**Entrega Final Proyecto BookPet**

**POLITÉCNICO GRANCOLOMBIANO.**

**Andrés Esteban García Ch.**

**Laura Daniela Solaque Susatama**

**Juan Norrison Salazar Sáenz**

**Yuri Tatiana Pineda Pinzón**

**21 de noviembre de 2018**

**Tabla de contenido**

1. [Nombre](#h.gjdgxs) del proyecto & Descripción.
2. [Objetivos.](#h.jfy6w6z61ip0)
3. Requerimientos.
4. Diseño
5. Metodología SCRUM
6. Prototipo Final
7. Lecciones Aprendidas

# Nombre del proyecto & Descripción

**Nombre Del Proyecto:**

BookPet.

**Descripción:**

Nuestra idea surge porque en los últimos años La comunidad pet friendly, ha crecido como espuma y ha logrado que la discriminación y el maltrato animal disminuya significativamente. El furor en el país y el mundo por nuestras mascotas ha sido tal, que restaurantes y centros comerciales han optado por abrir sus puertas a los pequeños acompañantes como una estrategia para atraer clientela. Incluso, algunos restaurantes en Bogotá han empezado a ofrecer menús gratuitos para ellos.

Teniendo en cuenta que las mascotas también se han involucrado más en la vida diaria del ser humano, ellos han sido muy importantes en el desarrollo social, intelectual y afectivo de los niños lo cual se ve reflejado positivamente en sus etapas posteriores, especialmente en su relación armoniosa con otros seres humanos, también los niños aprenden con sus mascotas que hay otras formas de comunicarse aparte del lenguaje verbal.

Los que convivimos con animales sabemos que criarlos es casi como criar a un hijo a pequeña escala, y por esto, dicha crianza requiere un compromiso firme y los conocimientos adecuados para que sea saludable; no todos contamos con el tiempo para ir todos los días a una veterinaria, necesitamos algo que nos pueda aportar de manera útil y económica, y como una ayuda para lograrlo nunca viene mal, aquí nace la necesidad de crear para ustedes una aplicación que le será útil para tratar con sus mascotas. Nuestra aplicación cuenta con diferentes módulos, donde les permitirá compartir, socializar y disfrutar de ella, de manera eficaz e innovadora.

# Objetivos

**Objetivo General:**

Generar una comunidad para los dueños de mascotas, la idea principal de la aplicación es que los diferentes usuarios puedan compartir e interactuar en un espacio que adicional los mantendrá al tanto de los últimos eventos y las últimas noticias de interés en relación a sus mascotas.

Vale la pena mencionar que el espacio será una aplicación móvil para dispositivos Android.

**Objetivos Específicos:**

Generar un sistema de información orientado a los dueños de mascotas.

Crear un sistema de información que permita mantener a los usuarios informados de los diferentes eventos para sus mascotas.

Generar un sistema de información con dos roles principales administrador y usuario, el administrador tendrá permisos CRUD sobre todo el sistema mientras que el usuario podrá visualizar todos los módulos de la página pero solo podrá realizar lo siguiente:

* Subir fotos de su mascota con ciertos datos.
* Eliminar fotos de su perfil.
* Subir clasificados.
* Enviar mensajes a otros usuarios.
* Ver los perfiles de otros usuarios.

Desarrollar un sistema de información que permita a los usuarios subir fotos de sus mascotas y en la misma poder cargar información, así mismo poder ver esta información (de él mismo y de otros usuarios).

# Requerimientos

Los siguientes son los requerimientos funcionales de la aplicación, es importante la definición de cada uno de estos ya que estos traducen el contrato o el alcance que tendrá el proyecto en general, así mismo describe los compromisos con los usuarios finales.

|  |  |
| --- | --- |
| **Categoría** | **Requerimiento** |
|  | El sistema debe permitir a los diferentes usuarios ingresar mediante Facebook o Gmail |
| El sistema debe permitir cerrar sesión al usuario |
|  |  |
| Administrativo | El sistema debe permitir al administrador registrar evento |
| El sistema debe permitir al administrador modificar evento |
| El sistema debe permitir al administrador consultar evento |
| El sistema debe permitir al administrador inactivar evento |
| El sistema debe permitir al administrador publicar información sobre competencias caninas |
| El sistema debe permitir al administrador consultar la información de las competencias caninas |
| El sistema debe permitir al administrador modificar la información de las competencias caninas |
| El sistema debe permitir al administrador inactivar competencias caninas |
| El sistema debe permitir al administrador consultar el historial de todos los usuarios |
| El sistema debe permitir al administrador inactivar los usuarios |
|  |  |
| usuarios | El sistema debe permitir al usuario, ingresar por medio de Facebook o Google |
| El sistema debe permitir al usuario registrar una o varias mascotas |
| El sistema debe permitir al usuario publicar fotos de su mascota |
| El sistema debe permitir al usuario publicar información acerca de la fotografía |
| El sistema debe permitir al usuario eliminar fotos, que hayan sido publicadas de su mascota |
| El sistema debe permitir al usuario realizar publicaciones en los clasificados |
| El sistema debe permitir al usuario modificar las publicaciones en sus clasificados |
| El sistema debe permitir al usuario eliminar las publicaciones en sus clasificados |
| El sistema debe permitir al usuario visualizar las publicaciones sobre eventos. |
| EL sistema debe permitir al usuario enviar mensajes a los diferentes usuarios que se encuentren interactuando en la aplicación |
| El sistema debe permitir al usuario visualizar las fotos publicadas por los otro usuarios |
|  |  |
| Eventos | El sistema debe permitir crear eventos |
| El sistema debe permitir modificar eventos |
| El sistema debe permitir eliminar eventos |
| El sistema debe permitir clasificar un evento dentro de las siguientes categorías: Clasificados, Jornadas de adopción y Competencias caninas |
|  |  |
| Chat | El sistema debe permitir la interacción entre dos usuarios, por medio de mensajes de texto. |
|  |  |
| Historial | El sistema debe permitir visualizar el historial de la mascota |
| El sistema debe permitir modificar el historial de la mascota |
| El sistema debe permitir inactivar el historial de la mascota |
|  |  |
| Publicaciones | El sistema debe permitir visualizar las publicaciones de las macotas de los diferentes usuarios |
|  |

**Historias De Usuario:**

Una historia de usuario es una representación de un [requisito](https://es.wikipedia.org/wiki/Requerimiento_(sistemas)) escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las [metodologías de desarrollo ágiles](https://es.wikipedia.org/wiki/Metodolog%C3%ADa_%C3%A1gil) para la especificación de requisitos.

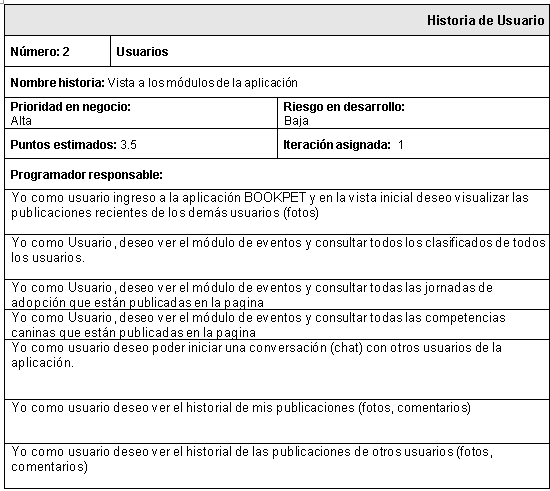
****

Ilustración 1: Historia De Usuario 1

**Diagrama de bases de datos:**

Un modelo de base de datos muestra la estructura lógica de la base, incluidas las relaciones y limitaciones que determinan cómo se almacenan los datos y cómo se accede a ellos. Los modelos de bases de datos individuales se diseñan en base a las reglas y los conceptos de cualquier modelo de datos más amplio que los diseñadores adopten.

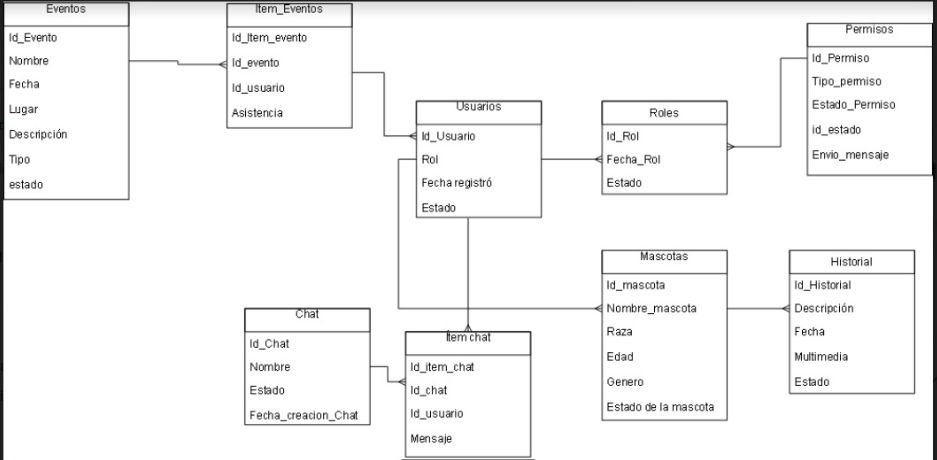
****

Ilustración 2: Diagrama De Base De Datos

**Detalle de requerimientos a través de diagramas de UML: secuencia, actividad, estados o casos de uso detallados:**

**Casos de uso:**

Un caso de uso es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. Los personajes o entidades que participarán en un diagrama de caso de uso se denominan actores.

Los siguientes son los casos de uso básicos en relación al proyecto BookPet.

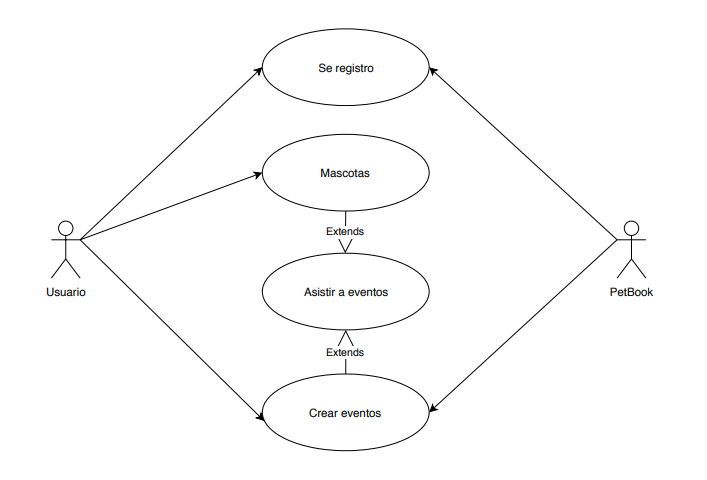


Ilustración 3: Caso De Uso Usuarios

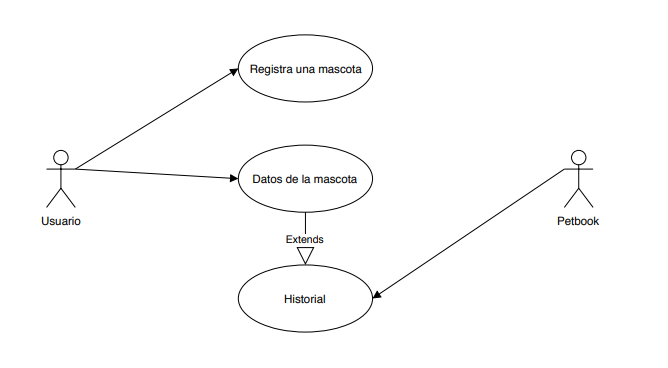


Ilustración 4: Caso De Uso Mascotas

**Diagrama De Estado:**

Un diagrama de estados, es un tipo de diagrama de comportamiento en el Lenguaje Unificado de Modelado (UML) que muestra transiciones entre diversos objetos.

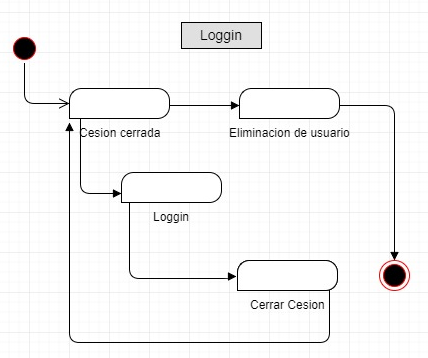


Ilustración 5: Diagrama De Estado Loggin

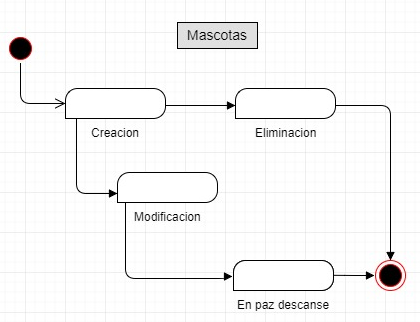


Ilustración 6: Diagrama De Estado Mascotas

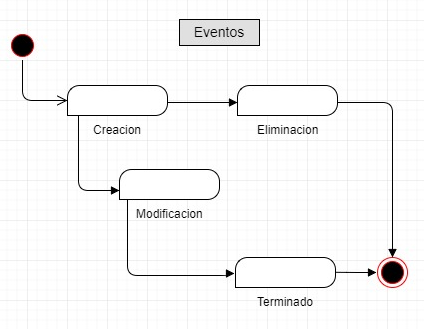


Ilustración 7: Diagrama De Estado Eventos

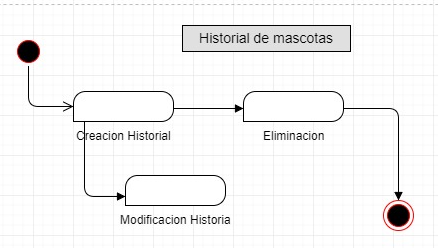


Ilustración 8: Diagrama De Estado Historial De Mascotas

**Diagrama De Secuencia:**

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. A menudo es útil para complementar a un [diagrama de clases](https://es.wikipedia.org/wiki/Diagrama_de_clases), pues el diagrama de secuencia se podría describir de manera informal como "el diagrama de clases en movimiento".

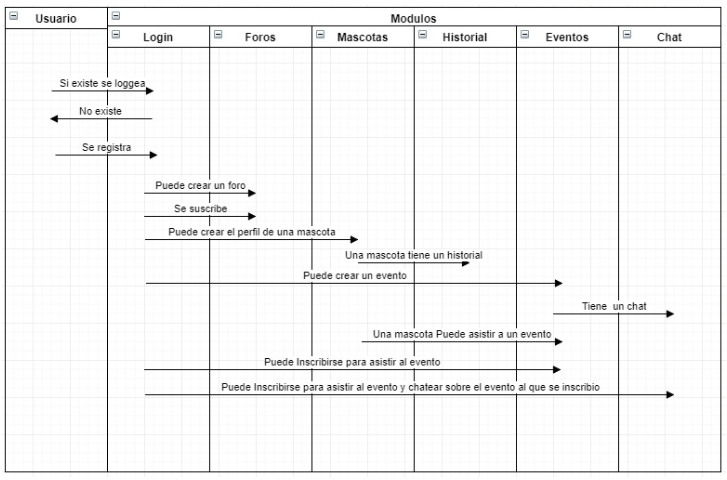


Ilustración 9: Diagrama De Secuencia

# Diseño

**Diagrama De Componentes:**

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes.

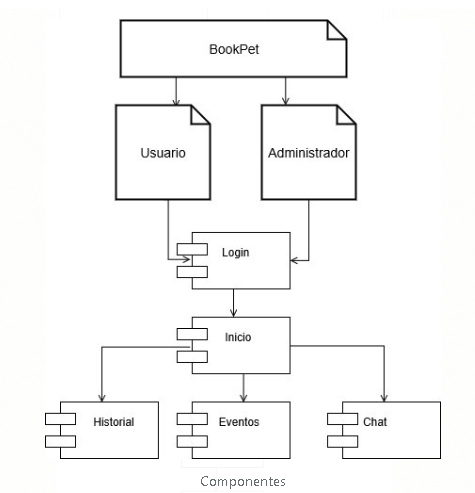


Ilustración 10: Diagrama De Componentes

**Diagrama De Despliegue:**

modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Un Nodo es un elemento de hardware o software.

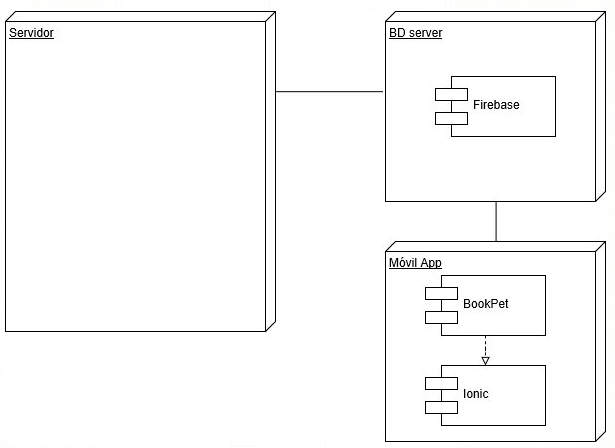


Ilustración 11: Diagrama De Despliegue

**Mockups:**

**Mockups Iniciales:**

Las siguientes imágenes muestran un bosquejo del diseño que tendrá la aplicación Bookpet, como se menciono anteriormente la aplicación será móvil y estará enfocada en dispositivos Android.

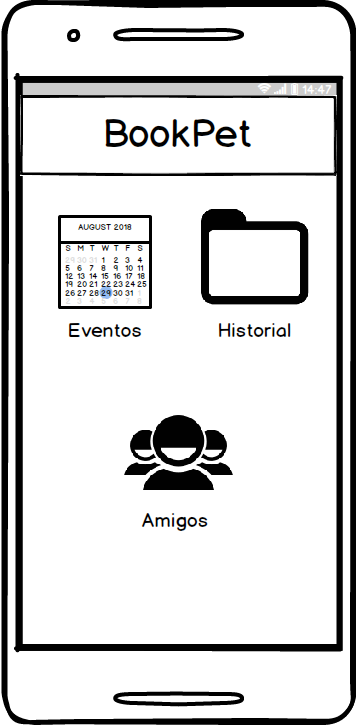


Ilustración 4: Mockup Módulos



Ilustración 3: Mockup Login



Ilustración 6: Mockup Eventos

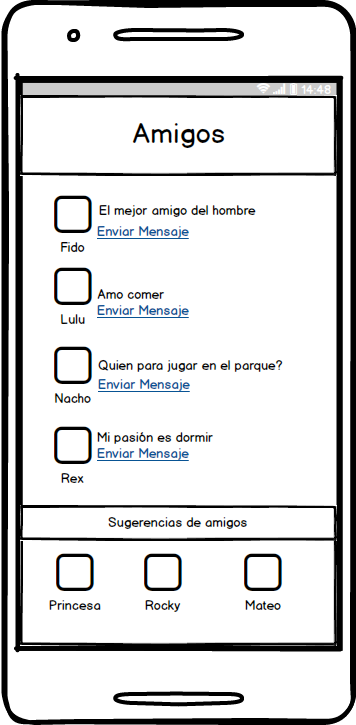


Ilustración 5: Mockup Socialización en línea



Ilustración 8: Mockup Competencias Caninas



Ilustración 7: Mockup Jornadas De Adopción



Ilustración 9: Mockup Clasificados

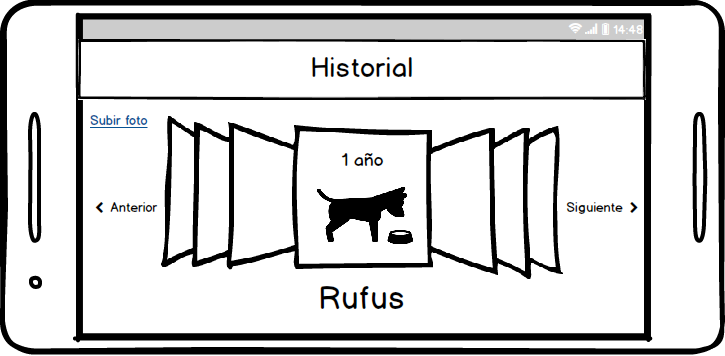


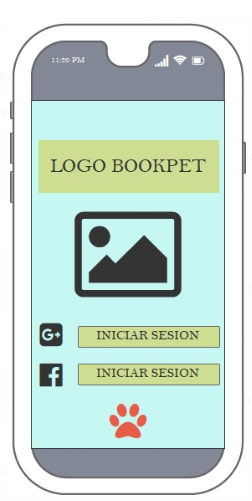
Ilustración 10: Mockup Competencias Caninas

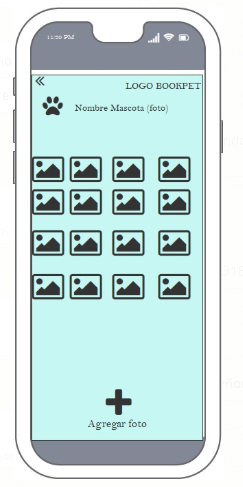
**Mockups Finales:**













# Metodología SCRUM

**Enlace a GitHub:**

<https://github.com/aegarcia1/Ingenier-a-del-Software---Proyecto>

**Product Backlog:**

https://github.com/aegarcia1/Ingenier-a-del-Software---Proyecto/issues

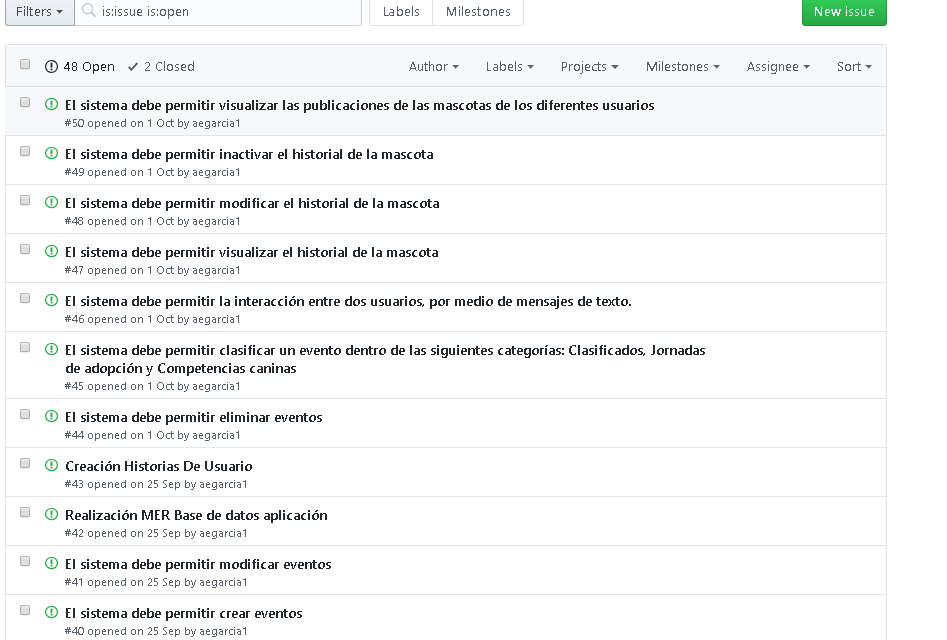


Ilustración 12: Backlog

**Backlog de cada sprint:**

Se detallan en cada una de las actas.

https://github.com/aegarcia1/Ingenier-a-del-Software---Proyecto/tree/master/Actas\_Seguimiento

**Gráfica Burndown:**

**Zenhub:**

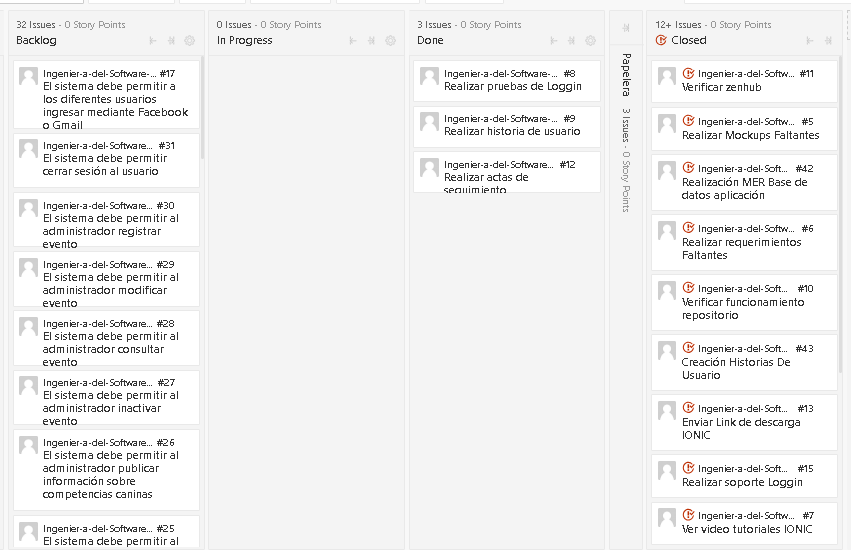
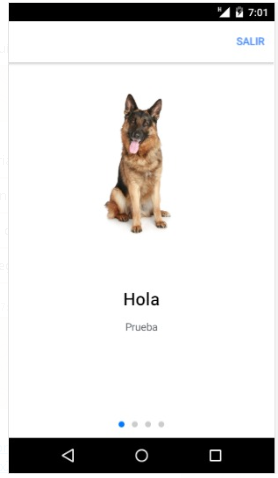
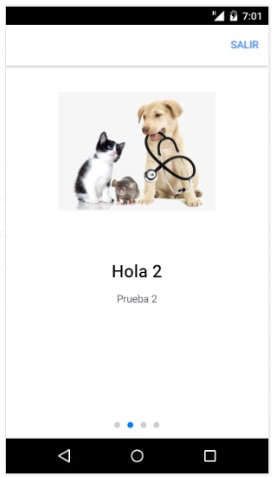


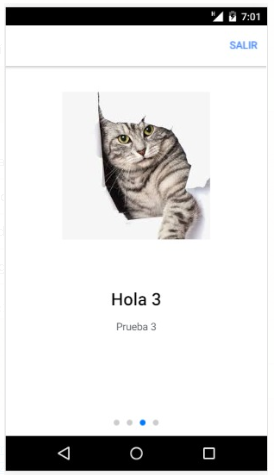
Ilustración 13: Tareas ZenHub

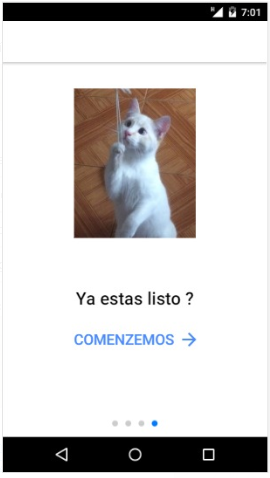
# Prototipo Final

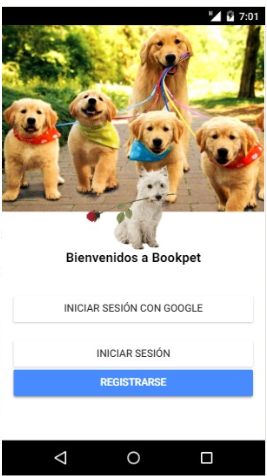
**Aplicación:**

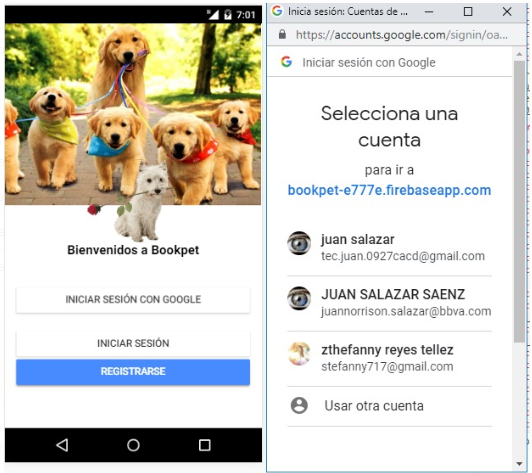


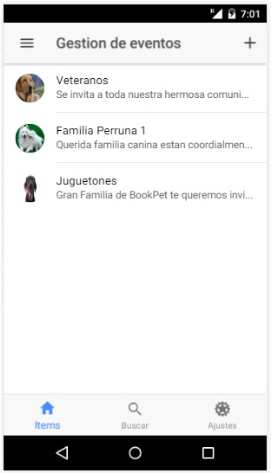


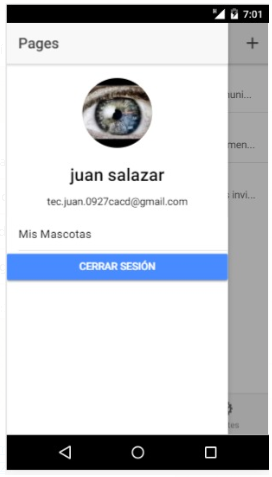


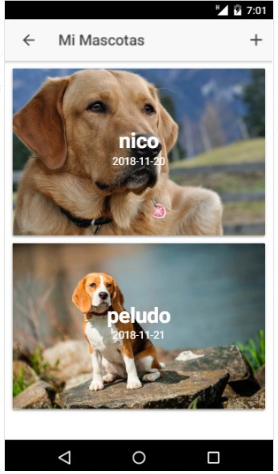


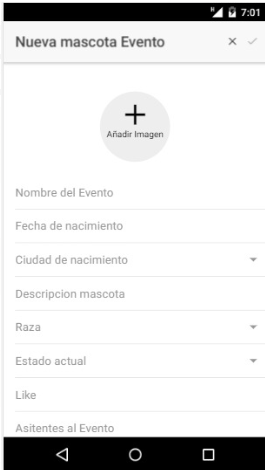




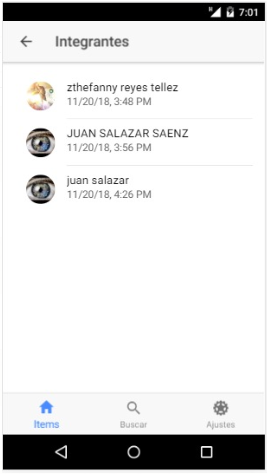


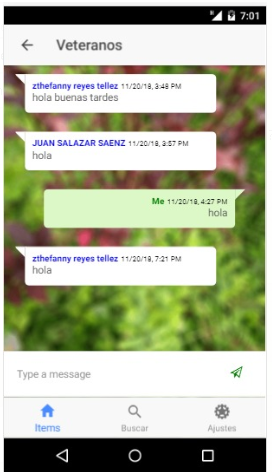


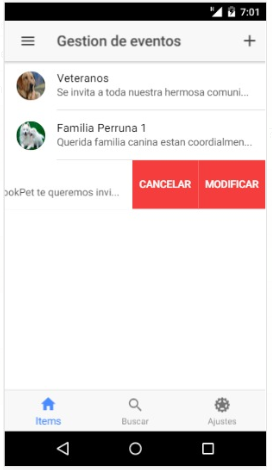


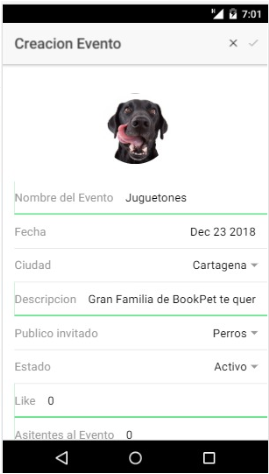


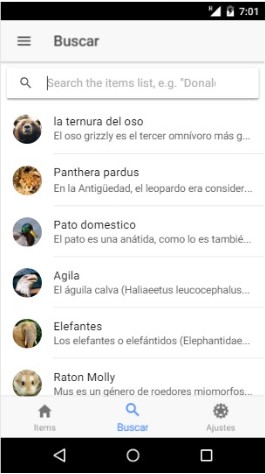












**Trabajo Futuro & Posibles Mejoras:**

A continuación, se presentan algunas mejoras y futuros que se pueden desarrollar como resultado de este proyecto, por exceder el alcance del proyecto, no han podido ser tratados con la suficiente profundidad. Además, se sugieren algunos desarrollos específicos para apoyar y mejorar el modelo y metodología propuestos. Entre las posibles mejoras y trabajos futuros se destacan:

• Realizar el estudio con la población total del personal que tiene mascotas o interés por animales, de esta forma se tendrá un resultado 100% representativo y no se dejará por fuera información que pudiera ser valiosa para la evolución de la aplicación.

• Mejorar la aplicación, con una interface más completa y amigable para el usuario, una sección de búsqueda por palabras clave de las dudas y áreas de especialización, así como un apartado de contacto.

• Capturar la mayor cantidad de perfiles de personal con información completa y llevar un detalle del monitoreo de las visitas y razones por las cuales consultaron el portal a manera de retroalimentación para enriquecer la información y funcionalidad de la aplicación

• Es necesario establecer notificación cuando haya algún tipo de eventos, publicaciones o foros de interés para el usuario

• Permitir tener chat privado, ya que actualmente es un chat publico entre las personas que se encuentren interesadas en los eventos vigentes

# Lecciones Aprendidas

**Requerimientos:** Durante el transcurso del proyecto se realizó un par de re-definiciones de estos requerimientos, esta fase es la base de proyecto , el contrato y los compromisos definidos con el cliente, para este tipo de metodología SCRUM, es vital definir cada uno de estos requerimientos con el Product Owner, sin embargo para el desarrollo del proyecto, nosotros mismos cumplíamos con este rol. Como se menciono anteriormente fue necesario realizar nuevas definiciones por lo cual para futuros proyectos sería bueno poder dedicar más tiempo en esta definición para no causar re trabajos.

Adicionalmente es importante definir que requerimientos se cubren en cada uno de los sprints, esto es bueno para darle un seguimiento al dueño del producto y así mismo poder ir cerrando requerimientos en el tablero de tareas, adicionalmente este cierre de requerimientos alimenta la gráfica BurnDown.

**Scrum:** La metodología utilizada o el marco de trabajo utilizado es muy interesante debido a que no solamente aplica para proyectos que involucren el desarrollo de Software, si no para proyectos en general, definitivamente es una metodología en que el trabajo en equipo e individual tiene mucha importancia, por la parte grupal cada persona perteneciente al Scrum Team debe tener un grado de responsabilidad muy alto, debido a que sus tareas pueden complementar o dar paso a otras fases que realizaran sus compañeros, por lo cual los tiempos y los compromisos es muy importantes que sean cumplidos.

Para el desarrollo del proyecto se tuvieron varios problemas con las fechas de entrega y asumir las responsabilidades en cada frente del proyecto, ya sea desarrollo, documentación, diseño, seguimiento entre otras estas dificultades se fueron resolviendo a medida que se utilizaban los conductos regulares correctos y los artefactos adecuados, ya sea la reunión diaria, el tablero de tareas y el seguimiento y apoyo que realizaba el rol de Scrum Master.

Adicionalmente otro componente esencial es el que cumple el Scrum Master, este rol es importante para gestionar requerimientos y para ayudar a resolver impedimentos que pueda tener el equipo, adicional es importante cumplir con varios roles dentro del proyecto, esto debido a la rotación de personal, esto ayuda a no perder habilidades necesarias para concluir el proyecto. Adicional fue bueno que todo el grupo de trabajo cumpliera con el rol de Scrum Master sin embargo el empalme del Scrum master actual con el anterior gastaba un poco de tiempo adicional.

Otro aspecto para mejorar es la definición y las responsabilidades del Sprint, esto debido a que se incluían muchas más tareas de las que el grupo podía cumplir, por lo cual se debe ser muy consciente con el alcance que se le da a la definición del Sprint, adicional tener en cuenta que dentro del tiempo de sprint se debe incluir la retrospectiva del mismo y las pruebas necesarias para demostrar su fiabilidad.

Para el tema de las actas, se debe tener en cuenta que una reunión diaria significa un acta nueva en el repositorio, por lo cual se debe seleccionar un formato sencillo y fácil de manejar, esto para no gastar tiempos adicionales que se pueden dedicar a actividades más críticas del proyecto.

**Desarrollo de software:** Este fue uno de los procesos más complicados en el desarrollo del proyecto, es importante tener en cuenta que si se escoge una plataforma y un ambiente de desarrollo nuevo, se debe tomar un tiempo adicional para aprender a manejar el mismo, en nuestro caso fue bueno ver tutoriales y documentación para avanzar en este proyecto, sin embargo una persona del grupo quedo recargada con estas tareas y como sabemos es una tarea larga y es la que más le da visibilidad al proyecto, por lo cual para futuros proyectos los demás miembros debemos incluir en las tareas del proyecto el capacitarnos mucho más para que la carga de esta actividad quede un poco más repartida y no dejar toda la responsabilidad sobre una persona, debido a que en este caso también se corre un riesgo muy grande y es que si este rol no puede desempeñar sus tareas ya sea por motivos personales, cambio de personal, incapacidad, entre otras, puede dejar el proyecto de manera incompleta y sería fallar los compromisos pactados con el cliente.

Por otro lado también se recomienda seguir unos patrones de diseño o una arquitectura de desarrollo óptima para el proyecto, esto nos da ventajas en el momento de la entrega, del mantenimiento, entre otros.

Anteriormente se menciona que en el desarrollo se implementa en nuevas tecnologías que el grupo de trabajo desconocía por lo cual fue necesario definir tiempos adicionales en el proyecto para capacitación sin embargo esto nos da más experiencia en otras tecnologías, para el caso del proyecto se realizo un ambiente mixto, donde el código se encontraba local pero los datos se almacenaban en una base de datos en la nube, esto nos da más herramientas en el momento de salir al mundo laboral, por lo cual cuando se pueda controlar es muy bueno innovar y probar nuevas tecnologías.

**Trabajo en grupo:** Como se mencionó anteriormente los diferentes miembros del grupo pasaron por diferentes roles, lo cual es bueno para el proyecto en caso de rotación de personal o que haya impedimentos para trabajar, por parte de algún miembro. Sin embargo falto profundizar mucho en los diferentes roles, debido a que cada uno tenía un rol principal y el secundario no se profundizo tanto.

Por otro lado se debe mejorar el grado de responsabilidad de la gran parte de los miembros, esto en tareas de seguimiento y tareas programadas, esto debido a que varias tareas dependen de la ejecución exitosa del alguna tarea anterior.

Ejemplo: Para realizar el test de un sprint, se debe tener lista la programación del mismo.

Adicionalmente y como fue mencionado anteriormente, la responsabilidad de la programación recayó principalmente en una sola persona, lo que causo que sus tareas fueran más cargadas y demandaran más tiempo, por lo cual para futuros proyectos todos los miembros deberían desarrollar tareas de programación.

Por otro lado el grupo tuvo que expulsar a un miembro por constantes fallas en los compromisos y en las tareas pactadas, definitivamente la recomendación inicial que mencionó la profesora en las clases es vital, se deben elegir muy bien los integrantes del grupo, y formar el mismo en base a las diferentes características y habilidades de los miembros, por lo cual para desarrollar un proyecto sería bueno tener un tiempo de adaptación o de “prueba”, esto para que el grupo este seguro de que miembros realmente le aportaran cosas positivas en el desarrollo del mismo. Esto evita malentendidos y estrés innecesario en un proyecto.