



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

“Sistema Web para comparar alimentos de
mascotas (perros y gatos)”.

Estudiante:

Iván Eduardo Martínez Villegas

Profesora Guía:

María Antonieta Soto Ch.

Enero 2019

Chillán, Chile



Sistema Web para comparar alimentos de mascotas (perros y gatos)



Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad del Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil en Informática.

El proyecto titulado “Sistema Web para comparar alimentos de mascotas (perros y gatos)” tiene como propósito agilizar los procesos de búsqueda y comparación de alimentos para mascotas, mediante el desarrollo de una Aplicación Web que permita, principalmente, a los propietarios de mascotas acceder a una plataforma en línea desde una computadora. Debe ofrecer las principales funcionalidades que tiene la mayoría de los sitios web, tales como permitir el registro de nuevos usuarios, el inicio de sesión y mostrar información de los diferentes alimentos que se publican junto con las tiendas que los ofrecen.

En términos de desarrollo, el Sistema consta de dos capas, la capa front-end y la capa back-end. La primera capa se desarrolla usando el framework Angular con el patrón de arquitectura Modelo-Vista-Controlador, mientras que para la segunda capa se utiliza el framework Slim PHP que permite realizar las peticiones a los diferentes servicios que se comunican con el Servidor. Este último recibe información de manera manual mediante la arquitectura REST, permitiendo la digitalización de tiendas y alimentos para todos los usuarios de la aplicación.

Finalmente, se obtiene un producto de software para computadoras, que contiene todas las funcionalidades propuestas y definidas para cumplir su propósito, de modo que puede ser usado por cualquier persona que tenga acceso a una computadora e Internet, aunque va dirigido especialmente a los dueños de mascotas, por el momento, de perros y gatos. Así, se da cumplimiento a todos los objetivos del proyecto.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Abstract

This project is presented to accomplish the requirements demanded by the University of Bío-Bío, certification process for the career of Ingeniería Civil en Informática.

The project entitled "Web system to compare pet food (dogs and cats)" aims to streamline the search and comparison of pet food, through the development of a Web application that allows, mainly, pet owners to access an online platform from a computer. It must offer the main features that most websites have, stories such as allowing the registration of new users, logging in and showing information on the different foods that are published along with the stores that offer them.

In terms of development, System consists of two layers, the front-end layer and the back-end layer. The first layer is developed using the Angular framework with the architecture model Model-View-Controller, while for the second layer the Slim PHP framework is used, which allows requests to be made to the different services that communicate with the Server. Server receives information manually through the REST architecture, allowing the digitalization of stores and food for all users of the application.

Finally, a computer software product is obtained, which contains all the proposed functionalities and definitions to fulfill its purpose, it can be used for anyone who has access to a computer and the Internet, although it has been specially directed to the owners of pets, for the moment, of dogs and cats. Thus, the objectives of the project are met.



Índice General

1. Introducción	10
2. Descripción del problema	12
2.1. Descripción de la organización.....	12
2.2. Ámbito y descripción del problema.....	12
3. Definición del proyecto.....	14
3.1. Objetivos del proyecto	14
3.2. Ambiente de Ingeniería de Software.....	14
3.3. Justificación del proyecto	17
4. Descripción de los requisitos específicos.....	19
4.1. Requisitos Funcionales	19
4.2. Requisitos No Funcionales	21
5. Estudio de Factibilidad del Proyecto.....	22
5.1 Factibilidad Técnica.....	22
5.2. Factibilidad Operativa.....	23
5.3 Factibilidad Económica	24
5.4 Conclusión del Estudio de Factibilidad	28
6. Modelamiento del sistema: Análisis de solución	29
6.1. Proceso de negocio con BPMN	29
6.2. Modelo basado en escenarios: Casos de Uso.....	31
6.3. Modelo de datos: Entidad Relación.....	35
7. Modelamiento del sistema: Diseño de solución.....	38
7.1. Modelo relacional	38
7.2. Diseño físico de la base de datos	39
7.3. Diseño orientado al consumo de servicios.....	39
7.4. Arquitectura de software.....	54
8. Pruebas	59
8.1. Especificación de las pruebas	59
8.2. Elementos para pruebas de sistema.....	61
8.3. Responsables de las pruebas	62



Sistema Web para comparar alimentos de mascotas (perros y gatos)

8.4. Planificación de las pruebas.....	62
8.5. Detalle de las pruebas	63
8.6. Conclusiones de pruebas efectuadas.....	64
9. Conclusiones	66
Referencias Bibliográficas.....	68
Anexos	69
Anexo 1. Encuesta para conocer el interés del público para el desarrollo del sistema.....	70
Anexo 2. Especificación de los Casos de uso	71
Anexo 3. Especificación de las pruebas del sistema realizadas.....	83
Anexo 4. Pauta de evaluación de la prueba de usabilidad	90



Índice de Tablas

Tabla 1. Resultados encuesta sobre el interés del público.....	18
Tabla 2. Requisitos funcionales del sistema.....	20
Tabla 3. Requisitos No funcionales del sistema.....	21
Tabla 4. Detalle de inversión en diseño y programación.....	24
Tabla 5. Detalle de costo anual en mano de obra.....	25
Tabla 6. Detalle de inversión fija para el año 0.....	26
Tabla 7. Detalle de costo anual de operación.....	26
Tabla 8. Detalle del flujo de caja.....	28
Tabla 9. Asociación entre casos de uso y requisitos funcionales.....	34
Tabla 10. Recursos y servicios de la aplicación.....	53
Tabla 11. Prueba de múltiple concurrencia de usuarios.....	59
Tabla 12. Prueba de sistema.....	60
Tabla 13. Prueba de usabilidad.....	61
Tabla 14. Responsables de las pruebas.....	62
Tabla 15. Resultados prueba de usabilidad.....	64
Tabla 16. Resultados del tiempo límite para realizar la prueba de usabilidad.....	64
Tabla 17. Caso de uso - Iniciar sesión.....	71
Tabla 18. Caso de uso - Validar sesión.....	72
Tabla 19. Caso de uso - Cerrar sesión.....	72
Tabla 20. Caso de uso - Registrarse.....	72
Tabla 21. Caso de uso - Agregar alimento.....	73
Tabla 22. Caso de uso - Modificar alimento.....	74
Tabla 23. Caso de uso - Eliminar alimento.....	74
Tabla 24. Caso de uso - Agregar tienda.....	75
Tabla 25. Caso de uso - Modificar tienda.....	76
Tabla 26. Caso de uso: Eliminar tienda.....	76
Tabla 27. Caso de uso - Revisar opiniones.....	76
Tabla 28. Caso de uso - Eliminar opinión inadecuada.....	77
Tabla 29. Caso de uso - Ver catálogo.....	77



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Tabla 30. Caso de uso – Buscar alimentos.	78
Tabla 31. Caso de uso - Agregar opinión.	79
Tabla 32. Caso de uso - Eliminar una opinión.	80
Tabla 33. Caso de uso - Crear cotización.	80
Tabla 34. Caso de uso - Modificar cotización.	81
Tabla 35. Caso de uso - Eliminar cotización.	81
Tabla 36. Caso de uso - Agregar alimento a cotización.	82
Tabla 37. Caso de uso - Eliminar alimento de cotización.	82
Tabla 38. Prueba de sistema - Crear una tienda.	83
Tabla 39. Prueba de sistema - Modificar una tienda.	84
Tabla 40. Prueba de sistema - Crear un alimento.	85
Tabla 41. Prueba de sistema - Modificar un alimento.	86
Tabla 42. Prueba de sistema - Registrarse.	87
Tabla 43. Prueba de sistema. Iniciar sesión.	88
Tabla 44. Prueba de sistema - Crear una opinión.	89
Tabla 45. Tareas a realizar para la prueba de usabilidad.	90
Tabla 46. Tiempos límite para realizar prueba de usabilidad.	91



Índice de Figuras

Figura 1. Diagrama del proceso de negocio utilizando BPMN.....	30
Figura 2. Diagrama de casos de uso.	33
Figura 3. Modelo entidad relación.....	37
Figura 4. Diseño físico de la base de datos.....	39
Figura 5. Interfaz principal de la aplicación.	41
Figura 6. Formulario para el inicio de sesión.	42
Figura 7. Formulario para el registro de un nuevo usuario.	42
Figura 8. Menú de navegación del usuario Administrador	43
Figura 9. Interfaz de gestión de tiendas.....	43
Figura 10. Interfaz de gestión de alimentos.....	44
Figura 11. Interfaz de revisión de opiniones.	45
Figura 12. Interfaz de cotizaciones del usuario registrado.	46
Figura 13. Elementos que permiten la búsqueda de alimentos.....	46
Figura 14. Arquitectura MVC de la aplicación.	58



1. Introducción

En nuestro país cada año el acceso a la Internet va aumentando, es así como la utilización de esta puede ser usada para diversas actividades, por lo que cada vez es mayor la cantidad de sitios web, en los cuales tanto organizaciones como personas dan a conocer aquella información que consideran más relevante, y así mantener a sus usuarios actualizados de los productos y/o servicios que ofrecen.

En paralelo, el negocio en torno a las mascotas crece día a día gracias a los emprendedores que buscan nuevos nichos para comercializar sus productos. Y, en este marco, sin duda uno de los temas que se ha vuelto interesante tanto en Chile como en el resto del mundo es el de los alimentos. Es que cada vez está más claro que un buen pellet les entrega a las mascotas todos los ingredientes necesarios para una dieta sana y balanceada, lo que en definitiva incide en que tengan una buena calidad de vida y una vida más larga.

Este proyecto busca desarrollar un sistema web que permita comparar alimentos para mascotas, en específico de perros y gatos, e incorpore las principales funcionalidades que poseen la mayoría de sitios web, tales como registro de nuevo usuario, inicio de sesión, búsqueda de productos y sus precios en diferentes tiendas, junto con funciones de gestión de tiendas y de alimentos para el perfil de administración.

En este documento se encuentra descrito el proceso realizado para obtener el producto propuesto. El siguiente capítulo presenta el contexto en el que se desarrolla este proyecto, el problema que se pretende resolver y de qué forma se debe solucionar. Posteriormente, el tercer capítulo estipula la definición del proyecto, en específico, los objetivos a seguir y el ambiente de ingeniería junto con las herramientas de desarrollo de software necesarias en el proyecto. El cuarto capítulo especifica los requerimientos de software propuestos para la construcción de la aplicación. Luego, el quinto capítulo corresponde al estudio de factibilidad que determina qué tan viable es la realización del sistema, en términos técnicos, operativos y económicos. El sexto capítulo analiza el sistema a desarrollar, usando una serie de modelos que permiten abstraer aquellos elementos propios de la realidad que resultan relevantes de identificar para llevar a cabo un análisis completo del problema para, posteriormente, dar pie a la etapa de diseño. El séptimo capítulo, por su parte, corresponde al diseño del software donde se construye la base de datos y se define el sistema como un consumidor de servicios,



Sistema Web para comparar alimentos de mascotas (perros y gatos)

incluyendo la especificación de los diseños de arquitectura (MVC y SOA). Posteriormente, el octavo capítulo corresponde a las pruebas realizadas en el sistema, que permiten verificar la calidad del mismo. Finalmente, el noveno capítulo presenta las conclusiones de este proyecto. Aparte, se dan a conocer las referencias bibliográficas usadas y los anexos que contienen información crucial sobre la cual fue elaborado este documento.



2. Descripción del problema

2.1. Descripción de la organización

La presente propuesta no se orienta a una organización en específico, sino que está orientada a un sector específico de individuos, el cual está constituido por aquellas personas que son propietarias de mascotas, particularmente perros y gatos, por lo cual la descripción de la Organización no tiene mayor relevancia en el proyecto.

2.2. Ámbito y descripción del problema

En esta sección, se contextualiza y describe el problema que aborda este proyecto, además del resultado que se espera obtener.

2.2.1. Descripción de la situación actual

Actualmente, son escasos y poco conocidos los sitios web que ofrecen productos alimenticios para mascotas que permitan a los dueños de estas decidir dónde conviene comprar. Por esta razón, contar con una plataforma integrada de búsqueda y comparación de productos alimenticios que permita encontrar alimentos para sus mascotas por un bajo costo se ve como un problema que podría encontrar una solución innovadora con apoyo de las Tecnologías de Información.

Entonces, se desea resolver este problema mediante el desarrollo de una aplicación que permita a los dueños de mascotas visitar una plataforma en línea, con el fin de poder agilizar los procesos de búsqueda, comparación y compra al que se enfrentan hoy en día. Esta agilización responde a problemas inherentes dentro de estos procesos para la toma informada de decisiones.

Lo anterior busca que los usuarios dueños de mascotas tengan mayor claridad acerca de los diferentes precios que puede llegar a tener un mismo alimento en las distintas tiendas.



2.2.2. Descripción del sistema a desarrollar

Se necesita un sistema que permita a los dueños de mascotas ingresar a una plataforma en línea desde sus computadoras. Esta aplicación debe contener diversas funcionalidades, tales como permitir el registro de usuarios, el inicio de sesión y mostrar los alimentos y las tiendas que los ofrecen. También se considera permitir a los usuarios dejar su opinión sobre el alimento que compran en un futuro y sobre la tienda donde lo hayan comprado.

La aplicación debe operar con los datos provistos en una base de datos propia cuya información se almacena a medida que se adhieren nuevas tiendas y alimentos al sistema.

En específico, la aplicación considera tres tipos de usuarios: Administrador, Usuario registrado y Usuario no registrado, quienes accederán constantemente. Para cada uno de estos tipos se hace uso de interfaces distintas, las cuales permiten realizar diversas tareas correspondientes a cada tipo de usuario.

En particular, el usuario Administrador se encarga principalmente de la gestión de tiendas, del catálogo de alimentos y de eliminar aquellas opiniones que puedan contener comentarios inapropiados de los usuarios registrados.

En el caso del usuario no registrado, este solo puede visitar el sitio para buscar y comparar alimentos para su mascota (el sistema le permite registrarse para convertirse en usuario registrado). Por otra parte, el usuario registrado puede dejar su opinión sobre el alimento y la tienda con que tenga experiencia de compra y realizar cotizaciones, las cuales en este proyecto se definen como un registro que recibe un nombre e incluye la información de comparación de los alimentos seleccionados, en específico, el precio que posee en la tienda seleccionada en un momento dado. También el usuario registrado puede realizar las mismas tareas que el usuario no registrado (excepto registrarse).

Adicionalmente, la aplicación contempla un buscador y despliega por pantalla un catálogo de alimentos con su imagen, nombre, marca y los distintos precios que poseen en las diferentes tiendas que los ofrezcan.

Por lo expuesto anteriormente, el objetivo de esta aplicación es dar a conocer a los usuarios en qué tiendas se encuentra a la venta el alimento que están buscando para su mascota y los diferentes precios que este puede llegar a tener, siendo esta una de las razones por la cual el sitio será desarrollado.



3. Definición del proyecto

3.1. Objetivos del proyecto

A continuación, se presenta el objetivo general y los objetivos específicos del proyecto, que deben ser cumplidos al término de este.

3.1.1. Objetivo General

Desarrollar una aplicación web para computadoras, que permita almacenar y desplegar información del catálogo de alimentos para mascotas de diferentes tiendas, e incorpore funcionalidades básicas que poseen la mayoría de sitios web, tales como registro e inicio de sesión, búsqueda y selección de productos, y que, además, incorpore datos útiles de los puntos de venta, permitiendo un acceso rápido a la información.

3.1.2. Objetivos específicos

El presente proyecto cuenta con tres objetivos específicos, los cuales se exponen a continuación:

- Determinar requerimientos para el desarrollo de un Sistema Web.
- Desarrollar interfaces atractivas, fáciles de mantener y operar, que permitan que el Sistema sea lo más amigable posible con los usuarios.
- Desarrollar un Sistema que permita la presentación y la actualización de información relacionada con la gestión de alimentos y puntos de venta.

3.2. Ambiente de Ingeniería de Software

A continuación, se indican las herramientas del ámbito del software que tienen relación con el desarrollo de este proyecto.

3.2.1. Herramientas de desarrollo del software (en orden alfabético):

- Apache: servidor web HTTP de código abierto, que implementa el protocolo HTTP/1.1.
- Composer: manejador de dependencias, paquetes y librerías de PHP.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

- Cygwin64 Terminal: consola de comandos (análoga a la de Windows).
- GitHub Desktop: gestor de repositorios de versiones para escritorio.
- Google Chrome: navegador.
- Mozilla Firefox: navegador.
- MySQL: Gestor de base de datos.
- Node.js: entorno de ejecución (capa de servidor).
- phpMyAdmin: Gestor de MySQL.
- Postman: API que permite el envío de peticiones HTTP REST (funcionalidad del Back-End) sin necesidad de desarrollar un cliente (Front-End). También puede desplegar la información que es enviada a la base de datos, por ejemplo, en formato JSON, brindando mayor claridad al desarrollador, para ver el código de una respuesta tras una petición, entre otras funciones.
- Sublime Text 3: editor de texto para escribir el código de la versión Web (no se usa IDE).
- Wamp Server: sistema de infraestructura (servidor web y gestor de base de datos), análogo a Xampp.
- Windows 8: sistema operativo.

3.2.2. Ambiente de desarrollo del software (en orden alfabético):

- Angular: framework para desarrollo de la capa Front-End [1], es utilizado para la creación y la mantención de aplicaciones Web de una sola página (SPA: single-page application) [12].
- REST (Transferencia de Estado Representacional): estilo de arquitectura cliente-servidor para el diseño de aplicaciones en red que, mediante el protocolo HTTP, es utilizado en la comunicación de datos en las aplicaciones Web.
- CSS: lenguaje de diseño gráfico para las hojas de estilo, complemento necesario para la capa Front.
- HTML: lenguaje de marcas de hipertexto, la base de la capa Front.
- PHP: lenguaje de programación para el desarrollo de la capa Back-End e interacción con la Base de Datos.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

- Slim Framework: micro framework para PHP que permite la escritura de aplicaciones Web y APIs [5].
- SPA (single-page application - aplicación de página única): es un tipo de aplicación Web donde todas las vistas se muestran en la misma página, es decir, todos los códigos HTML, CSS y JavaScript se cargan una sola vez, sin tener que recargar el navegador.
- TypeScript: lenguaje de programación para el desarrollo de modelos, componentes y servicios [3].

3.2.3. Relación de los elementos del ambiente de Ingeniería de SW

3.2.3.1. Preparación para el desarrollo de la capa Front-End

El equipo computacional posee Windows 8 como sistema operativo principal y cuenta con todas las herramientas software instaladas y configuradas para el desarrollo del sistema. El proceso de preparación y desarrollo se describe a continuación:

En primer lugar, se instala la última versión de NodeJS que es un entorno en tiempo de ejecución multiplataforma [6], para la capa del servidor basado en el motor V8 de Google, creado para Google Chrome, este último es el navegador de prueba para desplegar en línea el sistema.

Luego se instala Cygwin64 Terminal, que es una versión similar a la consola CMD que incorpora Windows para ejecutar líneas de comandos. Una vez abierto este programa, se instala el framework Angular CLI en su versión 5, que utiliza el lenguaje de programación TypeScript, para generar un proyecto base (componentes y servicios) y comenzar el desarrollo [4].

Se utiliza Sublime Text para el desarrollo del código (escritura/modificación), el cual es un editor de texto [7], en alternativa a un IDE como Netbeans, Eclipse u otro.

Además, se hace necesario instalar la última versión de Wamp Server (análogo a Xampp), el cual es un sistema de infraestructura de Internet, que permite subir páginas HTML a Internet (con hojas de estilos CSS y utilizando Bootstrap para el diseño web [8]), además de poder gestionar datos en ellas, y provee las siguientes herramientas: Apache, como



Sistema Web para comparar alimentos de mascotas (perros y gatos)

servidor web; MySQL, como gestor de BDs [11]; y PHP, como lenguaje de programación para el desarrollo del back-end. En el diseño de software, el back-end es la capa de acceso a datos, es decir, es la parte que procesa la entrada desde el front-end e interactúa con las BDs, y el front-end es la parte del software que interactúa con los usuarios. A medida que se escriban los métodos del back-end, se comprueba su funcionalidad con la herramienta Postman, la cual es una API que permite el envío (y prueba) de peticiones HTTP a una API REST sin necesidad de desarrollar un front-end [10].

Para complementar el párrafo anterior, se utiliza phpMyAdmin como herramienta para la administración de la base de datos en MySQL.

Por último, cabe mencionar que, a medida que se avanza con el desarrollo, se utiliza el servicio Web GitHub, que es un gestor de repositorios que permite gestionar el control de versiones del proyecto con la herramienta GitHub Desktop.

3.3. Justificación del proyecto

El presente proyecto se justifica sobre la base de un acotado estudio realizado mediante la aplicación de un instrumento cuyo propósito es conocer el interés del público propietario de mascotas para el desarrollo de la aplicación.

En particular, el instrumento corresponde a una encuesta conformada por nueve preguntas documentada en el Anexo 1, que se aplica a un conjunto de seis personas dueñas de mascotas. De esta manera, es posible conocer la opinión, perspectiva e interés de este público para llevar a cabo el desarrollo del proyecto. Los resultados de la aplicación de la encuesta se muestran en la Tabla 1.

Nº	Pregunta	Respuestas		Observaciones
1	Actualmente, ¿utiliza el computador con conexión a Internet?	Si: 6	No: 0	El hecho de que lo “posean” permite continuar la encuesta.
2	¿Tiene perro o gato como mascota?	Si: 6	No: 0	No se especifica el tipo de animal, pues solo interesa conocer la tenencia de una de estas especies.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

3	¿Qué tipo de alimento consume su mascota?	Comida casera: 1	Pellet: 5	Una persona no compra pellet y las marcas son reconocidas en la actualidad.
4	¿En dónde compra generalmente el alimento para su mascota?	Compra presencial: 4	Compra digital: 1	Una persona no realiza compra. Solo responden 5 personas.
5	¿Cómo se enteró de que en ese lugar estaba el producto para su mascota?	Contacto, presencial o medios: 4	A través de Internet: 1	Una persona no realiza compra. En general, cotizan presencialmente en el centro de la ciudad. La compra por Internet es en MercadoLibre.
6	Actualmente, ¿Se siente cómodo/a al comprar dicho alimento en ese lugar?	Si: 5	No: 0	Una persona no responde.
7	¿Estaría interesado/a en contar con una aplicación web que le permitiera buscar el alimento para su mascota?	Si: 5	No: 1	La mayoría de los encuestados manifiesta interés.
8	¿Le gustaría que esa aplicación le permitiera comparar y cotizar alimentos de diferentes marcas y precios en los distintos puntos de venta?	Si: 5	No: 1	Una persona desea conocer la ubicación de la tienda en el mapa (Esta idea se descarta para su desarrollo).
9	Pregunta dedicada a recibir comentarios	La mayoría de los encuestados, realizar una búsqueda y comparación de alimentos en forma presencial previa a la compra.		

Tabla 1. Resultados encuesta sobre el interés del público.

En virtud de los resultados obtenidos en la Tabla 1, se puede mencionar que las personas responden a favor del desarrollo de una aplicación que les permita comparar y cotizar alimentos (pellet) para mascotas. Por esta razón, se justifica el desarrollo del proyecto en pos de lograr los objetivos expuestos al comienzo del presente capítulo.



4. Descripción de los requisitos específicos

4.1.Requisitos Funcionales

A continuación, la Tabla 2 presenta los requisitos funcionales del sistema. En orden correlativo, se presentan los requisitos que dependen de la autenticación de un usuario (RF_01 hasta RF_03), del registro de un nuevo usuario (RF_04), luego los requisitos que son propios del Administrador del sistema (RF_05 hasta RF_11), después se encuentran los requisitos funcionales que están disponibles para los usuarios (usuarios no registrados RF_12 hasta RF_13, usuarios registrados RF12 hasta RF_17).

ID	Nombre	Descripción
RF_01	Iniciar sesión	El sistema debe mostrar el formulario de inicio de sesión al usuario que desea iniciar sesión en el sistema.
RF_02	Validar sesión	El sistema debe comparar las credenciales ingresadas, por el usuario con las registradas en la base de datos, para permitir o denegar su acceso al sistema.
RF_03	Cerrar sesión	El sistema debe terminar la sesión del usuario, redirigiéndolo a la interfaz principal del sistema.
RF_04	Registrarse	El sistema debe mostrar el formulario de registro de un nuevo usuario que desee registrarse en el sistema.
RF_05	Agregar alimento	El sistema debe permitir al Administrador ingresar un nuevo alimento al catálogo de productos del sistema.
RF_06	Modificar alimento	El sistema debe permitir al Administrador modificar un alimento existente en el catálogo de productos del sistema.
RF_07	Eliminar alimento	El sistema debe permitir al Administrador eliminar un alimento existente en el catálogo de productos del sistema.
RF_08	Agregar tienda	El sistema debe permitir al Administrador ingresar una nueva tienda al registro de tiendas del sistema.
RF_09	Modificar tienda	El sistema debe permitir al Administrador modificar una tienda existente en el registro de tiendas del sistema.
RF_10	Eliminar tienda	El sistema debe permitir al Administrador eliminar una tienda existente en el registro de tiendas del sistema.
RF_11	Eliminar opinión inadecuada	El sistema debe permitir al Administrador eliminar una opinión existente en el registro de opiniones del sistema, que considere inadecuada o no aceptable en relación a las demás.



Sistema Web para comparar y cotizar alimentos de mascotas (perros y gatos)

RF_12	Mostrar catálogo	El sistema debe mostrar a los usuarios el catálogo de alimentos registrados por el administrador en RF_04.
RF_13	Buscador de alimentos	El sistema debe buscar alimentos en función de la cadena de caracteres que ingrese el usuario en el buscador, o las características de la mascota (tales como: especie y grupo etario).
RF_14	Agregar opinión	El sistema debe permitir al usuario registrado agregar una nueva opinión acerca de un alimento y su tienda. La opinión considera la evaluación (calificación) y un comentario, tanto para el producto adquirido como para la tienda proveedora. El comentario corresponde a una experiencia del usuario como comprador del alimento.
RF_15	Eliminar opinión	El sistema debe permitir al usuario registrado eliminar una opinión existente en el registro de opiniones del sistema.
RF_16	Crear cotización.	El sistema debe permitir al usuario registrado realizar una cotización (definida en 2.2.2.), ingresando un nombre.
RF_17	Modificar cotización	El sistema debe permitir al usuario registrado modificar el nombre de la cotización, la cual se encuentra en el registro de cotizaciones del sistema.
RF_18	Eliminar cotización	El sistema debe permitir al usuario registrado eliminar una cotización existente en su registro de cotizaciones.
RF_19	Agregar alimento a cotización	El sistema debe permitir al usuario registrado agregar, en la cotización seleccionada, uno o más alimentos con el precio de la tienda seleccionada.
RF_20	Eliminar alimento de cotización	El sistema debe permitir al usuario registrado eliminar, en la cotización seleccionada, uno o más alimentos que se encuentren asociados a la cotización seleccionada.

Tabla 2. Requisitos funcionales del sistema.



Sistema Web para comparar y cotizar alimentos de mascotas (perros y gatos)

4.2.Requisitos No Funcionales

A continuación, la Tabla 3 presenta los requisitos No funcionales del sistema.

ID	Nombre	Descripción
RNF_01	Usabilidad	El sistema debe ser intuitivo y fácil de usar para sus usuarios (Administrador y usuarios en general). Para medir este requisito, se determina la efectividad a partir del número de usuarios que realizan de forma correcta cada tarea y se determina la eficiencia definiendo un tiempo límite para completar cada tarea.
RNF_02	Escalabilidad	El sistema debe adaptarse ante el crecimiento del entorno sin perder calidad en sus funcionalidades.
RNF_03	Seguridad	El sistema debe proteger adecuadamente el acceso del Administrador a información protegida. Además, los datos de las cuentas del sistema deben ser accedidos sólo a través de un usuario autenticado mediante el nombre de usuario y su contraseña. En la base de datos se almacena una versión cifrada de la contraseña resultado de la aplicación de una función Hash con Salt usando Bcrypt [13].
RNF_04	Estabilidad	El sistema debe evitar los fallos debidos a fuentes externas y, si no puede evitarlos, debe poder recuperarse sin ocasionar inconsistencias en los datos.
RNF_05	Eficiencia (tiempo de respuesta)	El sistema debe tardar menos de 15 segundos en mostrar las vistas con los datos solicitados, de lo contrario, el sistema debe mostrar mensajes de error.

Tabla 3. Requisitos No funcionales del sistema.



5. Estudio de Factibilidad del Proyecto

El Estudio de Factibilidad nos permite determinar si el Proyecto es viable analizando distintos ámbitos como lo son el Técnico, Operativo y Económico. Los resultados de cada estudio permiten concluir la Factibilidad del Proyecto.

5.1 Factibilidad Técnica

5.1.1 Lenguaje y herramientas software a utilizar.

La persona encargada de generar la solución cuenta con los conocimientos técnicos necesarios para llevar a cabo el desarrollo del sistema Web. Siendo estos los que a continuación se mencionan:

- i) Utilización del sistema operativo Windows 8.
- ii) Uso de Typescript como lenguaje de programación, en el marco de la utilización del framework **Angular**, aprendido en el curso de Angular 7 [2].
- iii) Uso de **phpMyAdmin** como sistema de gestión de bases de datos (SGBD) relacional con el servidor **MySQL**.

En resumen, se cuenta con el acceso a la tecnología de información y software mencionados.

5.1.2 Hardware

La persona encargada de generar la solución TI dispone del equipo hardware necesario para llevar a cabo el desarrollo del sistema Web. Dicho equipo hardware se constituye por un notebook Lenovo IdeaPad Z400, el cual posee la capacidad mínima necesaria para soportar la ejecución de los procesos que implica el desarrollo y la posterior utilización del sistema.

5.1.3 Soporte de la tecnología de información

Como se menciona en el punto 5.1.1., el desarrollo de la aplicación Web se enmarca en el uso del framework Angular, el cual se utiliza para crear y mantener aplicaciones web de una sola página.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Angular viene en desarrollo desde hace pocos años, por lo cual se le considera una tecnología reciente. Es mantenido en gran parte por Google y, en menor medida, por desarrolladores de todas partes del mundo que reportan y ayudan a las mejoras del framework.

5.1.4 Grado de maduración del framework

Desde Septiembre/2016, Angular 2 es la versión en la que miles de desarrolladores realizan sus aplicaciones ya que, desde ese año, se le considera una versión estable y comienza a formarse una comunidad de desarrolladores y a madurar el ecosistema de librerías.

Actualmente cuenta con el respaldo de Google y, por medio del sitio oficial [1] y otros foros de programación en la Web se puede consultar y recibir orientación y ayuda específica de la comunidad [9].

5.1.5 Conclusión

Desde el punto de vista técnico es totalmente factible y favorable el desarrollo del sistema, dado que se dispone, tanto del conocimiento en el framework y herramientas de software como de los elementos de hardware a utilizar. En conclusión, se tiene presente la posibilidad de tener un desarrollo confiable de la aplicación, ya que no parece haber inconvenientes de ningún tipo con los elementos mencionados en este estudio.

5.2. Factibilidad Operativa

5.1.1 Personas a quienes va dirigido el sistema

El sistema en línea va dirigido a los propietarios y dueños de mascotas, quienes ya poseen, al menos, conocimiento básico en el uso de aplicaciones para computadoras, smartphones y/o tablets, esto a consecuencia de la rápida evolución y facilidad de uso de la tecnología en nuestros días.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Por lo anterior, se tiene presente que los futuros usuarios del sistema ya cuentan con el nivel mínimo requerido para hacer uso de las distintas funciones que incorpora la aplicación a desarrollar.

5.2.2 Conclusión

Desde el punto de vista operativo es totalmente factible y favorable el desarrollo del sistema, ya que no se presentan inconvenientes ni se esperan obstáculos al momento de incorporar y poner en marcha el sistema Web.

5.3 Factibilidad Económica

5.3.1 Inversión

La inversión necesaria para el desarrollo del proyecto se presenta a continuación.

- a) **Equipo:** De acuerdo a lo expuesto en el punto 5.1.2, no se requiere de la compra de un computador (de escritorio o notebook) para el desarrollo y gestión del sitio Web.
- b) **Diseño y Programación del sitio Web:** Se requiere desarrollo para el diseño del Sitio Web y los servicios que el sistema utiliza. En específico, se desea implementar la arquitectura REST de petición-respuesta para permitir el intercambio de información entre las interfaces (Front-end) y los servicios (Back-end). A continuación, el detalle del cálculo (1 día laboral equivale a 9 HH).

Sueldo Ingeniero Civil en Informática ¹	\$ 1.088.100
Valor HH (Sueldo/180) Ingeniero Civil en Informática	\$ 6.045
HH en Análisis y Diseño (5 días):	45 HH
HH en Construcción (10 días):	90 HH
HH en Implantación (5 días):	45 HH
Total de HH:	180 HH
Inversión en mano de obra (Valor HH x Total HH)	\$1.088.100

Tabla 4. Detalle de inversión en diseño y programación.

¹ Fuente: <http://conexioningenieros.com/wp-content/uploads/2017/12/Estudio-Nacional-de-Sueldos-de-Ingenieros-2017.pdf>



Sistema Web para comparar alimentos de mascotas (perros y gatos)

- c) **Software:** El software a utilizar, expuesto en el punto 5.1.1, se encuentra disponible para la descarga gratuita en las Web oficiales, por tanto, no es necesario adquirir algún tipo de software.

5.3.2 Costo anual de operación y puesta en marcha

a) Dominio en Internet y servidor/hosting para la base de datos

El requisito mínimo para tener un sitio web en línea es el de poseer el dominio con el nombre del sitio web. Dado que el núcleo del proyecto consiste en el desarrollo de una plataforma Web, esta necesita de un servidor que pueda albergar las bases de datos que en ella se manejan. Se pretende utilizar para la plataforma un único dominio del tipo “.cl”. La cotización del plan de 5GB en servidoreshosting.cl, arroja un precio anual de **\$11.900** (IVA inc.), cuyo primer pago se realiza en el año 0 –para financiar el primer año del servicio- y culmina en el año 4 para financiar el último año.

b) Sueldo para el encargado de la mantención y actualización de datos en el sistema

Se requiere de un profesional encargado de la mantención de la información en el sitio Web. Para esto se requiere del trabajo de un Analista programador (mismo valor de Hora Hombre), que presta servicios en forma Mensual. A continuación, el detalle del cálculo.

Valor HH de Ingeniero Civil en Informática	\$ 6.045
HH en Mantención y Actualización	10 HH
Total de HH:	10 HH
Costo mensual (\$) en mano de obra (Valor HH x Total HH)	\$60.450
Costo anual (\$) en mano de obra (costo mensual x 12)	\$725.400

Tabla 5. Detalle de costo anual en mano de obra.



5.3.3 Evaluación Económica

5.3.3.1 Inversión fija (año 0)

En el año 0 se financia el (*) Dominio en Internet para el primer año de operación.

Mano de obra	\$1.088.100
(*) Dominio en Internet (hosting)	\$11.900
Inversión fija (\$)	\$ 1.100.000

Tabla 6. Detalle de inversión fija para el año 0.

5.3.3.2 Costo anual de operación y puesta en marcha

En el último año (5°) no se realiza pago por concepto de (*) Dominio en Internet.

(*) Dominio en Internet	\$11.900
Mantenimiento anual	\$725.400
Costo anual operacional (\$)	\$737.300

Tabla 7. Detalle de costo anual de operación.

5.3.4 Beneficios Económicos (Ingresos)

El valor agregado que entrega el sistema es brindar mayor satisfacción a los dueños de mascotas, pero estos no constituyen una fuente de ingreso, ya que lo que se busca es aumentar el número de ventas de las diferentes tiendas de alimentos de mascotas, y hacerles publicidad.

Por lo explicado anteriormente, las ganancias se obtendrán por medio de un método similar al de marketing de afiliación. El marketing de afiliación consiste en que el Administrador, en calidad de afiliado de los productos, recomienda los productos del vendedor en el Sistema Web y, posiblemente, realiza envío de tráfico web a la página de ventas del vendedor a través de un enlace de afiliado. Si los usuarios redirigidos terminan comprando dicho producto, automáticamente generan una comisión para el Administrador (Este método es una idea teórica).

La variante difiere al método, anteriormente explicado, en que se indica a los usuarios visitantes del sitio Web que mencionen el nombre de la aplicación, si es que deciden realizar la compra del producto.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Por lo tanto, se prevé la negociación de la presencia, en este espacio digital, de seis tiendas en el sistema para el primer año, con un aumento de cuatro tiendas por cada año transcurrido.

Por parte de las tiendas, se estima que estas generan un ingreso promedio mensual de \$300.000, por ende, su ingreso anual es de \$3.600.000 (\$300.000 x 12) y que, por simplicidad, el ingreso mensual promedio de las tiendas no varía en función del tiempo.

Por nuestra parte, dado que las comisiones no son muy altas, se plantea un 2,5% de comisión para efectuar la estimación del ingreso anual gracias al uso de nuestro método de marketing (variante del método de afiliación). Por esta razón, nuestro ingreso anual que se recauda es de \$90.000 x nro de tiendas adheridas.

Ingreso mensual promedio (tiendas)	\$300.000
Ingreso anual promedio (cada tienda)	\$3.600.000
Ingreso anual por tienda (2,5% com.)	\$90.000
Ingreso anual (\$) primer año	\$540.000

Tabla 8. Detalle de ingreso anual de operación.

5.3.5 Egresos y Gastos

Entre gastos generales se puede considerar únicamente los desembolsos provenientes del pago del servicio de Internet, pero se presume que no existirá un aumento en el monto a pagar mensualmente por este concepto. Por otra parte, gastos adicionales a considerar, como arriendo de instalaciones, electricidad, gas y agua, entre otros, no son considerados ya que el funcionamiento del proyecto se desarrolla dentro de las dependencias del Administrador y no suponen aumento en el monto a pagar por estos conceptos.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

5.3.6 Flujo de Caja del proyecto y cálculo del Valor Actual Neto (VAN)

Items / Año	0	1	2	3	4	5
Ingreso anual		\$ 540.000	\$ 900.000	\$ 1.260.000	\$ 1.620.000	\$ 1.980.000
Costo anual		\$ 737.300	\$ 737.300	\$ 737.300	\$ 737.300	\$ 725.400
Total		-\$ 197.300	\$ 162.700	\$ 522.700	\$ 882.700	\$ 1.254.600

Inversión Nuevo Sistema						
Equipo comput.	\$ -					
Desarrollo SW	\$ 1.088.100					
Capital de trabajo	\$ 11.900					
Total inversión	\$ 1.100.000					

Flujo de efectivo anual		-\$ 197.300	\$ 162.700	\$ 522.700	\$ 882.700	\$ 1.254.600
Valor Presente (VA)		-\$ 176.161	\$ 129.703	\$ 372.048	\$ 560.972	\$ 711.894

VAN	\$ 498.456
------------	-------------------

Tabla 9. Detalle del flujo de caja.

5.3.7. Conclusión

Desde el punto de vista económico resulta conveniente desarrollar este proyecto, puesto que la evaluación económica considera recursos que existen actualmente en dominio del Administrador, y no se ve necesario realizar una gran inversión (monto indicado de color rojo), ya que se dispone de recursos humanos y tecnológicos para este proyecto. Además, el resultado del Valor Actual Neto (VAN) da cuenta de que el proyecto es completamente factible, ya que es mayor a 0, por lo tanto, el proyecto es rentable para el Administrador.

5.4 Conclusión del Estudio de Factibilidad

Queda en evidencia en los puntos anteriores, que los estudios de factibilidad técnica, económica y operativa resultan ser favorables para el proyecto lo que determina su viabilidad. El proyecto resulta ser una buena herramienta para el Administrador, además a un costo no mayor que es recuperable al cuarto año en el horizonte temporal.



6. Modelamiento del sistema: Análisis de solución

6.1. Proceso de negocio con BPMN

6.1.1. Descripción del principal proceso de negocio

En esta sección, se presenta el **proceso de negocio** utilizando **BPMN** (Business Process Model and Notation), la cual es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow).

En particular, se contempla la solución al problema de búsqueda y comparación de alimentos haciendo uso de la aplicación. También se identifica a las **entidades que interactúan** en el proceso, lo que permite entender claramente el **proceso de negocio**. La situación detallada en el proceso general comienza con el acceso del Administrador al Sistema y termina cuando éste actualiza el catálogo en línea, poniendo a disposición de los usuarios, los alimentos ofrecidos por las diferentes tiendas adheridas.

6.1.2. Breve explicación del modelo

En la Figura 1, se observa el transcurso del **proceso de gestión** de alimentos y tiendas, por parte del **Administrador**, que comienza con la petición de la información de los alimentos que se van a mostrar, a las tiendas que venden alimentos para mascotas. Por otra parte, el Administrador puede recibir y revisar opiniones recientes en caso de que se registren nuevas entradas.

La **Tienda** recibe la petición y responde al Administrador con la información solicitada. Luego que el Administrador la recibe, este la ingresa en el Sistema. Una vez que se confirma el almacenamiento de la información, se actualiza el catálogo de alimentos quedando a disposición de todos los usuarios. Por otra parte, el **Usuario registrado** ingresa a la aplicación. Luego, realiza la búsqueda del alimento de interés. Posteriormente, compara los precios en las diferentes tiendas para después cotizar aquellos alimentos que eventualmente puede llegar a comprar, o bien para guardar un registro del precio del alimento que alguna vez tuvo una tienda en un momento específico. También puede ver información más detallada del alimento y dejar su opinión del alimento y de la tienda para conocimiento del Administrador y el resto de usuarios. De esta manera, quienes accedan a la aplicación



Sistema Web para comparar alimentos de mascotas (perros y gatos)

pueden tener una referencia de las características del alimento y de qué tan buen (o mal) proveedor es la tienda, conociendo la opinión de la comunidad.

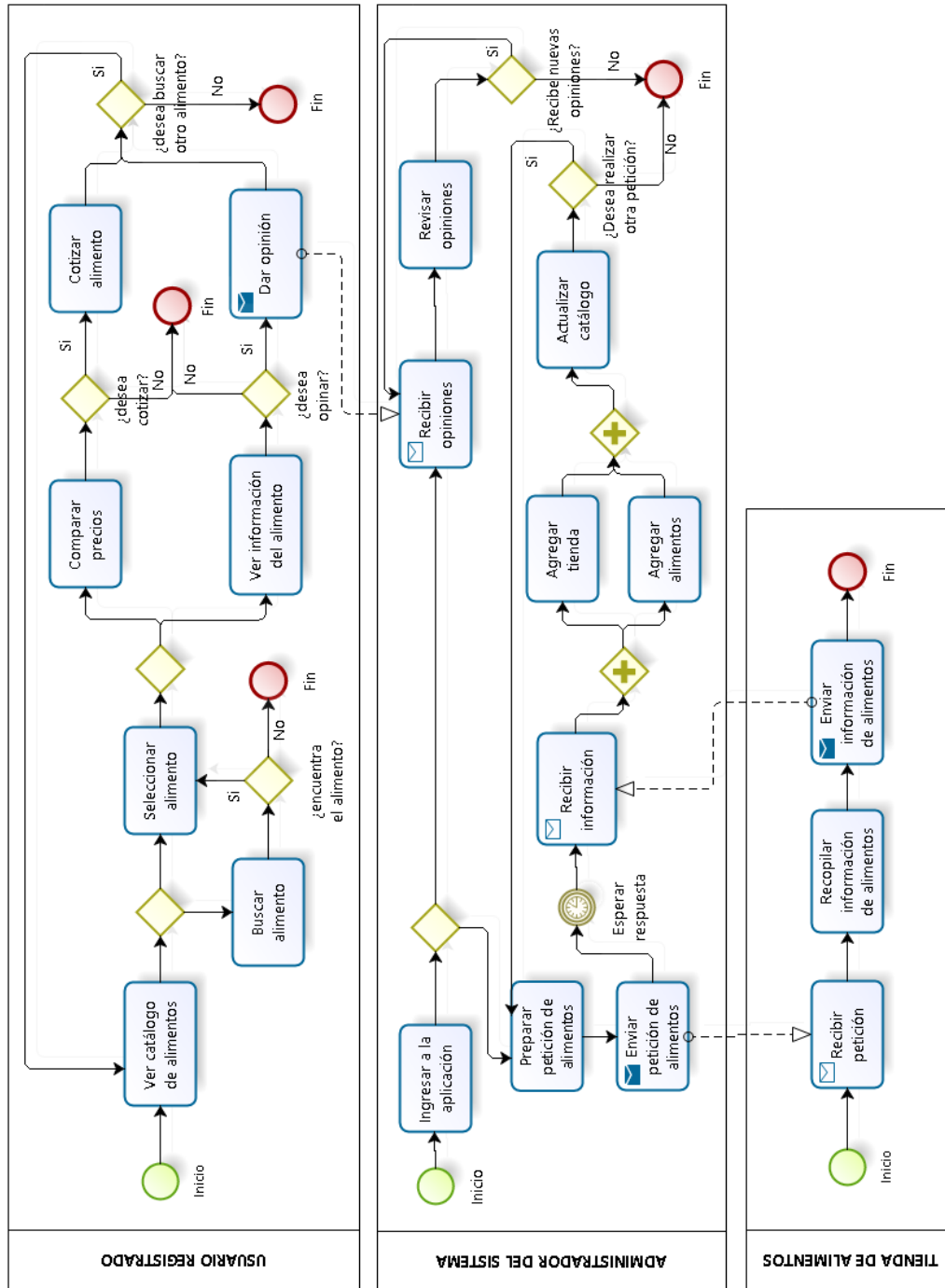


Figura 1. Diagrama del proceso de negocio utilizando BPMN.



6.2. Modelo basado en escenarios: Casos de Uso

El modelo basado en escenarios se utiliza para modelar la interacción entre el sistema y el usuario. Este modelo de análisis utiliza el Lenguaje Unificado de Modelado, más conocido como UML, y comienza con la creación de escenarios en la forma de diagrama de Casos de Uso.

A continuación, se dan a conocer los actores que interactúan directamente con el sistema, una breve explicación del diagrama de Casos de Uso, el diagrama en la Figura 2. Mientras que la especificación de estos se presenta en el Anexo 2.

6.2.1. Actores

1.1) Administrador:

Este actor es el encargado de gestionar la información del sistema, previo inicio de sesión. Además, tiene pleno conocimiento de las **tiendas** (locales de venta) que proveen los **alimentos** para mascotas que, posteriormente, podrán ser buscados por los Usuarios.

1.2) Usuario No registrado:

Este actor hace uso del sistema, no requiriendo del inicio de sesión para visitar el catálogo de alimentos, el cual una vez desplegado permite a este actor buscar productos y comparar precios. Además, este actor puede “ascender” en el sistema, registrándose para obtener una cuenta de usuario registrado (siguiente actor).

1.3) Usuario registrado:

Este actor hace uso del sistema, el cual además de buscar productos y comparar precios, puede (previo inicio de sesión) realizar/modificar cotizaciones y agregar opiniones a los diferentes alimentos y tiendas, esta última acción consiste en realizar un comentario y calificar tanto al producto como a la tienda.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

6.2.2. Breve explicación del diagrama

En el lado derecho del diagrama de la Figura 2, se muestra la interacción y el alcance que posee el perfil de Administrador con el Sistema. Este rol posee acceso a aquellas funcionalidades asociadas a la gestión (de tiendas y alimentos) y orden (supervisor de opiniones) del sistema.

Por otra parte, en el lado izquierdo de la Figura 2 se muestran las interacciones y el alcance que poseen los usuarios No registrados y registrados con el sistema. El primer rol posee un acceso muy limitado sólo con fines informativos, aunque puede “ascender” como se menciona en el punto 1.2) de la sección 6.2.1. Por otra parte, el segundo rol posee un acceso menos condicionado que el primer rol, ya que al estar registrado en el sistema tiene acceso a más funcionalidades tales como agregar/eliminar opiniones y realizar/modificar cotizaciones de alimentos.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

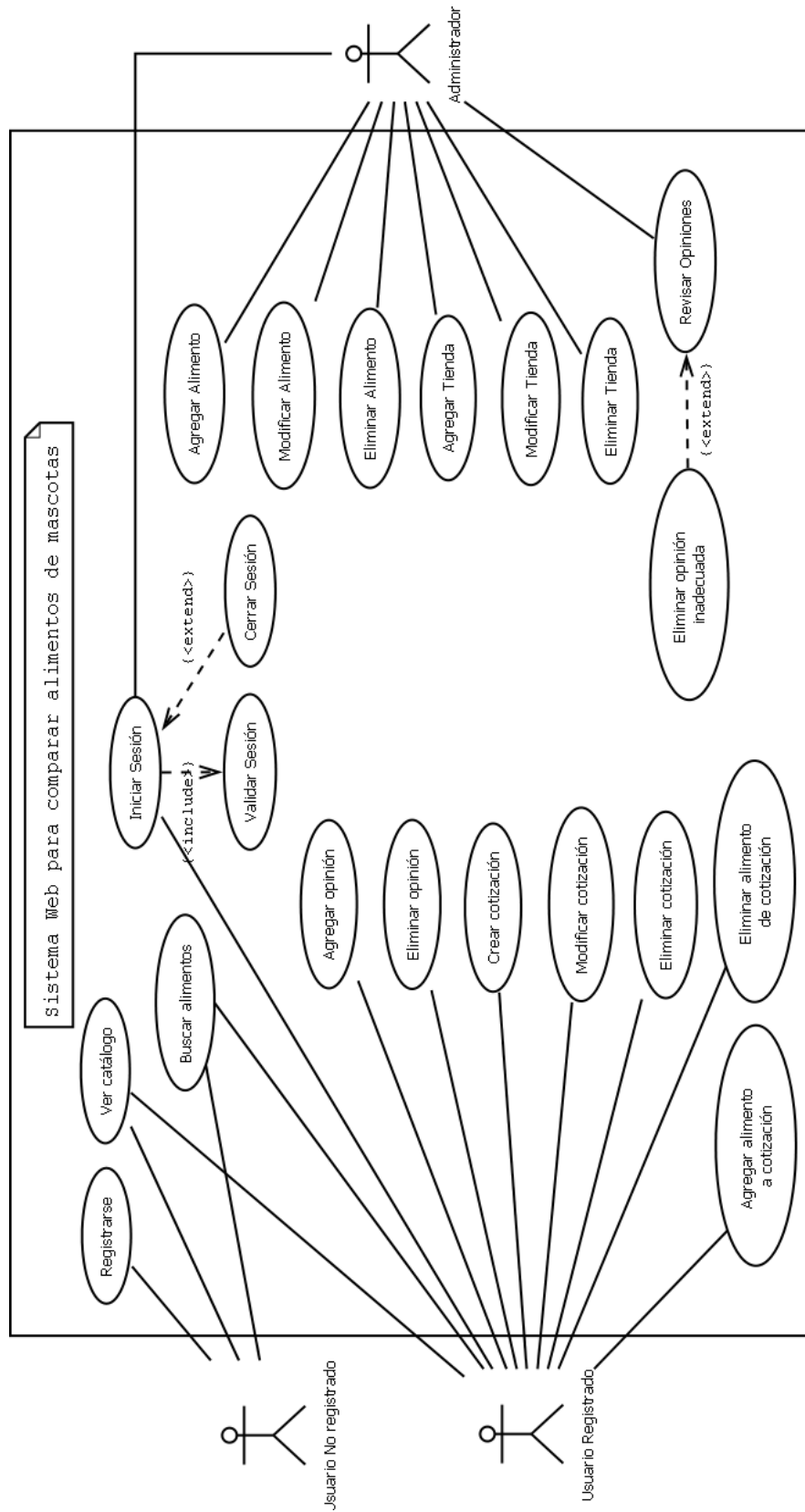


Figura 2. Diagrama de casos de uso.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

A cada Caso de Uso del diagrama se le asigna un ID (identificador) del tipo CDU_XX, donde XX corresponde a la numeración correlativa. Para mayor información, la especificación de los Casos de Uso se expone en el Anexo 2.

A continuación, en la Tabla 9 se dan a conocer las asociaciones entre cada Caso de Uso y los Requisitos Funcionales definidos en la Tabla 2.

ID	Nombre	RF asociado
CDU_01	Iniciar sesión	RF_01
CDU_02	Validar sesión	RF_02
CDU_03	Cerrar sesión	RF_03
CDU_04	Registrarse	RF_04
CDU_05	Agregar alimento	RF_05
CDU_06	Modificar alimento	RF_06
CDU_07	Eliminar alimento	RF_07
CDU_08	Agregar tienda	RF_08
CDU_09	Modificar tienda	RF_09
CDU_10	Eliminar tienda	RF_10
CDU_11	Revisar opiniones	RF_11
CDU_12	Eliminar opinión inadecuada	RF_11
CDU_13	Ver catálogo	RF_12
CDU_14	Buscar alimentos	RF_13
CDU_15	Agregar opinión	RF_14
CDU_16	Eliminar opinión	RF_15
CDU_17	Crear cotización	RF_16
CDU_18	Modificar cotización	RF_17
CDU_19	Eliminar cotización	RF_18
CDU_20	Agregar alimento a cotización	RF_19
CDU_21	Eliminar alimento de cotización	RF_20

Tabla 10. Asociación entre casos de uso y requisitos funcionales



6.3. Modelo de datos: Entidad Relación

En esta sección se presenta el modelo entidad-relación, que es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información con sus interrelaciones y propiedades (atributos). En particular, la Figura 3 cuenta con cinco entidades y cuatro relaciones.

A continuación, una breve descripción de cada entidad, y de cada relación:

6.3.1. Entidades

- a) **Cliente:** Esta entidad representa al Usuario (dueño de una mascota) que hace uso de la aplicación. Posee atributos de identificación personal (nombre, e-mail).
- b) **Mascota:** Esta entidad representa a la mascota que posee el Cliente. Sus atributos, con sus clave-valor, son los siguientes:
 - Especie → Valores: Perro o Gato,
 - Tamaño de la raza → Valores: Pequeña, Mediana o Grande
 - Edad (en años) la cual junto a la Especie permite determinar el Grupo etario (este último se muestra en la aplicación):
 - Especie Perro posee los Grupos etarios: Cachorro (0 a 1 año), Adulto (1 a 8 años), Senior (mayor a 8 años).
 - Especie Gato posee los Grupos etarios: Minino (0 a 1 año), Adulto (1 a 8 años), Senior (mayor a 8 años).
- c) **Alimento:** Esta entidad representa los productos que son ofrecidos por las tiendas de alimentos para mascotas, y que serán mostrados en la aplicación. Sus atributos son nombre, imagen, marca, precio y descripción.
- d) **Tienda:** Esta entidad representa al punto de venta que ofrece alimentos para mascotas. Sus atributos son nombre, ciudad, dirección, e-mail y teléfono.
- e) **Cotización:** Esta entidad representa la cotización que es un registro que contiene los alimentos seleccionados por el Cliente. Incluye el precio y las tiendas que los ofrecen. Sus atributos son un nombre que lo define el Cliente y la fecha en que se realiza.



6.3.2. Relaciones, cardinalidad y descripción.

- a) **Tener (1:1):** Un cliente tiene una mascota (por simplicidad).
Una mascota tiene un único dueño.
- b) **Realizar (1:N):** Un cliente realiza cero o muchas cotizaciones;
Una cotización es realizada por un único cliente.
- c) **Contener (N:N):** Una cotización contiene uno o muchos alimentos;
Un alimento es contenido en una o muchas cotizaciones.
- d) **Poseer (N:N):** Una tienda posee uno o muchos alimentos;
Un alimento es poseído por una o muchas tiendas.

Nota: En la aplicación, los alimentos son presentados en el catálogo considerando los siguientes atributos de la mascota: la especie (valores: Perro o Gato), el tamaño de la raza (valores: Pequeña, Mediana y Grande) y el grupo etario (valores en 6.3.1.b).



Sistema Web para comparar alimentos de mascotas (perros y gatos)

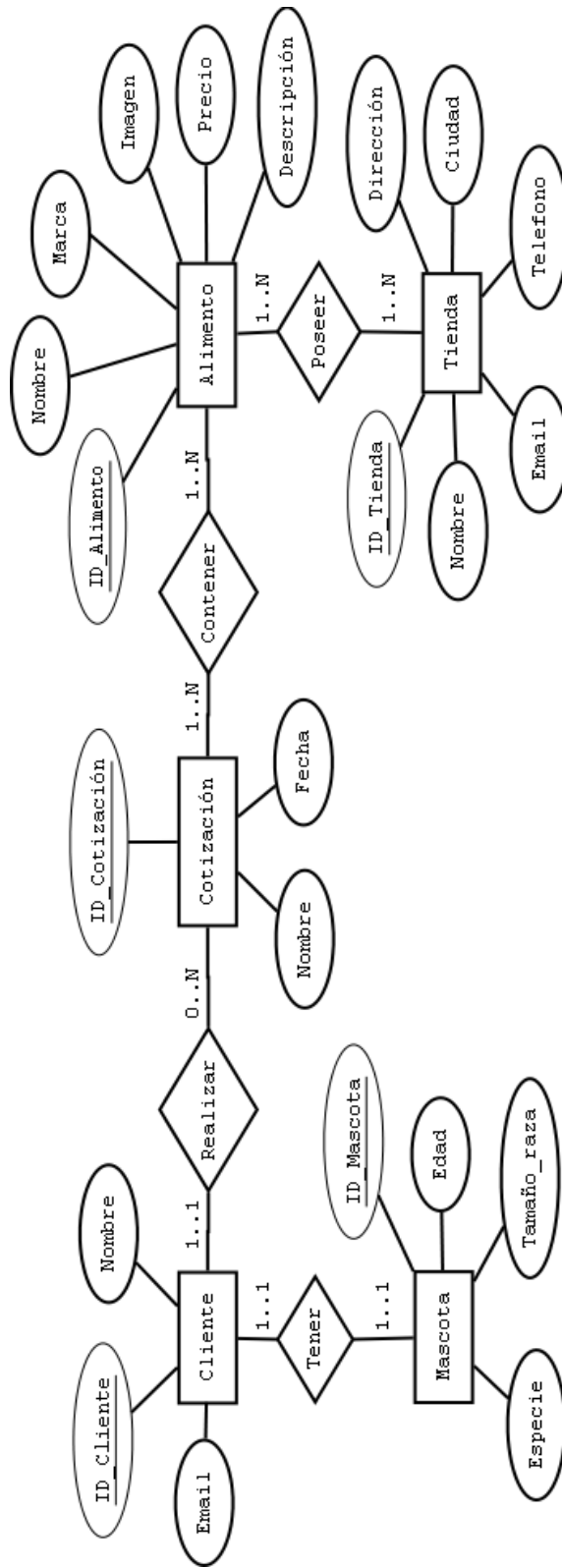


Figura 3. Modelo entidad relación.



7. Modelamiento del sistema: Diseño de solución

En este capítulo se presenta el diseño de la aplicación, enfocado en el desarrollo de la misma. Los puntos tratados a continuación corresponden a la elaboración del modelo relacional (continuación del modelo ER), una vista esquemática del diseño físico de la base de datos y el diseño orientado al consumo de servicios.

7.1. Modelo relacional

A continuación, se presentan las relaciones generadas a partir de las dependencias provenientes del modelo Entidad Relación como resultado de la cardinalidad entre entidades.

- a) Usuario (cliente) : ID_Usuario, Nombre, Email, Contraseña, Rol (perfil)
- b) Mascota : ID_Mascota, Especie, Edad, Tamano_raza
- c) Cotización : ID_Cotizacion, Nombre, Fecha, **FK_ID_Usuario**
- d) Tienda : ID_Tienda, Nombre, Ciudad, Direccion, Email, Telefono
- e) Alimento : ID_Alimento, Nombre, Marca, Animal, Grupo_etario, Tamano_raza, Kilos, Imagen, Precio, Descripción
- f) Cotizacion_Alimentos : **FK_ID_Cotizacion, FK_ID_Alimento, FK_ID_Tienda**
- g) Tienda_Alimentos : **FK_ID_Tienda, FK_ID_Alimento, Precio**

Cabe señalar que la relación **Cotización_Alimentos** corresponde al detalle de una cotización realizada por el Usuario resultado del registro de una comparación de precios, en esta relación el atributo FK_ID_Cotización se puede repetir, pero el FK_ID_Alimento no, permitiendo incluir más de un alimento en la cotización. Además, se añade el atributo FK_ID_Tienda para recuperar el precio que posee el alimento en la tienda seleccionada. Por otra parte, la relación **Tienda_Alimentos** corresponde al detalle del catálogo de alimentos que ofrece una Tienda, donde el atributo FK_ID_Tienda se puede repetir, pero el FK_ID_Alimento no. Además, se añade el atributo Precio ya que un mismo alimento se puede encontrar en más de una tienda, pero con diferentes precios.

Por último, estas dos relaciones resuelven el problema de la cardinalidad N:N mapeada en el modelo Entidad Relación.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

7.2. Diseño físico de la base de datos

El diseño físico permite construir la base de datos, y este proceso involucra lo que debe poseer cada instancia a partir de la información provista en las entidades. Por consiguiente, las entidades pasan a denominarse tablas, las instancias en filas (o tuplas) y los atributos en columnas.

A continuación, en la Figura 4 se muestra el diseño físico de la base de datos.



Figura 4. Diseño físico de la base de datos.

7.3. Diseño orientado al consumo de servicios

En esta sección se muestra la manera en que opera la aplicación, además de los recursos y servicios que deben ser consumidos para cumplir con los requisitos funcionales y casos de uso, asimilados al proceso del sistema.



7.3.1. Estructura de la aplicación

Cuando se ingresa localmente a la plataforma mediante la dirección <http://localhost:4200/>, se aprecia un menú de navegación, dos buscadores y el catálogo de alimentos (Figura 5). En el menú de navegación es posible iniciar sesión (Figura 6) o registrarse en el sistema (Figura 7). El inicio de sesión solicita la autenticación del usuario mediante E-mail y contraseña, mientras que el registro solicita el ingreso de un E-mail válido y una contraseña que luego debe ser escrita nuevamente para validar el registro.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

TRIPET

Catálogo
Tiendas
Mi Cuenta

Seleccione la especie

Seleccione el grupo etario

Buscar

Catálogo de alimentos

Productos por página

10

Ordenado por

Nombre Ascendente

Ordenado por

Nombre Ascendente

Nombre
Marca
Especie
Etapas
Precios

Filtrar por:

Alimento gato adulto Nutribalance Animal Planet 2,1 kg

Ver detalles

Animal Planet

Gato

Adulto

Seleccione una tienda

\$22,490 // Mayorista ALVI

Figura 5. Interfaz principal de la aplicación.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Formulario para iniciar sesión con correo electrónico. El formulario contiene el título "Iniciar sesión con su correo electrónico", un campo de texto para el "Correo electrónico", un campo de texto para la "Contraseña", y dos botones: "Iniciar sesión" (azul) y "Cancelar" (naranja).

Figura 6. Formulario para el inicio de sesión.

Formulario para registrarse con correo electrónico. El formulario contiene el título "Registrarse con su correo electrónico", un campo de texto para el "Correo electrónico", un campo de texto para la "Contraseña", un campo de texto para "Confirmar contraseña" con el texto "Vuelva a escribir su contraseña", y dos botones: "Registrarse" (azul) y "Cancelar" (naranja).

Figura 7. Formulario para el registro de un nuevo usuario.

Una vez que el usuario se ha autenticado, es decir, ha iniciado su sesión, se actualiza el menú de navegación de acuerdo al perfil del usuario donde:

- a) Si es el administrador (Figura 8) puede gestionar las tiendas adheridas (Figura 9) y el catálogo de alimentos (Figura 10), también puede revisar las opiniones (Figura 11) por los usuarios registrados, con el fin de eliminar comentarios inapropiados.
- b) Si es un usuario registrado puede realizar una cotización (RF_16 de la Tabla 2) que permita guardar alimentos resultado de un proceso de comparación de productos (Figura 12), en la cual se seleccionan los alimentos de interés y el precio (asociado a una tienda) que se desee añadir al registro.

Por otra parte, existen dos buscadores que permiten a los usuarios (no registrados y registrados) buscar alimentos en función del nombre del alimento (botón Buscar de color verde), o bien de la especie (perro o gato) acompañado del grupo etario del animal (botón Buscar de color azul), tal como se muestra en la Figura 13.



Sistema Web para comparar alimentos de mascotas (perros y gatos)



Figura 8. Menú de navegación del usuario Administrador

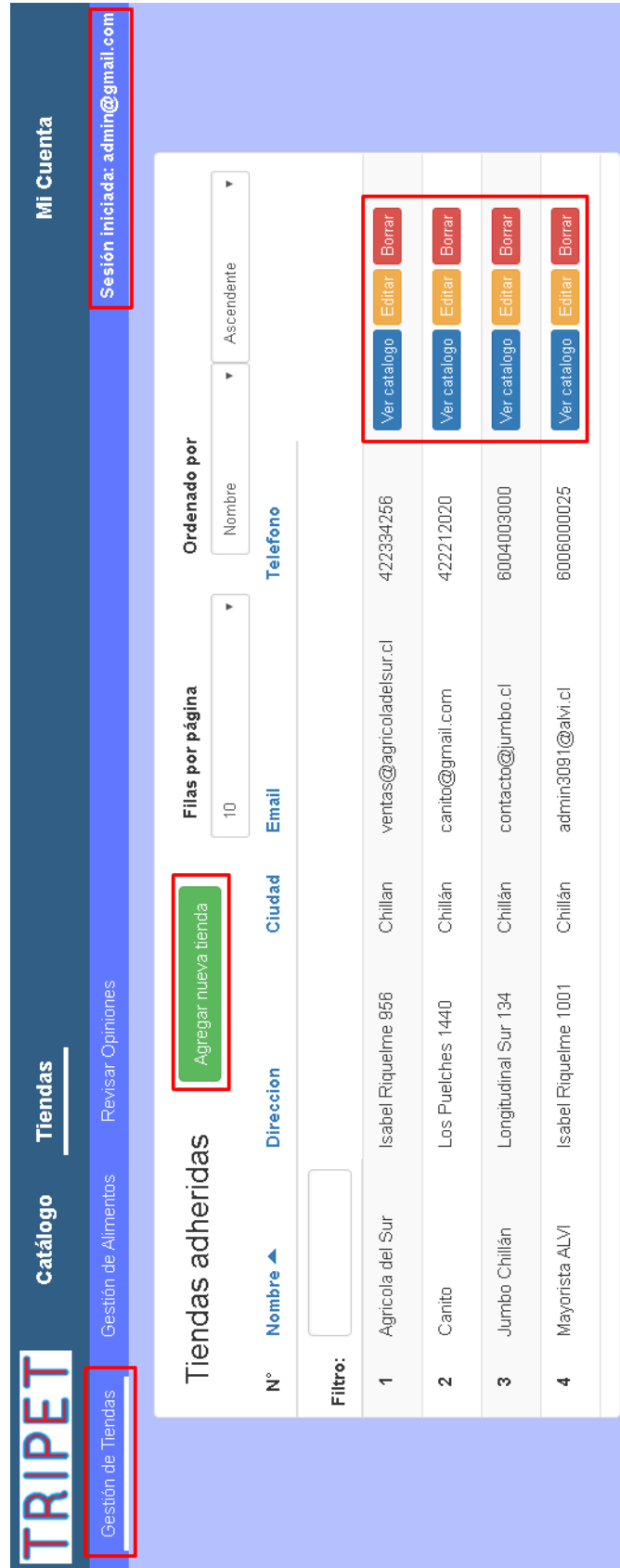


Figura 9. Interfaz de gestión de tiendas.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Sección iniciada: admin@gmail.com

Gestión de Tiendas Mis Catizaciones Revisar Opiniones

Alimentos registrados en el Sistema

Agregar nuevo alimento






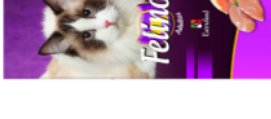
	Cat Chow Adulto 1K	Ver Editar Borrar
	Champion Adulto 1K	Ver Editar Borrar
	Master Cat 1K	Ver Editar Borrar
	Master Dog Cachorros 3K	Ver Editar Borrar
	Master Dog Senior 3K	Ver Editar Borrar
	Felino Mix 8K	Ver Editar Borrar

Figura 10. Interfaz de gestión de alimentos



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Gestión de Tiendas Gestión de Alimentos **Revisar Opiniones** Sesión Iniciada: admin@gmail.com

Opiniones registradas en el Sistema

Fecha	Usuario	Comentario alimento	Tienda	Comentario tienda	Acción
Dec 26, 2018	admin@gmail.com	Buen alimento	Agricola del Sur	Vendedores muy atentos, buena atención	Eliminar
Dec 26, 2018	juan@gmail.com	www	Agricola del Sur	eee	Eliminar
Dec 26, 2018	juan@gmail.com	Buen producto	Agricola del Sur	Buena atención, buenos vendedores	Eliminar
Dec 16, 2018	admin@gmail.com	Buen alimento	Agricola del Sur	Buena atención	Eliminar

Figura 11. Interfaz de revisión de opiniones.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

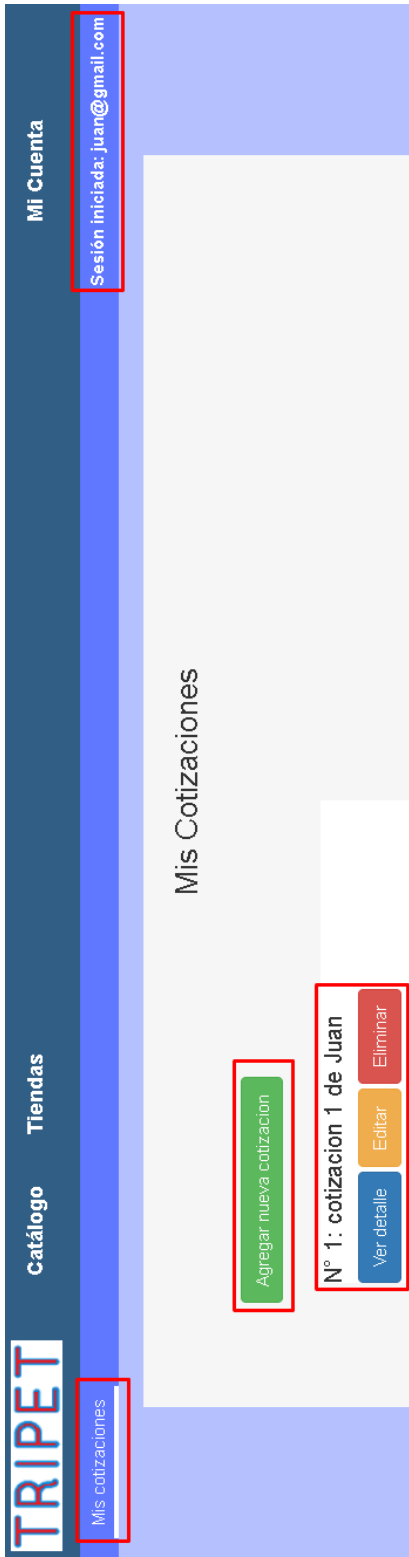


Figura 12. Interfaz de cotizaciones del usuario registrado.

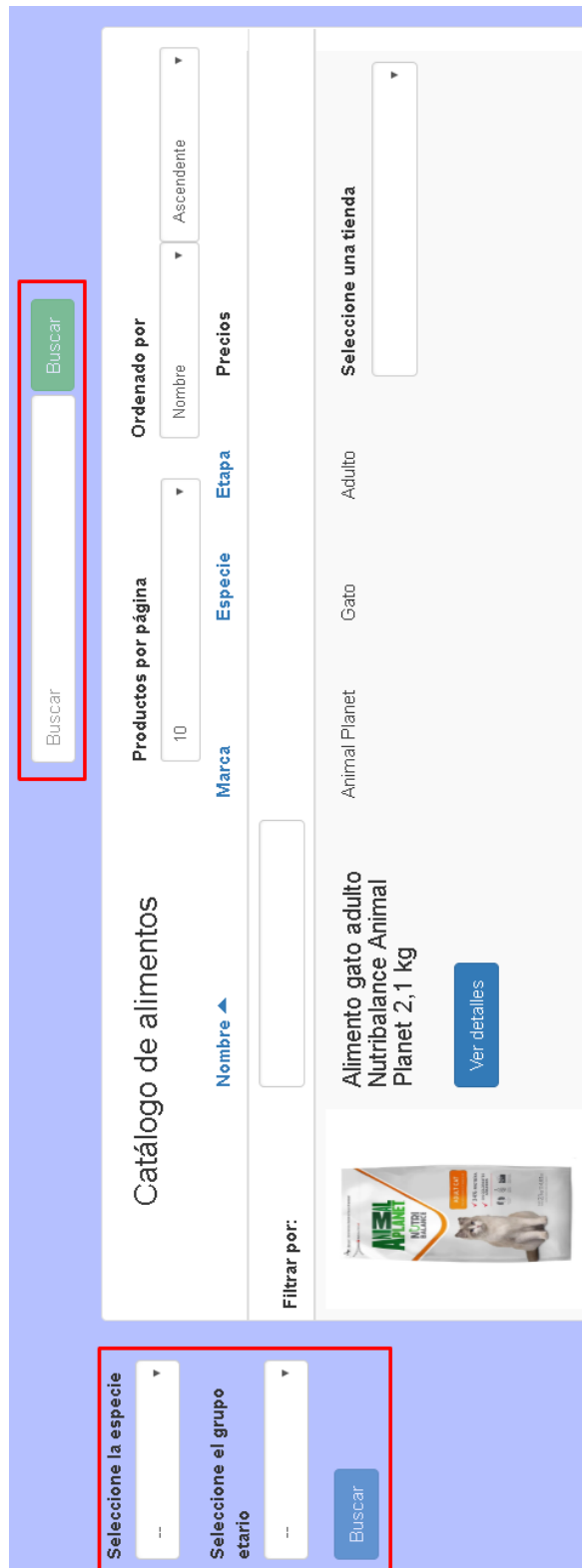


Figura 13. Elementos que permiten la búsqueda de alimentos.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

7.3.2. Recursos y servicios de la aplicación

Para cumplir con los casos de uso, y seguir la lógica de la plataforma de alimentos para mascotas, en la Tabla 10 se han definido los recursos y servicios que la aplicación ha de consumir.

ID	Recurso	Servicio	Observaciones	CDU asociado
Q_Inicio Sesión	Nueva sesión	Consulta que, a partir del e-mail y la contraseña de un usuario registrado, retorna un código y, un mensaje o un objeto usuario.	Si el código es 200, el Login fue correcto y el objeto es el Usuario que inicia sesión. Si el código es 404, el Login fue incorrecto y el mensaje es “Usuario No registrado”.	CDU_01 para iniciar sesión CDU_02 para validar sesión
A_Registrar Usuario	Nuevo usuario	Agregación que añade un nuevo usuario, a partir de un e-mail y una contraseña. Retorna código y mensaje.	Si el código es 200, el registro fue correcto y el mensaje es “Registrado”. Si el código es 404, el registro fue incorrecto y el mensaje es “No registrado”.	CDU_04 para registrarse
A_Insertar Alimento	Nuevo alimento	Agregación que añade un nuevo alimento al registro a partir del nombre, marca, especie, grupo etario, tamaño raza, kilos, imagen y descripción. Retorna código y mensaje.	Si el código es 200, el registro fue exitoso y el mensaje es “Alimento creado”. Si el código es 404, el registro no fue exitoso y el mensaje es “Alimento No creado”.	CDU_05 para agregar nuevo alimento
Q_Listar Alimentos	Alimentos	Consulta que retorna código y, un arreglo de objetos alimento o vacío.	Si el código es 200 entonces retorna un arreglo con los Alimentos registrados. Si el código es 404 entonces retorna un arreglo vacío y el mensaje “Alimentos no encontrados”.	CDU_13 para ver los alimentos en el catálogo.
Q_Listar Alimento	Alimento	Consulta que, a partir de un identificador,	Si el código es 200, la consulta fue exitosa y el objeto es un Alimento que cumple la condición.	CDU_13 para ver el detalle de un alimento del catálogo



Sistema Web para comparar alimentos de mascotas (perros y gatos)

		retorna código y, un mensaje o un objeto	Si el código es 404, la consulta no fue exitosa y el mensaje es “Alimento No disponible”.	
A_Actualizar Alimento	Alimento	Actualización que, mediante el identificador de un alimento, edita el nombre, marca, especie, grupo etario, tamaño raza, kilos, imagen y descripción. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Alimento actualizado”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Alimento No actualizado”.	CDU_06 para modificar un alimento existente
E_Eliminar Alimento	Alimento	Eliminación que, a partir del identificador de un alimento, lo elimina del registro. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Alimento eliminado”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Alimento No eliminado”.	CDU_07 para eliminar un alimento existente
A_Subir Imagen	Imagen	Agregación que añade una imagen al nuevo alimento. Retorna código, un mensaje y un nombre de archivo (si código es 200).	Si el código es 200, la consulta fue exitosa y el mensaje es “Archivo guardado”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Archivo No guardado”.	CDU_05 y CDU_06 ya que se puede agregar o modificar una imagen de un alimento, respectivamente
A_Insertar Tienda	Nueva tienda	Agregación que añade una nueva tienda al registro a partir del nombre, ciudad, dirección, e-mail y teléfono. Retorna código y mensaje.	Si el código es 200, el registro fue exitoso y el mensaje es “Tienda creada”. Si el código es 404, el registro no fue exitoso y el mensaje es “Tienda No creada”.	CDU_08 para agregar una tienda
Q_Listar Tiendas	Tiendas	Consulta que retorna código y un arreglo de objetos tienda.	Si el código es 200 entonces retorna un arreglo con las Tiendas registradas. Si el código es 404 entonces retorna un arreglo vacío y	CDU_13 ya que cada alimento del catálogo se ofrece en distintas tiendas



Sistema Web para comparar alimentos de mascotas (perros y gatos)

			un mensaje “Tiendas no encontradas”.	a precios diferentes
A_Actualizar Tienda	Tienda	Actualización que, mediante el identificador de una tienda edita el nombre, ciudad, dirección, e-mail y teléfono. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Tienda actualizada”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Tienda No actualizada”.	CDU_09 para modificar datos de una tienda existente
E_Eliminar Tienda	Tienda	Eliminación que, a partir del identificador de una tienda, la elimina del registro. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Tienda eliminada”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Tienda No eliminada”.	CDU_10 para eliminar una tienda existente
A_Insertar AlimentoEn Catálogo	Nuevo registro en Catálogo	Agregación que añade un registro en Tienda_Alimentos, a partir del identificador de la tienda, identificador y precio del alimento. Retorna código y mensaje.	Si el código es 200, el registro fue exitoso y el mensaje es “Alimento guardado en catálogo”. Si el código es 404, el registro no fue exitoso y el mensaje es “Alimento No guardado en catálogo”.	Relación con CDU_05 ya que el precio ingresado se almacena en el catálogo de la tienda
A_Modificar PrecioEn Catalogo	Catalogo Tienda	Actualización que, mediante el identificador de una tienda y el identificador de un alimento edita el precio del alimento en el catálogo de la tienda. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Precio actualizado en catálogo”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Precio NO actualizado en catálogo”.	Relación con CDU_06 ya que el precio se modificado se almacena en el catálogo de la tienda
E_Eliminar AlimentoDe Catalogo	Catalogo Tienda	Eliminación que, a partir del identificador de una tienda y del identificador de un	Si el código es 200, la consulta fue exitosa y el mensaje es “Alimento quitado del catálogo”.	Relación indirecta con CDU_07 ya que el alimento se elimina, pero no



Sistema Web para comparar alimentos de mascotas (perros y gatos)

		alimento elimina la asociación del alimento en la tienda. Retorna código y un mensaje.	Si el código es 404, la consulta no fue exitosa y el mensaje es “Alimento NO quitado del catálogo”.	de la relación Alimentos sino que del catálogo de la tienda
Q_Listar Catalogo DeUna Tienda	Catalogo Tienda	Consulta que, a partir del nombre de una tienda, retorna código y, un arreglo de objetos Alimento o un mensaje.	Si el código es 200, la consulta fue exitosa y retorna un arreglo con los Alimentos de la Tienda cuyo nombre coincide con el nombre dado. Si el código es 404 entonces retorna un arreglo vacío y el mensaje “Catálogo No disponible”.	CDU_13 para ver el catálogo específico de una tienda
Q_Listar Tiendas Ofrecen Alimento	Tiendas	Consulta que, a partir del identificador de un alimento, retorna código junto a un arreglo de objetos Tienda o un mensaje.	Si el código es 200, la consulta fue exitosa y retorna un arreglo con el identificador y nombre de las tiendas que poseen el identificador del alimento. Si el código es 404 entonces retorna un arreglo vacío y el mensaje “Tiendas No disponibles”.	CDU_13 para ver en el catálogo los precios que posee un alimento en distintas tiendas
Q_Buscar Alimento porNombre	Alimentos	Consulta que, a partir de una cadena de caracteres que represente el nombre de alimento o una parte de este, retorna un código y, un objeto o un mensaje.	Si el código es 200, la búsqueda fue exitosa y el objeto es un arreglo de Alimentos que cumplen la condición. Si el código es 404 entonces retorna un arreglo vacío y el mensaje “Alimentos No encontrados para el nombre dado”.	CDU_14 para buscar alimentos
Q_Buscar Alimento porEspecie yGrupo Etario	Alimentos	Consulta que, a partir de una especie y un grupo etario, retorna un código y, un mensaje o un objeto	Si el código es 200, la búsqueda fue exitosa y el objeto es un arreglo de Alimentos que cumplen las condiciones. Si el código es 404 entonces retorna un arreglo vacío y el mensaje “Alimentos No	CDU_14 para buscar alimentos



Sistema Web para comparar alimentos de mascotas (perros y gatos)

			encontrados para especie y grupo etario especificados”.	
A_Insertar Opinión	Nueva opinión	Agregación que añade una nueva opinión al registro a partir de la fecha, comentario alimento, nota alimento, comentario tienda, nota tienda, id usuario, id alimento e id tienda. Retorna código y mensaje.	Si el código es 200, el registro fue exitoso y el mensaje es “Opinión creada”. Si el código es 404, el registro no fue exitoso y el mensaje es “Opinión No creada”.	CDU_15 para agregar una nueva opinión
Q_Listar Opiniones	Opinión	Consulta que retorna código y, un mensaje o un arreglo de objetos opinión .	Si el código es 200, la consulta fue exitosa y retorna un arreglo de opiniones ordenadas por fecha en forma descendente. Si el código es 404 entonces retorna un arreglo vacío y el mensaje es “Opiniones no encontradas”.	CDU_11 para que el Administrador pueda revisar todas las opiniones recibidas
Q_Listar Opiniones de un Alimento	Opiniones	Consulta que, a partir del identificador de un alimento, retorna código y, un mensaje o un objeto.	Si el código es 200, la consulta fue exitosa y retorna un arreglo con las Opiniones que cumplen la condición (coincidencia de identificador del alimento) y el promedio de notas que ha recibido el alimento. Si el código es 404 entonces retorna un arreglo vacío de opiniones y el mensaje es “Alimento no registra opiniones”.	CDU_13 para que el usuario pueda ver las opiniones de otros usuarios en un alimento dado.
E_Eliminar Opinión	Opinión	Eliminación que, a partir del identificador de una opinión, la elimina del registro. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Opinión eliminada”. Si el código es 404, la consulta no fue exitosa y el	CDU_12 para que el Administrador pueda eliminar opiniones inapropiadas



Sistema Web para comparar alimentos de mascotas (perros y gatos)

			mensaje es “Opinión No eliminada”.	CDU_16 para que el usuario pueda eliminar alguna de sus opiniones.
A_Insertar Cotización	Cotización	Agregación que añade una nueva cotización, a partir del nombre y la fecha. Retorna código y mensaje.	Si el código es 200, el registro fue exitoso y el mensaje es “Cotización creada”. Si el código es 404, el registro no fue exitoso y el mensaje es “Cotización No creada”.	CDU_17 para que el usuario pueda realizar registros de comparación de precios
Q_Listar Cotizaciones DeUn Usuario	Cotización	Consulta que, a partir del identificador de un usuario, retorna código y, un mensaje o un arreglo de cotizaciones.	Si el código es 200, la consulta fue exitosa y retorna un arreglo con las cotizaciones que cumplen la condición (coincidencia de identificador del usuario). Si el código es 404 entonces retorna un arreglo vacío y el mensaje es “Usuario no registra cotizaciones”.	CDU_17 para que el usuario pueda ver su registro de cotizaciones.
A_Actualizar Cotización	Cotización	Actualización que, mediante el identificador de una cotización edita el nombre. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Cotización actualizada”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Cotización No actualizada”.	CDU_18 para modificar el nombre de una cotización existente.
E_Eliminar Cotización	Cotización	Eliminación que, a partir del identificador de una cotización, la elimina del registro. Además, elimina los alimentos asociados a esa cotización. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Cotización eliminada y alimentos cotizados eliminados”. Si el código es 404, la consulta no fue exitosa y el mensaje es “Cotización No eliminada”.	CDU_19 para eliminar una cotización y sus alimentos cotizados.
A_Agregar Alimento aCotización	Detalle Cotización	Agregación que, a partir de los identificadores de	Si el código es 200, el registro fue exitoso y el	CDU_20 para agregar un alimento a la



Sistema Web para comparar alimentos de mascotas (perros y gatos)

		una cotización y de un alimento, añade el alimento a la cotización. Retorna código y mensaje.	mensaje es “Alimento cotizado”. Si el código es 404, el registro no fue exitoso y el mensaje es “Alimento No cotizado”.	cotización deseada
E_Eliminar AlimentoDe Cotización	Detalle Cotización	Eliminación que, a partir de los identificadores de una cotización y de un alimento, elimina el alimento de la cotización. Retorna código y un mensaje.	Si el código es 200, la consulta fue exitosa y el mensaje es “Alimento eliminado de la cotización”. Si el código es 404, la consulta no fue exitosa y el mensaje es “No se pudo eliminar el alimento de la cotización”.	CDU_21 para eliminar un alimento de la cotización deseada

Tabla 11. Recursos y servicios de la aplicación.

Así como en la Tabla 9 se asocian los casos de uso a los diferentes requisitos funcionales, en la Tabla 10 cada recurso y servicio se encuentra asociado a uno o más casos de uso, razón por la cual quedan cubiertos todos los recursos y servicios definidos. La única excepción es el CDU_03 Cerrar sesión, ya que corresponde a una función dentro del Front-end que no necesita consumir un servicio.



7.4. Arquitectura de Software

7.4.1. Arquitectura Modelo Vista Controlador

El desarrollo del proyecto utiliza el patrón de arquitectura MVC, el cual separa los datos y la lógica de negocio -de una aplicación- de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. MVC se constituye a partir de tres componentes distintos (un modelo, una vista y un controlador) para separar, por un lado, la representación de la información, y por otro lado la interacción del usuario. Esto define la estructura de una aplicación. El uso de MVC implica seguir las ideas que este patrón promueve, tales como la reutilización de código y la separación de conceptos, con el propósito de facilitar el desarrollo de aplicaciones y su posterior mantenimiento.

A continuación, se definen los componentes de esta arquitectura.

- a) Modelo: es el componente responsable de la gestión de los datos de la aplicación. Responde a las solicitudes de la vista y a las instrucciones del controlador para actualizarse.
- b) Vista: es el componente responsable de la presentación de los datos en un formato particular, desencadenada por la decisión del controlador de presentar los datos. También captura eventos realizados por el usuario.
- c) Controlador: es el componente responsable de responder a las entradas del usuario y realiza interacciones en los objetos del modelo de datos. Además, este recibe entradas, las valida y luego realiza operaciones que modifican el estado del modelo de datos.

7.4.2. Relación entre el framework Angular y el patrón MVC

Angular no tiene un modelo-vista-controlador (MVC) bien definido, ya que el modelo y la vista mantienen una estrecha relación. Esto se debe al concepto “two-way data binding” que utiliza Angular, en el que la sincronización de los datos entre la vista y el modelo es totalmente dependiente, es decir, en la vista podemos modificar el modelo y viceversa.

Lo anterior implica que la independencia que se produce en un MVC clásico aquí no se produzca y, por lo tanto, tiende a llamarse modelo-vista vista-modelo (MVVM), porque no se tiene claridad acerca de su patrón de arquitectura. Actualmente, es tema de discusión entre los usuarios de Angular, en cómo considerar a este framework.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

No ocurre así con el modelo como lógica de negocio, como pueden ser los servicios y aquellos datos que se inyectan, que bien pueden ser considerados parte del modelo, ya que son totalmente independientes de la vista. Sin embargo, es posible adaptar el desarrollo con Angular utilizando el patrón MVC.

7.4.3. Diseño de arquitectura funcional

El proceso de desarrollo cumple con la estructura de arquitectura MVC, ya que la vista no extrae ni procesa datos en bruto; el controlador es el que obtiene los datos del modelo, a través de los servicios, y los incorpora en la vista, por lo tanto, los datos que solicita y recibe pueden ser incluidos fácilmente.

7.4.4. Arquitectura Orientada a Servicios

Adicionalmente, el desarrollo de la aplicación se intenta enmarcar en lo que se conoce como Arquitectura Orientada a Servicios (SOA), la cual cuenta con tres actores principales:

- (i) el **registro de servicios**: repositorio de servicios disponibles que permite visualizar la información que proveen los proveedores de servicios a los consumidores interesados.
- (ii) el **proveedor de servicios**: entidad accesible a través de la red que acepta y ejecuta consultas de consumidores y publica sus servicios en el registro de servicios para que el consumidor pueda realizar peticiones y acceder a estos.
- (iii) el **consumidor de servicios**: aplicación, un módulo de software u otro servicio que demanda (*consume*) la funcionalidad proporcionada por un servicio provisto por un proveedor, y la ejecuta en una interfaz definida.

Resulta importante señalar que la aplicación no posee un Registro de Servicios ya que no opera un servidor externo que aloje los servicios propiamente tal. Sin embargo, tanto el Proveedor como el Consumidor de servicios se encuentran bien definidos.

El Proveedor de servicios se constituye por el Back-End de la aplicación (desarrollado con Slim PHP) y la Base de Datos, ya que la primera posee el resultado de la construcción de los servicios y la segunda posee la información que se ha de consultar. Por otra parte, el



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Consumidor de Servicios es la aplicación propiamente tal (el Front-End desarrollado con Angular) que, a través de sus interfaces, permite a los usuarios realizar peticiones a los distintos servicios del Proveedor.

A continuación, se presenta un ejemplo real del empleo de las arquitecturas MVC y SOA en la aplicación, así como también las capas que la constituyen y cómo interactúan para procesar los recursos y servicios que deben ser consumidos para el funcionamiento del sistema.

La Figura 14 muestra cómo es el proceso secuencial de la información de los recursos y servicios que la aplicación ha de consumir.

Por ejemplo: Se desea listar el registro de alimentos en el catálogo de productos.

- (1) Un usuario ingresa a la aplicación en la ruta local <http://localhost:4200>,
- (2) La aplicación carga la vista principal en la ruta vacía, es decir, el path: “” que se encuentra almacenado en la variable “appRoutes” en el archivo app.routing.ts.

```
const appRoutes: Routes = [
  {path: "", component: CatalogoListComponent},
  ... //otras path y componentes
];
```

- (3) La variable “routing” (análoga a Router Provider en la Figura 14) tiene cargado el módulo de enrutamiento para la llamada a los Componentes (de la capa Controlador), el cual habilita y asocia el path “” con el componente “CatalogoListComponent”.

```
export const routing: ModuleWithProviders = RouterModule.forRoot(appRoutes);
```

- (4) El componente “CatalogoListComponent” importa los modelos para la creación de objetos y arreglos de este tipo de objetos, en nuestro ejemplo: CatalogoListComponent importa los modelos Alimento y Tienda.

```
import { Alimento } from '../models/alimento'; permite crear el arreglo “alimentos: Alimento[]”
import { Tienda } from '../models/tienda'; permite crear el arreglo “tiendas Tienda[]”
```




Sistema Web para comparar alimentos de mascotas (perros y gatos)

- (5) El componente “CatalogoListComponent” importa los servicios de la subcapa para invocar a los métodos del Back-end, los cuales realizan peticiones a la Base de Datos para consumir los recursos y servicios definidos en la Tabla 10. En nuestro ejemplo: CatalogoListComponent importa los modelos Alimento y Tienda.

```
import { AlimentoService } from '../services/alimento.service'; permite crear la variable “private _alimentoService: AlimentoService”
```

```
import { TiendaService } from '../services/tienda.service'; permite crear la variable “private _tiendaService: TiendaService”
```

Luego, el componente actual ejecuta la función getAlimentos() que, a su vez, hace uso del servicio “_alimentoService” para ejecutar el método getAlimentos() que este posee [método de la subcapa de Servicios].

- (6) En este punto actúa el framework Slim PHP que, siendo la base del Back-End, ejecuta el método del tipo GET que invoca al servicio de la ruta “/alimentos”.

```
$app->get('/alimentos', function() use($db, $app){
(7) // Consulta SQL a la base de datos → Select * from alimentos;
(8) // Respuesta de la base de datos → N tuplas en el array $alimentos,
    donde N es el número de registros en la tabla Alimento (con N≥0)
(9) // Codificación de la respuesta en formato JSON
    $result = array('code' => 200, 'data' => $alimentos);
    echo json_encode($result);
}
```

donde \$app es la aplicación de Slim en ejecución y \$db es la conexión a la base de datos.

En este punto, es importante recordar lo que dice la columna Observaciones del servicio con ID: “Q_ListarAlimentos” de la Tabla 10, que señala: “Si el código es 200 entonces retorna un arreglo con los Alimentos registrados”.

- (10) Posteriormente, el método getAlimentos() [el segundo que se menciona] recibe la respuesta en formato JSON y la devuelve al componente, que almacena la respuesta en el arreglo “alimentos: Alimento[]” declarada en (4).
- (11) Cabe señalar que este componente posee el atributo “templateURL” que permite asociar una vista (.html) para mostrar información. En este caso, la vista es “catalogo-list.html”, en la cual, a través de la directiva *ngFor se recorre el arreglo y en una



Sistema Web para comparar alimentos de mascotas (perros y gatos)

estructura llamada DataTable, que simula una tabla con paginación, se va agregando el detalle de los alimentos en el catálogo (imagen, nombre, especie, grupo etario, entre otros).

- (12) Finalmente, los alimentos se muestran a los usuarios a través de la vista.

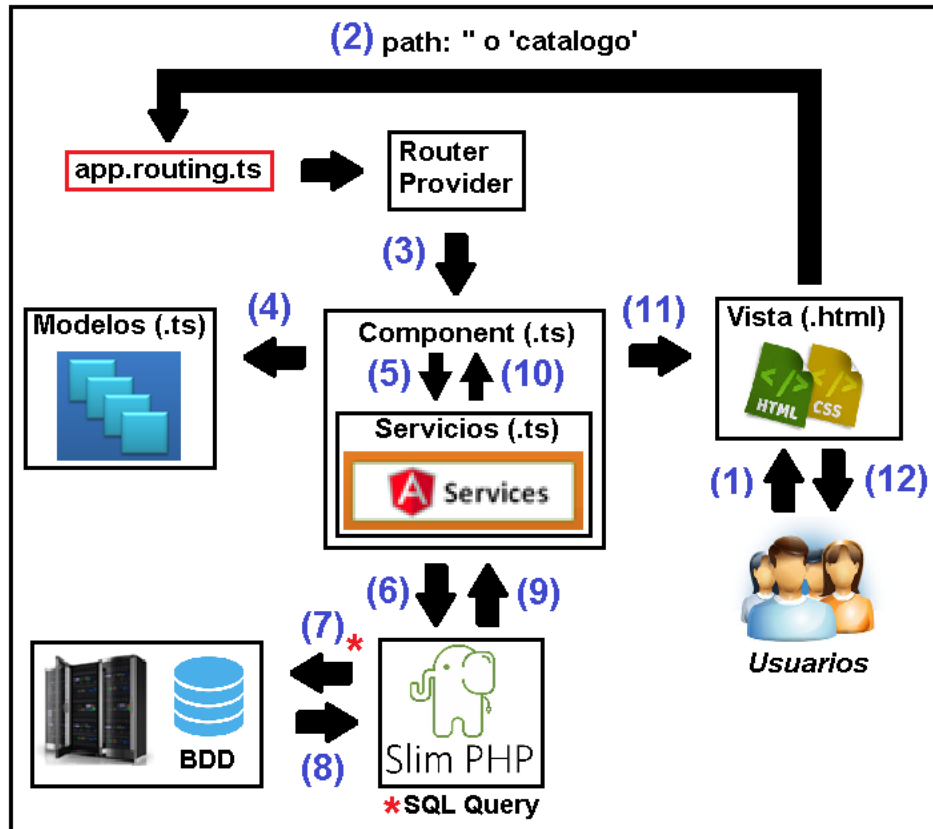


Figura 14. Arquitectura MVC de la aplicación.

Cabe señalar que el Componente (o Controlador) incluye la subcapa de Servicios, los cuales realizan las peticiones a los recursos y servicios que la aplicación ha de consumir, y de esta forma proveer datos a Vista. Este comportamiento de la arquitectura SOA se logra gracias al uso del framework Slim PHP que permite la construcción del Back-End.



8. Pruebas

En este capítulo se presentan las pruebas de software cuyo fin es proporcionar información objetiva e independiente sobre la calidad del producto.

En específico, se realizan tres tipos de pruebas y tienen como propósito analizar el comportamiento del sistema en relación a la comunicación que existe entre las dos capas principales. Por esta razón, se realizan peticiones desde la capa Front-end para obtener respuesta de los servicios que se encuentran en la capa Back-end.

8.1. Especificación de las pruebas

Las características puestas a prueba son las siguientes: Rendimiento, Funcionalidad e Interfaz y Navegación.

8.1.1. Prueba de múltiple concurrencia de usuarios

Característica a probar	Rendimiento
Nivel de prueba	Sistema
Objetivo de la prueba	Determinar si la aplicación web es capaz de controlar el acceso de distintos usuarios, bajo los siguientes escenarios: <ul style="list-style-type: none"> • Mínimo número de usuarios (tres usuarios), todos ejecutando la misma función. • La ejecución de la misma función implica que la o las consultas se ejecutan simultáneamente.
Enfoque para la definición de casos de prueba	Caja negra
Técnicas para la definición de casos de prueba	Múltiples usuarios ejecutarán una misma funcionalidad <ul style="list-style-type: none"> • Se utiliza un tamaño mínimo de cinco registros por tabla en la Base de datos con datos representativos
Actividades de prueba	Definir los casos de prueba considerando principalmente la realización de consultas <ul style="list-style-type: none"> • Aplicar los casos de prueba a los elementos especificados considerando la conexión de varios usuarios en el sistema • Registro de resultados
Criterios de cumplimiento	Los casos de prueba se ejecutan correctamente

Tabla 12. Prueba de múltiple concurrencia de usuarios.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

8.1.2. Prueba del Sistema

Característica a probar	Funcionalidad
Nivel de prueba	Sistema
Objetivo de la prueba	Asegurar la correcta navegación y funcionamiento del sistema
Enfoque para la definición de casos de prueba	Caja negra
Técnicas para la definición de casos de prueba	Para las funcionalidades más importantes se utilizarán datos válidos e inválidos, para verificar que: <ul style="list-style-type: none"> • Ocurran las validaciones correspondientes cuando los datos son inválidos • Al ejecutar un proceso los mensajes de éxito y fracaso se muestran efectivamente
Actividades de prueba	<ul style="list-style-type: none"> • Definir los casos de prueba considerando principalmente datos inválidos • Aplicar los casos de prueba a los elementos especificados • Registro de resultados • Corrección de errores
Criterios de cumplimiento	Los casos de prueba se ejecutan según lo esperado

Tabla 13. Prueba de sistema.

8.1.3. Prueba de Usabilidad

Característica a probar	Interfaz y navegación
Nivel de prueba	Sistema
Objetivo de la prueba	Determinar si el usuario es capaz de utilizar las funcionalidades de la aplicación web
Enfoque para la definición de casos de prueba	Caja negra
Técnicas para la definición de casos de prueba	Verificar que la aplicación web no presenta los siguientes problemas de usabilidad: <ul style="list-style-type: none"> • El uso del sistema no es intuitivo • Los mensajes de error no sugieren una posible solución • El seguimiento de los procesos se pierde
Actividades de prueba	Las pruebas de usabilidad serán realizadas por 3 usuarios: <ul style="list-style-type: none"> • Crear casos de prueba para que realicen los usuarios



Sistema Web para comparar alimentos de mascotas (perros y gatos)

	<ul style="list-style-type: none"> • Observar la ejecución de los procedimientos para posterior análisis • Realizar una encuesta de satisfacción con el software • Análisis posterior de los resultados
Criterios de cumplimiento	Los usuarios no tienen dificultad para utilizar el sistema

Tabla 14. Prueba de usabilidad.

8.2.Elementos para pruebas de sistema

Estos elementos de prueba van dirigidos a analizar las funcionalidades más importantes que posee la aplicación Web. Para estos efectos, se definen siete elementos de pruebas asociados a algunos casos de uso:

a) Módulo de Gestión (Actor administrador):

a. Sección de Tiendas.

i. Crear una Tienda → CDU_08.

ii. Editar una Tienda → CDU_09.

b. Sección de Alimentos.

i. Crear un Alimento (desde el catálogo de una tienda) → CDU_05.

ii. Editar un Alimento → CDU_06.

b) Módulo de Aplicación (usuario no registrado que luego se registra):

a. Sección de Registro y Autenticación

i. Registrarse → CDU_04 (usuario no registrado).

ii. Iniciar sesión → CDU_01.

b. Sección de Opiniones

i. Crear una opinión → CDU_15.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

8.3. Responsables de las pruebas

Tipo de prueba	Responsable (nombre)
Múltiple concurrencia de usuarios	Roberto Muñoz, Bastián Escalona, Jesica Martínez
Sistema	Roberto Muñoz, Bastián Escalona, Jesica Martínez
Usabilidad	Roberto Muñoz, Bastián Escalona y Jesica Martínez

Tabla 15. Responsables de las pruebas.

8.4. Planificación de las pruebas

No fue hasta las últimas dos semanas de desarrollo del proyecto, en que se lleva a cabo la realización de pruebas con el objetivo de crear un producto de calidad. Se establece contacto con tres personas para validar las funcionalidades del software, haciendo énfasis en las entradas y salidas de datos en el Sistema. Como en la etapa final el producto de software está terminado, se hicieron las pruebas de desempeño para ver el comportamiento en cuanto a las peticiones de los usuarios y así ver el funcionamiento de la aplicación web.



8.5. Detalle de las pruebas

- a) Pruebas de múltiple concurrencia de usuarios: Se ingresan tres usuarios simultáneamente a la aplicación con el objetivo de poner a prueba su funcionamiento a múltiples peticiones y determinar si este puede responder de forma consistente.
- b) Pruebas de sistema: Las pruebas se realizaron a las funcionalidades más importantes del sistema de acuerdo a los elementos definidos en el punto 8.2. Para mayor información dirigirse al Anexo 3 que posee el detalle de cada prueba.
- c) Prueba de usabilidad: En este caso, tres usuarios realizan navegaciones en la aplicación con el objetivo de evaluar si son capaces de entender el funcionamiento de la aplicación.

La prueba consiste en que los usuarios deben interactuar con la aplicación, realizando una serie de tareas. Se registran las observaciones y se realiza una encuesta para conocer la opinión y perspectiva de los usuarios (véase Anexo 4).

Los resultados de la evaluación se muestran en la Tabla 15, donde cada número representa la cantidad de usuarios que está de acuerdo, no lo está o mantiene una postura neutral con la afirmación planteada. Por último, en la Tabla 16 se da a conocer el número de personas que logra realizar cada prueba de manera correcta en el tiempo asignado.

Pregunta	De acuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	Observaciones
Entiende el vocabulario empleado en el menú, operaciones, vistas y tablas.	3			
Comprende los mensajes de error en los formularios y le resulta fácil corregirlos.	1	1	1	Faltan validaciones
Los formularios son simples, fáciles de entender y sencillos de completar.	2	1		
La aplicación tiene una respuesta rápida las acciones que se realizan.	3			
La estructura de la aplicación es simple y de fácil manejo.	3			



Sistema Web para comparar alimentos de mascotas (perros y gatos)

El catálogo que muestra los alimentos tiene un diseño adecuado.	3			
Los datos que muestra el catálogo de alimentos contienen información necesaria para la toma de decisiones	2	1		
La aplicación no presenta errores al ser utilizada		2	1	No se puede editar la opinión (funcionalidad omitida)

Tabla 16. Resultados prueba de usabilidad.

Pregunta 9. El tiempo límite es el adecuado para realizar cada tarea [Registrar el número de usuarios que fue capaz de realizar correctamente cada prueba].

Tareas a realizar (n° de usuarios)	Logrado			No logrado		
Registrarse (2)	2			0		
Iniciar sesión (3)	3			0		
Ingresar/Modificar/Eliminar una tienda (1)	1	1	1	0	0	0
Ingresar/Modificar/Eliminar un alimento (1)	1	1	1	0	0	0
Ingresar/Eliminar una opinión (2)	2	1		0	1	
Ingresar/Modificar/Eliminar una cotización (2)	2	2	2	0	0	0
Cerrar sesión (3)	3			0		

Tabla 17. Resultados del tiempo límite para realizar la prueba de usabilidad.

8.6. Conclusiones de pruebas efectuadas

Las pruebas de software son básicamente un conjunto de actividades dentro del desarrollo de un sistema que proporcionan información sobre la calidad del producto, con el objetivo de encontrar defectos, fallos de funcionamiento, problemas de usabilidad, etc. Frente a esto, se pueden tomar decisiones y solucionar conflictos a tiempo para aumentar el nivel de confianza con el usuario y evitar la aparición de futuros defectos.

En las etapas cercanas al final del desarrollo del software, se realizaron diversas pruebas que facilitaron la corrección de errores, pero se debió esperar a que estuvieran la mayoría de los requisitos implementados para hacer las pruebas con los usuarios, ya que los



Sistema Web para comparar alimentos de mascotas (perros y gatos)

requerimientos más básicos anteceden a los más complejos para que éstos últimos se puedan realizar.

A través de los resultados arrojados por las pruebas de múltiple concurrencia de usuarios, se puede concluir que la aplicación responde de manera consistente a las múltiples peticiones de usuarios simultáneos. Por otra parte, en relación a las pruebas de sistema, se puede concluir que el software entrega mensajes claros y concisos, propiciando un buen manejo en el funcionamiento del sistema. Adicionalmente, se realizaron pruebas de usabilidad a tres usuarios para evaluar qué tan intuitivo resulta ser el software, obteniendo resultados a través de la aplicación de una encuesta diseñada en relación a las tareas desarrolladas.

Por último, al concluir las pruebas con éxito, el sistema puede ejercer de manera óptima sus funcionalidades, entregando información correcta y adecuada para tener un buen análisis de datos y tomar decisiones al respecto.



9. Conclusiones

Desde el día de la propuesta del proyecto, el mercado digital de alimentos para mascotas no ha sido testigo de algún evento que revolucione la manera en que las tiendas promocionen y los clientes adquieran alimentos para sus mascotas. Se puede decir que el panorama permanece estático. Por esta razón, sigue considerándose una oportunidad desarrollar la aplicación denominada “Sistema Web para comparar alimentos de mascotas (perros y gatos)” pues responde a necesidades inherentes de los procesos de búsqueda y comparación, por ejemplo, agilizar el proceso de comparación que realizan los dueños de mascotas no yendo directamente a las diferentes tiendas, sino que realicen un proceso de comparación online desde sus hogares o lugar de trabajo, y que luego puedan ir directamente a la tienda que ofrece el producto que buscan de acuerdo a sus condiciones de presupuesto.

En primer lugar, mencionar que toda la información contenida y descrita en el presente documento logra el cumplimiento de los objetivos -general y específicos- y que, a partir de esto, se puede concluir lo siguiente:

- Durante el desarrollo del proyecto, surgieron nuevos requerimientos tales como agregar y eliminar opiniones, que permitieron añadir nuevas funcionalidades en pos de mejorar la utilización de la aplicación web.
- Tal como se había previsto, al momento de la ejecución de las pruebas de caja negra, se obtuvieron fallas en la aplicación, las que fueron prontamente corregidas permitiendo un uso más correcto de las funcionalidades del sistema.
- Se logra el íntegro cumplimiento de todos los requerimientos funcionales planteados en el capítulo 4. No obstante, se realizaron pruebas de múltiple concurrencia, de sistema y de usabilidad sobre un subconjunto de estos.

Por otra parte, es importante señalar que no fue considerado en este proyecto la programación de un algoritmo de búsqueda automática para capturar los alimentos de distintas tiendas en la web. Esta funcionalidad si bien permite la actualización en tiempo real de la información del sistema, se optó por realizar el ingreso manual de la información de las tiendas y los alimentos. Adicionalmente, tampoco se considera un perfil dedicado a las



Sistema Web para comparar alimentos de mascotas (perros y gatos)

tiendas para que sean estas las que gestionen su catálogo de alimentos y mantengan actualizados los precios y ofertas en las distintas épocas del año. Esta responsabilidad recae en el administrador del sistema. Por último, el desarrollo de una aplicación móvil que permita a los usuarios de teléfonos móviles la búsqueda, comparación y cotización de alimentos para mascotas.

Finalmente, hacer hincapié en que los procesos de planificación, ejecución y control realizados ponen a prueba al estudiante en relación al compromiso y tiempo que demanda realizar un proyecto en el cual no se puede dejar algo sin implementar, ya que lo que se desarrolla es un sistema que será utilizado por un grupo específico de personas (dueños de mascotas) y que requiere del cuidado y revisiones exhaustivas en todo momento, todo esto con el fin de desarrollar un producto de calidad y útil para los usuarios.



Referencias Bibliográficas

1. Angular.io. Página oficial del framework Angular. Recuperado el 30 de Noviembre de 2018, de <https://angular.io/>.
2. Curso de Angular 7. Desde cero hasta profesional. Recuperado el 30 de Noviembre de 2018, de <https://www.udemy.com/curso-de-angular-4-desde-cero-hasta-profesional/>
3. ¿Qué es TypeScript?, lenguaje de programación del Front-end. Recuperado el 09 de Noviembre de 2018, de <https://devcode.la/blog/que-es-typescript/>
4. Victor Robles Web. Blog sobre desarrollo web y administración de sistema. Recuperado el 07 de Diciembre de 2018, de <https://victorroblesweb.es/>
5. Documentación de Slim PHP. Recuperado el 07 de Diciembre de 2018, de <https://www.slimframework.com/docs/>
6. NodeJS. Descarga del entorno de ejecución para JavaScript. Recuperado el 03 de Septiembre de 2018, de <https://nodejs.org/es/>
7. Sublime Text. Descarga del editor de texto utilizado para el desarrollo del proyecto. Recuperado el 03 de Septiembre de 2018, de <https://www.sublimetext.com/>
8. Bootstrap. Construcción de aplicaciones web responsivas. Recuperado el 07 de Diciembre de 2018, de <https://getbootstrap.com/>.
9. StackOverFlow. Página oficial del sitio de preguntas y respuestas para programadores y profesionales de la informática. Recuperado el 30 de Noviembre de 2018, de <https://es.stackoverflow.com/>
10. Postman. Descarga de la aplicación para el desarrollo de API Rest. Recuperado el 04 de Septiembre de 2018, de <https://www.getpostman.com/>
11. Wamp Server, Apache, PHP, MySQL. Descarga del sistema de infraestructura análogo a Xampp. Recuperado el 04 de Septiembre de 2018, de <http://www.wampserver.com/>
12. Desarrollo web, Qué es una SPA. Recuperado el 07 de Diciembre de 2018, de <https://desarrolloweb.com/articulos/que-es-una-spa.html>
13. Bcrypt. Para realizar el hash de contraseñas utilizando Angular. Recuperado el 07 de Diciembre de 2018, de <https://www.npmjs.com/package/bcrypt>



Anexos



Anexo 1. Encuesta para conocer el interés del público para el desarrollo del sistema

1. Actualmente, ¿utiliza el computador con conexión a Internet?
 Si No Si la respuesta es sí, continuar. De otro modo, terminar.
2. ¿Tiene perro o gato como mascota?
 Si No
 Si la respuesta es sí, continuar. De otro modo, terminar o suponer que sí y continuar.
3. ¿Qué tipo de alimento consume su mascota?
 Comida casera Pellet, ¿de qué marca? _____
 Si la respuesta a la pregunta 3 es la primera opción, continuar en la pregunta 7.
 Si la respuesta a la pregunta 3 es la segunda opción, continuar en la pregunta 4.
4. ¿En dónde compra generalmente el alimento para su mascota?
 Tienda dedicada para mascotas Supermercado
 Ferretería o almacén (mini-market)
 Sitio web o app móvil de una tienda, si lo recuerda, ¿cuál? _____
 Otro lugar, ¿cuál? _____
5. ¿Cómo se enteró de que en ese lugar estaba el producto para su mascota?
 Contacto cercano (familiar, conocido) Medio de comunicación (TV, diario)
 Algún sitio web o aplicación móvil, ¿cuál? _____
 Otro, ¿cuál? _____
6. Actualmente, ¿Se siente cómodo/a al comprar dicho alimento en ese lugar?
 Si No, porque _____
7. ¿Estaría interesado/a en contar con una aplicación web que le permitiera buscar el alimento para su mascota? Si No
8. ¿Le gustaría que esa aplicación le permitiera comparar y cotizar alimentos de diferentes marcas y precios en los distintos puntos de venta? Si No
9. Espacio para recibir comentarios por parte del colaborador.



Anexo 2. Especificación de los Casos de uso

I) De autenticación y registro

Caso de uso: Iniciar sesión

ID	CDU_01
Actores	Usuario registrado o el Administrador
Descripción	El usuario inicia sesión ingresando sus credenciales de acceso y el sistema redirige al perfil que le corresponda.
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse registrado en la base de datos. 2. El usuario debe recordar sus credenciales de acceso (correo electrónico y contraseña).
Flujo principal	<ol style="list-style-type: none"> 1. El usuario oprime el botón “Iniciar sesión” del menú “Mi cuenta” 2. El sistema abre el formulario para el inicio de sesión 3. El usuario ingresa sus credenciales de acceso en el formulario 4. El usuario oprime el botón “Iniciar sesión”
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema valida las credenciales de acceso. 2. El sistema redirige al usuario a su perfil.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario ingresa credenciales erróneas 2. El usuario oprime el botón “Iniciar sesión” 3. El sistema notifica error en las credenciales de acceso del usuario 4. El usuario regresa al paso 3 del flujo principal.

Tabla 18. Caso de uso - Iniciar sesión.

Caso de uso: Validar sesión

ID	CDU_02
Actores	Usuario registrado o el Administrador
Descripción	Una vez que el usuario oprime el botón “Iniciar sesión” del formulario, el sistema verifica si las credenciales ingresadas se encuentran en los registros de usuarios y notifica el éxito o fracaso del Login.
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe encontrarse registrado en la base de datos. 2. El usuario debe recordar sus credenciales de acceso.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario oprime la opción “Iniciar sesión” del menú “Mi cuenta” 2. El usuario ingresa sus credenciales de acceso en el formulario y oprime el botón “Iniciar sesión” 3. El sistema realiza una consulta a la base de datos consultando por la existencia de las credenciales de acceso en el registro de usuarios registrados.
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema redirige al usuario a su perfil.
Flujo alternativo	<ol style="list-style-type: none"> 1. Si se ingresan credenciales erróneas, el sistema notifica que hay errores en el email y/o contraseña ingresados, denegando el acceso al sistema, pero manteniendo abierto el formulario de Login. Se regresa al paso 2 del flujo principal.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Tabla 19. Caso de uso - Validar sesión.

Caso de uso: Cerrar sesión

ID	CDU_03
Actores	Usuario registrado o el Administrador
Descripción	El usuario cierra su sesión iniciada y el sistema redirige a la página principal.
Pre-condiciones	1. El usuario debe encontrarse con la sesión iniciada
Flujo principal	1. El usuario se dirige al menú “Mi Perfil” 2. El usuario oprime la opción “Cerrar sesión”
Post-condiciones	1. El sistema cierra la sesión actual del usuario 2. El sistema redirige al usuario a la página principal (catálogo de alimentos)
Flujo alternativo	---

Tabla 20. Caso de uso - Cerrar sesión.

Caso de uso: Registrarse

ID	CDU_04
Actores	Usuario no registrado
Descripción	Un usuario no registrado se registra en el sistema, ingresando un correo electrónico, una contraseña y la confirmación de esta. Luego, el sistema notifica el resultado de la operación.
Pre-condiciones	1. El usuario desea registrarse en el sistema. 2. El usuario posee una cuenta de correo electrónico válida y una contraseña.
Flujo principal	1. El usuario se dirige al menú “Mi Cuenta” 2. El usuario oprime la opción “Registrarse” 3. El usuario ingresa su correo electrónico y su contraseña (dos veces) en el formulario 4. El usuario oprime el botón “Registrarse” para darse de alta en el sistema 5. El sistema verifica que el correo electrónico no se encuentre asociado a otra cuenta de usuario 6. El sistema registra las credenciales de acceso en su registro de usuarios registrados
Post-condiciones	1. El sistema notifica al usuario que su cuenta ha sido creada exitosamente.
Flujo alternativo	1. El sistema puede encontrar una coincidencia en el correo electrónico del usuario en otra cuenta de usuario 2. En ese caso, el sistema notifica que el correo electrónico ingresado ya se encuentra registrado y debe ingresar otro correo. Se regresa al paso 3 del flujo principal.

Tabla 21. Caso de uso - Registrarse.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

II) Administrador

A este actor se deben considerar los siguientes casos de uso: 01, 02 y 03.

Caso de uso: Agregar alimento (no existente, desde el catálogo de una tienda)

ID	CDU_05
Actores	Administrador
Descripción	El administrador ingresa un nuevo alimento a una tienda del sistema. Luego el sistema notifica que se ha ingresado con éxito.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	1. El administrador se dirige al menú “Gestión de Tiendas” 2. Se selecciona la tienda y oprime el botón “Ver catálogo” 3. El sistema redirige al catálogo de alimentos de la tienda seleccionada. 4. Se oprime el botón “Agregar nuevo alimento” 5. El sistema abre el formulario para el ingreso de un nuevo alimento 6. El administrador completa los campos asociados al nuevo alimento (nombre, marca, especie del animal, grupo etario, tamaño de la raza, kilos del alimento, descripción, precio y una imagen). 7. El sistema reconoce que todos los campos fueron llenados y habilita el botón Guardar. 8. El administrador oprime el botón “Guardar”
Post-condiciones	1. El sistema verifica que la información cumpla con los tipos de datos definidos. 2. El sistema verifica que el alimento no se encuentre ya registrado 3. El sistema inserta el nuevo alimento en el catálogo de alimentos 4. El sistema notifica que el registro ha sido creado exitosamente
Flujo alternativo	1. El sistema puede encontrar una coincidencia en los atributos del nuevo alimento en otro alimento registrado. 2. En ese caso, el sistema notifica al administrador que el alimento ya se encuentra registrado y debe modificar sus atributos.

Tabla 22. Caso de uso - Agregar alimento.

Caso de uso: Modificar alimento

ID	CDU_06
Actores	Administrador
Descripción	El administrador modifica los datos de un alimento registrado en el sistema, y este último notifica que se ha modificado con éxito.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	1. El administrador se dirige al menú “Gestión de Alimentos” 2. El administrador busca el alimento que desea modificar. 3. El administrador oprime el botón “Editar” en la fila en que se encuentre el alimento. 4. El sistema redirige al formulario cargando con los datos del alimento a modificar.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

	5. El administrador modifica los datos que desee y oprime el botón “Guardar cambios”.
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema verifica que la información cumpla con los tipos de datos definidos. 2. El sistema verifica que la nueva información del alimento no se encuentre ya registrada 3. El sistema actualiza los datos del alimento en el registro de alimentos 4. El sistema notifica al administrador que la modificación de los datos se ha realizado exitosamente
Flujo alternativo	<ol style="list-style-type: none"> 1. El sistema puede encontrar una coincidencia en los atributos del alimento que se modifica en otro alimento registrado. 2. En ese caso, el sistema notifica al administrador que los nuevos datos del alimento ya se encuentran registrados y sugiere modificar nuevamente los atributos.

Tabla 23. Caso de uso - Modificar alimento.

Caso de uso: Eliminar alimento

ID	CDU_07
Actores	Administrador
Descripción	El administrador elimina un alimento registrado en el sistema, y este último notifica que se ha eliminado con éxito.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador se dirige al menú “Gestión de Alimentos” 2. El administrador busca el alimento que desea eliminar. 3. El administrador oprime el botón “Eliminar” en la fila en que se encuentre el alimento. 4. El sistema solicita la confirmación de la acción Eliminar para evitar la eliminación por equivocación. 5. El administrador confirma la eliminación del alimento.
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema elimina el alimento. 2. El sistema notifica al administrador que la eliminación del alimento se ha realizado exitosamente. 3. El sistema recarga el registro de alimentos.
Flujo alternativo	1. El sistema pudiera no tener alimentos registrados, en ese caso, el sistema muestra en la vista un mensaje notificando el hecho, impidiendo la realización de este caso de uso.

Tabla 24. Caso de uso - Eliminar alimento.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Caso de uso: Agregar tienda

ID	CDU_08
Actores	Administrador
Descripción	El administrador del sistema, ingresa una nueva tienda al sistema, y este último notifica que se ha ingresado con éxito.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador se dirige al menú “Gestión de Tiendas” 2. El administrador oprime el botón “Agregar nueva tienda” 3. El sistema redirige al formulario para el ingreso de una nueva tienda. 4. El administrador completa los campos asociados a la nueva tienda (nombre, dirección, ciudad, email y teléfono). 5. El administrador oprime el botón “Guardar nueva tienda”
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema verifica que la información cumpla con los tipos de datos definidos. 2. El sistema verifica que la tienda no se encuentre ya registrada 3. El sistema inserta la nueva tienda en el registro de tiendas. 4. El sistema notifica al administrador que el registro ha sido creado exitosamente
Flujo alternativo	<ol style="list-style-type: none"> 1. El sistema puede encontrar una coincidencia en los atributos de la nueva tienda en otra tienda registrada. 2. En ese caso, el sistema notifica al administrador que la tienda ya se encuentra registrada y debe modificar sus atributos.

Tabla 25. Caso de uso - Agregar tienda.

Caso de uso: Modificar tienda

ID	CDU_09
Actores	Administrador
Descripción	El administrador modifica los datos de una tienda registrada en el sistema, y este último notifica que se ha modificado con éxito.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador se dirige al menú “Gestión de Tiendas” 2. El administrador busca la tienda que desea modificar. 3. El administrador oprime el botón “Editar” en la fila en que se encuentre la tienda. 4. El sistema redirige al formulario cargando con los datos de la tienda a modificar. 5. El administrador modifica los datos que desee y oprime el botón “Guardar cambios”.
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema verifica que la información cumpla con los tipos de datos definidos. 2. El sistema verifica que la nueva información de la tienda no se encuentre ya registrada 3. El sistema actualiza los datos de la tienda en el registro de tiendas.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

	4. El sistema notifica al administrador que la modificación de los datos se ha realizado exitosamente
Flujo alternativo	1. El sistema puede encontrar una coincidencia en los atributos de la tienda que se modifica en otra tienda registrada. 2. En ese caso, el sistema notifica al administrador que los nuevos datos de la tienda ya se encuentran registrados y sugiere modificar nuevamente los atributos.

Tabla 26. Caso de uso - Modificar tienda.

Caso de uso: Eliminar tienda

ID	CDU_10
Actores	Administrador
Descripción	El administrador elimina una tienda registrada en el sistema, y este último notifica que se ha eliminado con éxito.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	1. El administrador se dirige al menú “Gestión de Tiendas” 2. El administrador busca la tienda que desea eliminar. 3. El administrador oprime el botón “Eliminar” en la fila en que se encuentre la tienda. 4. El sistema solicita la confirmación de la acción Eliminar para evitar la eliminación por equivocación. 5. El administrador confirma la eliminación de la tienda.
Post-condiciones	1. El sistema elimina la tienda. 2. El sistema no elimina los alimentos asociados a esta tienda. 3. El sistema notifica al administrador que la eliminación de la tienda se ha realizado exitosamente. 4. El sistema recarga el registro de tiendas.
Flujo alternativo	--

Tabla 27. Caso de uso - Eliminar tienda.

Caso de uso: Revisar opiniones

ID	CDU_11
Actores	Administrador
Descripción	El administrador del sistema accede al registro de opiniones que reciben los alimentos y las tiendas.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	1. El administrador se dirige al menú “Revisar Opiniones” 2. El sistema redirige a la vista que contiene, en una tabla, el registro de todas las opiniones.
Post-condiciones	--
Flujo alternativo	1. El sistema pudiera no tener opiniones registradas, en ese caso, el sistema despliega un mensaje notificando la situación.

Tabla 28. Caso de uso - Revisar opiniones.



Caso de uso: Eliminar opinión inadecuada

ID	CDU_12
Actores	Administrador
Descripción	El administrador del sistema elimina una opinión que considere inadecuada por incluir palabras inapropiadas.
Pre-condiciones	1. El administrador debe tener sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El administrador se dirige al menú “Revisar Opiniones” 2. El sistema redirige a la vista que contiene, en una tabla, el registro de todas las opiniones. 3. El administrador busca la opinión que desea eliminar. 4. El administrador oprime el botón “Eliminar” en la fila en que se encuentre la opinión. 5. El sistema solicita la confirmación de la acción Eliminar para evitar la eliminación por equivocación. 6. El administrador confirma la eliminación de la opinión.
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema elimina la opinión seleccionada. 2. El sistema notifica al administrador que la eliminación de la opinión se ha realizado exitosamente. 3. El sistema recarga el registro de opiniones.
Flujo alternativo	1. El sistema pudiera no tener opiniones registradas, en ese caso, el sistema muestra en la vista un mensaje notificando el hecho, impidiendo la realización de este caso de uso.

Tabla 29. Caso de uso - Eliminar opinión inadecuada.

III) Usuario no registrado

A este actor se debe considerar, además, el caso de uso 04 y, si 04 está realizado, también los casos de uso 01, 02 y 03.

Caso de uso: Ver catálogo

ID	CDU_13
Actores	Cualquier usuario (registrado o no registrado)
Descripción	El usuario accede al catálogo de alimentos para ver cuáles existen en el sistema.
Pre-condiciones	No hay, aunque para este caso de uso es opcional que el usuario tenga su sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario entra a la aplicación. 2. El sistema se inicia en su vista principal, la cual es el catálogo de alimentos. 3. El usuario puede ver los alimentos registrados.
Post-condiciones	--
Flujo alternativo	1. El sistema pudiera no tener alimentos registrados, en ese caso, el sistema despliega un mensaje notificando la situación.

Tabla 30. Caso de uso - Ver catálogo.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Caso de uso: Buscar alimentos

ID	CDU_14
Actores	Cualquier usuario (registrado o no registrado)
Descripción	El usuario realiza búsqueda de alimentos en el catálogo de alimentos.
Pre-condiciones	No hay, aunque para este caso de uso es opcional que el usuario tenga su sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al catálogo de alimentos del sistema. 2. Hay dos posibles flujos ya que existen dos maneras de realizar la búsqueda, en la cual el usuario: <ol style="list-style-type: none"> 2.a.) escribe una cadena de caracteres que represente el nombre del alimento o una parte de este para realizar la búsqueda, o 2.b) selecciona dos parámetros para realizar la búsqueda: la especie de su mascota (Perro o Gato) y el grupo etario del animal. 3. El usuario oprime el botón “Buscar” para realizar la búsqueda.
Post-condiciones	<p>Puede ocurrir uno de dos posibles escenarios, esto depende del flujo 2.a) o 2.b).</p> <ol style="list-style-type: none"> 1. El sistema realiza la búsqueda en base a: <ol style="list-style-type: none"> 2.a) la cadena ingresada y muestra los alimentos cuyos nombres contengan dicha cadena, o 2.b) la especie y el grupo etario seleccionados y muestra los alimentos que cumplan estas condiciones. 3. El sistema recarga el componente Catálogo mostrando únicamente los alimentos que cumplen las condiciones.
Flujo alternativo	<ol style="list-style-type: none"> 1. El sistema pudiera no tener alimentos registrados, en ese caso, el sistema despliega un mensaje notificando la situación, impidiendo la realización de este caso de uso. <ol style="list-style-type: none"> 2.a) Para el primer flujo, el usuario pudiera ingresar una cadena de caracteres que ningún alimento posee, en ese caso, el sistema muestra en la vista un aviso notificando que no existen alimentos asociados a esa cadena. 2.b) Para el segundo flujo, pudiera no haber alimentos para los criterios seleccionados, en ese caso, el sistema muestra en la vista un aviso notificando que no se encontraron alimentos que cumplan esos criterios.

Tabla 31. Caso de uso – Buscar alimentos.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

IV) Usuario registrado

A este actor se deben considerar los siguientes casos de uso: 01, 02, 03, 13 y 14.

Caso de uso: Agregar opinión

ID	CDU_15
Actores	Usuario registrado
Descripción	El usuario registrado ingresa una nueva opinión al sistema, y este último notifica que se ha ingresado con éxito.
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe tener sesión iniciada en el sistema. 2. El usuario debe tener experiencia con el alimento y la tienda en donde este fue adquirido.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al catálogo de alimentos. 2. El usuario busca y selecciona un producto del catálogo oprimiendo el botón “Ver detalles” 3. En el detalle del alimento, el usuario puede generar una opinión oprimiendo el botón “Dejar opinión” 4. El sistema abre un formulario para el ingreso de una nueva opinión. 4. El usuario completa los campos asociados a la nueva opinión <ol style="list-style-type: none"> 4.1. Evaluar producto (ingresar una nota entre 1 y 7) 4.2. Comentar producto (ingresar un comentario) 4.3. Seleccionar tienda donde fue adquirido el producto 4.4. Evaluar tienda (ingresar un comentario) 4.5. Comentar tienda (ingresar una nota entre 1 y 7) 5. El usuario oprime el botón “Enviar opinión”
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema verifica que la información cumpla con los tipos de datos definidos. 2. El sistema inserta la nueva opinión en el registro de opiniones asociada al alimento y tienda seleccionados. 3. El sistema notifica al usuario que la opinión ha sido creada exitosamente
Flujo alternativo	<ol style="list-style-type: none"> 1. El sistema puede encontrar un error en el tipo de dato en algún dato del formulario. 2. En ese caso, el sistema notifica al usuario cual es el motivo del error y se le indica modificar lo que está generando el error.

Tabla 32. Caso de uso - Agregar opinión.

Caso de uso: Eliminar opinión

ID	CDU_16
Actores	Usuario registrado
Descripción	El usuario registrado (desde ahora usuario) elimina una opinión, y el sistema le notifica que se ha eliminado con éxito.
Pre-condiciones	1. El usuario debe tener sesión iniciada en el sistema.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

	2. El usuario debe recordar en que alimentos dejó la opinión que desea eliminar.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al catálogo de alimentos. 2. El usuario busca y selecciona un producto del catálogo oprimiendo el botón “Ver detalles” 3. En la tabla de opiniones situada bajo el detalle del alimento, el usuario puede ver si este tiene alguna opinión suya. 4. En caso de encontrar la opinión que desea eliminar entonces oprime el botón “Eliminar” en la fila en que esta se encuentre. 5. El sistema solicita la confirmación de la acción Eliminar para evitar la eliminación por equivocación. 6. El usuario confirma la eliminación de la opinión.
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema elimina la opinión seleccionada. 2. El sistema notifica al usuario que la eliminación de la opinión se ha realizado exitosamente. 3. El sistema recarga la tabla con las opiniones del alimento.
Flujo alternativo	El hecho de que el sistema no tenga opiniones registradas de un usuario en particular, impide la realización de este caso de uso.

Tabla 33. Caso de uso - Eliminar una opinión.

Caso de uso: Crear cotización (solo el nombre)

ID	CDU_17
Actores	Usuario registrado
Descripción	El usuario registrado (desde ahora: usuario) crea una nueva cotización en el sistema. Luego, el sistema notifica que la información se ha ingresado con éxito.
Pre-condiciones	1. El usuario debe tener sesión iniciada en el sistema.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al menú “Mis cotizaciones” 2. El usuario oprime el botón “Crear nueva cotización” 3. El sistema abre un formulario para que el usuario asigne un nombre a la cotización. 4. El usuario ingresa el nombre de la cotización y oprime el botón “Agregar cotización”. 5. El sistema verifica que el nombre de la cotización cumpla con el tipo de dato definido. 6. El sistema registra la nueva cotización en el registro de cotizaciones.
Post-condiciones	1. El sistema notifica al usuario la creación exitosa de la cotización
Flujo alternativo	1. El sistema puede encontrar un error en el tipo de dato en el nombre de la cotización. En ese caso, el sistema notifica al usuario cual es el motivo del error y se le indica modificar lo que ha ingresado en el formulario.

Tabla 34. Caso de uso - Crear cotización.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Caso de uso: Modificar cotización (solo el nombre)

ID	CDU_18
Actores	Usuario registrado
Descripción	El usuario registrado modifica el nombre de una de sus cotizaciones. Luego, el sistema notifica el éxito de la operación.
Pre-condiciones	1. El usuario debe tener sesión iniciada en el sistema. 2. El usuario debe poseer al menos una cotización en el sistema
Flujo principal	1. El usuario accede al menú “Mis cotizaciones” 2. El usuario oprime el botón “Editar” de la cotización seleccionada 3. El sistema redirige al formulario cargando el nombre actual de la cotización, sobre el cual el usuario puede realizar la modificación 4. El usuario modifica el antiguo nombre y oprime el botón “Guardar”
Post-condiciones	1. El sistema notifica con un mensaje el éxito de la operación 2. El sistema regresa al detalle del alimento recargando la tabla de opiniones de este
Flujo alternativo	El hecho de que el usuario no posea cotizaciones impide la realización de este caso de uso.

Tabla 35. Caso de uso - Modificar cotización.

Caso de uso: Eliminar cotización (junto con los alimentos cotizados)

ID	CDU_19
Actores	Usuario registrado
Descripción	El usuario registrado (desde ahora: usuario) elimina una cotización del sistema. Esta acción elimina el registro de alimentos cotizados. Luego, el sistema notifica que la operación se ha realizado con éxito.
Pre-condiciones	1. El usuario debe tener sesión iniciada en el sistema. 2. El usuario debe poseer al menos una cotización en el sistema
Flujo principal	1. El usuario accede al menú “Mis cotizaciones” 2. El sistema redirige a las cotizaciones que posee el usuario. 3. El usuario busca y oprime el botón “Eliminar” de la cotización que desee eliminar. 4. El sistema solicita la confirmación de la acción Eliminar para evitar la eliminación por equivocación. 5. El usuario confirma la eliminación de la cotización. 6. El sistema elimina la cotización seleccionada. 7. El sistema también elimina el registro de los alimentos cotizados de la cotización seleccionada.
Post-condiciones	1. El sistema notifica el éxito en la eliminación de la cotización. 2. El sistema recarga el registro de cotizaciones del usuario.
Flujo alternativo	El hecho de que el usuario no posea cotizaciones impide la realización de este caso de uso.

Tabla 36. Caso de uso - Eliminar cotización.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

Caso de uso: Agregar alimento a cotización

ID	CDU_20
Actores	Usuario registrado
Descripción	El usuario registrado (desde ahora: usuario) agrega un alimento en una de sus cotizaciones. Tras la operación, el sistema notifica el éxito de esta.
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe tener sesión iniciada en el sistema. 2. Debe existir alimentos en el catálogo.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al catálogo de alimentos. 2. El usuario busca un alimento y selecciona la tienda junto con el precio que desee cotizar. 3. El usuario selecciona la cotización en la que desee agregar el alimento 4. El usuario oprime el botón “Agregar a cotización” 5. El sistema agrega el alimento con el precio de una tienda en la cotización seleccionada
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema notifica con un mensaje el éxito de la operación y mantiene la vista actual.
Flujo alternativo	<ol style="list-style-type: none"> 1. El usuario podría no seleccionar un alimento y/o una tienda con el precio de este, en ese caso el sistema notifica con un aviso que hay un “Error en la selección”.

Tabla 37. Caso de uso - Agregar alimento a cotización.

Caso de uso: Eliminar alimento de cotización

ID	CDU_21
Actores	Usuario registrado
Descripción	El usuario registrado elimina un alimento en una de sus cotizaciones. Tras la operación, el sistema notifica el éxito de esta.
Pre-condiciones	<ol style="list-style-type: none"> 1. El usuario debe tener sesión iniciada en el sistema. 2. Debe existir al menos un alimento en la cotización seleccionada.
Flujo principal	<ol style="list-style-type: none"> 1. El usuario accede al menú “Mis cotizaciones” 2. El usuario busca y selecciona la cotización en la que desee eliminar un alimento 3. El sistema redirige al mini catálogo de alimentos cotizados 4. El usuario busca el alimento que desee eliminar y oprime el botón “Quitar de cotización” 5. El sistema elimina el alimento de la cotización seleccionada
Post-condiciones	<ol style="list-style-type: none"> 1. El sistema notifica con un mensaje el éxito de la operación y mantiene la vista actual.
Flujo alternativo	El hecho de que el usuario no posea alimentos en la cotización seleccionada impide la realización de este caso de uso.

Tabla 38. Caso de uso - Eliminar alimento de cotización.



Anexo 3. Especificación de las pruebas del sistema realizadas

1. Prueba de sistema: Crear una tienda

ID prueba	PS_01	Fecha	19-12-2018	
Caso de uso	CDU_08			
Objetivo	Verificar que el ingreso de una tienda sea correcto.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se crea una tienda con los siguientes datos: {“nombre”: Canito, “Direccion”: Los Puelches 1440, “Ciudad”: Chillán, “Email”: canito@gmail.com, “Telefono”: 422212020}.	El sistema verifica el formulario y de ser correcto muestra un mensaje de éxito y redirige al listado de tiendas.	Mensaje “Tienda creada exitosamente” y redirección al listado de tiendas.	Aprobado
Datos No válidos	Se crea una tienda con los mismos datos anteriores donde el nombre ya fue ingresado.	El sistema verifica el formulario y de ser incorrecto muestra un mensaje detallando el error.	Mensaje “Nombre de tienda ya registrado”	Aprobado

Tabla 39. Prueba de sistema - Crear una tienda.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

2. Prueba de sistema: Modificar una tienda

ID prueba	PS_02	Fecha	19-12-2018	
Caso de uso	CDU_09			
Objetivo	Verificar que la modificación de los datos de una tienda sea correcto.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se modifica la tienda “Canito” con los siguientes datos: {“nombre”: Super Canito, “Email”: supercanito@gmail.com}.	El sistema verifica los nuevos datos del formulario y de ser correctos muestra un mensaje de éxito y dirige al detalle de la tienda.	Mensaje “Tienda creada exitosamente” y redirección al listado de tiendas.	Aprobado
Datos No válidos	Se modifica la tienda “Canito” ingresando caracteres del abecedario en el teléfono de contacto.	El sistema verifica los nuevos datos del formulario y de ser incorrecto muestra un mensaje detallando el error.	Mensaje “Teléfono de contacto inválido (debe poseer solo números)”.	Aprobado

Tabla 40. Prueba de sistema - Modificar una tienda.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

3. Prueba de sistema: Crear un alimento (desde el catálogo de una tienda)

ID prueba	PS_03	Fecha	19-12-2018	
Caso de uso	CDU_05			
Objetivo	Verificar que el ingreso de un alimento sea correcto.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se crea un alimento con los siguientes datos: {"nombre": Doko, "Tipo": Perro, "Grupo etario": Cachorro, "Tamaño raza": Para todas las razas, "Kilos": 3K, "Descripción": Doko cachorro sabor carne de 3 kilos, "Precio": 2000}. Sin imagen.	El sistema verifica el formulario y de ser correcto muestra un mensaje de éxito y redirige al catálogo de la tienda.	Mensaje "Alimento creado exitosamente" y redirección al catálogo de la tienda.	Aprobado
Datos No válidos	Se crea un alimento con los mismos datos anteriores donde el nombre y la marca ya fueron ingresados.	El sistema verifica el nombre y la marca del formulario y de ser incorrectos muestra un mensaje detallando el error.	Mensaje "Nombre y marca de alimento ya registrados"	Aprobado

Tabla 41. Prueba de sistema - Crear un alimento.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

4. Prueba de sistema: Modificar un alimento

ID prueba	PS_04	Fecha	19-12-2018	
Caso de uso	CDU_06			
Objetivo	Verificar que la modificación de los datos de un alimento sea correcta.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se modifica el alimento “Doko” con los siguientes datos: {“nombre”: Doko 3K para cachorros, “Tamaño raza”: Para razas pequeñas}.	El sistema verifica que el formulario y de ser correcto muestra un mensaje de éxito y redirige al detalle del alimento.	Mensaje “Alimento modificado exitosamente” y redirección al detalle del alimento.	Aprobado
Datos No válidos	Se modifica el alimento “Doko para cachorros” con los siguientes datos: {“nombre”: 111111, “marca”: 222222, “precio”: Chillán}.	El sistema verifica el formulario y de ser incorrecto muestra un mensaje por cada error encontrado.	Mensaje “Nombre debe contener letras”, “Marca debe contener letras” y “El precio debe ser numérico”	Aprobado

Tabla 42. Prueba de sistema - Modificar un alimento.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

5. Prueba de sistema: Registrarse

ID prueba	PS_05	Fecha	19-12-2018	
Caso de uso	CDU_04			
Objetivo	Verificar que el registro de un nuevo usuario sea correcto.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se crean dos registros con los siguientes datos: {"correo electrónico": usuario@gmail.com, "contraseña": pantalla12} y {"correo electrónico": usuario2@gmail.com, "contraseña": televisor380}.	El sistema verifica el formulario y de ser correcto muestra un mensaje de éxito y redirige al listado de tiendas.	Mensaje "Registrado exitosamente" y redirección al catálogo de alimentos.	Aprobado
Datos No válidos	Se crean los mismos dos registros, donde los correos ya fueron ingresados.	El sistema verifica el formulario y de ser incorrecto muestra un mensaje detallando el error.	Mensaje "Correo electrónico ya registrado"	Aprobado

Tabla 43. Prueba de sistema - Registrarse.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

6. Prueba de sistema: Iniciar sesión

ID prueba	PS_06	Fecha	19-12-2018	
Caso de uso	CDU_01			
Objetivo	Verificar que el inicio de sesión sea correcto.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se inicia sesión con los dos registros creados en PS_05.	El sistema verifica el formulario y de ser correcto muestra un mensaje de éxito y redirige al catálogo principal.	Mensaje “Inicio de sesión exitoso” y redirección al catálogo principal.	Aprobado
Datos No válidos	Se inicia sesión con dos correos electrónicos que no se encuentran registrados. {usuario200@gmail.com, “contraseña”: pantalla200} y {usuario300@gmail.com, “contraseña”: televisor30}.	El sistema verifica el formulario y de ser incorrecto muestra un mensaje detallando el error.	En ambos casos se muestra el mensaje “Correo electrónico no registrado”	Aprobado

Tabla 44. Prueba de sistema. Iniciar sesión.



Sistema Web para comparar alimentos de mascotas (perros y gatos)

7. Prueba de sistema: Crear una opinión

ID prueba	PS_07	Fecha	19-12-2018	
Caso de uso	CDU_15			
Objetivo	Verificar que el ingreso de una nueva opinión sea correcto.			
Tipo de prueba	Caja negra			
Prueba	Datos de entrada	Resultado esperado	Resultado obtenido	Evaluación
Datos válidos	Se crea una opinión con los siguientes datos: {"Comentario alimento": Buen alimento, muy económico, "Nota alimento": 7, "Tienda": Super Canito, "Comentario tienda": Buena atención, servicio rápido, "Nota tienda": 7}.	El sistema verifica el formulario y de ser correcto muestra un mensaje de éxito y recarga las opiniones del alimento.	Mensaje "Opinión creada exitosamente" y recarga de las opiniones del alimento.	Aprobado
Datos No válidos	Se crea la misma opinión anterior, pero la calificación del alimento es 8, es decir, se encuentra fuera del rango [1,7].	El sistema verifica el formulario y de ser incorrecto muestra un mensaje con el error encontrado.	Mensaje "Error en la nota del alimento (valor entre 1 y 7)"	Aprobado

Tabla 45. Prueba de sistema - Crear una opinión.



Anexo 4. Pauta de evaluación de la prueba de usabilidad

Evaluación práctica de uso de sistema

La siguiente pauta tiene como propósito evaluar la facilidad de uso de la aplicación web. Consiste en desarrollar una serie de tareas que serán mencionadas a continuación y posteriormente, responder una encuesta para evaluar la calidad del software.

Listado de tareas

Tarea	Administrador	N° de usuarios	Tiempo límite
Registrarse		2	45 segundos
Iniciar sesión	1	2	45/30 segundos
Ingresar/Modificar/ Eliminar una tienda	1		45/45/20 segundos
Ingresar/Modificar/ Eliminar un alimento	1		45/45/20 segundos
Ingresar/Eliminar una opinión		2	30/30 segundos
Ingresar/Modificar/ Eliminar una cotización (solo el nombre)		2	20/20/20 segundos
Cerrar sesión	1	2	15 segundos

Tabla 46. Tareas a realizar para la prueba de usabilidad.



Encuesta de evaluación de uso de la aplicación para comparar alimentos de mascotas (perros y gatos)

1. Entiende el vocabulario empleado en el menú, operaciones, vistas y tablas.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
2. Comprende los mensajes de error en los formularios y le resulta fácil corregirlos
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
3. Los formularios son simples, fáciles de entender y por lo tanto sencillos de completar.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
4. La aplicación tiene una respuesta rápida a las acciones que se realizan.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
5. La estructura de la aplicación es simple y de fácil manejo.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
6. El catálogo que muestra los alimentos tiene un diseño adecuado.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
7. Los datos que muestra el catálogo de alimentos contienen información necesaria para la toma de decisiones.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
8. La aplicación no presenta errores al ser utilizada.
 - a) De acuerdo b) Ni de acuerdo ni en desacuerdo c) En desacuerdo
9. El tiempo límite es el adecuado para realizar cada tarea [Marcar tick (✓) o cruz (X)].

Tarea	Logrado			No logrado		
Registrarse						
Iniciar sesión						
Ingresar/Modificar/Eliminar una tienda						
Ingresar/Modificar/Eliminar un alimento						
Ingresar/Eliminar una opinión						
Ingresar/Modificar/Eliminar una cotización						
Cerrar sesión						

Tabla 47. Tiempos límite para realizar prueba de usabilidad.