o seleccionarlo de algún directorio si ya se dispone de él.



Figura 3.48: Asignación de Credenciales

Si se tiene que crear, se debe complementar el siguiente formulario de creación del certificado, teniendo en cuenta que se debe recordar la contraseña que se introduce para el certificado, pues es necesaria para realizar futuras exportaciones de la aplicación.

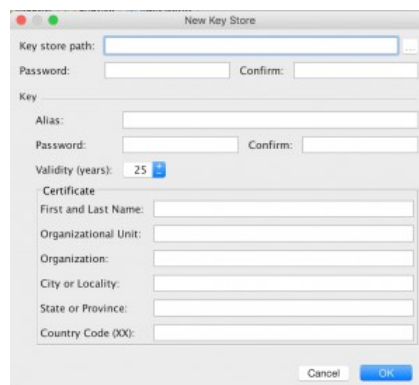


Figura 3.49: Formulario de Datos Básicos

Por último, se exporta la aplicación (fichero APK) y el certificado a un directorio de nuestro equipo o si se tiene sincronizada la computadora con el dispositivo móvil, se puede enviar directamente a la memoria interna del dispositivo móvil o a su memoria externa.



Figura 3.50: Exportación de Fichero

Luego de tenerla allí se procede a su instalación.

Para la instalación previamente se puede habilitar la opción orígenes desconocidos en el dispositivo móvil o en el momento de dar click sobre la aplicación a instalar saldrá un menú pidiendo que habilite dicha opción para poder proceder con la instalación.

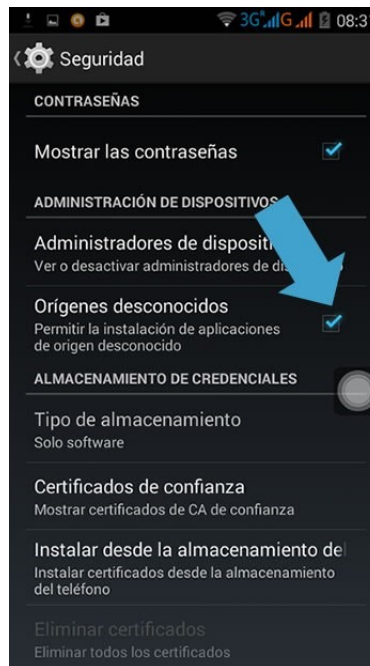


Figura 3.51: Configuración Permisos Smartphone

Es posible que en algunos equipos cuando se active la opción Orígenes Desconocidos muestre un mensaje confirmando si se está seguro de hacer esto, con el siguiente mensaje: El teléfono y los datos personales son más vulnerables a los ataques de aplicaciones de origen desconocido. El usuario acepta ser el único responsable de cualquier daño en el teléfono o pérdida de datos que se pueda derivar del uso de estas aplicaciones.

En el dispositivo móvil no se presentarán problemas si se sabe el origen de la app y se sabe el tipo de función que cumplirá. Por tanto, la aplicación creada para activar el comedero para mascotas, no tiene porque generar conflicto alguno en el dispositivo móvil.

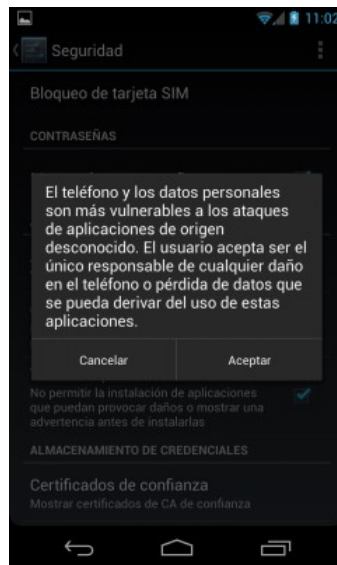


Figura 3.52: Advertencia de Seguridad

Se da click en “Aceptar” y ya se puede instalar la aplicación de origen desconocido.

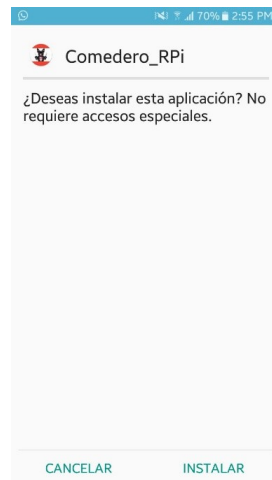


Figura 3.53: Instalación APK en Smartphone

Capítulo 4

PRUEBAS Y RESULTADOS

Elegir un comedero para mascotas implica que el amo debe tomar una decisión en beneficio de su mascota, la cual se debe realizar teniendo en cuenta las características físicas del animal, como el tamaño y la raza. Antes de resaltar los avances del diseño y el funcionamiento del comedero para mascotas, primero se debe destacar que los primeros pasos de la construcción se fundamentaron en dar con un recipiente que se adecuará a las necesidades de la mascota, que le ayudará a mantener una buena digestión y evitará infecciones o enfermedades en las articulaciones por las malas posturas en el momento de ingerir los alimentos. El tamaño del recipiente depende del tamaño de la mascota, puesto que este debe adaptarse a la cantidad de alimento a consumir según su talla y masa ideal. El comedero puede alimentar de forma organizada y en cantidades exactas a la mascota dependiendo de sus necesidades. Como se ha mencionado anteriormente, el comedero ha sido diseñado para beneficiar canes de razas pequeñas o gatos de igual tamaño, siendo esta la principal limitación del comedero. A la hora de elegir un perro muchos se deciden por perros de raza pequeña; se denominan así porque su peso se encuentra entre los 5 kg y 15 kg. Los perros de raza pequeña son muy populares gracias a su mayor adaptabilidad a los diferentes estilos de vida. Las razones de esta preferencia suelen ser muy diversas pero entre otras suele ser porque un perro de raza pequeña resulta ideal para vivir en un apartamento o en una casa pequeña. Otra de las ventajas es que resultan más económicos de mantener pues no necesitan comer tanto como los perros grandes ni requieren tanta dedicación.

La construcción del diseño se basa en piezas de acrílico, para su obtención fue utilizada una cortadora láser que permitiera realizar cortes y trazos precisos, garantizando precisión en la unión de todas las piezas. Los planos del diseño, luego de haber sido realizados en solidworks, pasan a CorelDraw para allí proporcionar el archivo a la cortadora láser y estipularle por medio de los bordes de colores, los cortes que se desean realizar.

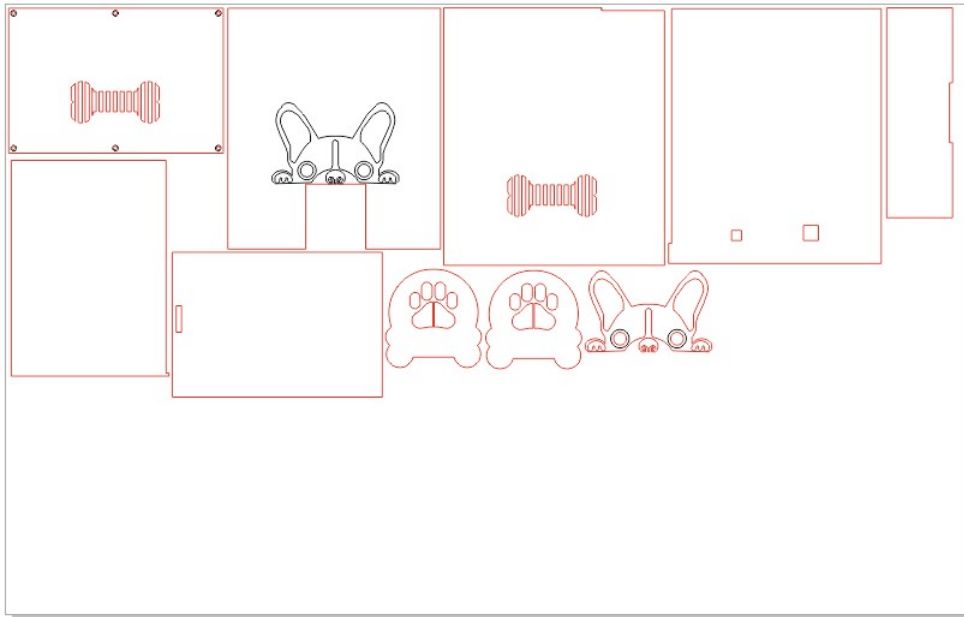


Figura 4.1: Piezas Corel Finas



Figura 4.2: Piezas Corel Cortes

Posteriormente a la impresión en acrílico, se efectúan los cortes en cartón paja para hacer la validación del diseño y estudiar sus posibles errores.

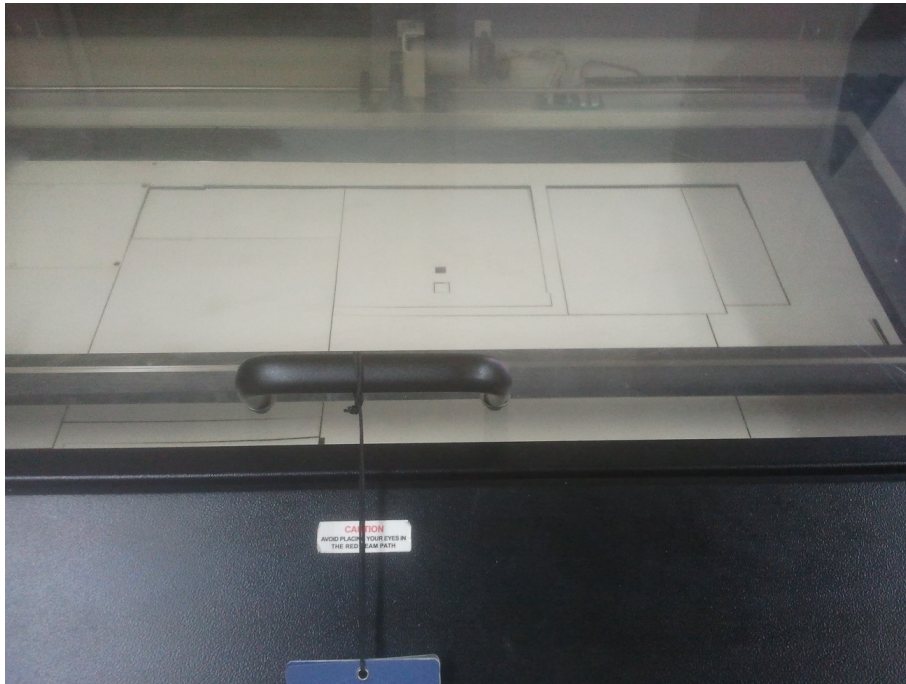


Figura 4.3: Primeros Cortes en Cortadora Láser



Figura 4.4: Prototipo Armado en Cartón

Al realizar la unión de todas las piezas se determinan errores menores que eran delimitados por el diámetro del cartón, ya que las medidas inicialmente descritas para cada cara eran las suministradas por el diámetro del acrílico que eran de 3 mm. Luego de hacer correcciones en algunos componentes

se pasa a realizar el corte de las piezas en el material final que es el acrílico, obteniendo los siguientes resultados:



Figura 4.5: Piezas Cortadas en Acrílico

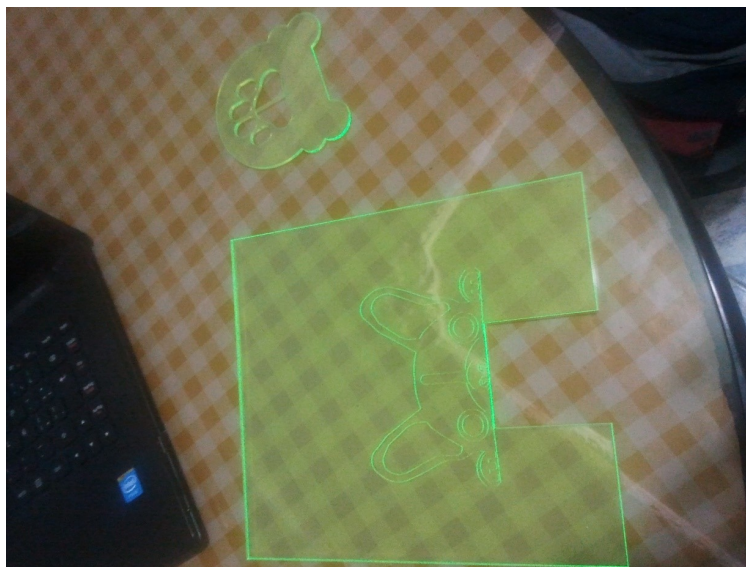


Figura 4.6: Piezas Finas Cortadas en Acrílico

Después de tener impresas las piezas, se unen dando origen a otra gran pieza que se complementará con tapas y otras piezas para terminar materializadas en el ensamble total. Para fijar cada una de las piezas se utiliza cloruro de metileno, el cual es un líquido volátil no inflamable, transparente, pesado e incoloro, con un olor suave parecido al éter, sin punto de inflamación, con arreglo a los procedimientos de ensayo normalizados, completamente miscible con una amplia gama de disolventes, elevado poder de disolución, excelentes propiedades de reciclado, altamente estable.

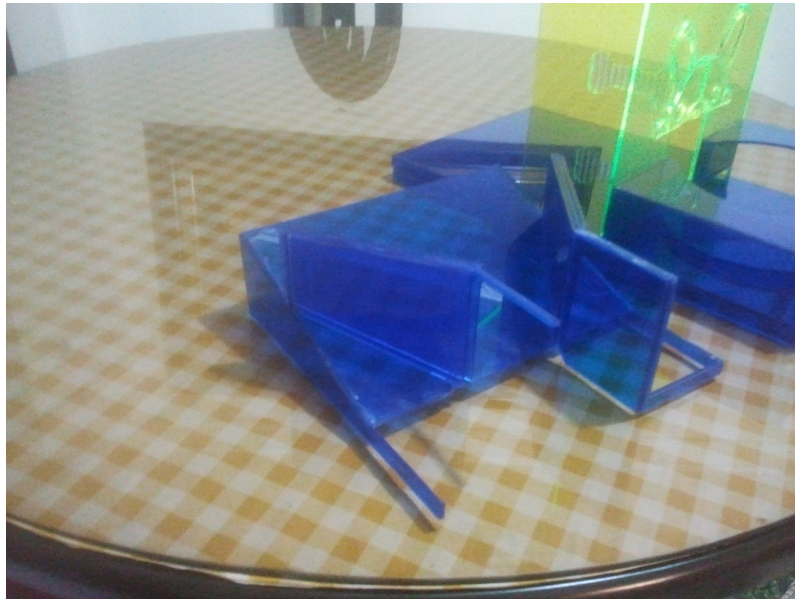


Figura 4.7: Ensamble de Piezas

Algunas piezas fueron impresas directamente por medio de una impresora 3D ya que los ángulos y sus bordes precisos no es posible obtenerlos mediante el uso de la cortadora láser.

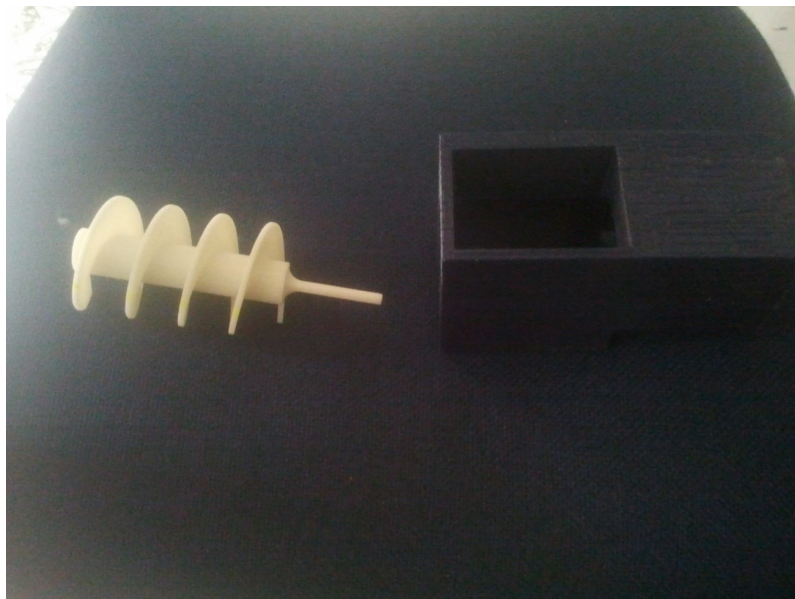


Figura 4.8: Tornillo y Base Impresos



Figura 4.9: Compartimento para Tornillo



Figura 4.10: Acople Tornillo en el Compartimento



Figura 4.11: Piezas para Adherir los Componentes

Luego de tener todas las partes y piezas se procede a realizar el ensamble total, haciendo un minucioso procedimiento de pegado que se debe ejecutar paso a paso para obtener el resultado final esperado, dando como resultado:



Figura 4.12: Comedero Ensamblado Totalmente



Figura 4.13: Vista Superior Comedero Ensamblado

Al tener el diseño terminado se le agrega la parte del circuito electrónico, compuesto por la raspberry pi, el circuito de potencia y un motor DC con caja reductora.

El comedero cuenta con un compartimento donde se aloja la Raspberry Pi, el circuito de potencia y el motor; este se encuentra protegido por una tapa sujeta con tornillos; esto con el fin de brindar seguridad a la mascota y que el circuito no se vea afectado por agentes externos. Algo más que se debe tener en cuenta es que se debe contar con una ip fija para la conexión del comedero, esto nos permite montar el servidor web y dirigir un dominio a esta IP sin tener que mantener actualizado el servidor DNS cada vez que cambie la IP, como ocurre con las IPs dinámicas.

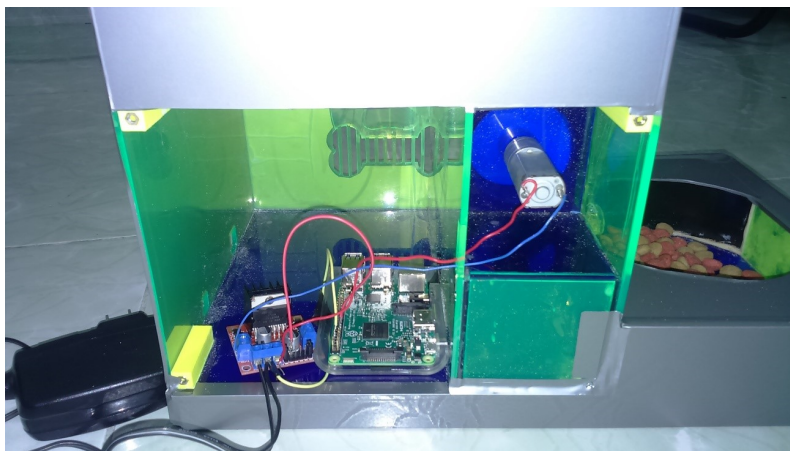


Figura 4.14: Vista Interna Comedero

Por último se pone en marcha la aplicación móvil y la revisión de la conexión con el servidor y de allí con la raspberry pi y el circuito electrónico, dando como resultado final el funcionamiento esperado.

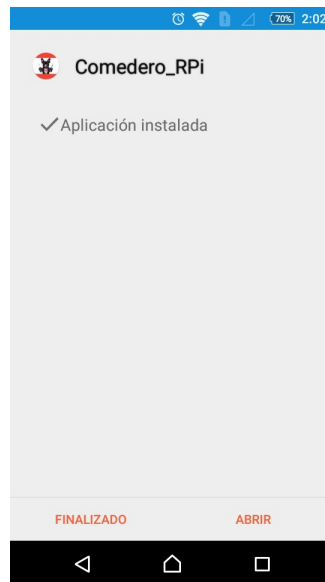


Figura 4.15: Instalación Aplicación en Smartphone



Figura 4.16: Interfaz Inicial Aplicación Comedero

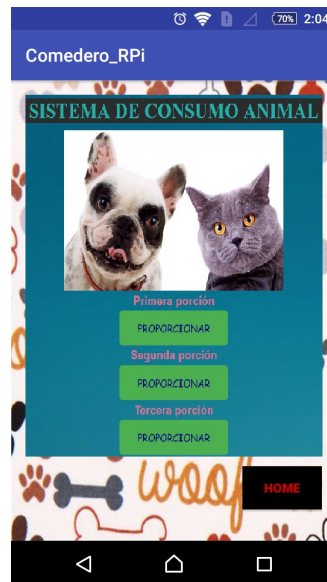


Figura 4.17: Interfaz Final de la Aplicación

Al proporcionar el alimento a la mascota (perro de raza pequeña o gato), este se acomoda a las dimensiones del diseño y se observa cómo puede alimentarse sin dificultad.



Figura 4.18: Prueba Funcionamiento Comedero

La gran ventaja del comedero es el ofrecer al dueño de la mascota la facilidad de alimentarla en tiempos determinados por él mismo. Para obtener un funcionamiento óptimo se destacaron los siguientes aspectos:

- Recopilación de información por medio de la observación, determinando con ello que la evolución nutricional se fundamenta en una adecuada alimentación en horas fijas y determinadas.
- El animal ingiere cierta cantidad de alimento dependiendo de su masa y la actividad física que este realiza.

Las pruebas se realizaron de forma real con un perro de raza pequeña, donde se saciaron sus necesidades alimenticias en horas determinadas, esto impidió que el animal se viera afectado por un largo lapso de tiempo, entre una y otra ración de comida. El dueño de mascota es libre de elegir la cantidad de alimento y horario en el que quiera alimentar a su mascota, se debe tener en cuenta que las características del

alimento influyen en su vitalidad, por eso se les debe tratar de proveer concentrado de la mayor calidad posible, puesto que su estómago es más pequeño, tiene menos apetito y es más sensible. Además es importante proporcionarle alimento de un tamaño adecuado, ya que croquetas de gran tamaño pueden generar inconvenientes a la hora de masticar. Cabe resaltar que el comedero ha sido diseñado para funcionar únicamente con concentrado que se acomode a las necesidades de la mascota de raza pequeña.

A continuación se muestra una tabla con los costos del proyecto:

Materiales	Recursos Propios	Recursos Universidad
Tarjeta Raspberry Pi	\$150,000	
Motor DC	\$12,000	
Cargador 12 V	\$8,000	
Cables para Conexión	\$5,000	
Módulo L298	\$15,000	
Impresión 3D	\$375,000	
Corte Láser	\$35,000	
Panel Acrílico	\$136,000	
Filamento ABS	\$200,000	
Software SolidWorks		Licencia UTP
Software Android Studio	Open Source	
Software Python	Open Source	
Software PHP	Open Source	
Software HTML	Open Source	
Software Apache	Open Source	
TOTAL	\$962,000	

Cuadro 4.1: Costos del Prototipo

Capítulo 5

CONCLUSIONES Y FUTUROS TRABAJOS

5.1. Conclusiones

En la actualidad es posible construir sistemas electrónicos con componentes asequibles y plataformas de fácil desarrollo como lo es Raspberry Pi. Esta plataforma está diseñada para facilitar la realización de prototipos que requieren un desarrollo de software avanzado, gracias a su capacidad de procesamiento, facilidad para acoplar dispositivos externos y bajo consumo energético, el desarrollo de este proyecto ha finalizado con un prototipo funcional y se ha comprobado que es posible construir un sistema de dosificación para mascotas controlado mediante una aplicación móvil utilizando esta tarjeta como componente principal del mismo. De acuerdo a los parámetros de diseño para la creación de un producto; como robustez, usabilidad, cognición, optimización, flexibilidad, estética y necesidades; se ha creado un prototipo funcional de dispensador de comida para animales.

Una de las ventajas de programar el tiempo de dosificación en lenguaje python, es su facilidad para cambiar el parámetro de tiempo de encendido; ya que python es un lenguaje fácil de aprender y simple en comparación con los otros lenguajes utilizados para el desarrollo de este proyecto. Esto se puede realizar con solo cambiar un valor en una línea de código.

La dosis que puede dosificar este comedero es bastante precisa, adicionalmente el tiempo de respuesta es inmediato, de esta manera el usuario puede tener la tranquilidad que se está realizando la alimentación de la mascota en el tiempo específico; sin embargo se debe tener en cuenta que para que este funcione de la manera adecuada, se debe proveer un concentrado de tamaño reducido, ya que utilizar concentrado de mayor tamaño puede generar obstrucciones en el canal de dosificación y en un caso extremo puede llegar a dañar el compartimiento y por ende el prototipo.

El servidor web instalado y configurado en la raspberry pi, se diseña para atender y responder a la peticiones del navegador que se soliciten usando el protocolo HTTP. Este servidor es uno de los elementos centrales de la red de trabajo, el cual permite la conexión de equipos remotos tales como el smartphone del usuario permitiendo el control, monitoreo y conexión con la aplicación móvil. El servidor web se enlaza directamente con la aplicación móvil, la cual dará las pautas de interacción del usuario con el comedero. La orden de ejecución para la activación del GPIO es dada desde el servidor web al código elaborado en PHP, quien a su vez hace un llamado al código PYTHON el cual coloca en HIGH el pin correspondiente. La unión de las herramientas webview (aplicación móvil) y la dirección ip permiten enlazar correctamente la app con el servidor el cual le responde al usuario con el contenido

que éste solicita.

El prototipo ha sido enfocado en el uso doméstico debido a su tamaño; sin embargo, es posible adaptar este a otros campos como podría ser el ganadero o agropecuario para la alimentación de diferentes tipos de animales.

5.2. Futuros Trabajos

Durante el desarrollo de este proyecto, se han encontrado algunos aspectos que pueden ser considerados a mejorar en un futuro, ellos son:

- Disminuir costos del prototipo mediante el reemplazo de componentes por tecnologías más económicas.
- Adicionar una cámara para la observación de la mascota.
- Incluir un sistema de transmisión de audio para que el amo pueda hablarle a su mascota, de esta manera realizar un llamado y que la mascota pueda reaccionar de una manera más amigable.
- Registrar los movimientos de alimentación en una base de datos y realizar gráficos estadísticos con dicha información.
- Integrar sensores que indiquen los niveles de carga de alimento y que se emita una alerta al amo cuando estos volúmenes se encuentren en niveles críticos.
- Mejorar la interfaz gráfica del diseño del comedero.
- Desarrollar el prototipo dispensador de agua complementario al sistema actual.
- Agregar un registro de validación de usuario y contraseña para mejorar la seguridad del dispositivo.

Bibliografía

- [1] Ángel Jiménez de Luis, “El valor de android,” *El Mundo*, 2016.
- [2] PrimaninSpirada, “Ayude a su perro come para vivir, no vivir para comer.”
- [3] S. de la República, “Ley 1774, 6 ene 2016,” 1 2016.
- [4] E. C. D. COLOMBIA, “Estatuto nacional de protección animal. ley 84 de 1989,” *Bogotá D.E., 27 de diciembre de 1989*.
- [5] S. J. M. Galán, “Animales en colombia, serán seres sintientes sujetos de derechos,” Noviembre 2015.
- [6] L. Tarde, “En 6 años la población de perros y gatos se duplicó,” Febero 2013.
- [7] E. Diario, “Aumentan perros callejeros,” Febrero 2013.
- [8] R. Dinero, “En colombia ¡los prefieren perros!,” Julio 2012.
- [9] C. Cruz Vázquez, “Mascotas: ¿amigos medicinales?,” *Alternativas en psicología*, vol. 14, no. 20, pp. 48–57, 2009.
- [10] C. K. Chandler, *Animal assisted therapy in counseling*. Routledge, 2012.
- [11] M. Nagasawa, S. Mitsui, S. En, N. Ohtani, M. Ohta, Y. Sakuma, T. Onaka, K. Mogi, and T. Kikusui, “Oxytocin-gaze positive loop and the coevolution of human-dog bonds,” *Science*, vol. 348, no. 6232, pp. 333–336, 2015.
- [12] R. Lozano Garzón, “Mascotas tienen a tope producción de alimentos balanceados,” *El Tiempo*, Septiembre 2014. Redacción Economía y Negocios.
- [13] Euromonitor, “European marketing data and statistics 2000,” 2000.
- [14] R. Dinero, “La comida para perros se vende como pan caliente,” 2015.
- [15] D. A. Dzanic, “The association of american feed control officials dog and cat food nutrient profiles: Substantiation of nutritional adequacy of complete and balanced pet foods in the united states,” *Journal of nutrition*, vol. 124, no. 12, p. 2535S, 1994.
- [16] M. Palmero Colado, “Requerimientos nutricionales del gato.,” *Anuario AAMeFe.*, pp. 95–97, 2003.
- [17] I. Burgos Costalargo, “Requerimientos nutricionales en perros,” *Alaskan Malamutes, Madrid - España*, 2013.
- [18] L. P. C. Case, D. P. Hirakawa, D. A. L. P. Case, D. P. Carey, and D. A. Hirakawa, *Nutrición canina y felina: manual para profesionales*. No. 619: 636.045, Harcourt Brace, 1997.

- [19] U. P. S.A., “¿qué es el acrilico?,” 2016.
- [20] D. systemes, “Las soluciones de solidworks abarcan todos los aspectos del proceso de desarrollo,” 2016.
- [21] G. G. Sergio, “El gran libro de solidworks,” *MARCOMBO ediciones técnicas. Barcelona, España*, 2008.
- [22] E. Upton and G. Halfacree, “Raspberry pi.: guía del usuario,” *Anaya Multimedia*, 2013.
- [23] R. Cellan-Jones, “A 15 pound computer to inspire young programmers,” May 2011.
- [24] R. P. FOUNDATION, “Raspberry pi 3 model b,” Feb. 2016.
- [25] M. Richardson and S. Wallace, *Getting started with raspberry PI*. °Reilly Media, Inc.", 2012.
- [26] A. Y. R. Hernández and R. C. Sánchez, “ConversiÓN a/d con protocolo spi para audiofrecuencias,” *JÓVENES EN LA CIENCIA*, vol. 1, no. 2, pp. 1470–1475, 2015.
- [27] G. Van Rossum and F. L. Drake Jr, “Tutorial python,” *Disponible gratuitamente em <http://python.org>*, 2004.
- [28] B. Croston, “Librería rpi.gpio,” 2016.
- [29] M. J. Kabir, *Servidor Apache 2*. 2003.
- [30] L. Welling and L. Thomson, *Desarrollo web con PHP y MySQL*. 2005.
- [31] I. S. Graham, *The HTML sourcebook*. John Wiley & Sons, Inc., 1995.
- [32] J. D. Gauchat, *El gran libro de HTML5, CSS3 y Javascript*. 2012.
- [33] M. Glass, Y. Le Scouarnec, and E. Naramore, *Desarrollo web con PHP, Apache y MySQL*. 2004.
- [34] E. Colombiano, “Después de la subasta de 4g, colombia queda con seis operadores,” 2013.
- [35] MINTIC, “Boletín trimestral de las tic cifras primer trimestre 2016,” 2016.
- [36] F. Castro, J. Benavides, *et al.*, “Actualización del estudio sobre la competencia en el mercado de telefonía móvil en colombia,” 2015.
- [37] ORACLE, “Java se downloads,” 2016.
- [38] A. Studio, “Android studio,” 2016.
- [39] S. Axel, “El gran libro de android,” *Madrid-España: Publishing Limited*, 2012.