



Universidad de las Ciencias Informáticas

Facultad 4

Sistema para el control del proyecto extensionista Corazón
Canino.

Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas.

Autor: Dayana Bolaños Banzo

Tutor: Ing. José Emilio Badia Valdés

La Habana, 2022

[FRASE]

“El tiempo vendrá en el que los hombres miren a los asesinos de los animales como hoy miran a los asesinos de los hombres”

Leonardo Da Vinci

[DEDICATORIA]

A ella que nunca ha dejado de estar ahí para mí, ni siquiera luego de tantos malos momentos.

[AGRADECIMIENTOS]

Inicialmente a mis padres por tenerme paciencia en esta travesía que tuvo altos y bajos, por los gatos en mí, por todos los viajes que tuvieron que darse hasta la UCI para llevarme cosas necesarias. A mi tutor José Emilio, por su apoyo y ayuda en esta locura. Profe Emilio, disculpe las veces que le escribía en las noches con dudas de la tesis, es el mejor GRACIASSS. A aquellos profesores que pese a tantos dolores de cabeza siempre se puede contar con ellos, como la profe Yadira, quien, aunque es algo exigente es la mejor de todos en su campo y como profesora de año. A mis amigos y compañeros en los que pude apoyarme siempre que lo necesitaba, y quisiera mencionar a aquellos que han venido arrastrándome por una pierna para poder llegar hasta donde estoy:

- Samira...espero y aparezcas en los agradecimientos de todos los graduados la verdad jajaja, eres genial.
- Jesús y Julio...les ha tocado aguantar mis boberías y lentitud para aprender por bastante tiempo.
- A Yanela que, aunque me dejó en la última vuelta jajaja también estuvo bastante tiempo a mi lado pasando dificultades.
- Milena, fuiste en todo el sentido "la buena influencia" jaja, dudo que me hubiera propuesto terminar rápido la tesis si no hubiera sido por verte trabajar en la tuya. Gracias.
- Daniel, Carlos y Ángel igualmente gracias por su ayuda.

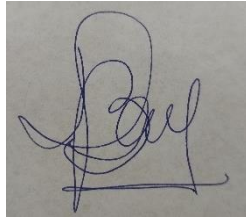
A la cuarentena por la cual muchos logramos pasar nuestros mundiales, a los profesores que estando lejos te respondían sin mucha demora tus dudas con tal de que sacaras la asignatura, a ambos tutores, y a mi universidad, gracias.

Y a ella, que sé, siempre estará ahí para mí.

[DECLARACIÓN DE AUTORÍA]

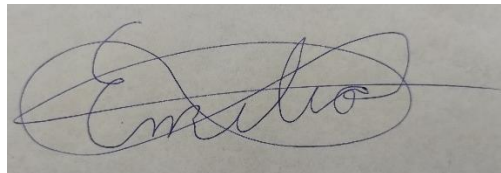
Declaramos ser autores de la presente tesis y reconocemos a la Universidad de Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los 16 días del mes de noviembre del año 2022.



Dayana Bolaños Banzo

(Autor)



Ing. José Emilio Badia Valdés

Tutor

[RESUMEN]

El origen del movimiento de defensa de los animales se sitúa en Inglaterra durante 1876, con la aprobación de la Ley contra la Crueldad de los Animales o *Cruelty to Animals Act*, la cual conjeturó una auténtica legislación sobre el derecho animal, para luego ser reemplazada por la *Protection Animal Act*. En el año 2021, Cuba dio un significativo paso dentro del mundo del cuidado y protección animal, luego de que el 26 de febrero de dicho año se aprobara el Decreto-Ley de Bienestar Animal por el Consejo de Estado, donde se definen las obligaciones y las responsabilidades de los propietarios de animales. A riendas de dicho decreto la Universidad de Ciencias Informáticas decidió dar inicio a un proyecto extensionista cuyo objetivo es mejorar la calidad de vida de los animales callejeros. Este proyecto se encuentra ralentizado debido a que su gestión se realiza totalmente a mano, arriesgándose a la pérdida de información almacenada o su destrucción. Para salvaguardar los datos y agilizar el proceso se propone el desarrollo de un software que gestione todas las tareas relacionadas al proyecto extensionista. Se decidió hacer uso de herramientas como la metodología XP, la cual viene ya con ciertas actividades predefinidas conveniente para un trabajo de diploma. Se hace uso de Python como lenguaje de programación y Django el *framework* a usar, viéndose implementado a través de Microsoft *Visual Studio Code* como editor de código, PostgreSQL para la base de datos y Mozilla Firefox al igual que Google Chrome como principales navegadores.

Palabras clave: animales, aplicación, código, extensionista, proyecto.

[ABSTRACT]

The origin of the movement for the defense of animals is located in England during 1876, with the approval of the Law against Cruelty to Animals or Cruelty to Animals Act, which conjectured an authentic legislation on animal rights, and later it was replaced by the Animal Protection Law. In the year 2021, Cuba took a significant step in the world of animal care and protection, after the Animal Welfare Decree-Law was approved by the Council of State on February 26th in the same year, where obligations and responsibilities of animal owners are declared. As a result of this decret the University of Informatics Sciences had to start an extension project whose objective is to improve the quality of life of stray animals. This project has slowed down because its management is done entirely by hand, risking the loss of stored information or its destruction. To save the data and speed up the process, the development of a software that manages all the tasks related to the extension project is proposed. You can make use of tools such as the XP methodology, which already comes with certain predefined activities convenient for a diploma work. It is used Python as the programming language, Django as the framework, and all this will be implemented through Microsoft Visual Studio Code as the code editor, PostgreSQL for the database and Mozilla Firefox as well as Google Chrome as the main browsers.

Keywords: animals, app, code, extensionist, project.

[INDICE GENERAL]

[FRASE]	I
[DEDICATORIA]	II
[AGRADECIMIENTOS]	III
[DECLARACIÓN DE AUTORÍA]	IV
[RESUMEN]	V
[ABSTRACT].....	VI
[INDICE GENERAL].....	VII
[INDICE DE ILUSTRACIONES]	X
[INDICE DE TABLAS].....	XI
[INTRODUCCIÓN]	1
CAPÍTULO 1: Fundamentación de la investigación, metodología y tecnologías empleadas.	5
Introducción	5
1.1 Estudio del estado del arte:	5
1.1.1 Conceptos y características de los componentes asociados al sistema de control del proyecto extensionista.	5
<input type="checkbox"/> Sistema de información	5
<input type="checkbox"/> Aplicación web	6
<input type="checkbox"/> Código QR	8
1.1.2 Homólogos	10
1.2 Herramientas y tecnologías a utilizar.	13
1.2.1 Metodologías para el desarrollo del software:	13
1.2.2 Bibliotecas	20
1.2.3 Herramientas a utilizar:	23
1.3 Conclusiones parciales	27
CAPÍTULO 2: Descripción de la solución propuesta	28

Introducción	28
2.1 Requisitos funcionales	28
2.2 Requerimientos no funcionales	30
2.3 Historias de usuarios	31
2.4 Estimación de esfuerzo por historias de usuario	41
2.4.1 Desarrollo del plan de iteraciones	42
2.5 Plan de duración de las iteraciones	42
2.5.1 Plan de entregas	43
2.6 Arquitectura de la solución	43
2.6.1 Patrón de diseño a utilizar	44
2.7 Tarjetas de contenidos, responsabilidad y colaboración	47
2.8 Modelo de datos	48
2.9 Conclusiones parciales	49
CAPÍTULO 3: Evaluación de la solución propuesta	50
Introducción	50
3.1 Tareas de ingeniería	50
3.1.1 Iteración I	50
3.1.2 Iteración II	52
3.1.3 Iteración III	55
3.2 Descripción y funcionamiento	56
3.3 Pruebas de aceptación	57
3.3.1 Iteración I	57
3.3.1 Iteración II	60
3.3.1 Iteración III	62
3.4 Pruebas de aplicación web	64
3.5 Pruebas unitarias	65
Conclusiones parciales	71

Conclusiones	72
Recomendaciones	73
Anexos	74
Referencias	115

[INDICE DE ILUSTRACIONES]

Ilustración 1 Aplicación Alerta Animal	10
Ilustración 2 Aplicación BACuba	11
Ilustración 3 Metodologías de desarrollo de software.....	15
Ilustración 4 Patrones de diseño de software (39)	45
Ilustración 5 Fragmento de código del patrón experto.....	46
Ilustración 6 Fragmento de código del patrón Bajo acoplamiento. Fuente: Elaboración propia.	47
Ilustración 7 Modelo de datos	49
Ilustración 8 Vista de la página principal	56
Ilustración 9 Prueba de aplicación #1 Autenticar correcto.....	64
Ilustración 10 Prueba de aplicación #2 Autenticar con datos incorrectos	64
Ilustración 11 Prueba de aplicación #4 Exportar PDF	65
Ilustración 12 Prueba de aplicación #5 Crear desparasitación	65

[INDICE DE TABLAS]

Tabla 1 Comparación. Metodología ágil vs Metodología Tradicional (Awad, 2005).....	15
<i>Tabla 2 Resultado de la selección de la metodología</i>	16
Tabla 3 Cuadro comparativo de XP, RUP, SCRUM (blogspot, 2017)	17
Tabla 4 Tabla comparativa entre VSCode, Sublime y Atom (Codementor, 2016).	24
Tabla 5 Tabla comparativa entre Visual Paradigm, Visio, Draw.io (Capterra, 2022)	25
Tabla 6 Comparativa entre PostgreSQL, MySQL y SQLite (Capterra, 2022)	27
Tabla 7 Historia de usuario #1	31
Tabla 8 Historia de usuario #2	32
Tabla 9 Historia de usuario #3	32
Tabla 10 Historia de usuario #4	33
Tabla 11 Historia de usuario #5	34
Tabla 12 Historia de usuario #6	35
Tabla 13 Historia de usuario #7	35
Tabla 14 Historia de usuario #8	36
Tabla 15 Historia de Usuario #9.....	37
Tabla 16 Historia de usuario #10	38
Tabla 17 Historia de usuario #11	38
Tabla 18 Historia de usuario #12	39
Tabla 19 Historia de usuario #13	39
Tabla 20 Historia de usuario #14	40
Tabla 21 Estimación de esfuerzo por historia de usuario	42
Tabla 22 Plan de duración de las iteraciones.....	42
Tabla 23 Plan de entregas	43
Tabla 24 Tarjeta CRC#1	47
Tabla 25 Tarjeta CRC#2	47
Tabla 26 Tarjeta CRC#3	48
Tabla 27 Tarjeta CRC#4	48
Tabla 28 Tarjeta CRC#5	48
Tabla 29 Tarea de investigación #1	51
Tabla 30 Tarea de investigación #2	51
Tabla 31 Tarea de investigación #3	51
Tabla 32 Tarea de investigación #4	52

Tabla 33 Tarea de investigación #5	52
Tabla 34 Tarea de investigación #7	53
Tabla 35 Tarea de investigación #8	53
Tabla 36 Tarea de investigación #9	54
Tabla 37 Tarea de investigación #10	54
Tabla 38 Tarea de investigación #11	54
Tabla 39 Tarea de investigación #46	56
Tabla 40 Prueba de aceptación #1.1	58
Tabla 41 Prueba de aceptación #1.2	58
Tabla 42 Prueba de aceptación #1.3	58
Tabla 43 Prueba de aceptación #1.4	59
Tabla 44 Prueba de aceptación #1.5	59
Tabla 45 Prueba de aceptación #1.5.2	60
Tabla 46 Prueba de aceptación #2.1	60
Tabla 47 Prueba de aceptación #2.1.2	60
Tabla 48 Prueba de aceptación #2.2	61
Tabla 49 Prueba de aceptación #2.3	61
Tabla 50 Prueba de aceptación #2.4	62
Tabla 51 Prueba de aceptación #2.5	62
Tabla 52 Prueba de aceptación #3	63
Tabla 53 Prueba de aceptación #3.1.2	63
Tabla 54 Prueba de aceptación #4	63

[INTRODUCCIÓN]

El año 1978 pasaría a la historia luego de que en el mundo surgiera la ley de protección animal, tras la proclamación de la primera Declaración Universal de los Derechos de los Animales. La cual fue una compilación creada por la Liga Internacional de los Derechos de los Animales y leída ante la Organización de las Naciones Unidas (UNESCO).

Entre las muchas contribuciones para la lucha animalista, esta iniciativa viene a reconocer que todos los animales tienen igual derecho a la existencia. En consecuencia, la acción del ser humano sobre ellos debería obedecer ciertos límites éticos. A este respecto, el preámbulo de dicha declaración determina lo siguiente (García, 2019):

“b) El hombre, en tanto que especie animal, no puede atribuirse el derecho a exterminar a los otros animales o explotarlos violando su derecho. Tiene la obligación de poner sus conocimientos al servicio de los animales.

c) Todos los animales tienen derecho a la atención, a los cuidados y a la protección del hombre.”

Casi 10 años después de la Declaración Universal de los derechos de los animales, se publicaría el Convenio Europeo para la protección de animales de compañía. Esta iniciativa impulsó las leyes de protección de animales en el mundo, en especial en los países miembros de la Unión Europea (UE) (1). También de forma significativa se tuvo un avance en lo que son las políticas públicas relacionadas a una tenencia responsable, como también el control de la sobrepoblación callejera.

España en febrero de este 2022 aprobó el anteproyecto de ley de protección animal, el cual busca prohibir el maltrato y sacrificio animal y su venta y exposición en tiendas; y exigirá a los que se interesen como nuevos dueños de mascotas a superar una formación de tenencia responsable. Recibiendo luz verde, se anuncia la reforma del Código Penal impulsada por la Dirección General de Protección Animal para que las penas por maltrato sean endurecidas a 24 meses de prisión con resultado de muerte y a 18 meses sin atención veterinaria.

En países latinoamericanos como Argentina, Colombia y México también existen leyes que protegen el bienestar animal. Existieron pocos años de diferencia para aprobar las leyes, pero se ha ido creando consciencia poco a poco en los territorios adyacentes a ellos.

En 1954 el Congreso Nacional de la República Argentina aprobó la Ley 14.346, la cual está en vigencia a día de hoy. Define que se entiende como maltrato y crueldad animal en dicho territorio, prohibiendo además el sacrificio de perros y gatos en la provincia de Buenos Aires (1).

Varios años después, concretamente en 2013, se publicaría en Ciudad de México la primera ley de protección animal de este país. Su texto prevé pena de prisión de hasta cuatro años, así como importantes sanciones económicas, para quienes cometan actos de crueldad en contra todos los animales no humanos, ya sean de compañía, silvestres o domesticados (1).

En 2018 Colombia también aprobaría la ley 1774 sobre el maltrato animal y la protección de animales en su territorio nacional. A partir de su publicación, se empiezan a considerar como delitos los actos de crueldad, maltrato o violencia en contra animales domésticos, silvestres o exóticos vertebrados. Las penas aplicables podrían variar de 12 a 36 meses en prisión (1).

El 26 de febrero del 2021 fue aprobado por el Consejo de Estado el Decreto-Ley de Bienestar Animal, norma jurídica que sustenta la implementación de la Política de Bienestar Animal en Cuba, aprobada el 19 de enero de 2021. Se establecen las responsabilidades de los Órganos del Estado y formas asociativas vinculadas al bienestar de los animales y los principios que rigen esta actividad. Además, la responsabilidad individual que tienen los propietarios, tenedores y poseedores de animales para satisfacer las necesidades básicas de los mismos y las obligaciones de los criadores al respecto (2).

El Decreto-Ley define las obligaciones de los médicos veterinarios para ejercer la asistencia veterinaria, bajo estrictas regulaciones estatales y el control de la autoridad competente con el objetivo de garantizar la prevención, diagnóstico, tratamiento y control de las enfermedades que afectan el bienestar animal y el cumplimiento de los programas de prevención y control de las enfermedades (2).

Se ratifica la necesidad del cumplimiento de las normas de bioseguridad y manejo zootécnico en los establecimientos dedicados a la crianza de animales productivos, según las características de cada especie. En el cuerpo de la norma quedan establecidas las obligaciones que deben cumplir las personas vinculadas a los animales productivos y de trabajo para evitar cualquier forma de maltrato animal y asegurar su bienestar en condiciones normales o en situaciones de desastres (2).

A partir de lo puesto en vigor en el Decreto-Ley 31/2021 “De Bienestar Animal”, la Universidad de Ciencias Informáticas buscando controlar la crecida población canina, que ha estado en constante aumento dentro del campus desde antes de la cuarentena, existiendo actualmente un aproximado de setenta perros sin esterilizar y recibir atención médica; dio inicio al proyecto extensionista “Corazón Canino”. Dicho proyecto surge con el objetivo de hacer cumplir los principios, deberes y reglas existentes con respecto al cuidado y la salud de los animales domésticos en terreno universitario, teniendo en cuenta como principio básico para la creación del proyecto el mencionado decreto-ley que define que *una salud* significa que la salud humana y la sanidad animal son interdependientes y están vinculadas a los ecosistemas en los cuales coexisten.

Dentro de los resultados esperados se encuentran los siguientes:

- .1 Mayor consciencia en la comunidad universitaria sobre el cuidado y protección animal.
- .2 Vacunación de animales, con su correspondiente registro.
- .3 Control del aumento la población canina
- .4 Disminución de la población canina en los terrenos de la universidad a través de adopciones.

Este proyecto tuvo inicio oficial el día 26 de marzo del 2022 con la primera campaña de desparasitación y el próximo domingo a este, el 3 de abril, la primera campaña de esterilización. Los datos obtenidos en dichas campañas, y las que le han proseguido desde entonces, se han realizado a mano por cada historial clínico de los perros y/o gatos atendidos durante dichas campañas. Cada uno de los animales lleva en su historial clínico: el nombre, no. identidad, sexo, peso, raza, color, esterilizado o no, vacunación, desparasitación y enfermedades que ha padecido. Al ser información guardada en historiales clínicos en papel, la posibilidad de extraviarse, mojarse, o necesidad de descartarse; además, el proceso es algo lento debido al cuidado que lleva de no cometer un error durante el mismo que provoque iniciar desde cero.

Teniendo en cuenta la situación problemática que se plantea y se encuentra alrededor del proyecto extensionista “Corazón Canino” se identifica como problema de la investigación ¿Cómo controlar la información generada por el proyecto extensionista corazón canino? Tomando como objeto de estudio: Sistema de control de información.

Buscando darle solución al problema de investigación que se plantea, se define como objetivo general: Desarrollar un sistema que permite el control de la población animal dentro de la universidad. Dentro de este mismo objetivo se toma como campo de acción: sistema de control de información de animales.

Resultado esperado:

- Sistema que permita el control de información de la población animal dentro de la universidad para agilizar el trabajo del proyecto corazón canino.

Método de investigación:

- Método histórico-lógico:

Lo histórico está relacionado con el estudio de la trayectoria real de los fenómenos y acontecimientos en el de cursar de una etapa o período. Lo lógico se ocupa de investigar las leyes generales del funcionamiento y desarrollo del fenómeno, estudia su esencia. Dígase durante el estudio del arte ambos se ponen de manifiesto, junto a la investigación de sus homólogos (3).

- Revisión documental:

Técnica de investigación cualitativa que se encarga de recopilar y seleccionar información a través de la lectura de documentos, libros, revistas, grabaciones, filmaciones, periódicos, bibliografías, etc. (4).

Objetivos específicos:

- Elaborar marco teórico a partir del estudio de diferentes sistemas utilizados para el control de animales.
 - Analizar estructura de los sistemas
 - Analizar funcionalidades más comunes
- Seleccionar y fundamentar las herramientas y lenguajes para el desarrollo de la solución.
- Definir requisitos funcionales y no funcionales.
- Diseñar el sistema para el control del proyecto extensionista corazón canino.
- Implementar y probar el sistema de gestión.

CAPÍTULO 1: Fundamentación de la investigación, metodología y tecnologías empleadas.

Introducción

En el presente capítulo se realizará un análisis de las principales características de los componentes que deben conformar el sistema de control del proyecto extensionista “Corazón Canino”. Además, se profundizará en el campo de bienestar animal en el mundo y la existencia de aplicaciones similares para realizar comparaciones, junto con los conceptos de los componentes que estén asociados a este proyecto.

Además, serán descritas las herramientas y las tecnologías para el desarrollo del sistema de control. Los patrones de diseño y las metodologías de desarrollo de software, por la importancia que tienen para la estructuración de la investigación y el software.

1.1 Estudio del estado del arte:

El estado del arte trata de una sección que compone a los trabajos de investigación a la que también se denomina “estado de la cuestión”. Consiste en la exposición de las teorías que existen acerca del tema que se está trabajando hasta el momento presente. Por lo tanto, hay que investigar qué corrientes de pensamiento, trabados de investigación, construcciones teóricas existen (5).

1.1.1 Conceptos y características de los componentes asociados al sistema de control del proyecto extensionista.

❖ Sistema de información

Se llama sistema de información (SI) a un conjunto de datos y elementos que interaccionan entre sí y que tienen un fin específico que, en general, tiene que ver con satisfacer una necesidad. Por ejemplo: El manejo de la base de datos de una biblioteca (6).

Los sistemas de información deben ser eficientes y de fácil ejecución porque procesan y almacenan una gran cantidad de datos. Estos datos son cargados, de forma automática o manual, en un soporte físico o digital y son usados para producir información que es útil para la ejecución de determinada actividad u objetivo. Toda empresa u organización, sea en el área de la medicina, los negocios, la química o las ciencias sociales, utiliza un sistema de información (6).

Características de un sistema de información (6):

- Se utiliza para recolectar, almacenar e incorporar datos.
- Puede ser un sistema formal, cuando utiliza medios basados en ordenadores o estructuras sólidas para alcanzar la meta o el objetivo, o un sistema informal, cuando utiliza estructuras más elementales o artesanales como, por ejemplo, un lápiz y un papel.
- Almacena información cualitativa (información no numérica) e información cuantitativa (variables numéricas).
- Está formado por datos que son introducidos manual o automáticamente para crear una base de datos.
- Utiliza encuestas, cuestionarios, observaciones de campo, censos o investigaciones para obtener datos.
- Debe ser evaluado y medido para ser actualizado o para corregir posibles errores.
- Debe ser seguro para evitar que la información recopilada sea extraviada o robada.
- Requiere de algún tipo de retroalimentación, esto quiere decir que la información sale del sistema y vuelve a ingresar con mayor elaboración o con más información.

Procesos de un sistema de información:

Todo sistema de este tipo requiere ciertos procesos internos indispensables, los que son (6):

1. Recopilación de datos: Proceso en el cual se reúnen todos los datos disponibles y necesarios. Esto puede hacerse de forma manual o a través de tecnologías como *scanners* o códigos de barra.
2. Alimentación de datos: Proceso en el que se clasifican los datos y se los almacena, esto puede darse en soportes físicos o digitales.
3. Procesamiento de datos: Proceso en el que se relacionan y transforman los datos en información disponible para su uso.
4. Distribución de información: Proceso en el que se hace uso de la información para un fin determinado, como puede ser la toma de decisiones o el planteo de objetivos.

❖ **Aplicación web**

Una aplicación Web es un sitio Web que contiene páginas con contenido sin determinar, parcialmente o en su totalidad. El contenido final de una página se determina sólo cuando el usuario solicita una página del servidor Web. Dado que el contenido final de la página varía de

una petición a otra en función de las acciones del visitante, este tipo de página se denomina página dinámica. Las aplicaciones Web se crean en respuesta a diversas necesidades o problemas. (7).

Usos comunes de las aplicaciones web

Las aplicaciones Web pueden tener numerosos usos tanto para los visitantes como para los ingenieros de desarrollo, entre otros (7):

- Permitir a los usuarios localizar información de forma rápida y sencilla en un sitio Web en el que se almacena gran cantidad de contenido.

Este tipo de aplicación web ofrece a los visitantes la posibilidad de buscar contenido, organizarlo y navegar por él de la manera que estimen oportuna. Entre los ejemplos se incluyen intranets de las empresas, Microsoft MSDN (www.msdn.microsoft.com) y Amazon.com (www.amazon.com).

- Recoger, guardar y analizar datos suministrados por los visitantes de los sitios.

En el pasado, los datos introducidos en los formularios HTML se enviaban como mensajes de correo electrónico a los empleados o a aplicaciones CGI para su procesamiento. Una aplicación Web permite guardar datos de formularios directamente en una base de datos, además de extraer datos y crear informes basados en la Web para su análisis. Ejemplos de ello son las páginas de los bancos en línea, las páginas de tiendas en línea, las encuestas y los formularios con datos suministrados por el usuario.

- Actualizar sitios Web cuyo contenido cambia constantemente.

Una aplicación Web evita al diseñador Web tener que actualizar continuamente el código HTML del sitio. Los proveedores de contenido, como los editores de noticias, proporcionan el contenido a la aplicación Web y ésta actualiza el sitio automáticamente. Entre los ejemplos, se incluyen The Economist (www.economist.com) y Cubadebate (www.cubadebate.cu).

Funcionamiento de una aplicación web

Una aplicación Web es un conjunto de páginas Web estáticas y dinámicas. Una página *Web estática* es aquella que no cambia cuando un usuario la solicita: el servidor Web envía la página al navegador Web solicitante sin modificarla. Por el contrario, el servidor modifica las *páginas Web dinámicas* antes de enviarlas al navegador solicitante. La naturaleza cambiante de este tipo de página es la que le da el nombre de dinámica. Por ejemplo, podría diseñar una página para que mostrara los resultados del programa de salud y dejara cierta información fuera (como el nombre del empleado y sus resultados) para calcularla cuando la página la solicite un empleado en particular (7).

Acceso a una base de datos

Un servidor de aplicaciones le permite trabajar con recursos del lado del servidor, como las bases de datos. Por ejemplo, una página dinámica puede indicar al servidor de aplicaciones que extraiga datos de una base de datos y los inserte en el código HTML de la página (7).

El uso de una base de datos para almacenar contenido permite separar el diseño del sitio Web del contenido que se desea mostrar a los usuarios. En lugar de escribir archivos HTML individuales para cada página, sólo se necesita escribir una página —o plantilla— para los distintos tipos de información que se desea presentar. Posteriormente, podrá cargar contenido en una base de datos y, seguidamente, hacer que el sitio Web recupere el contenido en respuesta a una solicitud del usuario. También puede actualizar la información en un único origen y, posteriormente, implantar ese cambio en todo el sitio Web sin necesidad de editar manualmente cada página (7).

❖ **Código QR**

Un código QR es un código de barras bidimensional cuadrado que puede almacenar los datos codificados. La mayoría del tiempo los datos es un enlace a un sitio web (URL) (8).

¿Cuáles son los diferentes tipos de códigos QR?

Los códigos QR pueden desencadenar varias acciones en el *smartphone* en el que se leen. Direccionar un usuario a un sitio web no es la única acción posible y algunos de ellos son la pena conocer (como guardar una tarjeta de visita o la conexión a redes inalámbricas) (9).

- Sitio web: Mediante la exploración de este tipo de códigos QR, los usuarios serán dirigidos a una página web y descubrirán el contenido disponible.

- Tarjeta comercial: Una tarjeta de contacto con los detalles que ha entrado no se almacenará automáticamente en la lista de contactos del Smartphone. Usted puede introducir sus nombres, dirección, número de teléfono, correo, electrónico, etc.
- Red inalámbrica: Puede crear códigos QR que contienen las credenciales de red inalámbricas. Después de la exploración, los teléfonos inteligentes se conectarán de forma automática.
- Enviar un SMS: Guarda el contenido y el número de teléfono del destinatario de un mensaje SMS. Después de la exploración, sólo tendría que confirmar antes de enviarlo.
- Enviar un correo electrónico: Esto funciona exactamente igual que el SMS de tipo Código QR. Sólo que esta vez, se introduce el contenido del correo electrónico, el tema y los destinatarios para habilitar el envío después de la digitalización.
- Llamar a un número de teléfono: Escriba un número de teléfono cuando se crea el código QR. Al escanear, se propondrán a los usuarios llamar al número de teléfono.
- Añadir un evento al calendario: Después de escanear estos códigos QR, se le preguntará si desea guardar el evento en el calendario de su *smartphone*. Al añadir el evento a su calendario, se le recordará de la fecha correcta.
- Geolocalización: Al crear estos códigos QR, que introduzca las coordenadas de latitud y longitud de un lugar. Mediante la exploración de ellos, los usuarios serán capaces de encontrar la ubicación en su aplicación de geolocalización favorito.
- Texto sin formato: Este es el tipo más simple Código QR. Un texto sin formato se codifica y se mostrará en la pantalla después de escanear. Puedes escribir lo que quieras.

❖ Bienestar animal

El bienestar animal es un tema complejo con múltiples dimensiones científicas, éticas, económicas, culturales, sociales, religiosas y políticas. Se trata de un asunto que suscita un interés creciente en la sociedad civil y constituye una de las prioridades de la Organización mundial de la sanidad animal (OIE). La OIE, a solicitud de sus Países Miembros, es la organización internacional responsable de la elaboración de normas en la materia (10).

Definición:

Según las normas internacionales de la OIE, el bienestar animal designa “el estado físico y mental de un animal en relación con las condiciones en las que vive y muere” (11).

La Organización Mundial de la Salud Animal considera que un animal se encuentra en un estado satisfactorio de bienestar cuando está sano, confortable y bien alimentado, puede expresar su comportamiento innato, y no sufre dolor, miedo o estrés (12)

1.1.2 Homólogos

Toma turno investigar acerca de otras aplicaciones o páginas web que se asemejen al mismo propósito del proyecto en marcha. Esta investigación ayudaría a consolidar las ideas y permitiría estructurar de manera más eficiente la aplicación web con las funcionalidades deseadas. Entre las funcionalidades a priorizar se encuentran el “gestionar ficha del animal”, el cual permitirá tener control de los perros y/o gatos en los terrenos de la universidad.

Dentro de las aplicaciones web que se destacan en este campo se encuentran:

1. **Alerta Animal** (España) (13)

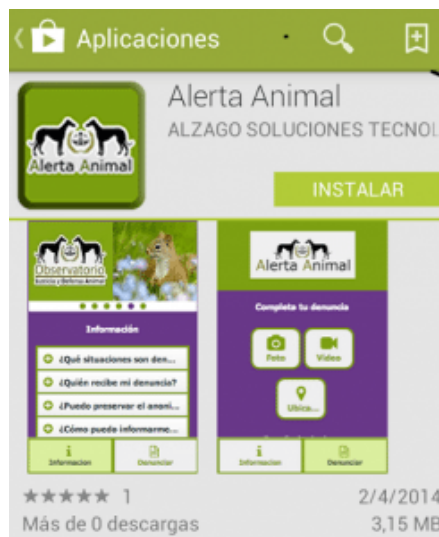


Ilustración 1 Aplicación Alerta Animal

Con esta nueva herramienta, que presenta un diseño funcional e intuitivo, cualquiera que sea testigo de un caso de maltrato animal podrá, de forma gratuita y anónima, ponerlo en conocimiento del equipo jurídico del Observatorio, cuyos expertos valorarán y emprenderán, si procede, las oportunas acciones legales. Permite también adjuntar fotografías o vídeos a la denuncia, indicar la ubicación exacta del suceso a través de coordenadas GPS y escribir un relato de los hechos (13).

En la primera pantalla de la aplicación, se describen los casos susceptibles de denuncia, no sólo los que afectan a animales de compañía, sino a cualquiera que dependa de las personas

para subsistir, además de especies salvajes, tanto en su hábitat como en cautividad. Desde esa misma pantalla, y a través de la pestaña Denunciar, se accede a una segunda ventana en la que se pueden cargar los vídeos o fotos que evidencien los hechos, localizar las coordenadas y describir el maltrato, antes de pulsar en la pestaña Enviar (13).

En una nota de prensa, el Observatorio Justicia y Defensa Animal destaca que «esta útil y novedosa herramienta en la lucha contra el abandono y el maltrato animal en España pretende profundizar en la sensibilización de la sociedad respecto a este grave problema y facilitar su denuncia para que estas situaciones no queden impunes». Desde su creación, el Observatorio ha presentado más de 400 denuncias por casos de maltrato animal en España y ha constatado la falta de iniciativa ciudadana a la hora de poner en su conocimiento «multitud de casos que se dan a diario en nuestro país», añade la nota (13).

Desgraciadamente es una aplicación de hace ya 8 años, la cual tiene origen y su GPS se centra en territorio español. Además, su prioridad es la denuncia de maltrato animal y aunque este es una de las funcionalidades importantes del sistema a desarrollar no es suficiente para darle solución la problemática del proyecto extensionista.

2. **BACuba** (Cuba) (14)

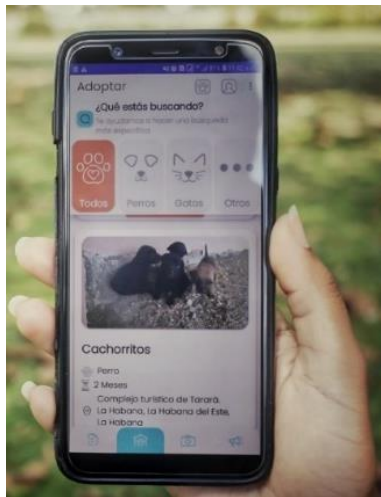


Ilustración 2 Aplicación BACuba

BACuba es la aplicación desarrollada en la Isla para velar, desde dispositivos móviles, por el bienestar animal en el territorio nacional. La herramienta está disponible en la plataforma de descarga gratuita nacional Apklis. así como en la tienda de aplicaciones Androide Play Store (14).

Dividida en secciones, la aplicación permite hacer Reportes y denuncias de maltrato o abandono, contactar tiendas, clínicas o veterinarios en la sección de Cuidados y Adoptar o acoger a todo tipo de mascotas para el bienestar de los animales en Cuba, explicó el sitio web de la emisora Radio Reloj (14).

En la sección de “Reportes” el usuario puede reportar animalitos necesitados en las calles, clasificando a su vez el reporte en función de las condiciones del mismo, ya sea perdido, abandonado, atropellado, etc. Así como también puede reportar un animalito que desee dar en adopción y que nuestros miembros se encarguen de buscarle siempre un adoptante adecuado. Para todo esto a su vez el cliente puede emplear filtros que le permiten ver estos reportes en función de la provincia y del municipio del país que desee (15).

En la sección de “Adoptar” el usuario puede ver todos aquellos animalitos que están en busca de un hogar y si desea adoptar uno puede hacerlo a través de una encuesta preliminar, luego de responder las preguntas estas llegan a nuestro equipo que se encarga de comunicarse con la persona para proseguir con la entrevista y poder adoptar el animalito (15).

En la sección “Cuidados” se muestran contactos de tiendas de mascotas y clínicas veterinarias a lo largo del país con sus respectivas formas de contactar y servicios especializados que brindan. Además, se muestran una serie de artículos de gran interés acerca del correcto cuidado a los animales, así como información pertinente acerca del Decreto-ley de protección animal y las vías de hacer denuncias ante casos de maltrato y abuso (15).

La sección “Noticias” mantiene al usuario informado en tiempo real de todas nuestras campañas, ferias y actividades que realizamos. Y en la sección “Mascotas” el cliente puede registrar sus animalitos y llevar al día sus vacunas y desparasitaciones (15).

Aunque esta aplicación es bastante completa, existe cierto descontrol en el momento de retener y guardar datos, esto debido a que está enfocada al manejo de la información por el público, además de ser en su mayoría meramente informativa, mientras que la aplicación que se desea crear será para gestión y control del proyecto “Corazón Canino”. Siendo esta la principal diferencia en sus funcionalidades, impidiendo que BACuba sea la necesaria para darle solución a la problemática.

1.2 Herramientas y tecnologías a utilizar.

Siempre que se vaya a desarrollar una aplicación web será necesario hacer uso de técnicas como los patrones de diseño, o las metodologías de desarrollo de software, además de decidir qué herramientas o editores de códigos serán usados a lo largo del desarrollo de dicha aplicación. Si bien parece simple y sencillo escoger que tecnologías a utilizar para desarrollar esto no podría estar más erróneo, considerando lo meticuloso que hay que ser cuando se analiza el problema a resolver y lo que se necesita. En el presente epígrafe se determinarán las herramientas a utilizar para darle solución a la problemática existente.

1.2.1 Metodologías para el desarrollo del software:

La metodología de desarrollo de software es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de software informático. Es importante señalar que existen varias, de manera que es una decisión de cada equipo (16).

Trabajar con una metodología es imprescindible por una cuestión de organización. No en vano, los factores tienen que estar ordenados y saber cómo se van a utilizar (16).

Por otra parte, las metodologías también sirven para controlar el desarrollo del trabajo. Esto sirve para minimizar los márgenes de errores y anticiparse a esa situación (16).

Otra ventaja de utilizar una metodología es que te hace ahorrar tiempo y gestionar mejor los recursos disponibles. Esto sucede tanto en metodologías a corto como a largo plazo. Cuando te decantes por un sistema, has de tener en cuenta este factor. Al final, uno de los elementos básicos es optimizar los recursos a tu alcance (16).

Finalmente, hay que hacer referencia al valor añadido. Hay metodologías más costosas, efectivamente, pero que facilitan que el resultado final sea mejor. Cuando tengas que ponderar los pros y los contras, te convendrá saber esta cuestión (16).

Tipos de metodologías para el desarrollo de software:

1. Metodologías tradicionales o rígidas en el desarrollo del software, son aquellas que establecen una disciplina de trabajo sobre el proceso de desarrollo del software, con el propósito de alcanzar un software más eficiente. Se caracterizan por definir y establecer total y rígidamente todos y cada uno de los requisitos al inicio de los proyectos de ingeniería de

software. Estas metodologías son poco flexibles y no permiten realizar cambios (17). Existen distintos tipos como, por ejemplo:

- a. *Waterfall* o En cascada: Se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. El *waterfall model* se utiliza, especialmente, en el desarrollo de software (18).
 - b. Proceso Unificado de Desarrollo (RUP): Es un proceso que puede especializarse para una gran variedad de sistemas de software, en diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización (19).
2. Metodologías ágiles: son enfoques para el desarrollo de productos que se ajustan a los valores y principios descritos en el Manifiesto Ágil para el desarrollo de software. Las metodologías ágiles pretenden ofrecer el producto adecuado, con una entrega incremental y frecuente de pequeños trozos de funcionalidad, a través de pequeños equipos multifuncionales auto organizados, lo que permite la retroalimentación frecuente del cliente y la corrección del curso según sea necesario. Existen distintos tipos como:
- a. Scrum: es también una metodología incremental que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (*planning sprint*), ejecución (*sprint*), reunión diaria (*daily meeting*) y demostración de resultados (*sprint review*). Cada iteración por estas etapas se denomina también *sprint*.
 - b. Programación extrema (XP): es una metodología de desarrollo de software basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya un *feedback* constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, *testing*, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación.



Ilustración 3 Metodologías de desarrollo de software

Con la recopilación de la anterior información se puede pasar a seleccionar la metodología a utilizar para el desarrollo del sistema para el control del proyecto extensionista “Corazón canino”. Para esto se tomará en cuenta la siguiente tabla en la que muestran ciertas características importantes a la hora de escoger la metodología para el desarrollo.

	Metodologías ágiles	Metodologías tradicionales
Acercamiento	Adaptativo	Predictivo
Medición del objetivo	Valor para el negocio	Conformación del plan
Tamaño del proyecto	Pequeño	Largo
Tipo de administración	Descentralizado	Autocrático
Perspectiva de cambios	Adaptable a cambio	Sostenible al cambio
Cultura del equipo	Liderazgo/Colaboración	Comanda/Controla
Documentación	Baja	Pesada
Orientada	Personas	Procesos
Ciclos de desarrollo	Numerosos	Limitados
Dominio de desarrollo	Impredecibles	Predecibles
Planificación inicial	Mínima	Comprensiva
Retorno de la inversión	Temprano en el proyecto	Al final del proyecto
Tamaño del equipo	Pequeño	Grande

Tabla 1 Comparación. Metodología ágil vs Metodología Tradicional (20)

El uso de la tabla 1 permitirá hacer un balance de lo que se necesita con ver los detalles del software a desarrollar.

	Metodologías ágiles	Metodologías tradicionales
Acercamiento	X	
Medición del objetivo	X	
Tamaño del proyecto	X	
Tipo de administración	X	
Perspectiva de cambios	X	
Cultura del equipo	X	
Documentación	X	
Orientada	X	
Ciclos de desarrollo	X	
Dominio de desarrollo		X
Planificación inicial	X	
Retorno de la inversión		X
Tamaño del equipo	X	
Total	11	2

Tabla 2 Resultado de la selección de la metodología

Según los resultados en la tabla 2, el software debe ser desarrollado siguiendo los puntos de las metodologías ágiles debido a que, de 13 parámetros, 11 fueron a favor de este tipo de metodología. De entre las metodologías ágiles se decide usar la XP. Para mayor seguridad en la toma de decisión a continuación una tabla comparativa entre las metodologías XP, SCRUM, RUP.

CRITERIOS	XP	RUP	SCRUM
TIPO DE FRAMEWORK	Basado en la adaptación, mayor flexibilidad, dinámica y funcional.	Análisis diseño implementación y documentación de sistemas orientados a objetos	Gestión y desarrollo de software basado en un proceso iterativo e incremental

TIPO DE REVISIÓN	Se debe integrar como mínimo una vez al día y realizar las pruebas sobre la totalidad del proceso.	En cada fase se realiza una o más iteraciones, perfeccionando así los objetivos. Si no se termina fase no se continua con la otra.	Sus pruebas se realizan al finalizar el proceso enfatizando en la reutilización de los componentes de los programas ya comprobados.
OBJETIVOS	Basado en dar prioridad a trabajos con resultados directos. -Satisfacer al cliente -Trabajo en grupo -Actúa sobre variables	Orientados a objetos que establecen las bases plantillas y ejemplos para todos los aspectos y fases de desarrollo de software.	Crea o redefine modelos existentes -Alto rendimiento ahorro de tiempo -Reducción de costos desarrollo -Mantener la calidad en todo el desarrollo.
TIPO DE DESARROLLO	Liviana y adaptable desarrollado por fases -planificación del proyecto -diseño -codificación -prueba	Proceso iterativo incremental por fases Inicio Elaboración Construcción transmisión	Desarrollo simple que requiere trabajo duro -Control de forma empírica y adaptable a La evaluación del proyecto.
FACILIDAD DE USO	Orientada para pequeños o medianos equipos -Para proyectos de riesgo fecha de entrega -No apto para muchas personas -Posibilidades de cambio.	Dirigido por casos de Uso - Establecimiento temprano de una buena arquitectura -iterativo e incremental -Incrementa el trabajo se divide en mini proyectos.	No se basa en el seguimiento de un plan -Modelo adaptable -Construcción incremental basada en iteraciones -No existe trabajo con diseño o abstracción

Tabla 3Cuadro comparativo de XP, RUP, SCRUM (21)

Metodología Programación Extrema (XP):

Programación Extrema o *Extreme Programming*; Conocida comúnmente por sus siglas en inglés XP. Es una metodología de desarrollo del software, enmarcada dentro de las metodologías ágiles de desarrollo. Constituye la metodología más utilizada dentro del grupo de las ágiles. Su objetivo principal es asegurar la producción de software con buena calidad y cubriendo las necesidades y requerimientos del usuario (22).

Características generales de XP:

Surge como posible solución a los problemas derivados del cambio en los requerimientos; esta metodología ofrece la posibilidad de cambiar los requisitos en cualquier momento de la vida de un proyecto, ya que es adaptable a estos cambios. Se centra en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Este método es típicamente atribuido a Kent Beck, Ron Jeffries y Ward Cunningham. El objetivo de XP son grupos pequeños y medianos de construcción de software en donde los requisitos aún son muy ambiguos, cambian rápidamente o son de alto riesgo. Además, sugiere que el lugar de trabajo sea una sala amplia, si es posible sin divisiones (en el centro los programadores, en la periferia los equipos individuales). XP es un proceso muy orientado a la implementación, en el que se genera poca documentación y en que la funcionalidad exacta del sistema final no se define nunca formal y contractualmente. Es por eso que este método es más aplicable para desarrollos internos (22).

Ciclo de vida ideal de XP:

El ciclo ideal de XP consta de seis fases:

- Exploración.
- Planificación de la Entrega (*Release*)
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

Proceso XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar, basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. El ciclo de desarrollo consiste de manera general en los siguientes pasos (22):

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las interacciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma, el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración (22).

Historias de usuario en XP.

Las historias de usuario son utilizadas por la metodología XP como una técnica para especificar los requisitos del mismo, tanto requisitos no funcionales como funcionales. Se trata de tarjetas en las cuales el usuario describe las características que el sistema debe poseer, de forma muy breve. Las características fundamentales del método son (22):

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas: frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión: Se aconseja escribir el código de la prueba antes de la codificación.
- Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Para que de esta manera el código sea revisado y discutido mientras se escribe, es más importante que la posible pérdida de productividad inmediata.
- Frecuente integración del equipo de programación en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario.

1.2.2 Bibliotecas

El desarrollo de una aplicación web cuenta con disímiles de lenguajes de programación, pero contando con las características que presenta el sistema a desarrollar.

Parámetro	Python	PHP
Aprendizaje	Python es mejor que PHP en proyectos a largo plazo.	PHP tiene una curva de aprendizaje baja, es fácil comenzar con PHP.
Estructura	Comparar con PHP, Python tiene un menor número de <i>Frameworks</i> . Los más populares son Django, <i>Flask</i> .	PHP tiene una gran cantidad de marcos. Los más populares son Laravel, Slim.
Sintaxis	La sintaxis es fácil de recordar, casi similar al lenguaje humano.	La sintaxis es poco común en comparación con Python, tiene una amplia gama de convenciones de nomenclatura.
Características clave	Menos número de línea de código, implementación rápida y escritura dinámica.	Código abierto y fácil implementación.
Tipo de idioma	Es un lenguaje de programación de propósito general.	Es un lenguaje de programación de desarrollo web.
Campo Popular	Tarea de aprendizaje automático, ciencia de datos, inteligencia artificial y automatización.	Elección del lenguaje en el desarrollo web.
Mantener	Comparar con PHP es más fácil de mantener.	Un poco difícil de mantener.
Ritmo de popularidad	Después de 2016, la popularidad de Python está aumentando rápidamente.	Al mismo tiempo, PHP pierde su popularidad en el desbordamiento de pila.
Velocidad	Los programas de Python son relativamente más lentos que PHP.	Las nuevas versiones de PHP han demostrado ser alrededor de 3 veces más rápidas que los programas de Python.
Depuración	La depuración en Python es más rápida que en PHP.	La depuración en PHP es mucho más lenta.
Integración de base de datos	Pobre integración de base de datos en comparación con PHP.	Proporciona a los usuarios acceso a más de 20 bases de datos.

Tabla 4 Tabla comparativa entre los lenguajes de Python y PHP (23).

Se decide tomar como lenguaje de programación Python.

Python es un tipo de lenguaje de código abierto pensado para desarrollar webs y aplicaciones informáticas. Fue creado a principios de los 90 por Guido van Rossum y, tres décadas después, sigue siendo el lenguaje más puntero. Ha sabido readaptarse a las novedades que fueron llegando y ha terminado por consolidarse como opción estrella de grandes compañías de éxito (24).

Elegir Python web te asegura no encasillarte en un único tipo de desarrollo web. Se trata de un código abierto cuyas posibilidades de creación son infinitas, así como sus aplicaciones directas (24).

Ventajas de elegir Python Web:

Estas ventajas varían en el momento de sentarse a programar webs, entre ellas están (24):

- Es lenguaje *open source*, lo que quiere decir que es de software libre. Podrás usarlo gratuitamente en todo momento.
- Es multiplataforma, válido para todo tipo de sistemas y dispositivos.
- Es entendible para todos los programadores. La forma de escribir código Python es ordenada y limpia, por lo que cualquiera sabrá entender y trabajar sobre tu estructura creada.
- Tiene un estilo flexible. Con Python web podrás crear multitud de elementos, desde listas hasta funciones más complejas.
- Estilo sano de programación. Python web está pensado para que funcione siguiendo unas reglas concretas, de forma que todo el mundo siga las mismas directrices.

Los términos de *frontend* y *backend*, tienen características únicas y distintivas, actúan como unidades que, por separado, interactúan entre sí para asegurar la funcionalidad de una aplicación web.

¿Qué es *frontend*?

El *frontend* es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos. Es esa parte de la página con la que interaccionan los usuarios de la misma, es todo el código que se ejecuta en el navegador

de un usuario, al que se le denomina una aplicación cliente, es decir, todo lo que el visitante ve y experimenta de forma directa (25).

¿Qué es el *backend*?

Mientras que el *frontend* es la capa de programación ejecutada en el navegador del usuario, el *backend* procesa la información que alimentará el *frontend* de datos. Es la capa de acceso a los datos, ya sea de un software o de un dispositivo en general, es la lógica tecnológica que hace que una página web funcione, lo que queda oculto a ojos del visitante. El *backend* de una solución, determina qué tan bien se ejecutará la aplicación y qué experiencia, positiva o negativa, obtendrá el usuario de su uso. El flujo de trabajo de un *backend* consiste en darle funciones a un sitio; mientras que el *frontend* hace un sitio estático, el *backend* después le da funciones y adapta el sistema programado a ese sitio web. (25).

Python web: Django.

Django es un framework de desarrollo web de código abierto, escrito en Python, que cumple en cierta medida el paradigma del Modelo Vista Controlador. La distribución principal de Django también aglutina aplicaciones que proporcionan un sistema de comentarios, herramientas para syndicar contenido vía RSS y/o *Atom*, "páginas planas" que permiten gestionar páginas de contenido sin necesidad de escribir controladores o vistas para esas páginas, y un sistema de redirección de URLs. (26)

Otras de las características de Django son (26):

- Un ORM (*Object Relational Mapping*)
- Aplicaciones "enchufables" que pueden instalarse en cualquier página gestionada con Django.
- Una API de base de datos robusta.
- Un sistema incorporado de "vistas genéricas" que ahorra tener que escribir la lógica de ciertas tareas comunes.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URLs basado en expresiones regulares.
- Un sistema "*middleware*" para desarrollar características adicionales; por ejemplo, la distribución principal de Django incluye componentes *middleware* que proporcionan cacheo, compresión de la salida, normalización de URLs, protección CSRF y soporte de sesiones.

- Soporte de internacionalización, incluyendo traducciones incorporadas de la interfaz de administración.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

Django template es el propio sistema de plantillas de Django. Hasta Django 1.8 era la única opción integrada disponible. Es una buena biblioteca de plantillas a pesar de que es bastante obstinada y tiene algunas idiosincrasias. Si no tiene una razón apremiante para elegir otro *backend*, debe usar DTL (*Django Template Language*), especialmente si está escribiendo una aplicación conectable y tiene la intención de distribuir plantillas. Las aplicaciones de contribución de Django que incluyen plantillas, como *django.contrib.admin*, usan el DTL (27).

1.2.3 Herramientas a utilizar:

Todo desarrollo de software requiere herramientas intermediarias que preferentemente faciliten el proceso. Los programas podrán variar según el trabajo que se vaya a realizar, ejemplo de esto la creación de diagramas de clases.

Para el desarrollo del sistema de el control del proyecto extensionista “Corazón canino” se hará uso de los siguientes programas:

- Editor de código: *Microsoft Visual Studio Code*

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, *TypeScript* y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) (28).

	Visual Studio Code	Atom	Sublime
URL	https://code.visualstudio.com/	https://atom.io/	https://www.sublimetext.com/
Costo	<ul style="list-style-type: none"> • Gratis 	<ul style="list-style-type: none"> • Gratis (Licencia MIT) 	<ul style="list-style-type: none"> • tarifa de licencia de \$ 70 con prueba gratuita
Desarrollador	<ul style="list-style-type: none"> • Microsoft 	<ul style="list-style-type: none"> • GitHub 	<ul style="list-style-type: none"> • Jon Skinner, Will Bond y Cesar Mariniti.
Plataformas	<ul style="list-style-type: none"> • OSX, Windows, Linux 	<ul style="list-style-type: none"> • OSX, Windows, Linux 	<ul style="list-style-type: none"> • OSX, Windows, Linux

Edición y flujo	<ul style="list-style-type: none"> Mucho más rápido y sin retrasos reales que Atom. Función de depuración excelente. Conveniente integración de Git. Integrada la herramienta <i>IntelliSense</i> para ayudar a autocomplete. 	<ul style="list-style-type: none"> Perfecta integración de Git Presenta el soporte de arrastrar/soltar archivos/carpetas en la vista de árbol. 	<ul style="list-style-type: none"> No presenta el soporte de arrastrar/soltar archivos/carpetas en la vista de árbol. Rapidez y facilidad para buscar un archivo y entre archivos. No existe complemento Git decente para realizar diferencias y cambios de escenarios.
Personalización	<ul style="list-style-type: none"> Presencia de funciones de personalización. 	<ul style="list-style-type: none"> Página de documentos sobre cómo anular estilos Excelente capacidad de anular la configuración según el tipo de archivo. 	<ul style="list-style-type: none"> Disponible “guardar en foco perdido”; “recordar espacios en blanco”, entre otros aspectos. Archivos de configuración son JSON simples.
Actuación	<ul style="list-style-type: none"> Rápido. 	<ul style="list-style-type: none"> Un poco lento. 	<ul style="list-style-type: none"> Fluido y rápido.

Tabla 5 Tabla comparativa entre VSCode, Sublime y Atom (29).

- Herramienta para diseño de prototipos: *Visual Paradigm*

Visual Paradigm ayuda a los equipos de desarrollo de softwares a capturar los requisitos correctos y transformarlos en diseños precisos, lo que ayuda a los desarrolladores a crear el software adecuado según los requisitos. Crea diagramas, *wireframes*, editor para arrastrar y soltar, y es una herramienta de colaboración.

	Visual Paradigm	Visio	Draw.io
Ideal para	Visual Paradigm ayuda a los equipos de desarrollo de softwares a capturar los requisitos correctos y transformarlos en	Microsoft Visio es una solución de diagramación con una robusta biblioteca de plantillas y formas, capacidades de conectividad de datos y herramientas de	<ul style="list-style-type: none"> Desarrolladores TI Administradores de redes Diseñadores, analistas de procesos

	diseños precisos, lo que ayuda a los desarrolladores a crear el software adecuado según los requisitos.	colaboración integradas para capacitar a los trabajadores remotos y equipos en cualquier lugar.	
Funcionalidades	<ul style="list-style-type: none"> • Creación de diagramas • Diseño y modelado del proceso • Gestión de comentarios • Herramientas de colaboración • Historial de revisión • Modelos 	<ul style="list-style-type: none"> • Creación de diagramas • Diseño y modelado del proceso • Gestión de comentarios • Gestión de visitantes • Gestión del diseño • Gestión del espacio • Herramientas de colaboración • Historial de revisión • Importación y exportación de datos • Modelos 	<ul style="list-style-type: none"> • Arrastrar y soltar de imágenes • Creación de diagramas • Creación de organigramas • Diseño y modelado del proceso • Herramientas de colaboración • Modelos
Plataforma	En la nube/para instalar: <ul style="list-style-type: none"> • Cloud, Windows, Mac 	En la nube/para instalar: <ul style="list-style-type: none"> • Cloud, Windows, Mac • Móvil: iOS, Android 	<ul style="list-style-type: none"> • En la nube/para instalar: • Cloud, Windows, Mac
Formación	<ul style="list-style-type: none"> • Documentación • En directo en línea 	<ul style="list-style-type: none"> • Seminarios web • Documentación 	<ul style="list-style-type: none"> • Seminarios web • Documentación • Vídeos
Asistencia	<ul style="list-style-type: none"> • Chat 	<ul style="list-style-type: none"> • E-mail/Help Desk • Preguntas frecuentes/foro • Base de conocimientos • Asistencia telefónica • Chat 	<ul style="list-style-type: none"> • E-mail/Help Desk • Preguntas frecuentes/foro • Base de conocimientos • Chat

Tabla 6 Tabla comparativa entre Visual Paradigm, Visio, Draw.io (30)

- Navegador:
Mozilla Firefox:

Es un navegador web desarrollado por la compañía Corporación Mozilla. Creado en el año 2002 por David Hyatt y Joe Hewitt, se trata de un navegador multiplataforma (teléfonos móviles, ordenadores, tablets y otros dispositivos) y disponible para Windows, Linux, Android, iOS y macOS. Desde su lanzamiento, este navegador ha ido adquiriendo popularidad hasta convertirse en el segundo navegador web más utilizado del mundo con más de 500 millones de usuarios (31).

Google Chrome:

Es el navegador web creado por la compañía Google INC. Considerado como el navegador más rápido del mundo, en poco tiempo que tiene desarrollado Google Chrome cumple con sus objetivos principales, rápido, seguro, práctico, estable y con un sentido minimalista único, que le brinda al usuario la mayor comodidad a la hora de navegar por la web. (32).

- Base de datos: PostgreSQL

PostgreSQL es un gestor que trabaja con bases de datos relacionales y que está orientado a objetos. Se trata de un programa de código abierto u *open source*, es decir, no está bajo el control de ninguna compañía particular, sino que cuenta con una comunidad de desarrolladores que trabajan en mejorar el programa de forma desinteresada (33).

Características principales de PostgreSQL

Las principales características de PostgreSQL se resumen en las siguientes (33):

- Consiste en un gestor de bases de datos relacionales, soporta diferentes tipos de datos y está orientado a objetos
- Es de código abierto, es decir, cuentan con una comunidad de desarrolladores que implementan mejoras o resuelven errores de forma altruista.
- Emplea un lenguaje SQL basado en el estándar ISO/IEC. Por ello, permite importar consultas y código de scripts de otras bases de datos.
- Cumple con el modelo ACID, es decir, los datos almacenados cuentan con las características de Atomicidad, Consistencia, Aislamiento y Durabilidad.
- Ofrece opciones que otras bases de datos no tienen, como la posibilidad de crear tablas heredadas, esquemas o *triggers*.
- No solo permite crear procedimientos en lenguaje SQL, sino que también permite utilizar otros lenguajes de programación como Python o Pearl.
- Se pueden añadir nuevas funciones o extensiones desarrolladas por terceros, o incluso por uno mismo.
- También se caracteriza por ofrecer una gran escalabilidad vertical.

	PostgreSQL	MySQL	SQLite
Funcionalidades	<ul style="list-style-type: none"> • Controles o permisos de acceso • Conversión de base de datos • Copia de seguridad y recuperación • Gestión de almacenamiento de datos • Gestión de diccionario de datos • Permite la programación de múltiples idiomas • Replicado de datos 	<ul style="list-style-type: none"> • Controles o permisos de acceso • Copia de seguridad y recuperación • Gestión de almacenamiento de datos • Replicado de datos 	<ul style="list-style-type: none"> • Conversión de base de datos • Copia de seguridad y recuperación • Gestión de almacenamiento de datos • Permite la programación de múltiples idiomas • Replicado de datos
Plataforma	Cloud, SaaS, Web, Mac (desktop), Windows (desktop)	Cloud, SaaS, Web, Mac (desktop), Windows (desktop)	Cloud, SaaS, Web, Mac (desktop), Windows (desktop), Linux (desktop), Windows (local), Linux (local)

Tabla 7 Comparativa entre PostgreSQL, MySQL y SQLite (30)

1.3 Conclusiones parciales

- Teniendo en cuenta la no existencia de una aplicación que satisfagan las necesidades del proyecto extensionista “Corazón Canino” es necesario que se desarrolle una aplicación web la cual cumpla con todo lo requerido por el proyecto. Se incluyen las demandas de maltrato, las fichas animales junto al lector y generador de código QR, además del resto de funcionalidades solicitadas.
- Se hará uso del editor de código *Visual Studio Code*, siendo Mozilla Firefox o Google Chrome aquellos que se encargarán de visualizar la aplicación web.
- Python como lenguaje de programación y junto a él, Django como *framework* de desarrollo.
- La metodología de desarrollo ágil es la más indicada para ser utilizada, esto ayudará a que el proyecto sea terminado de forma más rápida para luego poder darle uso, específicamente la metodología ágil XP.

CAPÍTULO 2: Descripción de la solución propuesta

Introducción

A lo largo del presente capítulo se busca obtener las características principales haciendo uso de los requisitos funcionales (RF) y los requisitos no funcionales (RnF). Estos describen los atributos de calidad y funcionalidades que debe presentar el software a desarrollar.

Anteriormente en el capítulo 1, se determinó que como metodología a usar la XP era perfecta como guía para el desarrollo del software; lo que conlleva a la utilización de las “historias de usuarios” (HU) como principal herramienta para la descripción detallada de los requisitos funcionales y la confección del plan de iteraciones. A través del uso de dicho plan de iteraciones se dará un estimado del tiempo requerido para culminar el desarrollo del sistema de control, y apoyándose en el uso de los patrones de diseño, se facilitará luego el diseño de las fichas técnicas de los animales.

Teniendo en cuenta la situación problemática planteada anteriormente y el problema a resolver se le ha buscado como propuesta de solución una aplicación web en la cual el usuario pueda acceder al historial clínico de los animales con solo escanear el código QR que estos tendrán como identificador en sus collares.

2.1 Requisitos funcionales.

La definición de un sistema o sus componentes es aquello que se entiende por RF en la ingeniería de software. Estos describen aquella función que un software a desarrollar debe realizar, ya sea manipulación de datos, cálculos, procesos de negocios y otros. Son de ayuda en el momento de capturar los comportamientos planificados para un sistema, esto podría ser expresado como un servicio, tarea o función que el software a desarrollar debe realizar (34).

Entre los requisitos funcionales a desarrollar planteados por el usuario están:

- RF 1: Crear ficha de animal.
- RF 2: Modificar ficha de animal.
- RF 3: Eliminar ficha de animal.
- RF 4: Mostrar el animal.
- RF 5: Generar código QR de la ficha del animal.
- RF 6: Crear enfermedad animal.
- RF 7: Modificar enfermedad animal.

- RF 8: Eliminar enfermedad animal.
- RF 9: Mostrar enfermedad.
- RF 10: Crear vacunas.
- RF 11: Modificar vacuna.
- RF 12: Eliminar vacuna.
- RF 13: Filtrar vacuna
- RF 14: Crear medicamento.
- RF 15: Modificar medicamento.
- RF 16: Eliminar medicamento.
- RF 17: Mostrar medicamento.
- RF 18: Crear denuncia
- RF 19: Modificar denuncia
- RF 20: Eliminar denuncia
- RF 21: Mostrar denuncia
- RF 22: Filtrar denuncia
- RF 23: Crear visitas
- RF 24: Modificar visitas
- RF 25: Eliminar visitas
- RF 26: Mostrar visitas
- RF 27: Filtrar visitas
- RF 28: Crear eventos
- RF 29: Modificar eventos
- RF 30: Eliminar eventos
- RF 31: Mostrar eventos
- RF 32: Filtrar eventos
- RF 33: Crear información
- RF 34: Modificar información
- RF 35: Eliminar información
- RF 36: Mostrar información
- RF 37: Crear asociados
- RF 38: Modificar asociados
- RF 39: Eliminar asociados
- RF 40: Mostrar Asociados

- RF 41: Crear contacto
- RF 42: Modificar contactos
- RF 43: Eliminar contactos
- RF 44: Mostrar contacto
- RF 45: Autenticar usuario
- RF 46: Generar reportes integrales
- RF 47: Generar graficas
- RF 48: Crear desparasitación
- RF 49: Modificar desparasitación
- RF 50: Eliminar desparasitación
- RF 51: Filtrar desparasitación
- RF 52: Cerrar sesión
- RF 53: Iniciar sesión como invitado

2.2 Requerimientos no funcionales.

Los requisitos no funcionales (RnF) definen los atributos de calidad que el software a desarrollar debe cumplir. Su esencialidad asegura la efectividad y eficiencia del software en plano general. En caso de cometer un error en el momento de seleccionar los RnF de un sistema, podría conllevar al no cumplimiento de todas las necesidades del usuario (34). Para este trabajo se definieron los siguientes RnF:

Usabilidad:

- RNF 1: El sistema debe ser intuitivo y de fácil manipulación para el usuario.
- RNF 2: La interfaz deberá mostrarse ordenada para mejor manipulación por el usuario.

Software:

- RNF 3: Debe ser capaz de ejecutarse en un navegador web (Chrome)

Apariencia o interfaz externa:

- RNF 4: La interfaz del visor debe estar compuesta principalmente por los colores: Blanco, Azul.
- RNF 5: El visor debe ocupar el 100% del navegador, quedando centrado el modelo.

Rendimiento:

- RNF 6: El tiempo de respuesta de la solicitud del usuario que sea menor a un segundo.

Seguridad:

- RNF 7: La longitud de la clave de seguridad del administrador deberá ser de mínimo 8 caracteres y debe incluir símbolos, al menos una mayúscula y al menos un número.

2.3 Historias de usuarios.

Las historias de usuario son parte de un enfoque ágil que ayuda a cambiar el enfoque de escribir sobre los requisitos a hablar sobre ellos. Todas las historias de usuario ágiles incluyen una oración escrita o dos y, más importante aún, una serie de conversaciones sobre la funcionalidad deseada (35).

Los puntos estimados están vistos por semanas, de modo que 1=1 semana, 0.25 = 2 días (aproximadamente)

Tabla 8 Historia de usuario #1

Historia de usuario	
Número: 1	Nombre: Crear ficha de animal
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá crear fichas médicas de los animales comunitarios. Estas fichas tendrán como información a agregar: <u>nombre, número de identidad, sexo, peso, raza, color, esterilizado o no, vacunación, desparasitación y enfermedades que ha padecido.</u>	
Observaciones:	
<ul style="list-style-type: none"> • El crear deberá ser sencillo, un botón flotante estático que al presionarlo acceda al formulario a llenar y al concluir de llenar la información se presione un botón para guardar la información o cancelar. 	
Interfaz:	

Tabla 9 Historia de usuario #2

Historia de usuario	
Número: 2	Nombre: Modificar ficha de animal
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá modificar las fichas médicas de los animales comunitarios. Estas fichas tendrán como información a modificar: <u>nombre, número de identidad, sexo, peso, raza, color, esterilizado o no, vacunación, desparasitación y enfermedades que ha padecido.</u>	
Observaciones:	
<ul style="list-style-type: none"> Para modificar presionar un botón en una columna de “Opciones” que permita acceder al formulario ya lleno y en este poder hacer las modificaciones. 	
Interfaz:	

Tabla 10 Historia de usuario #3

Historia de usuario	
Número: 3	Nombre: Eliminar ficha de animal
Usuario: Administrador	

Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá eliminar las fichas médicas de los animales comunitarios localizadas en el “Listado de fichas”	
Observaciones: <ul style="list-style-type: none"> Para eliminar presionar el botón correspondiente en la columna de “Opciones” y que muestre un mensaje de aviso y seguridad de eliminar. 	
Interfaz: 	

Tabla 11 Historia de usuario #4

Historia de usuario	
Número: 4	Nombre: Mostrar el animal
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá mostrar en un modal toda la información de la ficha animal seleccionada.	
Observaciones: <ul style="list-style-type: none"> En la columna “Opciones” se encontrará el botón correspondiente para poder mostrar la ficha animal. Y luego deberá aparecer en una ventana flotante todos los datos que le corresponden a la ficha elegida. 	
Interfaz:	

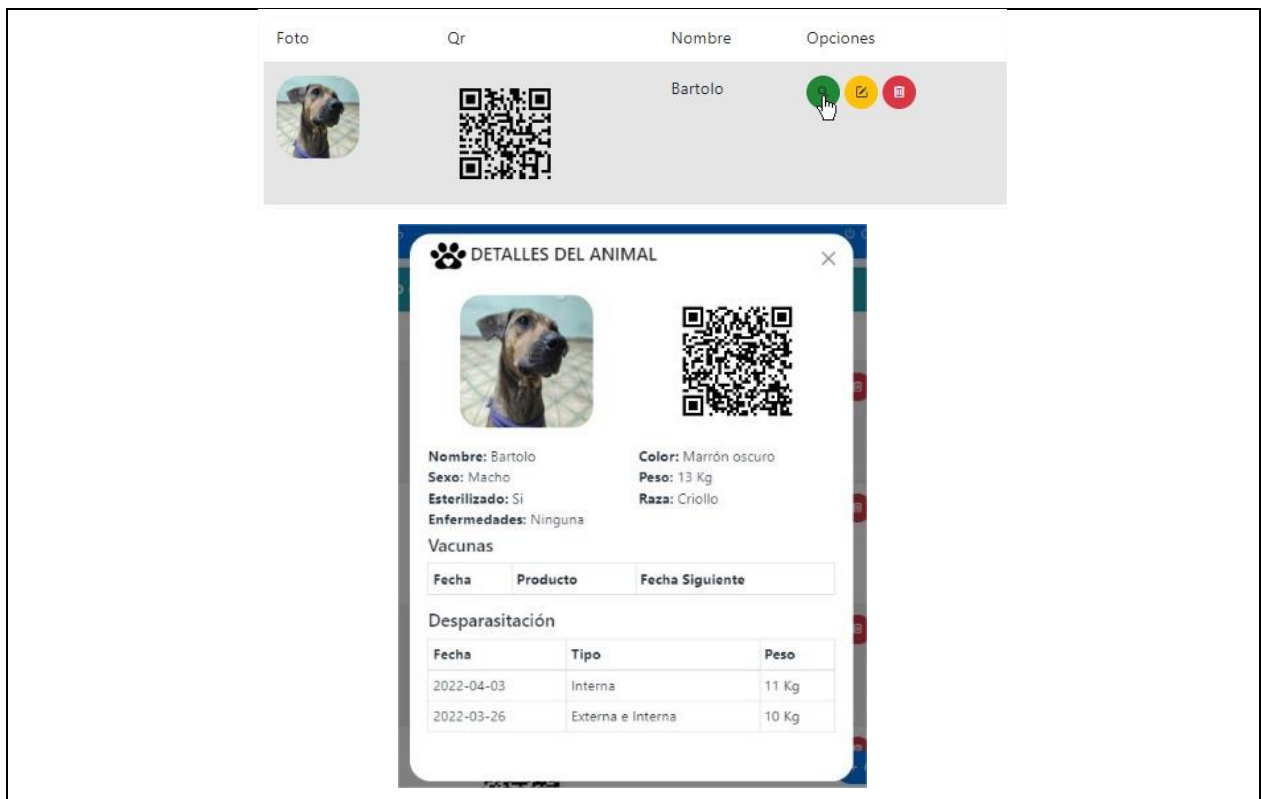


Tabla 12 Historia de usuario #5

Historia de usuario	
Número: 5	Nombre: Generar código QR de la ficha animal
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.4	Iteración asignada: 1
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá generar un código QR luego de guardar la información de una nueva ficha animal.	
Observaciones:	
<ul style="list-style-type: none"> EL QR será generado automáticamente al guardar sin necesidad de presionar ningún botón en específico. 	
Interfaz:	

Tabla 13 Historia de usuario #6


Historia de usuario	
Número: 6	Nombre: Crear denuncia
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
<p>Descripción:</p> <p>El sistema le dará acceso al usuario a información relacionada a las denuncias de maltrato animal que se pueden realizar, contactos, y datos importantes sobre el tema. Podrá crear una denuncia responsable rellorando los siguientes campos: tipo de denuncia, descripción, ubicación, características del animal y fotos de evidencia.</p>	
<p>Interfaz:</p>	

Tabla 14 Historia de usuario #7

Historia de usuario

Número: 7	Nombre: Modificar denuncia
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá modificar la información de las denuncias al presionar el botón correspondiente a “Modificar” en la columna de “Opciones” y luego de modificarla podrá guardarla.	
Interfaz: 	

Tabla 15 Historia de usuario #8

Historia de usuario	
Número: 8	Nombre: Eliminar denuncia
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá eliminar la denuncia al presionar un botón colocado en la columna de “Opciones”. Deberá aparecer un mensaje de seguridad para eliminar.	
Interfaz: 	

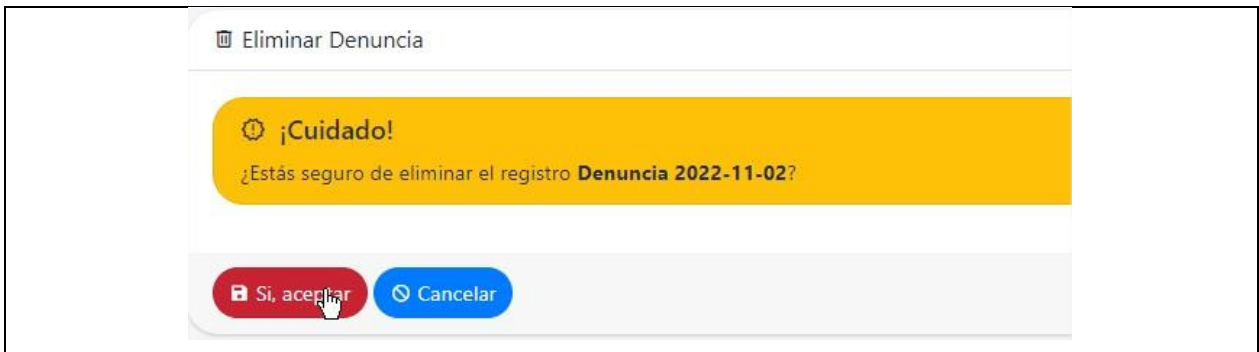


Tabla 16 Historia de Usuario #9

Historia de usuario	
Número: 9	Nombre: Mostrar denuncia
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
<p>Descripción:</p> <p>El sistema permitirá ver los detalles de las denuncias al presionar un botón que le corresponda. Y deberá mostrar dichos detalles en otra ventana.</p>	
<p>Interfaz:</p>	

Tabla 17 Historia de usuario #10

Historia de usuario	
Número: 10	Nombre: Filtrar denuncias
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.1	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema deberá permitir filtrar las denuncias por fechas para mayor facilidad en el momento de encontrar la deseada entre intervalos de fechas.	
Interfaz: 	

Tabla 18 Historia de usuario #11

Historia de usuario	
Número: 11	Nombre: Generar gráficas
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.25	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
Descripción: Crear interfaces gráficas donde se reflejen los datos (<u>Porcentaje de animales por sexo</u> , <u>Porcentaje de animales por esterilizados</u> , <u>Porcentaje de animales con vacuna</u> y <u>Porcentaje de animales con desparasitación</u>) actuales.	
Interfaz:	

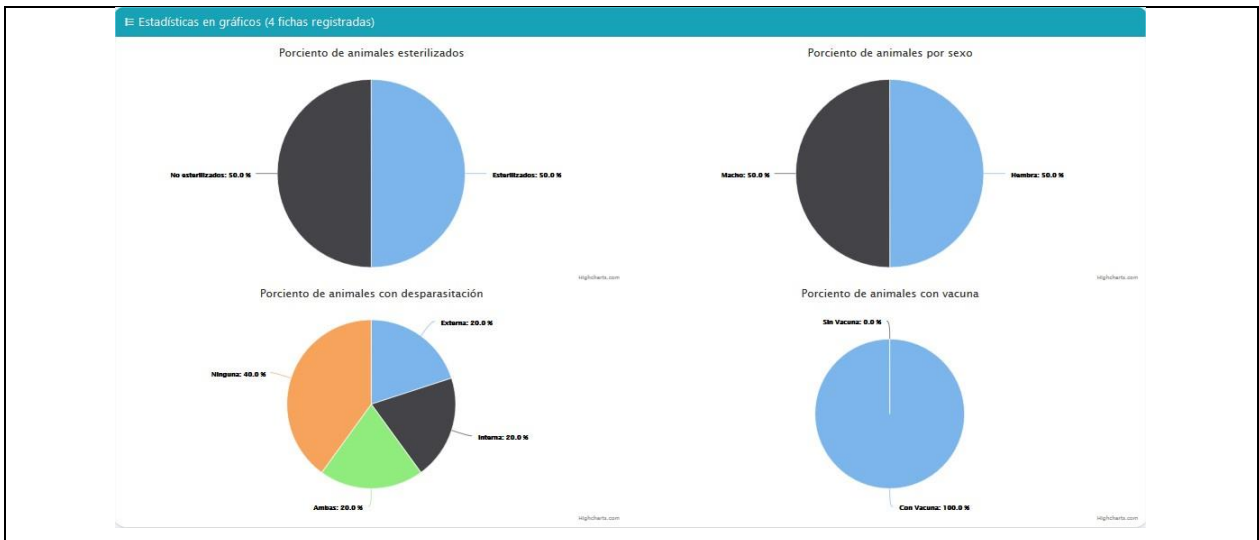


Tabla 19 Historia de usuario #12

Historia de usuario	
Número: 12	Nombre: Autenticar usuario
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.05	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
Descripción: El usuario en caso de ser administrador deberá autenticarse al rellenar los campos de "Usuario" y "Contraseña", en este caso solo existirá un usuario y una contraseña.	
Interfaz:	

Tabla 20 Historia de usuario #13

Historia de usuario

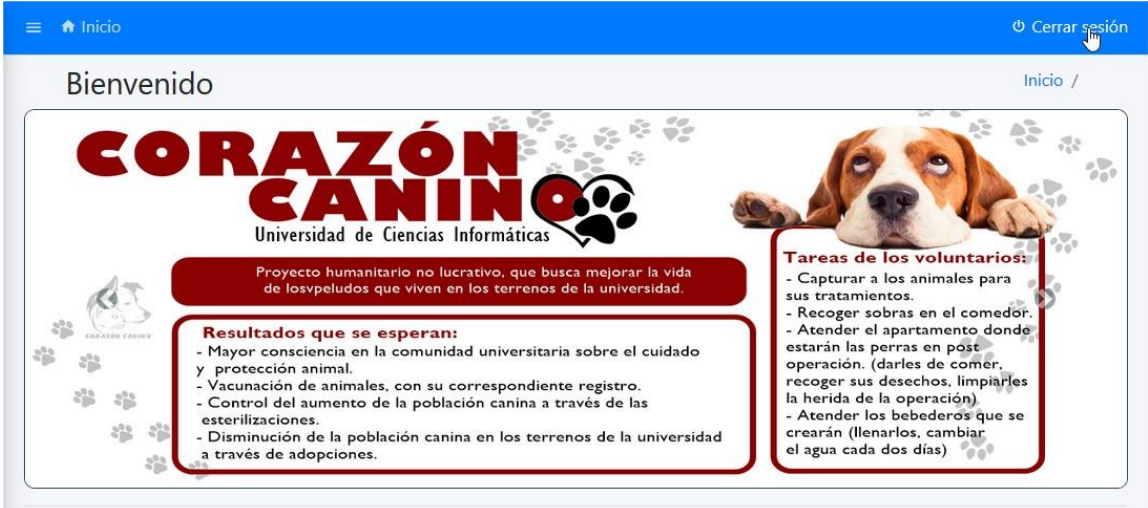
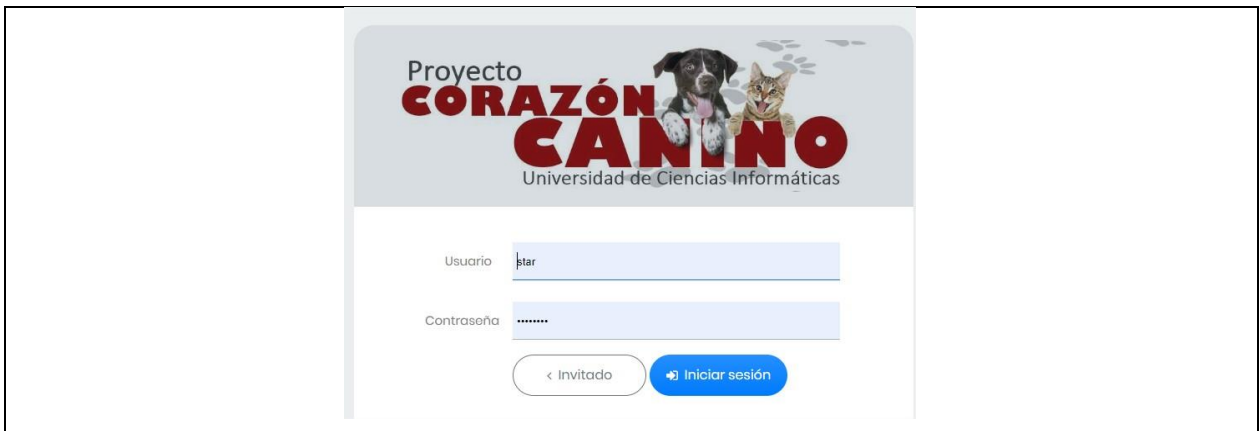
Número: 13	Nombre: Cerrar sesión
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.05	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
Descripción: Cuando el administrador haya culminado sus tareas en la página deberá acceder al botón “Cerrar sesión” localizado en la esquina superior derecha y hacer clic en ella, para automáticamente se cierre la sesión.	
Interfaz:	
	

Tabla 21 Historia de usuario #14

Historia de usuario	
Número: 14	Nombre: Iniciar sesión como “Invitado”
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.05	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
Descripción: El usuario en caso de ser únicamente invitado o usuario promedio deberá acceder a la página presionando el botón blanco “Invitado” y podrá tener vista de la página sin los permisos que otorga ser administrador.	
Interfaz:	



El resto de las HU, se encuentran en los Anexos.

2.4 Estimación de esfuerzo por historias de usuario.

La estimación de esfuerzo es planteada por la metodología XP a partir de las historias de usuario, determina que para desarrollar el software se facilite la estimación de tiempo necesaria.

Iteración		Historias de usuario	Puntos estimados (semanas)
1	1	Gestión de ficha animal	2
2	2	Gestionar visitas	0.5
	3	Gestionar eventos	0.5
	4	Gestionar información	0.5
	5	Gestionar denuncias	0.5
3	6	Gestionar enfermedades de animales	0.25
	7	Gestionar vacunas	0.25
	8	Gestionar medicamentos	0.25
	9	Gestionar asociados	0.25
	10	Gestionar contactos	0.20
	11	Generar reportes integrales	0.25
	12	Generar gráficas	0.25
	13	Gestionar desparasitación	0.25
	14	Autenticar usuario	0.05
	15	Cerrar Sesión	0.05

	16	Iniciar sesión como "Invitado"	0.05
Total			6.1

Tabla 22 Estimación de esfuerzo por historia de usuario

2.4.1 Desarrollo del plan de iteraciones.

Estando ya definidas las historias de usuarios y con la realización previa de la estimación de esfuerzos, se procede a planificar la etapa de implementación de sistema. Esto teniendo como objetivo hacer ver la duración que tendrá cada iteración, en orden y prioridad de las HU.

2.5 Plan de duración de las iteraciones.

Iteración	Historias de usuario	Puntos estimados (semanas)
1	1 Gestión de ficha animal	2
2	2 Gestionar visitas	2
	3 Gestionar eventos	
	4 Gestionar información	
	5 Gestionar denuncias	
	6 Gestionar enfermedades de animales	2.1
	7 Gestionar vacunas	
	8 Gestionar medicamentos	
	9 Gestionar asociados	
	10 Gestionar contactos	
	11 Generar reportes integrales	
	12 Generar gráficas	
	13 Gestionar desparasitación	
	14 Autenticar usuario	
	15 Cerrar Sesión	
	16 Iniciar sesión como "Invitado"	
Total		6.1

Tabla 23 Plan de duración de las iteraciones.

2.5.1 Plan de entregas

Planificación de la entrega, es una práctica en donde el Cliente presenta las características deseadas a los programadores, y los programadores estiman la dificultad. Teniendo las estimaciones de costo y sabiendo la importancia de las características, el cliente establece un plan para el proyecto. Los planes iniciales de entregas son necesariamente imprecisos: ni las prioridades, ni las estimaciones son sólidas y tampoco sabremos qué tan rápido trabaja el equipo hasta que empiece a trabajar. Sin embargo, incluso el primer plan de entrega es lo suficientemente preciso como para tomar decisiones, y el equipo XP revisa de forma regular el plan (36).

Iteraciones	Fecha de inicio	Fecha de fin
Iteración 1	9 de mayo de 2022	23 de mayo de 2022
Iteración 2	24 de mayo de 2022	7 de junio de 2022
Iteración 3	8 de junio de 2022	15 de junio de 2022

Tabla 24 Plan de entregas

2.6 Arquitectura de la solución

Una arquitectura de solución es una materialización de una arquitectura de referencia, en la que se detalla a nivel técnico los componentes involucrados en la configuración de una solución empresarial, y que ofrece una “fotografía” en la que dichos componentes interactúan entre sí, respetando los lineamientos y reglas de gobierno documentados en la arquitectura de referencia. El criterio para evaluar si una arquitectura de solución es apropiada a una entidad se basa esencialmente en dos elementos: que sea una respuesta tecnológica a las necesidades definidas en el alcance de una arquitectura de referencia, y que su diseño respete la misma (37).

¿Qué significa MVT de Django (38)?

- M está escrito como *Model*, que tiene la misma función que M en MVC, y es responsable de interactuar con la base de datos y procesar datos.
- V se escribe como *View*, que tiene la misma función que C (*Controller*) en MVC. Recibe solicitudes, procesa negocios y devuelve respuestas.
- T está escrito como *Template*, que tiene la misma función que V en MVC, y es responsable de encapsular y construir el HTML que se devolverá.

2.6.1 Patrón de diseño a utilizar.

Los patrones de diseño o *design patterns*, son una solución general, reutilizable y aplicable a diferentes problemas de diseño de software. Se trata de plantillas que identifican problemas en el sistema y proporcionan soluciones apropiadas a problemas generales a los que se han enfrentado los desarrolladores durante un largo periodo de tiempo, a través de prueba y error (39).

¿Por qué usar patrones de diseño?

El gran crecimiento del sector de las tecnologías de la información ha hecho que las prácticas de desarrollo de software evolucionen. Antes se requería completar todo el software antes de realizar pruebas, lo que suponía encontrarse con problemas. Para ahorrar tiempo y evitar volver a la etapa de desarrollo una vez que este ha finalizado, se introdujo una práctica de prueba durante la fase de desarrollo. Esta práctica se usa para identificar condiciones de error y problemas en el código que pueden no ser evidentes en ese momento. En definitiva, los patrones de diseño te ayudan a estar seguro de la validez de tu código, ya que son soluciones que funcionan y han sido probados por muchísimos desarrolladores siendo menos propensos a errores (39).

Tipos de patrones de diseño de software:

Los patrones de diseño más utilizados se clasifican en tres categorías principales, cada patrón de diseño individual conforma un total de 23 patrones de diseño. Las tres categorías principales son (39):

- Patrones creacionales
- Patrones estructurales
- Patrones de comportamiento



Ilustración 4 Patrones de diseño de software (39)

Django: *Template*

Al ser un marco web, Django necesita una forma conveniente de generar HTML dinámicamente. El enfoque más común se basa en plantillas. Una plantilla contiene las partes estáticas de la salida HTML deseada, así como alguna sintaxis especial que describe cómo se insertará el contenido dinámico (40).

Un proyecto de Django se puede configurar con uno o varios motores de plantillas (o incluso cero si no usa plantillas). Django incluye *backends* integrados para su propio sistema de plantillas, creativamente denominado lenguaje de plantillas de Django (DTL), y para la popular alternativa Jinja2. Los *backends* para otros lenguajes de plantilla pueden estar disponibles de terceros. También puede escribir su propio *backend* personalizado (40).

Django define una API estándar para cargar y renderizar plantillas independientemente del *backend*. La carga consiste en encontrar la plantilla para un identificador determinado y preprocesarla, generalmente compilándola en una representación en memoria. Representar significa interpolar la plantilla con datos de contexto y devolver la cadena resultante (40).

Django *Template* es un documento de texto o una cadena de Python marcada con el Lenguaje de Django *Template*. Algunos constructos son reconocidos e interpretados por el motor de plantillas. Los principales son las variables y las etiquetas. Una plantilla se representa con un contexto. La representación reemplaza las variables con sus valores, que se buscan en el contexto, y ejecuta etiquetas. Todo lo demás se emite tal cual (41).

GRASP

General Responsibility Software Patterns (GRASP) da principios generales para asignar responsabilidades y se utiliza sobre todo en la realización de diagramas de interacción (42). Entre los principales patrones de GRASP, se hizo uso de:

Experto: Determina la clase que debe asumir una responsabilidad a partir de la información que posee cada una.

Ejemplo de Patrón experto en la página web está la clase Ficha, la cual contiene las funcionalidades que son necesarias para acceder a la información de un proyecto.

```

39 class Ficha(models.Model):
40     SEXO_CHOICES = (
41         ('m', 'Macho'),
42         ('h', 'Hembra'),
43     )
44     nombre = models.CharField(max_length=255, verbose_name="Nombre", validators=[validate_upper_nombre])
45     color = models.CharField(max_length=255, verbose_name="Color")
46     raza = models.CharField(max_length=255, verbose_name="Raza")
47     foto = models.ImageField(upload_to='fotos/', verbose_name="Foto")
48     sexo = models.CharField(max_length=255, choices=SEXO_CHOICES, verbose_name="Sexo")
49     esterilizado = models.BooleanField(default=False, verbose_name="Esterilizado")
50     peso = models.FloatField(verbose_name="Peso en KG", validators=[
51         MinValueValidator(0, 'Debe ser positivo')], null=True, blank=True)
52     enfermedades = models.TextField(verbose_name='Enfermedades o lesiones',
53                                   null=True, blank=True)
54     qr = models.CharField(max_length=900, blank=True, null=True, verbose_name="Código Qr")
55     date_creation = models.DateField(auto_now_add=True, null=True, blank=True,
56                                   verbose_name='Fecha de registro')
57
58     def __str__(self):
59         return self.nombre
60
61     def toJSON(self):
62         json = model_to_dict(self)
63         json['foto'] = self.foto.url
64         json['sexo'] = 'Macho' if self.sexo == 'm' else 'Hembra'
65         json['esterilizado'] = 'Si' if self.esterilizado else 'No'
66         json['enfermedades'] = 'Ninguna' if not self.enfermedades else self.enfermedades
67         json['vacunas'] = [i.toJSON() for i in self.vacuna_set.all()] if self.vacuna_set.exists() else []
68         json['desparasitaciones'] = [i.toJSON() for i in
69                                   self.desparasitacion_set.all()] if self.desparasitacion_set.exists() else
70
71         json['qr'] = self.qr_relativo()
72         return json
73
74     def mostrar_foto(self):
75         return mark_safe('')
77
78     def link_foto(self):
79         return mark_safe(f'<a href="{self.foto.url}"> {self.mostrar_foto()}</a>')
80
81     mostrar_foto.short_description = 'Vista previa'
82     mostrar_foto.allow_tags = True

```

Ilustración 5 Fragmento de código del patrón experto.

Bajo acoplamiento (43):

El patrón indica que se debe tratar de mantener el nivel de acoplamiento bajo, minimizando el conocimiento que unas clases deben tener de otras.

- Cuando se toma una decisión, si está implica establecer una nueva relación entre las clases, se debe evaluar si realmente es necesario, porque al hacerlo se debe aumentar el nivel de acoplamiento.
- Patrón evaluativo (reduce el impacto de los cambios y aumenta la reutilización)

Ejemplo de bajo acoplamiento en el código de la página web:

```
class DesparasitacionListView(LoginRequiredMixin, generic.ListView, ):
    model = Desparasitacion
    template_name = 'desparasitacion_list.html'
    queryset = Desparasitacion.objects.all()
    success_url = reverse_lazy('desparasitacion-list')

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['create_url'] = reverse_lazy('desparasitacion-add')
        context['entity'] = 'Desparasitación'
        context['title'] = 'Desparasitaciones'
        return context

    def get_queryset(self):
        qs = super(DesparasitacionListView, self).get_queryset()
        if self.request.GET:
            if not self.request.GET.get('initial') or not self.request.GET.get('end'):
                return qs
            qs = qs.filter(fecha_range=[self.request.GET.get('initial'), self.request.GET.get('end')])
        return qs
```

Ilustración 6 Fragmento de código del patrón Bajo acoplamiento. Fuente: Elaboración propia.

2.7 Tarjetas de contenidos, responsabilidad y colaboración.

Tarjeta CRC	
Clase: FichaListView	
Responsabilidad:	Colaboración:
get_context_data, dispatch, post, get_queryset	ListView

Tabla 25 Tarjeta CRC#1

Tarjeta CRC	
Clase: FichaCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView

Tabla 26 Tarjeta CRC#2

Tarjeta CRC

Clase: FichaUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView

Tabla 27 Tarjeta CRC#3

Tarjeta CRC	
Clase: FichaDetailsView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView

Tabla 28 Tarjeta CRC#4

Tarjeta CRC	
Clase: VisitaListView	
Responsabilidad:	Colaboración:
get_context_data, get_queryset	ListView

Tabla 29 Tarjeta CRC#5

El resto de las tarjetas CRC, se encuentran en los Anexos.

2.8 Modelo de datos

El modelado de datos es el proceso de creación de una representación visual o esquema que define los sistemas de recopilación y administración de información de cualquier organización. Este esquema o modelo de datos ayuda a las diferentes partes interesadas, como analistas de datos, científicos e ingenieros, a crear una vista unificada de los datos de una organización. El modelo esboza los datos que recoge la empresa, la relación entre los distintos conjuntos de datos y los métodos que se usarán para almacenarlos y analizarlos (44).

Un modelo en Django es un tipo especial de objeto que se guarda en la base de datos. Una base de datos es una colección de datos. Es un lugar en el cual se almacenará la información sobre usuarios, las entradas de blog, etc. Y se usará el *Object relational mapping* (ORM) de base de datos predeterminado de Django (45).

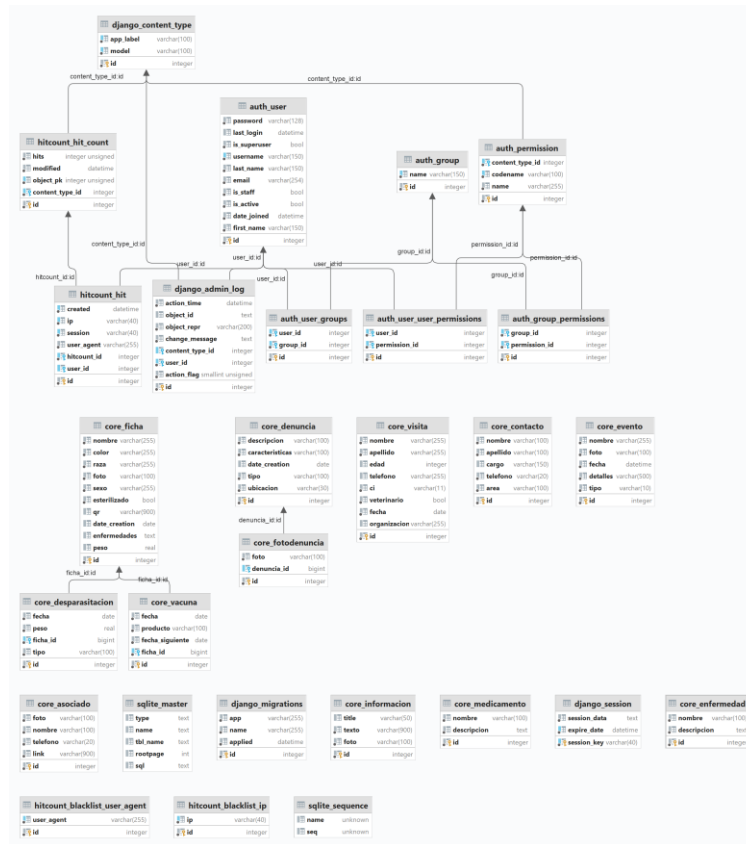


Ilustración 7 Modelo de datos

2.9 Conclusiones parciales.

- Se identificaron hasta 53 requisitos funcionales y 7 requisitos no funcionales.
- A partir de los RF se detallaron los mismos en las historias de usuarios, concluyendo un total de 3 iteraciones y un estimado de 6 semanas para el desarrollo.
- Se realizó la descripción de las tarjetas CRC, con la especificación de clases presentes, con un total de 45 clases.

CAPÍTULO 3: Evaluación de la solución propuesta

Introducción

En el presente capítulo a partir de las historias de usuario ya creadas se desarrollarán las tareas de ingeniería, las cuales, desde el punto de vista del programador o desarrollador, se descomponen y describen. Por otro lado, se estará realizando la descripción y funcionamiento de la aplicación web vista desde la página de inicio. Cuando la descripción este realizada a detalle, se podrá pasar a una nueva etapa, en la cual se incluye la muestra de códigos principales del programa lo que permitirá un mejor desenvolvimiento al momento de probar el programa. Finalmente se llega a las pruebas, las cuales, entre las recomendaciones hechas por la metodología XP, se encuentra la prueba de aceptación, la que forma parte del esquema de trabajo.

3.1 Tareas de ingeniería

La metodología XP propone, que, por cada iteración planeada, se describan las tareas de ingeniería correspondientes a las historias de usuario obtenidas. Permitiendo, desde el punto de vista del programador, ver los pasos para dar solución a las funcionalidades requeridas para el usuario. Es necesario decir que una historia de usuario puede presentar varias tareas de ingeniería, las cuales serán representadas a continuación.

3.1.1 Iteración I

Dentro de la primera iteración se encuentran las funcionalidades relacionadas a las historias de usuario 1, 2, 3, 4, 5. Las tareas definidas para la iteración son:

- Tarea N.1: Crear ficha de animal.
- Tarea N.2: Modificar ficha de animal.
- Tarea N.3: Eliminar ficha de animal.
- Tarea N.4: Mostrar el animal.
- Tarea N.5: Generar código QR de la ficha del animal.

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Crear ficha de animal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 9 de mayo de 2022	Fecha de fin: 11 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	

<p>Descripción:</p> <p>Se implementa un botón, nombrado “Crear Ficha”, que al ser presionado accederá al formulario a llenar para crear la nueva ficha animal. Luego deberá estar implementado el botón “Guardar Ficha”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de fichas.</p>
--

Tabla 30 Tarea de investigación #1

Tarea	
Número de tarea: 2	Número de Historia de usuario: 2
Nombre de la tarea: Modificar ficha de animal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 11 de mayo de 2022	Fecha de fin: 12 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
<p>Descripción:</p> <p>Se implementa un botón icono localizado en las “Opciones” en la tabla del Listado de Fichas, dicho botón amarillo, será el que permita modificar los formularios rellenos anteriormente en el “Crear Ficha”. Implementar además el botón “Guardar Ficha” (su respectivo icono) el cual guardará los cambios hechos, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de fichas.</p>	

Tabla 31 Tarea de investigación #2

Tarea	
Número de tarea: 3	Número de Historia de usuario: 3
Nombre de la tarea: Eliminar ficha de animal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.3
Fecha de inicio: 12 de mayo de 2022	Fecha de fin: 14 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
<p>Descripción:</p> <p>Se implementa el botón rojo con el icono de <i>basura</i> para eliminar la ficha animal escogida que aparecerá en la columna de “Opciones” del Listado de Fichas. Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.</p>	

Tabla 32 Tarea de investigación #3

Tarea	
Número de tarea: 4	Número de Historia de usuario: 4
Nombre de la tarea: Mostrar el animal.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 14 de mayo de 2022	Fecha de fin: 18 de mayo de 2022

Programador responsable: Dayana Bolaños Banzo
Descripción: Se implementa el botón “Mostrar”, el cual estará representado con un botón verde y centrado el icono de una lupa. Al presionar el botón se ampliará el modal con toda la información del animal. Para minimizar el modal se implementa el botón cerrar (X) en la esquina superior derecha.

Tabla 33 Tarea de investigación #4

Tarea	
Número de tarea: 5	Número de Historia de usuario: 5
Nombre de la tarea: Generar código QR de la ficha del animal.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 18 de mayo de 2022	Fecha de fin: 23 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Al crear la ficha animal, se implementa que el sistema genere automáticamente el código QR de modo que almacené el link con la información y datos de la ficha animal a la que pertenece.	

Tabla 34 Tarea de investigación #5

3.1.2 Iteración II

Dentro de la segunda iteración se encuentran las funcionalidades relacionadas al área de denuncia, visitas, eventos e información, las cuales corresponden a las historias de usuario de la 6 a la 10, 18, 19 y 20. Las tareas definidas para la iteración son:

- Tarea N.6: Crear denuncia
- Tarea N.7: Modificar denuncia.
- Tarea N.8: Eliminar denuncia.
- Tarea N.9: Mostrar denuncia.
- Tarea N.10: Filtrar denuncia.
- Tarea N.11: Crear visitas
- Tarea N.12: Modificar visitas
- Tarea N.13: Eliminar visitas
- Tarea N.14: Mostrar visitas
- Tarea N.15: Filtrar visitas
- Tarea N.16: Crear eventos
- Tarea N.18: Modificar eventos
- Tarea N.18: Eliminar eventos

- Tarea N.19: Mostrar eventos
- Tarea N.20: Filtrar eventos
- Tarea N.21: Crear información
- Tarea N.22: Modificar información
- Tarea N.23: Eliminar información
- Tarea N.24: Mostrar información

Tarea	
Número de tarea: 6	Número de Historia de usuario: 6
Nombre de la tarea: Crear denuncia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 24 de mayo de 2022	Fecha de fin: 25 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón, nombrado “Crear Denuncia”, que al ser presionado accederá al formulario a llenar para crear la denuncia. Luego deberá estar implementado el botón “Guardar Denuncia”, el cual será presionado al terminar de rellenar el formulario y automáticamente se enviará al correo almacenado en el sistema, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de denuncias.	

Tabla 35 Tarea de investigación #7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 7
Nombre de la tarea: Modificar denuncia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 25 de mayo de 2022	Fecha de fin: 26 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Acceder al botón amarillo en “Opciones” que corresponder a “Modificar denuncia” que permitirá realizarle cambios a la información de la denuncia, en los datos <u>Descripción</u> , <u>Ubicación</u> , <u>Características del animal</u> . Se procede a guardar dando clic en el botón “Guardar Denuncia”, o en caso contrario para descartar los cambios el botón “Cancelar”.	

Tabla 36 Tarea de investigación #8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 8
Nombre de la tarea: Eliminar denuncia	

Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 26 de mayo de 2022	Fecha de fin: 27 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Acceder al botón rojo en “Opciones” que corresponder a “Eliminar denuncia”, luego de presionarlo se mostrará la alerta y pregunta de seguridad (¡Cuidado!, ¿Estás seguro de eliminar [nombre]?) para confirmar con el “Sí, aceptar” o descartar la eliminación con el “Cancelar”	

Tabla 37 Tarea de investigación #9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 9
Nombre de la tarea: Mostrar denuncia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 27 de mayo de 2022	Fecha de fin: 28 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Acceder al botón verde en “Opciones” que corresponder a “Mostrar denuncia”, lo cual procederá a mostrar toda la información referente a la denuncia que no está visible en la tabla, como <u>Tipo de denuncia</u> , <u>Descripción</u> , <u>Ubicación de los hechos</u> , y <u>Tipo de abandono</u> . Para luego retroceder solo se deberá presionar el botón “Atrás”	

Tabla 38 Tarea de investigación #10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 10
Nombre de la tarea: Filtrar denuncia	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 28 de mayo de 2022	Fecha de fin: 29 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un filtro por fecha que permitirá mostrar únicamente las denuncias en dependencia de las fechas especificadas.	

Tabla 39 Tarea de investigación #11

El resto de las tablas de tareas de investigación de la iteración II se encuentran en los Anexos.

3.1.3 Iteración III

Dentro de la tercera iteración se encuentran las funcionalidades relacionadas a las enfermedades, vacunas, medicamentos, asociados, contactos, autenticar usuario, generar reportes integrales, generar gráficas, desparasitación, cerrar sesión y entrar como invitado. Las tareas definidas para la iteración son:

- Tarea 25: Crear enfermedad animal.
- Tarea 26: Modificar enfermedad animal.
- Tarea 27: Eliminar enfermedad animal.
- Tarea 28: Mostrar enfermedad.
- Tarea 29: Crear vacunas.
- Tarea 30: Modificar vacuna.
- Tarea 31: Eliminar vacuna.
- Tarea 32: Filtrar vacuna
- Tarea 33: Crear medicamento.
- Tarea 34: Modificar medicamento.
- Tarea 35: Eliminar medicamento.
- Tarea 36: Mostrar medicamento.
- Tarea 37: Crear asociados
- Tarea 38: Modificar asociados
- Tarea 39: Eliminar asociados
- Tarea 40: Mostrar Asociados
- Tarea 41: Crear contacto
- Tarea 42: Modificar contactos
- Tarea 43: Eliminar contactos
- Tarea 44: Mostrar contacto
- Tarea 45: Autenticar usuario
- Tarea 46: Generar reportes integrales
- Tarea 47: Generar gráficas
- Tarea 48: Crear desparasitación
- Tarea 49: Modificar desparasitación
- Tarea 50: Eliminar desparasitación
- Tarea 51: Filtrar desparasitación

- Tarea 52: Cerrar sesión
- Tarea 53: Iniciar sesión como “Invitado”

Tarea	
Número de tarea: 47	Número de Historia de usuario: 11
Nombre de la tarea: Generar gráficas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 13 de junio de 2022	Fecha de fin: 13 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Las gráficas (“Por ciento de animales esterilizados”, “Por ciento de animales por sexo”, Por ciento de animales con desparasitación” y “Por ciento de animales con vacunas”) se generarán automáticamente en su apartado de <u>Gráficos</u> siempre y cuando se tenga acceso a internet y estas se puedan mantener actualizadas.	

Tabla 40 Tarea de investigación #46

El resto de las tareas de investigación de la Iteración III, se encuentran en los Anexos.

3.2 Descripción y funcionamiento

Con el uso del lenguaje de programación python y su *framework* de desarrollo Django, se hace uso de plantillas que permiten agilizar el proceso de desarrollo y darle una estructura más profesional a la página web, tal como se puede observar en la “Figura 5”

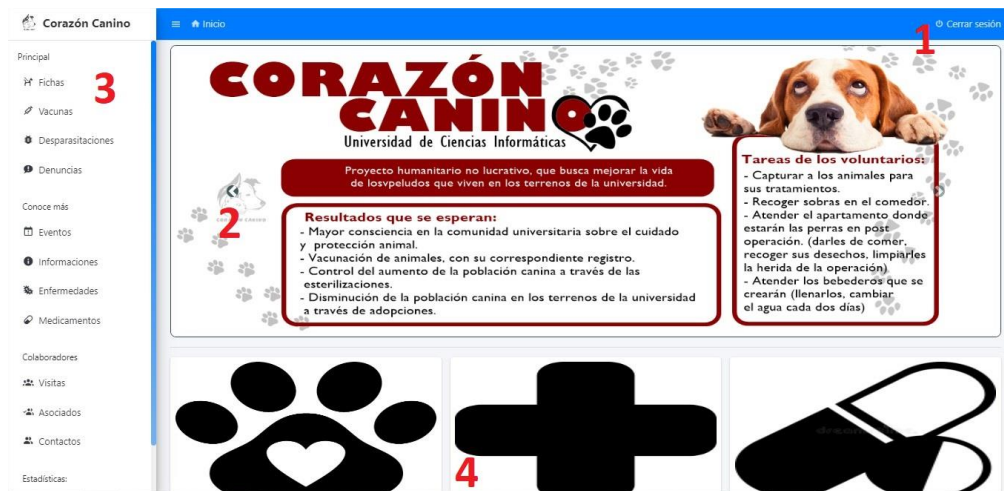


Ilustración 8 Vista de la página principal

Dicha plantilla cuenta con:

1. Una barra superior en la esquina derecha está el botón de “Iniciar/Cerrar sesión” y en la izquierda el icono junto a la caseta de Inicio, permite desplegar el menú lateral que cuenta con las principales opciones con las que cuenta la página del proyecto extensionista.
2. Debajo de la barra superior se encuentra el carrusel informativo de la página. Este permite al usuario ver la información cambiante en un plazo de 5 segundos. Si se desea cambiar la imagen antes del tiempo programado, solo necesita dar clic en la punta de flecha que está centrada en los literales internos del carrusel.
3. En el menú desplegable se encuentran divididas las sesiones ente “Principal” donde se encuentran las principales funcionalidades del proyecto, “Conoce más” área informativa para mantener al usuario actualizado, y “Colaboradores”, esta última tendrá contactos de aquellos que forman parte del proyecto extensionista. Al inicio del menú desplegable se encuentra el logo del proyecto extensionista junto a su nombre “Corazón canino”, en caso de encontrarse en el área de ficha animal y se desee regresar rápido a la página de bienvenida, solo será necesario tocar el botón del nombre el cual permitirá moverse rápidamente a este destino.
4. Otra forma de acceder a las funcionalidades sin desplegar el menú lateral, se encuentra un poco más abajo, donde están igualmente agrupados. Divididos de forma similar al menú desplegable: “Ficha Animal”, “Salud”, y “Colaboradores”.

3.3 Pruebas de aceptación

La metodología XP, con la cual se ha trabajado a lo largo del proyecto propone las pruebas de aceptación, las cuales permitirán comprobar que los requisitos propuestos por el cliente tienen un correcto funcionamiento.

3.3.1 Iteración I

Caso de prueba de aceptación	
Código: HU_P1.1	Historia de usuario: 1
Nombre: Crear ficha de animal	
Descripción: Prueba para la funcionalidad: Crear ficha animal en el Listado de fichas.	
Condiciones de ejecución: -	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede al “Listado de fichas” o solo “Listado” en el apartado de <u>Ficha animal</u>. 2. Se selecciona el botón “Crear Ficha” 3. Se muestra el apartado de “Crear una Ficha” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Ficha”. 	

<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se muestra en el “Listado de fichas” la nueva ficha creada. • Resultado satisfactorio
--

Tabla 41 Prueba de aceptación #1.1

Caso de prueba de aceptación	
Código: HU_P2.1	Historia de usuario: 2
Nombre: Modificar ficha de animal	
Descripción: Prueba para la funcionalidad: Modificar ficha animal en el Listado de fichas.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Ya deberá existir una ficha creada que pueda ser modificada. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a “Fichas” en “Principal”. 2. Se selecciona el botón amarillo en la columna de “Opciones” para modificar. 3. Se muestra el formulario al que se le realizarán modificaciones. 4. Se selecciona el botón “Guardar ficha” 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra en el “Listado de fichas” la ficha ya modificada. • Resultado satisfactorio. 	

Tabla 42 Prueba de aceptación #2.1

Caso de prueba de aceptación	
Código: HU_P3.1	Historia de usuario: 3
Nombre: Eliminar ficha de animal	
Descripción: Prueba de la funcionalidad: Eliminar ficha animal en el Listado de ficha.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Ya deberá existir una ficha creada que pueda ser eliminada. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a “Fichas” en “Principal”. 2. Se selecciona el botón rojo en la columna de “Opciones” para eliminar. 3. Se muestra el mensaje de seguridad. 4. Se selecciona el botón “Si, aceptar” para proseguir en la eliminación. 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se elimina del “Listado de fichas” la ficha seleccionada. • Resultado satisfactorio. 	

Tabla 43 Prueba de aceptación #3.1

Caso de prueba de aceptación

Código: HU_P4.1	Historia de usuario: 4
Nombre: Mostrar el animal	
Descripción: Prueba de la funcionalidad: Mostrar el animal que alberga la ficha en el "Listado de ficha"	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Ya deberá existir una ficha creada que pueda ser mostrada. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a "Fichas" en "Principal". 2. Se selecciona el botón verde con el icono de lupa, en la columna de "Opciones" para mostrar. 3. Se muestran todos los datos de la ficha animal en un modal. 4. Se selecciona el botón (X) en la esquina superior derecha para cerrar. 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra la ficha seleccionada del "Listado de fichas". • Resultado satisfactorio. 	

Tabla 44 Prueba de aceptación #4.1

Caso de prueba de aceptación	
Código: HU_P5.1	Historia de usuario: 5
Nombre: Generar código QR de la ficha del animal.	
Descripción: Prueba de la funcionalidad: Generar código QR de la ficha animal del "Listado de fichas"	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Haber accedido al botón "Crear ficha" 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se crea la ficha animal luego de rellenar el formulario. 2. Se accede al botón "Guardar ficha" 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra en el "Listado de fichas" el QR. • Resultado no satisfactorio. 	

Tabla 45 Prueba de aceptación #5.1

Caso de prueba de aceptación	
Código: HU_P5.2	Historia de usuario: 5
Nombre: Gestionar ficha de animal: Generar código QR de la ficha del animal.	
Descripción: Prueba de la funcionalidad: Generar código QR de la ficha animal del "Listado de fichas"	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Haber accedido al botón "Crear ficha" 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se crea la ficha animal luego de rellenar el formulario. 	

2. Se accede al botón “Guardar ficha”
Resultados esperados: <ul style="list-style-type: none"> • Se muestra en el “Listado de fichas” el QR. • Resultado satisfactorio.

Tabla 46 Prueba de aceptación #5.2

3.3.1 Iteración II

Caso de prueba de aceptación	
Código: HU_P6.1	Historia de usuario: 6
Nombre: Crear denuncia	
Descripción: Prueba para la funcionalidad: Crear denuncia en el “Listado de denuncias”.	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a “Denuncias” en “Principal” o “Denuncia” en el apartado de <u>Ficha animal</u> del menú principal. 2. Se selecciona el botón “Crear Denuncia” 3. Se muestra el apartado de “Crear Denuncia” donde se llenará el formulario. 4. Se selecciona el botón “Enviar Denuncia”. 	
Resultados esperados: <ul style="list-style-type: none"> • Se envía correctamente la denuncia reflejándose en el “Listado de denuncias” • Resultado no satisfactorio 	

Tabla 47 Prueba de aceptación #6.1

Caso de prueba de aceptación	
Código: HU_P6.2	Historia de usuario: 6
Nombre: Crear denuncia	
Descripción: Prueba para la funcionalidad: Crear denuncia en el “Listado de denuncias”.	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a “Denuncias” en “Principal”. 2. Se selecciona el botón “Crear Denuncia” 3. Se muestra el apartado de “Crear Denuncia” donde se llenará el formulario. 4. Se selecciona el botón “Enviar Denuncia”. 	
Resultados esperados: <ul style="list-style-type: none"> • Se envía correctamente la denuncia reflejándose en el “Listado de denuncias” • Resultado satisfactorio 	

Tabla 48 Prueba de aceptación #6.2

Caso de prueba de aceptación	
Código: HU_P7.1	Historia de usuario: 7
Nombre: Modificar denuncia	
Descripción: Prueba para la funcionalidad: Modificar denuncia en el “Listado de denuncias”	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Existencia de la denuncia a modificar. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Se accede a “Denuncias” en “Principal”. Se selecciona el botón amarillo en la columna de “Opciones” para modificar. Se muestra el formulario al que se le realizarán modificaciones. Se selecciona el botón “Guardar denuncia” 	
Resultados esperados:	
<ul style="list-style-type: none"> Se modifica correctamente la denuncia reflejándose en el “Listado de denuncias” Resultado satisfactorio 	

Tabla 49 Prueba de aceptación #7.1

Caso de prueba de aceptación	
Código: HU_P8.1	Historia de usuario: 8
Nombre: Eliminar denuncia	
Descripción: Prueba para la funcionalidad: Eliminar denuncia en el “Listado de denuncias”	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Existencia de la denuncia a eliminar. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Se accede a “Denuncias” en “Principal”. Se selecciona el botón rojo con el icono de basurero, en la columna “Opciones” de la denuncia a eliminar. Se muestra el mensaje de alerta. Se selecciona el botón “Si, aceptar” para proseguir con la eliminación. 	
Resultados esperados:	
<ul style="list-style-type: none"> Se elimina la denuncia seleccionada del “Listado de denuncias” Resultado satisfactorio. 	

Tabla 50 Prueba de aceptación #8.1

Caso de prueba de aceptación	
Código: HU_P9.1	Historia de usuario: 9
Nombre: Mostrar denuncia	
Descripción: Prueba para la funcionalidad: Mostrar denuncia del “Listado de denuncias”	

<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> Existencia de la denuncia a mostrar.
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> Se accede a “Denuncias” en “Principal”. Se selecciona el botón verde con el icono de lupa, en la columna de “Opciones” para mostrar. Se muestra la información referente a la denuncia en el visual “Detalles de la denuncia”. Se selecciona el botón “Atrás” para regresar al “Listado de denuncias”
<p>Resultados esperados:</p> <ul style="list-style-type: none"> Se muestran todos los detalles de la denuncia seleccionada en el “Listado de denuncias”. Resultado satisfactorio.

Tabla 51 Prueba de aceptación #9.1

Caso de prueba de aceptación	
Código: HU_P10.1	Historia de usuario: 10
Nombre: Filtrar denuncia	
Descripción: Prueba para la funcionalidad: Filtrar denuncia del “Listado de denuncias”	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> Existencia de denuncias a filtrar. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> Se accede a “Denuncias” en “Principal”. Se establecen las fechas límites en el filtro de fechas localizado en la esquina superior derecha del “Listado de denuncias” Se selecciona el botón “Filtrar” 	
<p>Resultados esperados:</p> <ul style="list-style-type: none"> Se muestran las denuncias correspondientes a las fechas puestas en el filtro. Resultado satisfactorio. 	

Tabla 52 Prueba de aceptación #10.1

El resto de las pruebas de aceptación de la Iteración II, se encuentran en los Anexos.

3.3.1 Iteración III

Caso de prueba de aceptación	
Código: HU_P11.1	Historia de usuario: 11
Nombre: Generar gráficas	
Descripción: Prueba para la funcionalidad: Generar gráficas en el apartado de “Estadísticas en gráficos”	
Condiciones de ejecución: Ninguna	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Se crea la ficha animal las vacunas las desparasitaciones. 2. Se guardan.
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se generan automáticamente los porcentos en el gráfico de: Por ciento de animales esterilizados, Por ciento de animales por sexo, Por ciento de animales con desparasitación, Por ciento de animales con vacuna. • Resultado no satisfactorio.

Tabla 53 Prueba de aceptación #11.1

Caso de prueba de aceptación	
Código: HU_P11.2	Historia de usuario: 11
Nombre: Generar gráficas	
Descripción: Prueba para la funcionalidad: Generar gráficas en el apartado de “Estadísticas en gráficos”	
Condiciones de ejecución: Ninguna	
Pasos de ejecución:	
<ol style="list-style-type: none"> 3. Se crea la ficha animal las vacunas las desparasitaciones. 4. Se guardan. 	
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se generan automáticamente los porcentos en el gráfico de: Por ciento de animales esterilizados, Por ciento de animales por sexo, Por ciento de animales con desparasitación, Por ciento de animales con vacuna. • Resultado satisfactorio. 	

Tabla 54 Prueba de aceptación #11.2

Caso de prueba de aceptación	
Código: HU_P12	Historia de usuario: 12
Nombre: Autenticar usuario	
Descripción: Prueba de la funcionalidad: Autenticar usuario administrador de la página.	
Condiciones de ejecución: Ninguna	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder al “Iniciar sesión”. 2. Llenar los parámetros “Usuario” y “Contraseña”. 3. Seleccionar el botón “Iniciar sesión”. 	
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se inicia sesión correctamente como administrador de la página. • Resultado satisfactorio. 	

Tabla 55 Prueba de aceptación #12.1

El resto de las pruebas de aceptación de la Iteración III, se encuentran en los Anexos.

3.4 Pruebas de aplicación web

La prueba de Aplicaciones Web, es el proceso de actividades relacionadas con la meta de descubrir errores en el contenido, la facilidad de uso, el desempeño, la función, la capacidad, la navegabilidad y la seguridad de la *WebApp*. Es importante para que los usuarios finales estén contentos con el contenido y la función del producto, por eso es importante aplicar este proceso, para eliminar la mayor cantidad de errores posibles (46).

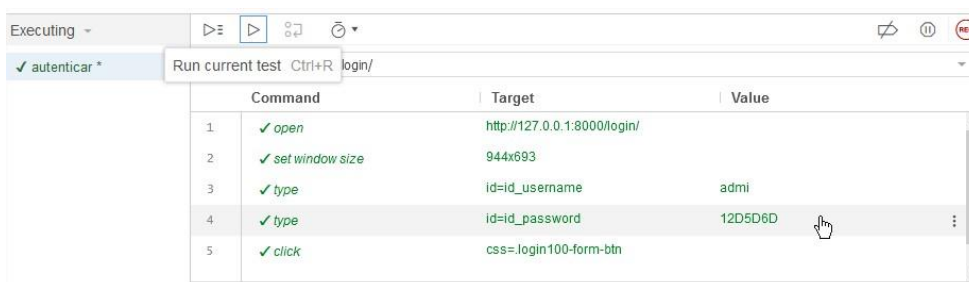
Para poder realizar dichas pruebas se hizo uso inicialmente de *Selenium IDE (Integrated Development Environment)*, el cual, es una herramienta de automatización que permite grabar, editar y depurar pruebas, también conocido como *Selenium Recorder* (47).

✓ Prueba #1 Autenticar (Correcto)

1	✓ open	http://127.0.0.1:8000/login/	
2	✓ set window size	944x693	
3	✓ type	id=id_username	admiuci
4	✓ type	id=id_password	1@2B3C4D
5	✓ click	css=.login100-form-btn	

Ilustración 9 Prueba de aplicación #1 Autenticar correcto

✓ Prueba #2 Autenticar (con datos incorrectos-salió satisfactorio)



	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/login/	
2	✓ set window size	944x693	
3	✓ type	id=id_username	admi
4	✓ type	id=id_password	12D5D6D
5	✓ click	css=.login100-form-btn	

Ilustración 10 Prueba de aplicación #2 Autenticar con datos incorrectos

✓ Prueba #4 Exportar PDF (Generar reporte)

	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/estadisticas/reports/	
2	✓ set window size	1382x744	
3	✓ click	css=.btn	

Ilustración 11 Prueba de aplicación #4 Exportar PDF

✓ Prueba #7 Crear desparasitación (con datos correctos)

http://127.0.0.1:8000/desparasitacion/add/

	Command	Target	Value
1	✓ open	http://127.0.0.1:8000/desparasitacion/add/	
2	✓ set window size	944x693	
3	✓ click	id=id_ficha	
4	✓ select	id=id_ficha	label=Bartolo
5	✓ click	css=#id_ficha > option:nth-child(2)	
6	✓ click	id=id_fecha	
7	✓ type	id=id_fecha	0002-11-08
8	✓ type	id=id_fecha	0020-11-08
9	✓ type	id=id_fecha	0202-11-08
10	✓ type	id=id_fecha	2022-11-08
11	✓ click	id=id_tipo	
12	✓ select	id=id_tipo	label=Externa
13	✓ click	css=#id_tipo > option:nth-child(2)	
14	✓ click	id=id_peso	
15	✓ type	id=id_peso	15
16	✓ click	css=.bg-gradient-primary	

Ilustración 12 Prueba de aplicación #5 Crear desparasitación

El resto de las pruebas estará en los anexos.

3.5 Pruebas unitarias

Las pruebas unitarias son de muy bajo nivel y se realizan cerca de la fuente de la aplicación. Consisten en probar métodos y funciones individuales de las clases, componentes o módulos que usa tu software. En general, las pruebas unitarias son bastante baratas de automatizar y se pueden ejecutar rápidamente mediante un servidor de integración continua (48).

El *Selenium* IDE, permite la realización de pruebas de aplicación, al concluir las permite que estas sean exportadas como "Python pytest", código que puede ser visualizado en el editor de código para realizar correctamente las pruebas unitarias.

Entre las pruebas unitarias realizadas se encuentran:

✓ Prueba #1 Autenticar (Correcto)

```
class TestPruebadeloginincorrect():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_pruebadeloginincorrect(self):
        self.driver.get("http://127.0.0.1:8000/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.CSS_SELECTOR, ".ml-auto .d-none").click()
        self.driver.find_element(By.ID, "id_username").click()
        self.driver.find_element(By.ID, "id_username").send_keys("admiuci")
        self.driver.find_element(By.ID, "id_password").send_keys("1@2B3C4D")
        self.driver.find_element(By.CSS_SELECTOR, ".login100-form-btn").click()
```

✓ Prueba #2 Autenticar (con datos incorrectos-salió satisfactorio)

```
class TestPruebadeloginincorrect():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_pruebadeloginincorrect(self):
        self.driver.get("http://127.0.0.1:8000/login/")
```

```
self.driver.set_window_size(944, 692)
self.driver.find_element(By.ID, "id_username").send_keys("admiuci")
self.driver.find_element(By.ID, "id_password").send_keys("1@2B3C5")
self.driver.find_element(By.CSS_SELECTOR, ".login100-form-btn").click()
```

✓ Prueba #3 Generar gráfico

```
class TestPruebadelgrafico():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_pruebadelgrafico(self):
        self.driver.get("http://127.0.0.1:8000/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.CSS_SELECTOR, ".chart-sidebar > p").click()
        element = self.driver.find_element(By.LINK_TEXT, "Medicamentos")
        actions = ActionChains(self.driver)
        actions.move_to_element(element).perform()
```

✓ Prueba #4 Exportar PDF (Generar reporte)

```
class TestExportarPDF():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_exportarPDF(self):
        self.driver.get("http://127.0.0.1:8000/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.CSS_SELECTOR, ".report-sidebar > p").click()
```



```
self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
```

✓ Prueba #5 Crear ficha animal (correctamente)

```
class TestCrearfichaanimal():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_crearfichaanimal(self):
        self.driver.get("http://127.0.0.1:8000/ficha/list/")
        self.driver.set_window_size(944, 692)
        self.driver.find_element(By.LINK_TEXT, "Crear Ficha").click()
        self.driver.find_element(By.ID, "id_nombre").click()
        self.driver.find_element(By.ID, "id_nombre").send_keys("LOla")
        self.driver.find_element(By.ID, "id_color").send_keys("Negra")
        self.driver.find_element(By.ID, "id_raza").send_keys("Sata")
        self.driver.find_element(By.ID, "id_foto").click()
        self.driver.find_element(By.ID, "id_foto").send_keys("D:\\DAYANA\\1. Proyecto\\Fotos\\Corazón
canino\\Dunia\\IMG_20220610_142218.jpg")
        self.driver.find_element(By.ID, "id_sexo").click()
        dropdown = self.driver.find_element(By.ID, "id_sexo")
        dropdown.find_element(By.XPATH, "//option[. = 'Hembra']").click()
        self.driver.find_element(By.CSS_SELECTOR, "option:nth-child(3)").click()
        self.driver.find_element(By.ID, "id_esterilizado").click()
        self.driver.find_element(By.ID, "id_peso").click()
        self.driver.find_element(By.ID, "id_peso").send_keys("14")
        self.driver.find_element(By.ID, "id_peso").send_keys("13")
        self.driver.find_element(By.ID, "id_peso").send_keys("12")
        self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()
```

✓ Prueba #6 Eliminar ficha animal (correctamente)

```
class TestEliminarfichaanimal():
```

```

def setup_method(self, method):
    self.driver = webdriver.Firefox()
    self.vars = {}

def teardown_method(self, method):
    self.driver.quit()

def test_eliminarfichaanimal(self):
    self.driver.get("http://127.0.0.1:8000/ficha/list/")
    self.driver.set_window_size(1382, 744)
    self.driver.find_element(By.CSS_SELECTOR, "tr:nth-child(4) .btn-danger > .mdi").click()
    self.driver.find_element(By.CSS_SELECTOR, ".btn-danger").click()

```

✓ Prueba #7 Crear desparasitación (con datos correctos)

```

class TestCreardesparasitacin():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_creardesparasitacin(self):
        self.driver.get("http://127.0.0.1:8000/desparasitacion/list/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.LINK_TEXT, "Crear Desparasitación").click()
        self.driver.find_element(By.ID, "id_ficha").click()
        dropdown = self.driver.find_element(By.ID, "id_ficha")
        dropdown.find_element(By.XPATH, "//option[. = 'Bartolo']").click()
        self.driver.find_element(By.CSS_SELECTOR, "#id_ficha > option:nth-child(2)").click()
        self.driver.find_element(By.ID, "id_fecha").click()
        self.driver.find_element(By.ID, "id_fecha").send_keys("0002-11-08")
        self.driver.find_element(By.ID, "id_fecha").send_keys("0020-11-08")
        self.driver.find_element(By.ID, "id_fecha").send_keys("0202-11-08")
        self.driver.find_element(By.ID, "id_fecha").send_keys("2022-11-08")

```

```

self.driver.find_element(By.ID, "id_tipo").click()
dropdown = self.driver.find_element(By.ID, "id_tipo")
dropdown.find_element(By.XPATH, "//option[. = 'Externa e Interna']").click()
self.driver.find_element(By.CSS_SELECTOR, "#id_tipo > option:nth-child(4)").click()
self.driver.find_element(By.ID, "id_peso").click()
self.driver.find_element(By.ID, "id_peso").send_keys("20")
self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()

```

Prueba #8 Crear denuncia (con datos incompletos-salió satisfactorio)

```

class TestCreardenuncia():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_creardenuncia(self):
        self.driver.get("http://127.0.0.1:8000/denuncia/list/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.LINK_TEXT, "Crear Denuncia").click()
        self.driver.find_element(By.ID, "id_descripcion").click()
        self.driver.find_element(By.ID, "id_descripcion").send_keys("dddddddddddddddddddddddddddddddd")
        self.driver.find_element(By.ID, "id_ubicacion").click()
        self.driver.find_element(By.ID,
" id_ubicacion").send_keys("dddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd")
        self.driver.find_element(By.ID, "id_descripcion").click()
        self.driver.find_element(By.ID, "id_descripcion").click()
        self.driver.find_element(By.ID, "id_descripcion").click()
        self.driver.find_element(By.ID,
" id_descripcion").send_keys("dddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddd")
        self.driver.find_element(By.ID, "id_caracteristicas").click()

```

```
self.driver.find_element(By.ID,
"id_caracteristicas").send_keys("dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd")
self.driver.find_element(By.NAME, "image").click()
self.driver.find_element(By.NAME, "image").send_keys("D:\\DAYANA\\1. Proyecto\\Fotos\\Corazón
canino\\IMG_20220524_201717.jpg")
self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()
element = self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary")
actions = ActionChains(self.driver)
actions.move_to_element(element).perform()
element = self.driver.find_element(By.CSS_SELECTOR, "body")
actions = ActionChains(self.driver)
actions.move_to_element(element, 0, 0).perform()
self.driver.find_element(By.CSS_SELECTOR, ".swal2-confirm").click()
```

(se escribieron las “d” para poder poner rápido el ejemplo.)

El resto de las pruebas unitarias se encuentran en los anexos.

Conclusiones parciales

Abordados los temas principales de implementación y las pruebas de aceptación realizadas. En la primera iteración se realizaron 5 tareas de ingeniería, en la segunda 19, y en la tercera 29, dando un total de 53 tareas de investigación.

Posteriormente luego de realizar un total de 56 pruebas de aceptación, se encontraron 3 errores, los cuales ya se encuentran solucionados de forma efectiva. Finalmente, de las pruebas de aceptación realizadas, 50 fueron satisfactorias, finalizando con un correcto cumplimiento de los objetivos propuestos.

Conclusiones

A partir del desarrollo del presente trabajo de diploma se llega a las siguientes conclusiones generales:

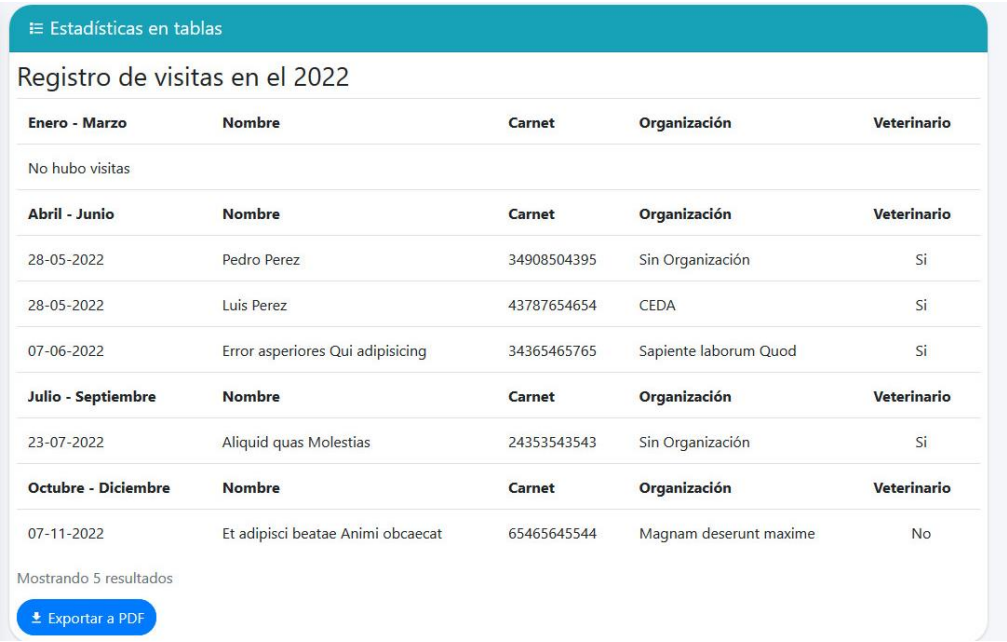
- El estudio de sistemas homólogos permitió conocer sobre las diversas y numerosas aplicaciones cuya función se inclina al bienestar animal, pero ninguna da solución en su totalidad a la problemática planteada en la investigación.
- El estudio de las herramientas y tecnologías como: la metodología XP, Python y Django; permitió la selección más adecuada para el desarrollo de la propuesta de solución.
- La selección de los patrones de diseño *Template* y el patrón arquitectónico MVT permitió el desarrollo del sistema para el control del proyecto extensionista Corazón Canino.
- La realización de pruebas permitió la validación del sistema y que este permita el control de la población animal dentro de la universidad.
- El sistema obtenido, ayuda a mejorar el proyecto extensionista Corazón canino, porque permite controlar e informar en tiempo real todo lo relacionado a los animales comunitarios y eventos a realizar por el propio proyecto.

Recomendaciones

- Creación de una aplicación móvil (APK) con un lector QR integrado.
- Configuración de alertas del sistema para cuando se realice una nueva operación.
- Configuración de alertas del sistema para avisos sobre vacunas o desparasitaciones próximas.
- Crear una API para futuras integraciones con otros sistemas.

Anexos

Historias de usuarios:

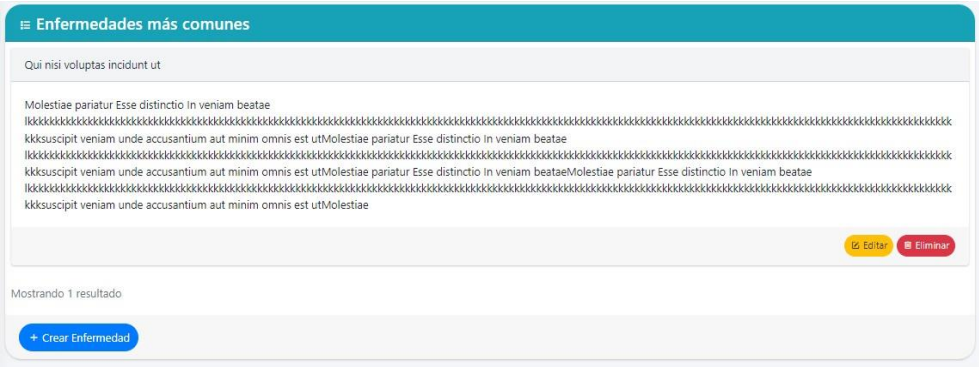
Historia de usuario	
Número: 23	Nombre: Generar reportes integrales
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Medio
Puntos estimados: 0.25	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
<p>Descripción:</p> <p>Se mostrarán en tabla de forma organizada y por <u>Mes</u>, <u>Nombre</u>, <u>Carnet</u>, <u>Organización</u> y <u>Veterinario</u>, todas las visitas que se han realizado en el año, y las que incluso ya están planeadas. Contará además con el botón "Exportar a PDF", el cual le permitirá al administrador realizar una exportación a PDF de lo reflejado en la tabla.</p>	
<p>Interfaz:</p>  <p>The screenshot shows a web interface with a teal header bar containing a hamburger menu icon and the text 'Estadísticas en tablas'. Below the header, the title 'Registro de visitas en el 2022' is displayed. The main content area contains a table with columns: 'Enero - Marzo', 'Nombre', 'Carnet', 'Organización', and 'Veterinario'. The table is divided into sections for different months: 'Enero - Marzo' (with a message 'No hubo visitas'), 'Abril - Junio', 'Julio - Septiembre', and 'Octubre - Diciembre'. Each section contains a list of visits with their respective details. At the bottom of the table, it says 'Mostrando 5 resultados' and there is a blue button labeled 'Exportar a PDF'.</p>	

Historia de usuario	
Número: 15	Nombre: Gestionar enfermedades de animales.
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.25	Iteración asignada: 3

Programador responsable: Dayana Bolaños Banzo

Descripción:
 El sistema mostrará las enfermedades que serán para información del usuario visitante, pero solo el administrador podrá crear, modificar, eliminar, buscar, la cual llevará nombre y descripción.

Interfaz:




Historia de usuario

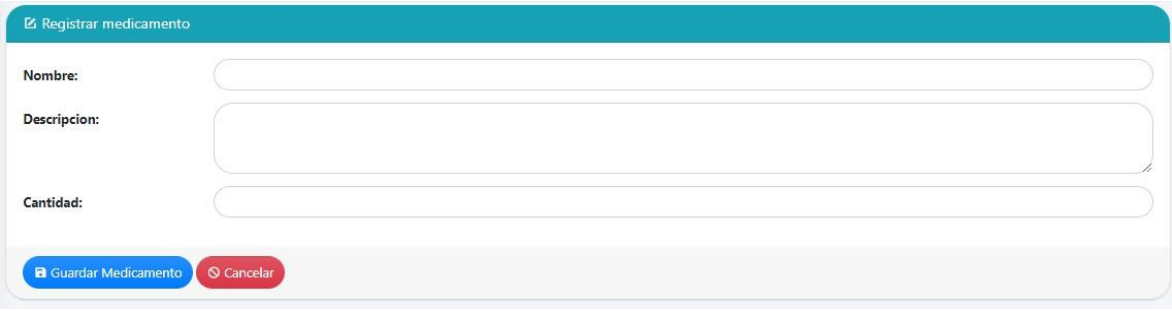
Número: 16	Nombre: Gestionar vacunas
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.25	Iteración asignada: 3

Programador responsable: Dayana Bolaños Banzo












Descripción:
 El sistema permitirá crear, modificar, eliminar, buscar y listar vacunas que se agregarán a las fichas de los animales comunitarios. Estas tendrán como información a agregar y/o modificar: Ficha, Fecha, Producto, Siguiente vacuna.

Interfaz:



Historia de usuario	
Número: 17	Nombre: Gestionar medicamentos.
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.25	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema mostrará los medicamentos que serán para información del usuario visitante, pero solo el administrador podrá crear, modificar, eliminar, buscar o listar el medicamento, el cual llevará <u>Nombre y Descripción</u> .	
Interfaz: 	

Historia de usuario	
Número: 18	Nombre: Gestionar visitas
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema permitirá crear, eliminar, modificar, listar y buscar las visitas a realizar por miembros de las organizaciones y veterinarios. Llevaría nombre, apellidos, carnet de identidad, organización a la que pertenece, si es veterinario y fecha de visita.	
Interfaz:	

+ Crear Listado de visita				
Fecha	Nombre	Carnet	Organización	Opciones
07-11-2022	Et adipisci beatae Animi obcaecat	65465645544	Magnam deserunt maxime	  
23-07-2022	Aliquid quas Molestias	24353543543	Sin Organización	  
07-06-2022	Error asperiores Qui adipiscing	34365465765	Sapiente laborum Quod	  
28-05-2022	Pedro Perez	34908504395	Sin Organización	  
28-05-2022	Luis Perez	43787654654	Sin Organización	  

Mostrando 5 resultados

Historia de usuario













Número: 19	Nombre: Gestionar eventos
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	

Descripción:

El sistema permitirá crear, modificar, eliminar, visualizar y filtrar eventos próximos relacionados al proyecto extensionista como campañas de desparasitación o esterilización.

Sus campos a llenar serán: Tipo de evento, Fecha y horario, Organización que la encabeza, Descripción.

Interfaz:

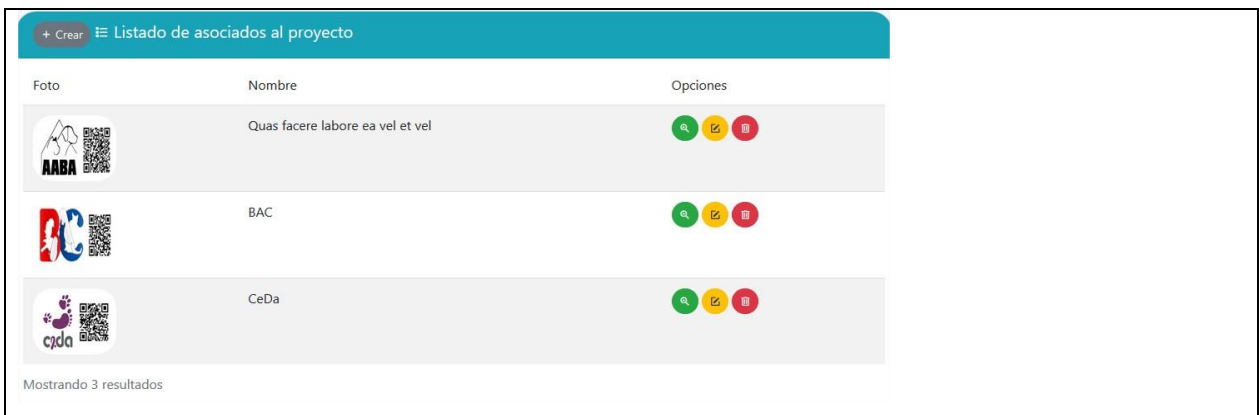
+ Crear Listado de eventos		
dd / mm / aaaa	dd / mm / aaaa	Filtrar
 <p>Excepteur tempora nulla vitae nihil soluta reprehenderit: Tipo: Campaña de desparasitación 26 de Agosto de 2022 a las 05:52</p> <p>  </p>	 <p>Est ut aliquip perferendis nostrum lorem aliquip illo: Tipo: Campaña de esterilización 30 de Junio de 2022 a las 05:51</p> <p>  </p>	 <p>Magna consequatur Eum consequatur officia dolor voluptate nobis: Tipo: Feria de adopciones 26 de Junio de 2022 a las 05:51</p> <p>  </p>

Historia de usuario

Número: 20	Nombre: Gestionar información
Usuario: Administrador	
Prioridad en negocio: Media	Riesgo en desarrollo: Bajo

Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema mostrará imágenes informativas que el administrador subirá regularmente. Se podrá crear la información agregando una «imagen», está se podrá modificar y eliminar. Además tendrá un título y descripción opcional.	
Interfaz:	
	

Historia de usuario	
Número: 21	Nombre: Gestionar Asociados
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.25	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	
Descripción: El sistema mostrará los datos de las organizaciones asociadas al proyecto extensionista, permitirá crear, modificar o eliminar los mismos. Entre los campos a llenar estará: <u>Nombre de la organización</u> , <u>Teléfono</u> , <u>Link de contacto</u> y <u>la imagen insertada con el logo de la organización</u> .	
Interfaz:	



Historia de usuario

Número: 22	Nombre: Gestionar contactos
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.20	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	

Descripción:

El sistema mostrará los datos de los miembros y representantes del proyecto extensionista, permitirá crear, modificar o eliminar los mismos.

Entre los campos a llenar estará: Nombre y apellidos, Facultad o área, Cargo y Contacto.

Interfaz:



Historia de usuario

Número: 24	Nombre: Gestionar desparasitación
Usuario: Administrador	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 0.25	Iteración asignada: 3
Programador responsable: Dayana Bolaños Banzo	

Descripción:

El sistema permitirá crear, modificar, eliminar y filtrar vacunas que se agregarán a las fichas de los animales comunitarios. Estas tendrán como información a agregar y/o modificar: Ficha, Fecha, Tipo (externa o interna, o ambas), Peso (Kg).

Interfaz:

Fecha	Ficha	Tipo	Peso (Kg)	Opciones
09-08-2022	Dolor itaque rerum libero et et minima ut voluptatem	Externa e Interna	56,0	 
08-08-2022	Est elit volupta	Externa	25,0	 
08-08-2022	Dolor itaque rerum libero et et minima ut voluptatem	Interna	23,0	 

Mostrando 3 resultados

Tarjetas CRC

Tarjeta CRC	
Clase: VisitaCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView

Tarjeta CRC	
Clase: VisitaUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView

Tarjeta CRC	
Clase: VisitaDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView

Tarjeta CRC	
Clase: EventoListView	
Responsabilidad:	Colaboración:
get_context_data, get_queryset	ListView

Tarjeta CRC	
Clase: EventoCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView

Tarjeta CRC	
Clase: EventoUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView

Tarjeta CRC	
Clase: EventoDeleteView	

Responsabilidad:	Colaboración:
get_context_data	DeleteView

Tarjeta CRC	
Clase: InformacionListView	
Responsabilidad:	Colaboración:
get_context_data	ListView

Tarjeta CRC	
Clase: InformacionCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView

Tarjeta CRC	
Clase: InformacionUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView

Tarjeta CRC	
Clase: InformacionDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView

Tarjeta CRC	
Clase: ContactoListView	
Responsabilidad:	Colaboración:
get_context_data	ListView

Tarjeta CRC	
Clase: ContactoCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView

Tarjeta CRC	
Clase: ContactoUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView

Tarjeta CRC	
Clase: ContactoDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView

Tarjeta CRC	
Clase: AsociadoListView	
Responsabilidad:	Colaboración:
get_context_data	ListView

Tarjeta CRC	
Clase: AsociadoCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView

Tarjeta CRC	
Clase: AsociadoUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView

Tarjeta CRC	
Clase: AsociadoDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView

Tarjeta CRC	
Clase: EnfermedadListView	
Responsabilidad:	Colaboración:
get_context_data	ListView

Tarjeta CRC	
Clase: EnfermedadCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView, LoginRequiredMixin

Tarjeta CRC	
Clase: EnfermedadUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView, LoginRequiredMixin

Tarjeta CRC	
Clase: EnfermedadDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView, LoginRequiredMixin

Tarjeta CRC	
Clase: DenunciaListView	
Responsabilidad:	Colaboración:
get_context_data, get_queryset	LoginRequiredMixin, ListView

Tarjeta CRC	
Clase: DenunciaUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView, LoginRequiredMixin

Tarjeta CRC	
Clase: DenunciaDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView, LoginRequiredMixin

Tarjeta CRC	
Clase: VacunaListView	

Responsabilidad:	Colaboración:
get_context_data, get_queryset	ListView, LoginRequiredMixin

Tarjeta CRC	
Clase: VacunaCreateView	
Responsabilidad:	Colaboración:
get_context_data	CreateView, LoginRequiredMixin

Tarjeta CRC	
Clase: VacunaUpdateView	
Responsabilidad:	Colaboración:
get_context_data	UpdateView, LoginRequiredMixin

Tarjeta CRC	
Clase: VacunaDeleteView	
Responsabilidad:	Colaboración:
get_context_data	DeleteView, LoginRequiredMixin

Tarjeta CRC	
Clase: DesparasitacionListView	
Responsabilidad:	Colaboración:
get_context_data, get_queryset	ListView, LoginRequiredMixin

Tarjeta CRC	
Clase: DesparasitacionCreateView	
Responsabilidad:	Colaboración:
get_context_data	LoginRequiredMixin, CreateView

Tarjeta CRC	
Clase: DesparasitacionUpdateView	
Responsabilidad:	Colaboración:
get_context_data	LoginRequiredMixin, UpdateView

Tarjeta CRC	
Clase: DesparasitacionDeleteView	
Responsabilidad:	Colaboración:
get_context_data	LoginRequiredMixin, DeleteView

Tarjeta CRC	
Clase: MedicamentoListView	
Responsabilidad:	Colaboración:
get_context_data	ListView

Tarjeta CRC	
Clase: MedicamentoCreateView	
Responsabilidad:	Colaboración:
get_context_data	LoginRequiredMixin, CreateView

Tarjeta CRC	
Clase: MedicamentoUpdateView	
Responsabilidad:	Colaboración:
get_context_data	LoginRequiredMixin, UpdateView

Tarjeta CRC	
Clase: MedicamentoDeleteView	
Responsabilidad:	Colaboración:
get_context_data	LoginRequiredMixin, DeleteView

Tarjeta CRC	
Clase: LoginFormView	
Responsabilidad:	Colaboración:
get_context_data, dispatch	LoginView

Tarjeta CRC	
Clase: LogoutRedirectView	
Responsabilidad:	Colaboración:
dispatch	RedirectView

Tareas de investigación (Iteración 2)

Tarea	
Número de tarea: 11	Número de Historia de usuario: 18
Nombre de la tarea: Crear visitas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 29 de mayo de 2022	Fecha de fin: 30 de mayo de 2022
Programador responsable:	
Descripción: Se implementa un botón, nombrado “Crear Visita”, que al ser presionado accederá al formulario a llenar para dar registro de la nueva visita. Luego deberá estar implementado el botón “Guardar Visita”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de visitas.	

Tarea	
Número de tarea: 12	Número de Historia de usuario: 18
Nombre de la tarea: Modificar visitas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 30 de mayo de 2022	Fecha de fin: 30 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón icono localizado en las “Opciones” en la tabla del Listado de visita, dicho botón amarillo, será el que permita modificar los formularios rellenos anteriormente en el “Crear Visita”. Implementar además el botón “Guardar Visita” (su respectivo icono) el cual guardará los cambios hechos, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de visita.	

Tarea	
Número de tarea: 13	Número de Historia de usuario: 18
Nombre de la tarea: Eliminar visitas	
Tipo de tarea: Dayana Bolaños	Puntos estimados: 0.13
Fecha de inicio: 30 de mayo de 2022	Fecha de fin: 31 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón rojo con el icono de <i>basura</i> para eliminar la visita escogida que aparecerá en la columna de “Opciones” del “Listado de visita”. Al presionar el botón aparecerá la interfaz con el mensaje	

de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.

Tarea	
Número de tarea: 14	Número de Historia de usuario: 18
Nombre de la tarea: Mostrar visitas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 31 de mayo de 2022	Fecha de fin: 31 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón “Mostrar” en la columna de “Opciones”, el cual estará representado con un botón verde y centrado el icono de una lupa. Al presionar el botón se ampliará el modal con toda la información de la visita. Para minimizar el modal se implementa el botón cerrar (X) en la esquina superior derecha y el botón “Cerrar” en el borde inferior derecho.	

Tarea	
Número de tarea: 15	Número de Historia de usuario: 18
Nombre de la tarea: Filtrar visitas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 31 de mayo de 2022	Fecha de fin: 31 de mayo de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un filtro por fecha que permitirá mostrar únicamente las visitas a la universidad en dependencia de las fechas puestas en el filtro.	

Tarea	
Número de tarea: 16	Número de Historia de usuario: 19
Nombre de la tarea: Crear eventos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.11
Fecha de inicio: 31 de mayo de 2022	Fecha de fin: 1 de junio de 2022
Programador responsable:	
Descripción: Se implementa un botón, nombrado “Crear Evento”, que al ser presionado accederá al formulario a llenar para crear el nuevo evento a desarrollar. Luego deberá estar implementado el botón “Guardar Evento”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de eventos.	

--

Tarea	
Número de tarea: 17	Número de Historia de usuario: 19
Nombre de la tarea: Modificar eventos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.11
Fecha de inicio: 1 de junio de 2022	Fecha de fin: 2 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón amarillo nombrado "Editar" el cual estará en el recuadro del evento a modificar. Implementar el botón "Guardar Evento" (su respectivo icono) el cual guardará los cambios hechos en el formulario del evento, o por otra parte el botón "Cancelar" para regresarse a la página del Listado de eventos	

Tarea	
Número de tarea: 18	Número de Historia de usuario: 19
Nombre de la tarea: Eliminar eventos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.11
Fecha de inicio: 2 de junio de 2022	Fecha de fin: 2 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón rojo con el icono de <i>basura</i> para eliminar el evento escogido que aparecerá en el "Listado de eventos". Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón "Sí, aceptar", para cancelar y regresar atrás, implementar el botón "Cancelar".	

Tarea	
Número de tarea: 19	Número de Historia de usuario: 19
Nombre de la tarea: Mostrar eventos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.11
Fecha de inicio: 2 de junio de 2022	Fecha de fin: 3 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón "Ver más", el cual estará representado con un botón verde. Al presionar el botón se ampliará la información estirándose el recuadro del evento mostrando la descripción del mismo. Para volver a la vista anterior sin que la descripción esté asomada, se presiona una vez más el botón verde.	

Tarea	
Número de tarea: 20	Número de Historia de usuario: 19
Nombre de la tarea: Filtrar eventos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.08
Fecha de inicio: 2 de junio de 2022	Fecha de fin: 3 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un filtro por fecha que permitirá mostrar únicamente los eventos en dependencia de las fechas especificadas en el filtro.	

Tarea	
Número de tarea: 21	Número de Historia de usuario: 20
Nombre de la tarea: Crear información	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha de inicio: 3 de junio de 2022	Fecha de fin: 4 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón, nombrado "Crear Información", que al ser presionado accederá al formulario a llenar para crear la nueva información. Luego deberá estar implementado el botón "Guardar Información", el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón "Cancelar" para regresarse a la página del Informaciones.	

Tarea	
Número de tarea: 22	Número de Historia de usuario: 20
Nombre de la tarea: Modificar información	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 4 de junio de 2022	Fecha de fin: 5 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón amarillo nombrado "Editar" el cual estará en el recuadro de la información a modificar. Implementar el botón "Guardar Información" (su respectivo icono) el cual guardará los cambios hechos en el formulario de la información, o por otra parte el botón "Cancelar" para regresarse a la página de Informaciones.	

Tarea

Número de tarea: 23	Número de Historia de usuario: 20
Nombre de la tarea: Eliminar información	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 5 de junio de 2022	Fecha de fin: 6 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón rojo "Eliminar" con el icono de <i>basura</i> , localizada en el recuadro de la información que se desea desechar. Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón "Sí, aceptar", para cancelar y regresar atrás, implementar el botón "Cancelar".	

Tarea	
Número de tarea: 24	Número de Historia de usuario: 20
Nombre de la tarea: Mostrar información	
Tipo de tarea: Desarrollo	Puntos estimados: 0.13
Fecha de inicio: 6 de junio de 2022	Fecha de fin: 07 junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón "Ver más", el cual estará representado con un botón verde. Al presionar el botón se ampliará la información estirándose el recuadro informativo mostrando la descripción del mismo. Para volver a la vista anterior sin que la descripción este asomada, se presiona una vez más el botón verde.	

Tareas de investigación (Iteración 3)

Tarea	
Número de tarea: 25	Número de Historia de usuario: 15
Nombre de la tarea: Crear enfermedad animal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 8 de junio de 2022	Fecha de fin: 8 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón, nombrado "Crear Enfermedad", que al ser presionado accederá al formulario a llenar para crear un nuevo apartado de información de enfermedades más comunes. Luego deberá estar implementado el botón "Guardar Enfermedad", el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón "Cancelar" para regresarse a la página de Enfermedades más comunes	

Tarea	
Número de tarea: 26	Número de Historia de usuario: 15
Nombre de la tarea: Modificar enfermedad animal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 8 de junio de 2022	Fecha de fin: 8 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón amarillo nombrado “Editar” el cual estará en el recuadro de la enfermedad común a modificar. Implementar el botón “Guardar Enfermedad” (su respectivo icono) el cual guardará los cambios hechos en el formulario de la enfermedad, o por otra parte el botón “Cancelar” para regresarse a la página de Enfermedades más comunes.	

Tarea	
Número de tarea: 27	Número de Historia de usuario: 15
Nombre de la tarea: Eliminar enfermedad animal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 8 de junio de 2022	Fecha de fin: 8 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón rojo “Eliminar” con el icono de <i>basura</i> para eliminar el recuadro con la información de la enfermedad escogida la cual se encuentra en el listado de “Enfermedades más comunes”. Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.	

Tarea	
Número de tarea: 28	Número de Historia de usuario: 15
Nombre de la tarea: Mostrar enfermedad	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 8 de junio de 2022	Fecha de fin: 9 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	

Descripción:

Se implementa el botón “Detalles”, el cual estará representado con un botón verde. Al presionar el botón se mostrará toda la información de la información en un modal. Para cerrar dicho modal se implementa el botón característico (X) en la esquina superior derecha del modal, o el botón “Cerrar” en el borde inferior derecho.

Tarea	
Número de tarea: 29	Número de Historia de usuario: 16
Nombre de la tarea: Crear vacunas	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 9 de junio de 2022	Fecha de fin: 9 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	
Se implementa un botón, nombrado “Crear Vacuna”, que al ser presionado accederá al formulario a llenar para crear la nueva vacuna. Luego deberá estar implementado el botón “Guardar Vacuna”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de Vacunas.	

Tarea	
Número de tarea: 30	Número de Historia de usuario: 16
Nombre de la tarea: Modificar vacuna	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 9 de junio de 2022	Fecha de fin: 9 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	
Se implementa un botón icono localizado en las “Opciones” en la tabla del “Listado de vacunas”, dicho botón amarillo, será el que permita modificar los formularios rellenos anteriormente en el “Crear Vacuna”. Implementar además el botón “Guardar Vacuna” (su respectivo icono) el cual guardará los cambios hechos, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de vacunas.	

Tarea	
Número de tarea: 31	Número de Historia de usuario: 16
Nombre de la tarea: Eliminar vacuna	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 9 de junio de 2022	Fecha de fin: 9 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	

Descripción: Se implementa el botón rojo con el icono de <i>basura</i> para eliminar la vacuna escogida que aparecerá en la columna de “Opciones” del “Listado de vacunas”. Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.
--

Tarea	
Número de tarea: 32	Número de Historia de usuario: 16
Nombre de la tarea: Filtrar vacuna	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 9 de junio de 2022	Fecha de fin: 10 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un filtro por fecha que permitirá mostrar únicamente las vacunas en dependencia de las fechas especificadas en el filtro.	

Tarea	
Número de tarea: 33	Número de Historia de usuario: 17
Nombre de la tarea: Crear medicamento	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 10 de junio de 2022	Fecha de fin: 10 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón, nombrado “Crear Medicamento”, que al ser presionado accederá al formulario a llenar para registrar el nuevo medicamento. Luego deberá estar implementado el botón “Guardar Medicamento”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página del Medicamentos disponibles.	

Tarea	
Número de tarea: 34	Número de Historia de usuario: 17
Nombre de la tarea: Modificar medicamento	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 10 de junio de 2022	Fecha de fin: 10 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	

Se implementa un botón amarillo nombrado "Editar" el cual estará en el recuadro del medicamento a modificar. Implementar el botón "Guardar Medicamento" (su respectivo icono) el cual guardará los cambios hechos en el formulario del medicamento, o por otra parte el botón "Cancelar" para regresarse a la página de Medicamentos disponibles.

Tarea	
Número de tarea: 35	Número de Historia de usuario: 17
Nombre de la tarea: Eliminar medicamento	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 10 de junio de 2022	Fecha de fin: 10 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	
Se implementa el botón rojo "Eliminar" con el icono de <i>basura</i> para desechar el recuadro con la información del medicamento localizado en el listado de "Medicamentos disponibles". Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón "Sí, aceptar", para cancelar implementar el botón "Cancelar", lo cual servirá también para ir hacia atrás.	

Tarea	
Número de tarea: 36	Número de Historia de usuario: 17
Nombre de la tarea: Mostrar medicamento	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 10 de junio de 2022	Fecha de fin: 11 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	
Se implementa el botón "Detalles", el cual estará representado con un botón verde. Al presionar el botón se mostrará toda la información de los medicamentos en un modal. Para cerrar dicho modal se implementa el botón característico (X) en la esquina superior derecha del modal, o el botón "Cerrar" en el borde inferior derecho.	

Tarea	
Número de tarea: 37	Número de Historia de usuario: 21
Nombre de la tarea: Crear asociados	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 11 de junio de 2022	Fecha de fin: 11 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	

Se implementa un botón, nombrado “Crear Asociado”, que al ser presionado accederá al formulario a llenar para crear el nuevo asociado al proyecto. Luego deberá estar implementado el botón “Guardar Asociado”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de asociados al proyecto.

Tarea	
Número de tarea: 38	Número de Historia de usuario: 21
Nombre de la tarea: Modificar asociados	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 11 de junio de 2022	Fecha de fin: 11 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	
Se implementa un botón icono localizado en las “Opciones” en la tabla del Listado de asociados al proyecto, dicho botón amarillo, será el que permita modificar los formularios rellenos anteriormente en el “Crear Asociado”. Implementar además el botón “Guardar Asociado” (su respectivo icono) el cual guardará los cambios hechos, o por otra parte el botón “Cancelar” para regresarse a la página del Listado de asociados al proyecto.	

Tarea	
Número de tarea: 39	Número de Historia de usuario: 21
Nombre de la tarea: Eliminar asociados	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 11 de junio de 2022	Fecha de fin: 11 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción:	
Se implementa el botón rojo con el icono de <i>basura</i> para eliminar la asociación escogida que aparecerá en la columna de “Opciones” del “Listado de asociados al proyecto”. Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.	

Tarea	
Número de tarea: 40	Número de Historia de usuario: 21
Nombre de la tarea: Mostrar asociados	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 11 de junio de 2022	Fecha de fin: 12 de junio de 2022

Programador responsable: Dayana Bolaños Banzo
Descripción: Se implementa el botón “Mostrar” en la columna de “Opciones”, el cual estará representado con un botón verde y centrado el icono de una lupa. Al presionar el botón se ampliará el modal con la información faltante de los asociados. Para minimizar el modal se implementa el botón cerrar (X) en la esquina superior derecha, y el botón “Cerrar” en el borde inferior derecho.

Tarea	
Número de tarea: 41	Número de Historia de usuario: 22
Nombre de la tarea: Crear contacto	
Tipo de tarea: Desarrollo	Puntos estimados: 0.05
Fecha de inicio: 12 de junio de 2022	Fecha de fin: 12 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón, nombrado “Crear Contacto”, que al ser presionado accederá al formulario a llenar para crear el nuevo contacto del proyecto. Luego deberá estar implementado el botón “Guardar Contacto”, el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón “Cancelar” para regresarse a la página de Contactos.	

Tarea	
Número de tarea: 42	Número de Historia de usuario: 22
Nombre de la tarea: Modificar contactos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.05
Fecha de inicio: 12 de junio de 2022	Fecha de fin: 12 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón icono localizado en las “Opciones” en la tabla de Contactos, dicho botón amarillo, será el que permita modificar los formularios rellenos anteriormente en el “Crear Contacto”. Implementar además el botón “Guardar Contacto” (su respectivo icono) el cual guardará los cambios hechos, o por otra parte el botón “Cancelar” para regresarse a la página de Contactos.	

Tarea	
Número de tarea: 43	Número de Historia de usuario: 22
Nombre de la tarea: Eliminar contactos	

Tipo de tarea: Desarrollo	Puntos estimados: 0.05
Fecha de inicio: 12 de junio de 2022	Fecha de fin: 12 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón rojo con el icono de <i>basura</i> para eliminar el contacto escogido que aparecerá en la columna de “Opciones” del listado de “Contactos”. Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.	

Tarea	
Número de tarea: 44	Número de Historia de usuario: 22
Nombre de la tarea: Mostrar contactos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.03
Fecha de inicio: 12 de junio de 2022	Fecha de fin: 12 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón “Mostrar” en la columna de “Opciones”, el cual estará representado con un botón verde y centrado el icono de una lupa. Al presionar el botón se ampliará el modal con la información faltante de los contactos. Para minimizar el modal se implementa el botón cerrar (X) en la esquina superior derecha, y el botón “Cerrar” en el borde inferior derecho.	

Tarea	
Número de tarea: 45	Número de Historia de usuario: 12
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06
Fecha de inicio: 12 de junio de 2022	Fecha de fin: 13 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Implementar un botón nombrado “Iniciar sesión” que luego de rellenar los campos de “Usuario” y “Contraseña” permita acceder a la página como administrador con sus respectivos permisos.	

Tarea	
Número de tarea: 46	Número de Historia de usuario: 23
Nombre de la tarea: Generar reportes integrales	
Tipo de tarea: Desarrollo	Puntos estimados: 0.06

Fecha de inicio: 13 de junio de 2022	Fecha de fin: 13 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Los reportes se generarán automáticamente en tiempo real que se agreguen las visitas en el apartado de visitas. Se implementa el botón "Exportar a PDF", el cual permitirá realizar una exportación de la información anual de visitas a la universidad por organizaciones o veterinarios.	

Tarea	
Número de tarea: 48	Número de Historia de usuario: 24
Nombre de la tarea: Crear desparasitación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.05
Fecha de inicio: 13 de junio de 2022	Fecha de fin: 14 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón, nombrado "Crear Desparasitación", que al ser presionado accederá al formulario a llenar para crear el registro de la nueva desparasitación puesta al animal. Luego deberá estar implementado el botón "Guardar Desparasitación", el cual será presionado al terminar de rellenar el formulario, o por otra parte el botón "Cancelar" para regresarse a la página de Desparasitaciones	

Tarea	
Número de tarea: 49	Número de Historia de usuario: 24
Nombre de la tarea: Modificar desparasitación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.05
Fecha de inicio: 14 de junio de 2022	Fecha de fin: 14 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un botón icono localizado en las "Opciones" en la tabla de Desparasitaciones, dicho botón amarillo, será el que permita modificar los formularios rellenos anteriormente en el "Crear Desparasitación". Implementar además el botón "Guardar Desparasitación" (su respectivo icono) el cual guardará los cambios hechos, o por otra parte el botón "Cancelar" para regresarse a la página de Desparasitaciones.	

Tarea	
Número de tarea: 50	Número de Historia de usuario: 24
Nombre de la tarea: Eliminar desparasitación	

Tipo de tarea: Desarrollo	Puntos estimados: 0.05
Fecha de inicio: 14 de junio de 2022	Fecha de fin: 14 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón rojo con el icono de <i>basura</i> para eliminar la ficha animal escogida que aparecerá en la columna de “Opciones” del listado de “Desparasitaciones” Al presionar el botón aparecerá la interfaz con el mensaje de seguridad, para aceptar implementar el botón “Sí, aceptar”, para cancelar implementar el botón “Cancelar”.	

Tarea	
Número de tarea: 51	Número de Historia de usuario: 24
Nombre de la tarea: Filtrar desparasitación	
Tipo de tarea: Desarrollo	Puntos estimados: 0.03
Fecha de inicio: 14 de junio de 2022	Fecha de fin: 14 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa un filtro por fecha que permitirá mostrar únicamente las desparasitaciones en dependencia de las fechas especificadas en el filtro.	

Tarea	
Número de tarea: 52	Número de Historia de usuario: 13
Nombre de la tarea: Cerrar sesión	
Tipo de tarea: Desarrollo	Puntos estimados: 0.03
Fecha de inicio: 14 de junio de 2022	Fecha de fin: 15 de junio de 2022
Programador responsable: Dayana Bolaños Banzo	
Descripción: Se implementa el botón “Cerrar sesión” el cual será presionado para salir de la sesión de administración de la página.	

Tarea	
Número de tarea: 53	Número de Historia de usuario: 14
Nombre de la tarea: Iniciar sesión como “Invitado”	
Tipo de tarea: Desarrollo	Puntos estimados: 0.03
Fecha de inicio: 1 de junio de 2022	Fecha de fin: 15 de junio de 2022

Programador responsable: Dayana Bolaños Banzo
--

Descripción:

Se implementa el botón "Invitado" el cual al ser presionado dará acceso a la página sin los permisos de administrador.
--

Pruebas de aceptación (Iteración II)

Caso de prueba de aceptación	
Código: HU_P18.1.1	Historia de usuario: 18
Nombre: Gestionar visitas: Crear visitas	
Descripción: Prueba de la funcionalidad: Crear visitas en el "Listado de visita"	
Condiciones de ejecución: Ninguna	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede al "Listado de visita" o solo "Visitas" en el apartado de <u>Colaboradores</u> en el menú principal. 2. Se selecciona el botón "Crear Visita" 3. Se muestra el apartado de "Registrar una Ficha" donde se llenará el formulario. 4. Se selecciona el botón "Guardar Visita". 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra en el "Listado de visita" la nueva visita creada. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P18.2.1	Historia de usuario: 18
Nombre: Gestionar visitas: Modificar visitas	
Descripción: Prueba para la funcionalidad: Modificar visitas en el "Listado de visita".	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Ya deberá existir una visita creada que pueda ser modificada. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede al "Listado de visitas" 2. Se selecciona el botón amarillo de la columna de "Opciones" para modificar. 3. Se muestra el formulario al que se le realizarán modificaciones. 4. Se selecciona el botón "Guardar Visita" 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra en el "Listado de visita" la visita ya modificada. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P18.3.1	Historia de usuario: 18
Nombre: Gestionar visitas: Eliminar visitas	
Descripción: Prueba de funcionalidad: Eliminar visitas del "Listado de visita"	

Condiciones de ejecución:
<ul style="list-style-type: none"> Ya deberá existir una visita que pueda ser eliminada.
Pasos de ejecución:
<ol style="list-style-type: none"> Se accede al “Listado de visita” Se selecciona el botón rojo en la columna de “Opciones” para eliminar. Se muestra el mensaje de seguridad. Se selecciona el botón “Si, aceptar” para proseguir en la eliminación.
Resultados esperados:
<ul style="list-style-type: none"> Se elimina del “Listado de visita” la visita seleccionada. Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P18.4.1	Historia de usuario: 18
Nombre: Gestionar visitas: Mostrar visitas	
Descripción: Prueba de funcionalidad: Mostrar la visita del “Listado de visita”	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Ya deberá existir una visita creada para que pueda ser mostrada. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Se accede al “Listado de visita”. Se selecciona el botón verde con el icono de lupa, en la columna de “Opciones” para mostrar. Se muestran todos los datos de la visita en un modal. Se selecciona el botón (X) en la esquina superior derecha para cerrar o el botón “Cerrar” 	
Resultados esperados:	
<ul style="list-style-type: none"> Se muestra la visita seleccionada del “Listado de visita”. Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P18.5.1	Historia de usuario: 18
Nombre: Gestionar visitas: Filtrar visitas	
Descripción: Prueba para la funcionalidad: Filtrar visitas del “Listado de visita”	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Existencia de visitas para filtrarlas. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Se accede al “Listado de visita” en el apartado “Colaboradores”. Se establecen las fechas límites en el filtro de fechas localizado en la esquina superior derecha del “Listado de visita” 	

3. Se selecciona el botón “Filtrar”
Resultados esperados: <ul style="list-style-type: none"> • Se muestran las visitas correspondientes a las fechas correspondientes. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P19.1.1	Historia de usuario: 19
Nombre: Gestionar eventos: Crear eventos	
Descripción: Prueba para la funcionalidad: Crear eventos para el proyecto en el “Listado de eventos”	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a “Eventos” en “Conoce más”. 2. Se selecciona el botón “Crear Evento” 3. Se muestra el apartado de “Registrar un evento” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Evento”. 	
Resultados esperados: <ul style="list-style-type: none"> • Se registra correctamente el evento en el “Listado de eventos” • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P19.2.1	Historia de usuario: 19
Nombre: Gestionar eventos: Modificar eventos	
Descripción: Prueba para la funcionalidad: Modificar eventos del proyecto.	
Condiciones de ejecución: <ul style="list-style-type: none"> • Existencia de eventos. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder a “Eventos” en “Conoce más” 2. Se selecciona el botón amarillo “Editar” del evento a modificar. 3. Se muestra el formulario al que se le realizarán las modificaciones. 4. Se selecciona el botón “Guardar Evento”. 	
Resultados esperados: <ul style="list-style-type: none"> • Se modifica correctamente el evento seleccionado del “Listado de eventos”. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P19.3.1	Historia de usuario: 19

Nombre: Gestionar eventos: Eliminar eventos
Descripción: Prueba para la funcionalidad: Eliminar eventos del proyecto.
Condiciones de ejecución: <ul style="list-style-type: none"> Existencia de eventos que puedan ser eliminados.
Pasos de ejecución: <ol style="list-style-type: none"> Acceder a “Eventos” en “Conoce más”. Se selecciona el botón rojo “Eliminar” del evento a borrar. Se muestra el mensaje de advertencia. Se selecciona el botón “Si, aceptar”.
Resultados esperados: <ul style="list-style-type: none"> Se elimina correctamente el evento seleccionado. Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P19.4.1	Historia de usuario: 19
Nombre: Gestionar eventos: Mostrar eventos	
Descripción: Prueba para la funcionalidad: Mostrar eventos	
Condiciones de ejecución: Existencia de eventos que puedan ser mostrados.	
Pasos de ejecución: <ol style="list-style-type: none"> Acceder a “Eventos” en “Conoce más”. Se selecciona el botón verde “Ver más” del evento a mostrar. Se muestra un modal con toda la información del evento. Se selecciona el botón “Cerrar” o (X) para cerrar el modal. 	
Resultados esperados: <ul style="list-style-type: none"> Se muestra toda la información sobre el evento seleccionado del “Listado de eventos” Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P19.5.1	Historia de usuario: 19
Nombre: Gestionar eventos: Filtrar eventos	
Descripción: Prueba para la funcionalidad: Filtrar eventos.	
Condiciones de ejecución: <ul style="list-style-type: none"> Existencia de eventos que puedan ser filtrados por fechas. 	
Pasos de ejecución: <ol style="list-style-type: none"> Se accede al “Listado de eventos” en el apartado “Conoce más”. 	

2. Se establecen las fechas límites en el filtro de fechas localizado en la esquina superior derecha del “Listado de eventos”
3. Se selecciona el botón “Filtrar”
Resultados esperados:
<ul style="list-style-type: none"> • Se filtran correctamente los eventos mostrándose solo los correspondientes a las fechas. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P20.1.1	Historia de usuario: 20
Nombre: Gestionar información: Crear información	
Descripción: Prueba para la funcionalidad: Crear información	
Condiciones de ejecución: Ninguna	
Pasos de ejecución:	
1. Se accede a “Informaciones” en “Conoce más”.	
2. Se selecciona el botón “Crear Información”	
3. Se muestra el apartado de “Crear Información” donde se llenará el formulario.	
4. Se selecciona el botón “Guardar Información”.	
Resultados esperados:	
<ul style="list-style-type: none"> • Se registra correctamente la información. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P20.2.1	Historia de usuario: 20
Nombre: Gestionar información: Modificar información	
Descripción: Prueba para la funcionalidad: Modificar información	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Existencia de información que pueda ser modificada. 	
Pasos de ejecución:	
1. Acceder a “Informaciones” en “Conoce más”	
2. Se selecciona el botón amarillo “Editar” de la información a modificar.	
3. Se muestra el formulario al que se le realizarán las modificaciones.	
4. Se selecciona el botón “Guardar Información”.	
Resultados esperados:	
<ul style="list-style-type: none"> • Se modifica correctamente la información seleccionada del apartado “Informaciones”. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P20.3.1	Historia de usuario: 20
Nombre: Gestionar información: Eliminar información	
Descripción: Prueba para la funcionalidad: Eliminar información	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Existencia de eventos que puedan ser eliminados. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Acceder a “Informaciones” en “Conoce más”. Se selecciona el botón rojo “Eliminar” de la información a borrar. Se muestra el mensaje de advertencia. Se selecciona el botón “Si, aceptar”. 	
Resultados esperados:	
<ul style="list-style-type: none"> Se elimina correctamente la información seleccionada. Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P20.4.1	Historia de usuario: 20
Nombre: Gestionar información: Mostrar información	
Descripción: Prueba para la funcionalidad: Mostrar información	
Condiciones de ejecución:	
Existencia de información que pueda ser mostrada.	
Pasos de ejecución:	
<ol style="list-style-type: none"> Acceder a “Informaciones” en “Conoce más”. Se selecciona el botón verde “Ver más” de la información a mostrar. Se muestra un modal con toda la información. Se selecciona el botón “Cerrar” o (X) para cerrar el modal. 	
Resultados esperados:	
<ul style="list-style-type: none"> Se muestra toda la información seleccionada. Resultado satisfactorio. 	

Pruebas de aceptación (Iteración III).

Caso de prueba de aceptación	
Código: HU_P15.1.1	Historia de usuario: 15
Nombre: Gestionar enfermedades de animales: Crear enfermedad animal	

Descripción: Prueba para la funcionalidad: Crear enfermedad animal
Condiciones de ejecución: Ninguna
Pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder a “Enfermedades” en “Conoce más” o en el menú principal en “Enfermedades” del apartado <u>Salud</u>. 2. Se selecciona el botón “Crear Enfermedad” 3. Se muestra el apartado de “Registrar Enfermedad” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Enfermedad”.
Resultados esperados: <ul style="list-style-type: none"> • Se registra correctamente la enfermedad entre las “Enfermedades más comunes”. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P15.2.1	Historia de usuario: 15
Nombre: Gestionar enfermedades de animales: Modificar enfermedad animal	
Descripción: Prueba para la funcionalidad: Modificar enfermedad animal	
Condiciones de ejecución: <ul style="list-style-type: none"> • Existencia de una enfermedad que pueda ser modificada. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder a “Enfermedades” en “Conoce más” o en el menú principal en “Enfermedades” del apartado <u>Salud</u>. 2. Se selecciona el botón amarillo “Editar” de la enfermedad a modificar. 3. Se muestra el formulario al que se le realizarán las modificaciones. 4. Se selecciona el botón “Guardar Enfermedad”. 	
Resultados esperados: <ul style="list-style-type: none"> • Se realiza la modificación de la enfermedad correctamente. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P15.3.1	Historia de usuario: 15
Nombre: Gestionar enfermedades de animales: Eliminar enfermedad animal	
Descripción: Prueba para la funcionalidad: Eliminar enfermedad animal.	
Condiciones de ejecución: <ul style="list-style-type: none"> • Existencia de las enfermedades que puedan ser eliminadas. 	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Acceder a “Enfermedades” en “Conoce más” o en el menú principal en “Enfermedades” del apartado <u>Salud</u>. 2. Se selecciona el botón rojo “Eliminar” de la enfermedad a borrar. 3. Se muestra el mensaje de advertencia. 4. Se selecciona el botón “Si, aceptar”.
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se elimina correctamente la enfermedad seleccionada. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P15.4.1	Historia de usuario: 15
Nombre: Gestionar enfermedades de animales: Mostrar enfermedad	
Descripción: Prueba para la funcionalidad: Mostrar enfermedad	
Condiciones de ejecución: Existencia de enfermedades que puedan ser mostradas.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Acceder a “Enfermedades” en “Conoce más” o en el menú principal en “Enfermedades” del apartado <u>Salud</u>. 2. Se selecciona el botón verde “Ver más” de la enfermedad a mostrar. 3. Se muestra un modal con toda la información de la enfermedad. 4. Se selecciona el botón “Cerrar” o (X) para cerrar el modal. 	
Resultados esperados: <ul style="list-style-type: none"> • Se muestra toda la información sobre la enfermedad seleccionada entre las “Enfermedades más comunes” • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P16.1.1	Historia de usuario: 16
Nombre: Gestionar vacunas: Crear vacunas	
Descripción: Prueba para la funcionalidad: Crear vacunas para las fichas registradas	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a “Vacunas” en “Principal” o “Vacunas” en <u>Ficha Animal</u> del menú principal. 2. Se selecciona el botón “Crear Vacuna” 3. Se muestra el apartado de “Crear Vacuna” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Vacuna”. 	

Resultados esperados: <ul style="list-style-type: none"> • Se crea correctamente la vacuna de la ficha animal especificada. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P16.2.1	Historia de usuario: 16
Nombre: Gestionar vacunas: Modificar vacuna	
Descripción: Prueba para la funcionalidad: Modificar vacuna	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Existencia de vacuna para poder modificarla. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a “Vacunas” en “Principal”. 2. Se selecciona el botón amarillo en la columna de “Opciones” para modificar. 3. Se muestra el formulario al que se le realizarán modificaciones. 4. Se selecciona el botón “Guardar Vacuna” 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra en el “Listado de vacunas” la vacuna ya modificada. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P16.3.1	Historia de usuario: 16
Nombre: Gestionar vacunas: Eliminar vacuna	
Descripción: Prueba para la funcionalidad: Eliminar vacuna	
Condiciones de ejecución:	
Existencia de vacuna para poder eliminar.	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a “Vacunas” en “Principal”. 2. Se selecciona el botón rojo en la columna de “Opciones” para eliminar. 3 Se muestra el mensaje de seguridad. 4 Se selecciona el botón “Si, aceptar” para proseguir en la eliminación. 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se elimina de “Listados de vacunas” la vacuna del animal seleccionada. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P16.4.1	Historia de usuario: 16

Nombre: Gestionar vacunas: Filtrar vacuna
Descripción: Prueba para la funcionalidad: Filtrar vacuna.
Condiciones de ejecución: <ul style="list-style-type: none"> Existencia de vacunas que puedan ser filtradas por fechas.
Pasos de ejecución: <ol style="list-style-type: none"> Se accede al “Listados de vacunas” en el apartado “Principal”. Se establecen las fechas límites en el filtro de fechas localizado en la esquina superior derecha del “Listados de vacunas” Se selecciona el botón “Filtrar”
Resultados esperados: <ul style="list-style-type: none"> Se realiza el filtrado con éxito mostrando las vacunas correspondientes a las fechas. Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P17.1.1	Historia de usuario: 17
Nombre: Gestionar medicamentos: Crear medicamento	
Descripción: Prueba para la funcionalidad: Crear medicamento	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none"> Acceder a “Medicamentos” en “Conoce más” o en el menú principal en “Medicamentos” del apartado <u>Salud</u>. Se selecciona el botón “Crear Medicamento” Se muestra el apartado de “Registrar medicamentos donde se llenará el formulario. Se selecciona el botón “Guardar Medicamento”. 	
Resultados esperados: <ul style="list-style-type: none"> Se registra correctamente el medicamento en los “Medicamentos disponibles”. Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P17.2.1	Historia de usuario: 17
Nombre: Gestionar medicamentos: Modificar medicamento	
Descripción: Prueba para la funcionalidad: Modificar medicamento	
Condiciones de ejecución: <ul style="list-style-type: none"> Existencia de medicamentos que puedan ser modificados. 	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Acceder a “Medicamentos” en “Conoce más” o en el menú principal en “Medicamentos” del apartado <u>Salud</u>. 2. Se selecciona el botón amarillo “Editar” del medicamento a modificar. 3. Se muestra el formulario al que se le realizarán las modificaciones. 4. Se selecciona el botón “Guardar Enfermedad”.
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se realiza la modificación del medicamento correctamente. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P17.3.1	Historia de usuario: 17
Nombre: Gestionar medicamentos: Eliminar medicamento	
Descripción: Prueba para la funcionalidad: Eliminar medicamento	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Existencia de medicamentos que puedan ser eliminados. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder a “Medicamentos” en “Conoce más” o en el menú principal en “Medicamentos” del apartado <u>Salud</u>. 2. Se selecciona el botón rojo “Eliminar” de la enfermedad a borrar. 3. Se muestra el mensaje de advertencia. 4. Se selecciona el botón “Si, aceptar”. 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se elimina correctamente el medicamento seleccionado. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P17.4.1	Historia de usuario: 17
Nombre: Gestionar medicamentos: Mostrar medicamento	
Descripción: Prueba para la funcionalidad: Mostrar medicamento	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Existencia de medicamentos que puedan ser mostrados. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1-Acceder a “Medicamentos” en “Conoce más” o en el menú principal en “Medicamentos” del apartado <u>Salud</u>. 2-Se selecciona el botón verde “Detalles” del medicamento a mostrar. 3-Se muestra un modal con toda la información del medicamento. 	

4-Se selecciona el botón “Cerrar” o (X) para cerrar el modal.
Resultados esperados: <ul style="list-style-type: none"> • Se muestra toda la información sobre el medicamento seleccionado entre los “Medicamentos disponibles” • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P21.1.1	Historia de usuario: 21
Nombre: Gestionar asociados: Crear asociados	
Descripción: Prueba para la funcionalidad: Crear asociados	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a “Asociados” en “Colaboradores” 2. Se selecciona el botón “Crear Asociado” 3. Se muestra el apartado de “Crear Asociado” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Asociado”. 	
Resultados esperados: <ul style="list-style-type: none"> • Se crea correctamente el nuevo asociado de la organización. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P21.2.1	Historia de usuario: 21
Nombre: Gestionar asociados: Modificar asociados	
Descripción: Prueba para la funcionalidad: Modificar asociados	
Condiciones de ejecución: Existencia de asociados para poder modificarlos.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Se accede a “Asociados” en “Colaboradores”. 2. Se selecciona el botón amarillo en la columna de “Opciones” para modificar. 3. Se muestra el formulario al que se le realizarán modificaciones. 4. Se selecciona el botón “Asociado ” 	
Resultados esperados: <ul style="list-style-type: none"> • Se muestra en “Listado de asociados al proyecto” el asociado ya modificado. • Resultado satisfactorio. 	

Caso de prueba de aceptación

Código: HU_P21.3.1	Historia de usuario: 21
Nombre: Gestionar asociados: Eliminar asociados	
Descripción: Prueba para la funcionalidad: Eliminar asociados	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Existencia de asociados para poder eliminarlos. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Se accede a “Asociados” en “Principal”. Se selecciona el botón rojo en la columna de “Opciones” para eliminar. Se muestra el mensaje de seguridad. Se selecciona el botón “Si, aceptar” para proseguir en la eliminación. 	
Resultados esperados:	
<ul style="list-style-type: none"> Se elimina del “Listado de asociados al proyecto” el asociado seleccionado. Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P21.4.1	Historia de usuario: 21
Nombre: Gestionar asociados: Mostrar asociados	
Descripción: Prueba para la funcionalidad: Mostrar asociados	
Condiciones de ejecución:	
<ul style="list-style-type: none"> Existencia de asociados para poder mostrarlos. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> Se accede a “Asociados” en “Colaboradores”. Se selecciona el botón verde con el icono de lupa en la columna de “Opciones” del “Listado de asociados al proyecto”. Se muestra un modal con toda la información referente al asociado. Se selecciona el botón “Cerrar” o (X) para cerrar el modal. 	
Resultados esperados:	
<ul style="list-style-type: none"> Se muestra toda la información del asociado correctamente. Resultados satisfactorios. 	

Caso de prueba de aceptación	
Código: HU_P22.1.1	Historia de usuario: 22
Nombre: Gestionar contactos: Crear contacto	
Descripción: Prueba para la funcionalidad: Crear contacto	
Condiciones de ejecución: Ninguna	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Se accede a “Contactos” en “Colaboradores” 2. Se selecciona el botón “Crear Contacto” 3. Se muestra el apartado de “Crear Contacto” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Contacto”.
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se crea correctamente el nuevo contacto de la organización. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P22.2.1	Historia de usuario: 22
Nombre: Gestionar contactos: Modificar contactos	
Descripción: Prueba para la funcionalidad: Modificar contactos	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • Existencia de contacto para poder modificarlo. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Se accede a “Contactos” en “Colaboradores”. 2. Se selecciona el botón amarillo en la columna de “Opciones” para modificar. 3. Se muestra el formulario al que se le realizarán modificaciones. 4. Se selecciona el botón “Guardar Contacto” 	
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se muestra en “Contactos” el contacto modificado. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P22.3.1	Historia de usuario: 22
Nombre: Gestionar contactos: Eliminar contactos	
Descripción: Prueba para la funcionalidad: Eliminar contactos	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • Existencia de contactos para poder eliminarlos. 	
<p>Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Se accede a “Contactos” en “Colaboradores”. 2. Se selecciona el botón rojo en la columna de “Opciones” para eliminar. 3. Se muestra el mensaje de seguridad. 4. Se selecciona el botón “Si, aceptar” para proseguir en la eliminación. 	

Resultados esperados: <ul style="list-style-type: none"> • Se elimina de “Contactos” el contacto del proyecto seleccionado. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P22.4.1	Historia de usuario: 22
Nombre: Gestionar contactos: Mostrar contacto	
Descripción: Prueba para la funcionalidad: Mostrar contacto	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Existencia de contacto para poder mostrarlo. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Se accede a “Contactos” en “Colaboradores”. 2. Se selecciona el botón verde con el icono de lupa en la columna de “Opciones” de los “Contactos”. 3. Se muestra un modal con toda la información referente al contacto. 4. Se selecciona el botón “Cerrar” o (X) para cerrar el modal. 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se muestra toda la información del contacto correctamente. • Resultados satisfactorios. 	

Caso de prueba de aceptación	
Código: HU_P23	Historia de usuario: 23
Nombre: Generar reportes integrales	
Descripción: Prueba para la funcionalidad: Generar reportes integrales del proyecto.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Haber registrado visitas en el último año. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Crear visitas en el apartado “Listado de visitas” en “Colaboradores” 2. Acceder a “Reportes” la tabla “Registro de visitas en el # (número del año actual)” 3. Seleccionar botón “Exportar a PDF” 	
Resultados esperados:	
<ul style="list-style-type: none"> • Se exporta correctamente el registro de visitas del año actual. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P24.1.1	Historia de usuario: 24
Nombre: Gestionar desparasitación: Crear desparasitación	

Descripción: Prueba para la funcionalidad: Crear desparasitación para las fichas registradas
Condiciones de ejecución: Ninguna
Pasos de ejecución: 1. Se accede a “Desparasitaciones” en “Principal” o “Desparasitaciones” en <u>Ficha Animal</u> del menú principal. 2. Se selecciona el botón “Crear Desparasitación” 3. Se muestra el apartado de “Crear desparasitación” donde se llenará el formulario. 4. Se selecciona el botón “Guardar Desparasitación”.
Resultados esperados: <ul style="list-style-type: none"> • Se crea correctamente la desparasitación de la ficha animal especificada. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P24.2.1	Historia de usuario: 24
Nombre: Gestionar desparasitación: Modificar desparasitación	
Descripción: Prueba para la funcionalidad: Modificar desparasitación	
Condiciones de ejecución: <ul style="list-style-type: none"> • Existencia de desparasitación para poder modificarla. 	
Pasos de ejecución: 1. Se accede a “Desparasitaciones” en “Principal”. 2. Se selecciona el botón amarillo en la columna de “Opciones” para modificar. 3. Se muestra el formulario al que se le realizarán modificaciones. 4. Se selecciona el botón “Guardar Desparasitación”	
Resultados esperados: <ul style="list-style-type: none"> • Se muestra en “Desparasitaciones” la desparasitación ya modificada. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P24.3.1	Historia de usuario: 24
Nombre: Gestionar desparasitación: Eliminar desparasitación	
Descripción: Prueba para la funcionalidad: Eliminar desparasitación	
Condiciones de ejecución: <ul style="list-style-type: none"> • Existencia de desparasitación para poder eliminarla. 	
Pasos de ejecución: 1. Se accede a “Desparasitaciones” en “Principal”. 2. Se selecciona el botón rojo en la columna de “Opciones” para eliminar.	

<p>3. Se muestra el mensaje de seguridad.</p> <p>4. Se selecciona el botón “Si, aceptar” para proseguir en la eliminación.</p>
<p>Resultados esperados:</p> <ul style="list-style-type: none"> • Se elimina de “Desparasitaciones” la desparasitación del animal seleccionada. • Resultado satisfactorio.

Caso de prueba de aceptación	
Código: HU_P24.4.1	Historia de usuario: 24
Nombre: Gestionar desparasitación: Filtrar desparasitación	
Descripción: Prueba para la funcionalidad: Filtrar desparasitación en “Desparasitaciones”	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Existencia de desparasitaciones realizadas. 	
Pasos de ejecución:	
<p>1. Se accede al “Desparasitaciones” en el apartado “Principal”.</p> <p>2. Se establecen las fechas límites en el filtro de fechas localizado en la esquina superior derecha de “Desparasitaciones”</p> <p>3. Se selecciona el botón “Filtrar”</p>	
Resultados esperados:	
<ul style="list-style-type: none"> • Se filtran correctamente las desparasitaciones mostrando únicamente las afines a las fechas especificadas. • Resultado satisfactorio. 	

Caso de prueba de aceptación	
Código: HU_P13	Historia de usuario: 13
Nombre: Cerrar sesión	
Descripción: Prueba para la funcionalidad: Cerrar sesión como administrador.	
Condiciones de ejecución:	
<ul style="list-style-type: none"> • Estar autenticado. 	
Pasos de ejecución:	
<p>1. Seleccionar el botón “Cerrar sesión”</p>	
Resultados esperados:	
<ul style="list-style-type: none"> • Se cierra correctamente la sesión. • Resultado satisfactorio. 	

Caso de prueba de aceptación

Código: HU_P14	Historia de usuario: 14
Nombre: Iniciar sesión como invitado	
Descripción: Prueba para la funcionalidad: Iniciar sesión como invitado de la página.	
Condiciones de ejecución: Ninguna	
Pasos de ejecución: <ol style="list-style-type: none">1. Acceder a la página de “Corazón Canino”2. Seleccionar botón “Invitado”	
Resultados esperados: <ul style="list-style-type: none">• Se accede correctamente a la página “Corazón Canino”• Resultado satisfactorio.	

Pruebas de aplicación web

Crear visita

	Command	Target	Value
✓ Crear visita*	1 ✓ open	http://127.0.0.1:8000/visitas/add/	
	2 ✓ set window size	944x693	
	3 ✓ click	id=id_nombre	
	4 ✓ type	id=id_nombre	Laura
	5 ✓ type	id=id_apellido	Castillo
	6 ✓ type	id=id_edad	25
	7 ✓ type	id=id_telefono	5635458756
	8 ✓ type	id=id_ci	99051225364
	9 ✓ type	id=id_organizacion	PASOS
	10 ✓ click	css=.bg-gradient-primary	

Crear visita (con datos incorrectos)

1	✓ open	http://127.0.0.1:8000/visitas/add/	
2	✓ set window size	944x693	
3	✓ click	id=id_nombre	
4	✓ type	id=id_nombre	25
5	✓ type	id=id_apellido	Lopez
6	✓ type	id=id_edad	52
7	✓ type	id=id_telefono	53024561
8	✓ type	id=id_ci	99051523624
9	✓ click	id=id_organizacion	
10	✓ type	id=id_organizacion	PASOS

11	✓ click	css=.bg-gradient-primary	
12	✓ click	css=.swal2-confirm	
13	✓ click	id=id_nombre	
14	✓ type	id=id_nombre	Lora
15	✓ click	css=.bg-gradient-primary	

Eliminar desparasitación

		Command	Target	Value
✓ eliminar desparasitación*	1	✓ open	/	
	2	✓ set window size	944x693	
	3	✓ click	css=.nav-item:nth-child(1) > .nav-link	
	4	✓ click	linkText=Fichas	
	5	✓ click	css=.mdi-menu	
	6	✓ click	css=.desparasitacion-sidebar > p	
	7	✓ click	css=tr:nth-child(1) .btn-danger > .mdi	
	8	✓ click	css=.btn-danger	

Modificar visita

		Command	Target	Value
✓ Modificar visita*	1	✓ open	/	
	2	✓ set window size	944x693	
	3	✓ click	css=.col-12:nth-child(2) a:nth-child(2)	
	4	✓ click	css=tr:nth-child(1) .btn-warning > .mdi	
	5	✓ click	id=id_edad	
	6	✓ type	id=id_edad	35
	7	✓ click	css=.bg-gradient-primary	

Generar gráfico

		Command	Target	Value
✓ Generar gráfico*	1	✓ open	/	
	2	✓ set window size	944x693	
	3	✓ click	css=.nav-item:nth-child(1) > .nav-link	
	4	✓ click	css=.chart-sidebar > p	

Crear medicamento (con información incompleta)

✓ Crear medicamento*		http://127.0.0.1:8000	
	Command	Target	Value
1	✓ open	/	
2	✓ click	css=.col-12:nth-child(3).text-gray-dark:nth-child(3) > a	
3	click	linkText=Crear Medicamento	
4	click	id=id_nombre	
5	type	id=id_nombre	5631
6	click	id=id_descripcion	
7	type	id=id_descripcion	Usamos este metodo para informarl es sobre esta prueba en cuestión
8	click	css=.bg-gradient-primary	
9	click	css=.swal2-confirm	

Pruebas unitarias

Modificar denuncia (con datos incompletos - satisfactorio)

```
class TestModificardenunciacondatosincompletos():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_modificardenunciacondatosincompletos(self):
        self.driver.get("http://127.0.0.1:8000/denuncia/list")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.CSS_SELECTOR, "tr:nth-child(3) .btn-warning > .mdi").click()
        self.driver.find_element(By.ID, "id_ubicacion").click()
        self.driver.find_element(By.ID, "id_ubicacion").click()
        self.driver.find_element(By.ID, "id_ubicacion").click()
        self.driver.find_element(By.ID, "id_ubicacion").send_keys("Universidad ")
        self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()
        self.driver.find_element(By.CSS_SELECTOR, ".swal2-confirm").click()
```

Prueba Eliminar denuncia

```
class TestEliminardenuncias():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_eliminardenuncias(self):
        self.driver.get("http://127.0.0.1:8000/denuncia/list")
```



```

self.driver.set_window_size(1382, 744)
self.driver.find_element(By.CSS_SELECTOR, "tr:nth-child(3) .btn-danger > .mdi").click()
self.driver.find_element(By.CSS_SELECTOR, ".btn-danger").click()

```

Prueba: Crear contacto con datos incorrectos (prueba satisfactoria)

```

class TestCrearcontactocondatosincorrectos():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_crearcontactocondatosincorrectos(self):
        self.driver.get("http://127.0.0.1:8000/contacto/add/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.ID, "id_nombre").click()
        self.driver.find_element(By.ID, "id_nombre").send_keys("56")
        self.driver.find_element(By.ID, "id_apellido").send_keys("56")
        self.driver.find_element(By.ID, "id_area").click()
        dropdown = self.driver.find_element(By.ID, "id_area")
        dropdown.find_element(By.XPATH, "//option[. = 'Facultad 1']").click()
        self.driver.find_element(By.CSS_SELECTOR, "option:nth-child(2)").click()
        self.driver.find_element(By.ID, "id_cargo").click()
        self.driver.find_element(By.ID, "id_cargo").send_keys("Presidente")
        self.driver.find_element(By.ID, "id_telefono").click()
        self.driver.find_element(By.ID, "id_telefono").send_keys("53054361")
        self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()
        self.driver.find_element(By.CSS_SELECTOR, ".swal2-confirm").click()

```

Prueba crear visita con datos correctos

```

class TestCrearvisita():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):
        self.driver.quit()

    def test_crearvisita(self):
        self.driver.get("http://127.0.0.1:8000/")
        self.driver.set_window_size(1382, 744)
        self.driver.find_element(By.LINK_TEXT, "Visitas").click()
        self.driver.find_element(By.LINK_TEXT, "Crear Visita").click()
        self.driver.find_element(By.ID, "id_nombre").click()
        self.driver.find_element(By.ID, "id_nombre").send_keys("Laura")
        self.driver.find_element(By.ID, "id_apellido").click()
        self.driver.find_element(By.ID, "id_apellido").send_keys("Contreras")
        self.driver.find_element(By.ID, "id_edad").send_keys("15")
        self.driver.find_element(By.ID, "id_telefono").send_keys("5236458522")
        self.driver.find_element(By.ID, "id_ci").send_keys("56584623298")
        self.driver.find_element(By.ID, "id_organizacion").send_keys("Ceda")
        self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()

```

Prueba Crear visita con datos incorrectos (satisfactoria)

```

class TestCrearvisitacondatosincorrectos():
    def setup_method(self, method):
        self.driver = webdriver.Firefox()
        self.vars = {}

    def teardown_method(self, method):

```

```
self.driver.quit()

def test_crearvisitacondatosincorrectos(self):
    self.driver.get("http://127.0.0.1:8000/visitas/add/")
    self.driver.set_window_size(1382, 744)
    self.driver.find_element(By.ID, "id_nombre").click()
    self.driver.find_element(By.ID, "id_nombre").send_keys("52")
    self.driver.find_element(By.ID, "id_apellido").send_keys("Perez")
    self.driver.find_element(By.ID, "id_edad").send_keys("45")
    self.driver.find_element(By.ID, "id_telefono").send_keys("5656523555")
    self.driver.find_element(By.ID, "id_ci").send_keys("5889998775")
    self.driver.find_element(By.ID, "id_organizacion").send_keys("CEDA")
    self.driver.find_element(By.CSS_SELECTOR, ".bg-gradient-primary").click()
    self.driver.find_element(By.CSS_SELECTOR, ".swal2-confirm").click()
```

Referencias

1. **García, Francisco María.** misanimales. [En línea] 04 de septiembre de 2019. <https://misanimales.com/leyes-proteccion-animales-mundo/> .
2. **CUBADEBATE.** *Cubadebate.* [En línea] 26 de febrero de 2021. <http://www.cubadebate.cu/noticias/2021/02/26/cuba-aprueba-decreto-ley-de-bienestar-animal/>.
3. **Ecured.** Ecured. *Método de análisis histórico-lógico.* [En línea] 05 de 06 de 2022. https://www.ecured.cu/M%C3%A9todo_de_an%C3%A1lisis_hist%C3%B3rico-l%C3%B3gico.
4. **QuestionPro.** QuestionPro. *Qué es la investigación documental.* [En línea] 06 de 05 de 2022. <https://www.questionpro.com/blog/es/investigacion-documental/>.
5. **Tesis y máster.** *Estado del arte: Qué es y ejemplo.* [En línea] 2022. <https://tesisymasters.com.co/estado-del-arte-ejemplo/>.
6. **Uriarte, Julia Máxima.** Caracteristicas.co. *Sistema de información.* [En línea] 26 de agosto de 2021. <https://www.caracteristicas.co/sistema-de-informacion/>.
7. **Adobe.** Adobe. *Aspectos básicos de una página web.* [En línea] 4 de mayo de 2021. <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html>.
8. **UnitagQR.** UnitagQR. *Qué es un código QR?* [En línea] <https://www.unitag.io/es/qrcode/what-is-a-qrcode>.
9. —. UnitagQR. *?Cuáles son los diferentes tipos de código QR?* [En línea] <https://www.unitag.io/es/qrcode/what-are-the-different-types-of-qrcodes>.
10. **OIE.** Organización mundial de sanidad animal (OIE). *Bienestar animal.* [En línea] 2021. <https://www.oie.int/es/que-hacemos/sanidad-y-bienestar-animal/bienestar-animal/#ui-id-1>.
11. —. Organización Mundial de sanidad animal(OIEA). *Acceso en línea al código terrestre.* [En línea] 2021. https://www.oie.int/es/que-hacemos/normas/codigos-y-manuales/acceso-en-linea-al-codigo-terrestre/?id=169&L=1&htmfile=titre_1.7.htm.

12. *Introduction to the recommendations for animal welfare*. Health, World Organization of Animal. Paris, Francia : s.n., 2008. World Organization for Animal Health (OIE),. págs. 235-236.
13. EFEverde, Redacción. EFE:Verde. *Alerta Animal, una App para denunciar vía móvil casos de maltrato*. [En línea] 03 de 04 de 2014. <https://www.efeverde.com/noticias/alerta-animal-una-app-para-denunciar-via-movil-casos-de-maltrato/>.
14. Boffill, Anylisec Grau. TVYumuri. *BACuba, una app para el bienestar animal*. [En línea] 21 de 02 de 2022. <https://www.tvyumuri.cu/tecnologia/bacuba-una-app-para-el-bienestar-animal/>.
15. Rivero, Adyz Lien. VISTAR. *BACuba: primera aplicación móvil por el bienestar animal en Cuba*. [En línea] 10 de 02 de 2022. <https://vistarmagazine.com/bacuba-primera-aplicacion-movil-por-el-bienestar-animal-en-cuba/>.
16. Carlemany, Universitat. Universitat Carlemany. *Metodologías de desarrollo de software*. [En línea] 06 de abril de 2022. <https://www.universitatcarlemany.com/actualidad/metodologias-de-desarrollo-de-software>.
17. GINZO. Ginzo. *Tipos de metodología clásica en desarrollo de software*. [En línea] <https://ginzo.tech/blog/metodologia-clasica-desarrollo-software/>.
18. IONOS. Digital IONOS. *El modelo en cascada: desarrollo secuencial de software*. [En línea] 19 de 03 de 2019. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>.
19. Ecured. [En línea] 05 de 05 de 2022. https://www.ecured.cu/Proceso_unificado_de_desarrollo.
20. *A Comparison between Agile and Traditional Software Development Methodologie*. Awad, M. A. Australia : s.n., 2005, Western Australia. : School of Computer Science y software Engineering, págs. vid. págs. 13,14.
21. blogspot. Con la tecnología de Blogger . *Metodología*. [En línea] 21 de 11 de 2017. <http://socorrozegarra.blogspot.com/2017/11/cuadro-comparativo-modelo-xp-scrum-rup.html>.

22. Ecured. *Programación Extrema o XP*. [En línea] 05 de 05 de 2022. https://www.ecured.cu/Programaci%C3%B3n_Extrema_o_XP.
23. @JASHKOTHARI1. Geeks for geeks. *Python vs php*. [En línea] 30 de 05 de 2022. <https://www.geeksforgeeks.org/python-vs-php/>.
24. School, Tokio. Tokio School. [En línea] 20 de 05 de 2020. <https://www.tokioschool.com/noticias/curso-python-web-disenar-web/>.
25. Bautista García, Iván Jahel. servnet. *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?* [En línea] 30 de 03 de 2021. <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programaci%C3%B3n-de-una-aplicaci%C3%B3n-web>.
26. Ecured. *Django*. [En línea] 13 de 05 de 2022. <https://www.ecured.cu/Django>.
27. django. *Templates*. [En línea] 13 de 05 de 2022. <https://docs.djangoproject.com/en/4.0/topics/templates/>.
28. Visual Studio Code. [En línea] 21 de 04 de 2022. <https://code.visualstudio.com/docs>.
29. Codementor. *Best Text Editor? Atom vs Sublime vs Visual Studio Code vs Vim*. [En línea] 27 de diciembre de 2016. <https://codementorio.medium.com/best-text-editor-atom-vs-sublime-vs-visual-studio-code-vs-vim-85b651d36665>.
30. Capterra. *postgresql*. [En línea] 21 de 04 de 2022. <https://www.capterra.do/software/48791/postgresql>.
31. Arimetrics. *Qué es Firefox*. [En línea] 21 de 04 de 2022. <https://www.arimetrics.com/glosario-digital/firefox>.
32. ConceptoDefinición. *Google Chrome*. [En línea] 18 de 07 de 2019. <https://conceptodefinicion.de/google-chrome/>.
33. ayudaley. *Qué es PostgreSQL y sus principales ventajas*. [En línea] 21 de 04 de 2022. https://ayudaleyprotecciondatos.es/bases-de-datos/que-es-postgresql-ventajas/#Que_es_PostgreSQL.
34. Guru99. *Functional vs Non Functional Requirements*. [En línea] 12 de 02 de 2020. <https://www.guru99.com/functional-vs-non-functional-requirements.html>.

35. Scrum Mexico. [En línea] 02 de 08 de 2018. <https://scrum.mx/informate/historias-de-usuario>.
36. Extreme programming. *Documentación*. [En línea] 29 de 05 de 2022. <https://iswugxp.wordpress.com/documentacion-de-xp/>.
37. Constaín Rengifo, Sylvia Cristina, y otros. *G.SIS.04 Guía de arquitectura de soluciones tecnológicas*. 2019.
38. programador clic. [En línea] 22 de 05 de 2022. <https://programmerclick.com/article/41921755798/>.
39. profile. profile. *?Qué son los patrones de diseño de software?* [En línea] 24 de junio de 2020. https://profile.es/blog/patrones-de-diseno-de-software/#%C2%BFQue_son_los_patrones_de_diseno_design_patterns.
40. django. django. *Templates*. [En línea] 05 de 06 de 2022. <https://docs.djangoproject.com/en/4.0/topics/templates/>.
41. Django. Django. *Templates*. [En línea] 10 de 10 de 2022. <https://docs.djangoproject.com/es/4.1/topics/templates/>.
42. Cartagena99. Ingeniería de software II . *Tema 6 Patrones de diseño*. [En línea] 2011. <https://www.cartagena99.com/recursos/alumnos/apuntes/Patrones%20de%20Diseno.pdf>.
43. Carrasco, José Carlos. DocPlayer. *Patrones: bajo acoplamiento, alta cohesión, experto y creador*. [En línea] 2021. <https://docplayer.es/207734180-Patrones-bajo-acoplamiento-alta-cohesion-experto-y-creador.html>.
44. Amazon. Qué es el modelado de datos? *Amazon*. [En línea] 15 de 10 de 2022. <https://aws.amazon.com/es/what-is/data-modeling/>.
45. DjangoGirl. Tutorial de Django Girls. *Modelos en Django*. [En línea] 15 de 10 de 2022. https://tutorial.djangogirls.org/es/django_models/.
46. Luzkarly Vázquez, Vanessa Daboin . Ingeniería del Software . *Prueba de Aplicaciones Web*. [En línea] 17 de 10 de 2017. [Citado el: 08 de 11 de 2022.] <http://ingenieriadelsoftware-uvm.blogspot.com/2017/10/prueba-de-aplicaciones-web.html>.

47. Erik Anaya. Inmediatum. *¿Qué es Selenium y para qué sirve?* [En línea] 20 de 05 de 2020. [Citado el: 08 de 11 de 2022.] <https://inmediatum.com/blog/piensa-digital/que-es-selenium-y-para-que-sirve/>.
48. Pittet, Sten. Atlassian. *Los distintos tipos de pruebas de software* . [En línea] 07 de 11 de 2022. <https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>.
50. Council, Farm Animal Welfare. *Second Report on Priorities for Research and Development in Farm Animal Welfare*. Londres : s.n., 1992, 1993.
51. Manteca, X., Mainau, E. y Temple, D. FAWEC. *¿Qué es el bienestar animal?* [En línea] <https://www.fawec.org/es/fichas-tecnicas/23-bienestar-general/21-que-es-el-bienestar-animal>.
52. digité. digité. *¿Qué es la metodologías ágil?- Descripción general del desarrollo de software ágil y modelos ágiles.* [En línea] 2022. <https://www.digite.com/es/agile/metodologia-agil/>.
53. Ortiz, Marlady. Ingeniería de Software. Modelos de Desarrollo de Software. *Modelo incremental*. [En línea] 2017. <http://isw-udistrital.blogspot.com/2012/09/ingenieria-de-software-i.html>.
54. Universidades, Santander. Santander Becas. *Metodologías de software de desarrollo de software ¿qué son?* [En línea] 21 de 12 de 2020. <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>.
55. ComparaSoftware. ComparaSoftware. *Metodología XP: La Mejor Vía para el Desarrollo de Software.* [En línea] 08 de agosto de 2020. <https://blog.comparasoftware.com/metodologia-xp/>.
56. —. ComparaSoftware. *Programación Extrema: Pros y Contras.* [En línea] 21 de 08 de 2020. <https://blog.comparasoftware.com/programacion-extrema-ventajas-desventajas/>.
57. Capterra. [En línea] 21 de 04 de 2022. <https://www.capterra.do/compare/166985/145716/draw-io/vs/visual-paradigm>.
58. Mozilla. Resources for Developers. *¿Qué es JavaScript?* [En línea] 21 de 04 de 2022. https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript.

59. Castro Gallego, Juan David. Platzi. *React vs. Angular vs. Vue vs. Svelte | ¿Cómo elegir tu próxima herramienta frontend? ¿Cuál es mejor?* [En línea] 08 de 2021. <https://platzi.com/blog/react-angular-vue-svelte/>.
60. tiThink Technology Consulting. *7 razones para utilizar React.* [En línea] 14 de 11 de 2018. <https://www.tithink.com/es/2018/11/14/7-razones-para-utilizar-react/>.
61. epitech. *NestJS: ¿Qué es y por qué es tan interesante este framework Node.js?* [En línea] 19 de 02 de 2022. <https://www.epitech-it.es/nestjs-que-es/>.
62. Ecured. [En línea] 12 de 05 de 2022. <https://www.ecured.cu/Python>.
63. Carmona, Juan García. Juan García Carmona. *GRASP: Alta cohesión y bajo acoplamiento* . [En línea] 07 de 09 de 2012. <https://juan-garcia-carmona.blogspot.com/2012/09/grasp-alta-cohesion-y-bajo-acoplamiento.html>.
64. QAlified. *Introducción a Selenium Testing.* [En línea] 12 de 03 de 2022. <https://qalified.com/es/introduccion-a-selenium-testing/>.