Universidad ORT Uruguay

Facultad de Ingeniería

Proyecto: Obligatorio Ingeniería de Software en la Práctica

Entregado como requisito para la aprobación del Obligatorio de Ingeniería de Software en la Práctica

Autores:

Francisco Martinez (233126)

Federico Caimi (176902)

Federico Jacobo (194257)

Mathias Talón (208990)

Tutores:

Luis Barrague

Bruno Ferrari

Valentin Moscone

2020

Índice

Primer entrega 3
- **Análisis del problema** 3
- **Proceso** 3
  - Metodología 3
  - User stories 4
  - Sprints 5
  - Tablero de sprint 6
  - Backlog 6
  - Gestión de riesgos 10
- **Uso de tecnología** 12
  - Repositorio en github 12
  - Uso de cámara 12
  - GPS 12
  - Push notifications 13
  - Mapas 13
  - Algoritmo de coincidencia 13
- **Gestión de calidad** 14
- **Bocetos** 15
- **Sprint 0** 16

Segunda entrega 17
- **Aplicación del proceso** 17
  - Sprint 0 18
    - *Resultado* 18
    - *Informe* 18
    - *Retro* 18
    - *Comentarios* 19
  - Sprint 1 20
    - *Resultado* 20
    - *Informe* 20
    - *Retro* 20
    - *Comentarios* 21
  - Sprint 2 22
    - *Resultado* 22
    - *Informe* 22
    - *Retro* 22
    - *Comentarios* 23
  - Sprint 3 24
    - *Resultado* 24
    - *Informe* 24
    - *Retro* 24
    - *Comentarios* 25
  - Sprint 4 26

Resultado *26*
Informe *27*
Comentarios *27*
Sprint 5 28
Resumen de velocidad del equipo 29
Estimaciones del equipo 30
Sprint 1 *30*
Sprint 2 *31*
Sprint 3 *32*
Tablero de sprint 33
Uso de los repositorios 34
Repositorio Frontend 34
Repositorio Backend 34
Diseño y arquitectura del sistema **35**
Backend 35
Diagrama de componentes 35
Diagrama de clases 36
Esquema de base de datos 36
Cobertura de código 37
Frontend 38
Manual de Instalación 39
Aspectos a mejorar **41**
Refactorizar código del frontend 41
Guardar fotos en firebase 41
Agregar un logout 41
Desactivar notificaciones 41
Notificaciones automáticas filtradas por rango 42
Diseño UI / UX del sistema **43**
Pantalla principal 43
Mis mascotas 45
Mascotas similares 46
Formularios 48

Primer entrega

Análisis del problema

Cualquier persona que tenga o haya tenido una mascota comprende que el riesgo de que esta se extravíe es grande. Cuando algo así sucede, los dueños recurren a realizar ‘’cadenas de mensajes” en las redes cuyo éxito depende exclusivamente de la cantidad de amistades de esta persona y de que la misma llegue a algún individuo que haya tenido contacto con la mascota perdida.
Otro dilema surge cuando una persona encuentra un animal perdido y quiere devolverlo a su dueño. Ésta generalmente realiza una publicación similar a la anterior pero con menos posibilidades de ser útil, ya que el éxito depende de que el mensaje llegue a una persona específica (el dueño de la mascota).

La poca eficiencia de estos métodos nos llevó a idear una aplicación que facilite y automatice este proceso de búsqueda, aumentando así las posibilidades de que una mascota perdida se reencuentre con su dueño.
De esta manera surgió **APPet**, una aplicación que brinda un canal de comunicación rápido y confiable entre el dueño de una mascota extraviada y la persona que la encuentra.

La aplicación contempla ambos casos mencionados anteriormente. Por un lado, un sujeto que pierde una mascota puede notificarlo en la aplicación dejando los datos del animal y los suyos propios. Por otro lado, si una persona encuentra un animal perdido que identifica como doméstico, puede crear otro tipo de reporte para notificar que el mismo ha sido visto o encontrado.

APPet se encarga de cotejar estas publicaciones para encontrar coincidencias y permitir el reencuentro de la mascota con su dueño.

Proceso

Metodología

Optamos por gestionar el proceso de la creación de nuestra app mediante metodologías ágiles.
Estas metodologías permiten mitigar los cambios e imprevistos en el plan. Esto es importante ya que la entrega tiene un plazo fijo y en base a este plazo se debe fijar el alcance. Éste puede cambiar de acuerdo a problemas que pueden ir surgiendo y el diseño ágil hace posible solucionar estos cambios en el alcance.

Dentro de estas metodologías, optamos por Scrum. Después de analizar las posibilidades y tomando en cuenta las características de este proyecto (fecha de entrega fija, alcance eventualmente variable, experiencia del equipo, entre otras), concluimos que esta es la metodología que mejor se adapta a nuestra situación.

Para gestionar nuestro proyecto basado en Scrum, utilizamos la herramienta Jira que nos proporciona un conjunto de funcionalidades que nos facilitan el proceso.

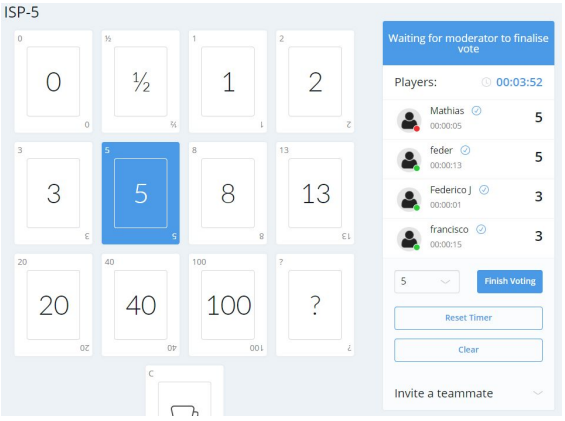
User stories

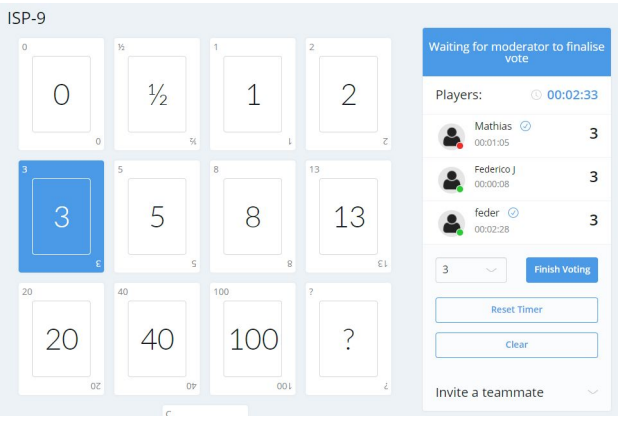
En primer lugar, definimos correctamente los requerimientos de la aplicación. Con esto de base, creamos las **User stories** (que representan valor para los usuarios del sistema) y que conforman el **Product Backlog** de la aplicación.

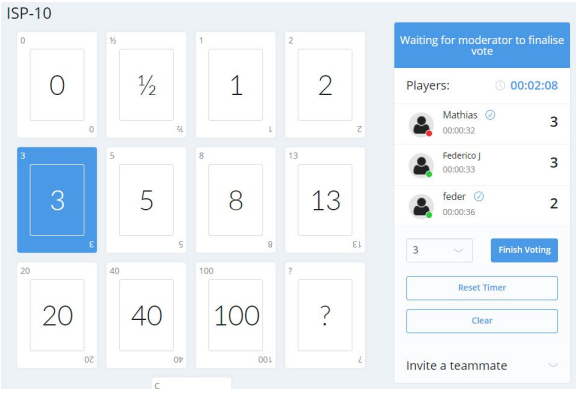
Para la construcción de dichas **User stories** utilizamos la estructura de id, título, descripción, etiquetas y esfuerzo. Teniendo especial cuidado en que quede claro tanto el usuario asociado al requerimiento como la funcionalidad en sí.

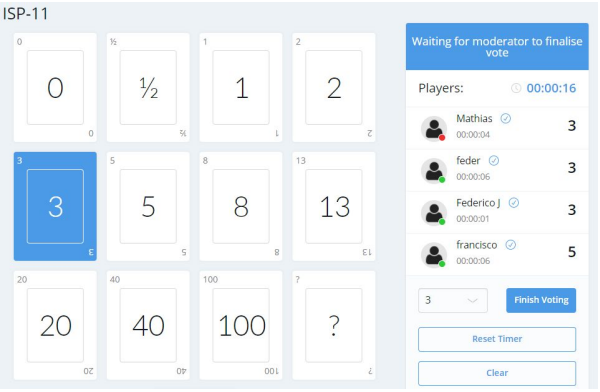
Para estimarlas utilizamos como unidad de medida los **Story points** y nos apoyamos en la herramienta planitpoker, que permite estimar historias de usuario utilizando la técnica de Planning Poker.
Dichas estimaciones las llevamos a cabo en las sesiones de **Grooming**.

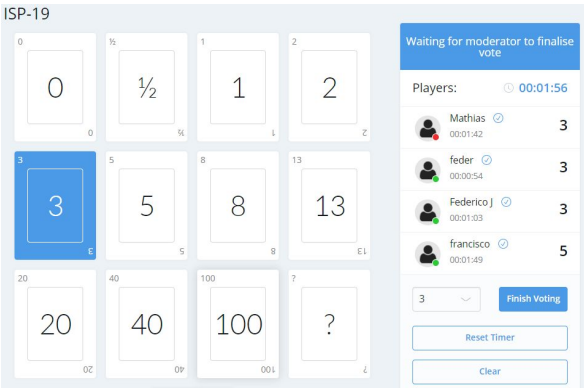
A continuación adjuntamos evidencias de nuestra primera sesión de **Grooming**.

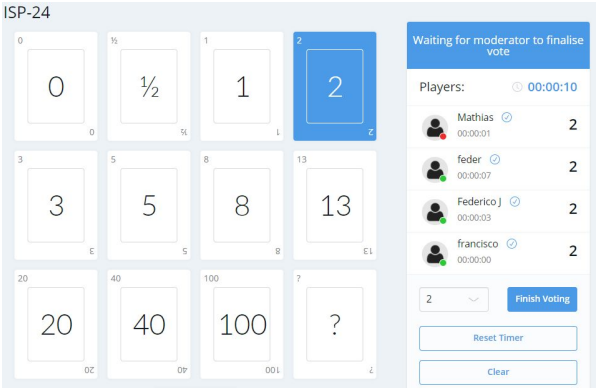












Sprints

Dado que la fecha de entrega del sistema ya está definida y es fija, estaremos siguiendo una planificación **Date driven**, esto quiere decir que nuestros releases ya están fijados y serán hechos en la fecha pactada.
Ahora, para llegar a realizar dichos releases en tiempo y forma, el equipo trabajará en Sprints de **una semana de duración**, esto se ha decidido así ya que creemos que es necesario tener un seguimiento del desarrollo desde cerca, y sprints cortos nos permiten poder corregir detalles del proceso de manera continua, asegurando así que el proceso estará en continua mejora durante el desarrollo del sistema.

Previo a comenzar un sprint, el equipo tendrá una sesión de grooming, en la cual se estimarán las **User stories** que se pretenden desarrollar en el siguiente **Sprint**.

Una vez terminada la estimación, se tendrá una sesión de **Sprint planning**, en la cual el equipo decidirá en base a las **User stories** estimadas qué se desarrollará en dicho sprint.

Para que todos los integrantes del equipo estén alineados con el trabajo que se está realizando, se realizarán instancias de **Daily** todos los días, con el objetivo de que todos sepan que están haciendo sus compañeros y si alguno está bloqueado con algo.
Además, siguiendo con el mismo objetivo, cada sprint contará con un tablero, en el que se podrá visualizar el trabajo de cada uno de los miembros del equipo en tiempo real (ver sección 'Tablero de sprint' por más detalles).

Una vez terminado el sprint, el equipo realizará una sesión de **Retrospective**, en la cual se verá que salió bien y que se puede mejorar en el proceso.

Dado que no hay una interacción directa con un cliente, el equipo ha decidido que no es necesario tener una instancia de **Review (Demo)** del sistema al final de cada sprint.

Tablero de sprint

Para lograr los objetivos de cada **Sprint** y completar las **User stories** planificadas es necesario organizar el trabajo del equipo de la mejor manera posible. Para ello, Jira ofrece un tablero donde el equipo puede ver en todo momento el estado de las **User stories** del sprint activo.
Este tablero consta de cuatro columnas:

- **TO DO**: donde inician todas las **User stories**. Las mismas están esperando que se les asigne un integrante del equipo para realizarla o si ya fue asignado que este comience a trabajar en ella.
- **IN PROGRESS**: aquí se encuentran las historias que están siendo desarrolladas.
- **TO TEST**: una vez desarrolladas, las **User stories** pasan a esta columna donde será testeado su correcto funcionamiento.
- **DONE**: una vez finalizada, pasa a esta columna. Aquí están las historias ya finalizadas e integradas a la rama develop con éxito.

Este tablero permite optimizar el trabajo del equipo logrando la máxima eficiencia posible. Al poder visualizar el flujo de trabajo, se evitan cuellos de botella y otros atrasos.
Para ver en detalle la estructura del tablero, se puede ver un ejemplo del tablero aquí.

Backlog

Como ya mencionamos, el backlog es el artefacto de Scrum que contiene las **User stories** que corresponden a todos los requerimientos que la aplicación contempla.
Se creó el **Backlog** siguiendo un plan incremental, esto quiere decir que hay **User stories** que se basan en que ya hay una base implementada y en sobre eso se agregará funcionalidad.

Teniendo como base las funcionalidades que queríamos que cumpla la aplicación, validadas por el cliente, pasamos a crear una primera versión de este backlog.
Esta versión comprendía varias **User stories** épicas, es decir, que son muy grandes para tomar como una sola.

Dichas **User stories** fueron divididas para crear una segunda versión que fuese más granular y que a su vez sigan aportando valor al cliente.

Una vez definidas las **User stories**, notamos que en algunas se necesitaba de una investigación previa por parte del equipo ya que no se tenía conocimiento sobre las tecnologías que estas requerían.
Para solucionar este problema se decidió realizar un primer **Sprint 0** el cual sólo tiene **Spikes** (stories dedicadas a investigar ciertas tecnologías que nos permitirán realizar **User stories** futuras)

Para dejar claro el proceso incremental seguido, utilizaremos como ejemplo la funcionalidad que permite a una persona que encuentra una mascota perdida hacer un reporte.

Para este caso, la historia original que se creó fue la siguiente:

Está historia se corresponde exactamente con el requerimiento que fue presentado al cliente pero al analizarla salta a la vista que comprende tres funcionalidades que, si bien están relacionadas, se pueden separar.

- Como usuario logueado, registrar que se localizó una mascota que se cree perdida.
- Poder agregarle a este registro una imagen
- Poder agregarle a este registro la ubicación GPS.

Una vez realizada esta división, se obtienen las tres historias de usuario finales:

Como usuario no registrado, quiero poder reportar una mascota vista

Attach | Create subtask | Link issue

Description

El usuario logueado, debe ser capaz de reportar que localizó una mascota que cree que está perdida

Como usuario logueado, quiero poder reportar una mascota vista y adjuntar una foto de la misma

Attach | Create subtask | Link issue

Description

El usuario logueado, debe ser capaz de reportar que localizó una mascota que cree que está perdida y adjuntar una foto

Como usuario logueado, quiero poder reportar una mascota vista y agregar ubicación GPS

Attach | Create subtask | Link issue

Description

El usuario logueado, debe ser capaz de reportar que localizó una mascota que cree que está perdida y agregar ubicacion GPS

De estas últimas dos historias de usuario se crearán dos spikes. La primera corresponde a la investigación sobre cómo sacar fotos y utilizar la cámara. La segunda a cómo utilizar la ubicación GPS del celular.
Para finalizar, cada miembro del equipo dividirá las historias de usuario que se le asignen en tasks, según su criterio, para facilitar y organizar su ejecución.

A continuación adjuntamos evidencia del backlog con todas las **User stories** del sistema definidas, el **Sprint 0** comenzado y la estimación de las **User stories** que se planean llevar a cabo en el **Sprint 1**.

Sprint 0 6 incidencias

0 6 0 Páginas vinculadas 0 •••

Recolectar información y experiencia que nos pueda ayudar al momento de comenzar a desarrollar el sistema.

30/sep/20 2:24 PM • 07/oct/20 2:24 PM

FC FM FJ MT •••

SPIKE: Uso y persistencia de imágenes	FM	ISP-31	1
SPIKE: Uso de push notifications	FJ	ISP-32	1
SPIKE: Uso de GPS	MT	ISP-33	1
SPIKE: Uso de cámara	FM	ISP-34	1
SPIKE: Android studio	FC	ISP-35	1
SPIKE: Uso de material design	FC	ISP-36	1

Backlog 26 incidencias

Crear sprint •••

Como usuario registrado, quiero poder loguearme al sistema	ISP-9	3
Como usuario no registrado, quiero poder registrarme en el sistema	ISP-10	3
Como usuario logueado, quiero poder reportar a mi mascota perdida	ISP-11	3
Como usuario no registrado, quiero poder reportar una mascota vista	ISP-19	3
Como usuario no registrado, quiero poder ver todos los eventos (reportes de mascota perdida/encontrada) del sistema	ISP-5	3
Como usuario logueado, quiero poder marcar como encontrada una mascota que reporté como perdida	ISP-24	2
Como usuario no registrado, quiero poder ver los eventos del sistema filtrados por ubicación	ISP-6	-
Como usuario no registrado, quiero poder ver los eventos del sistema ordenados por prioridad	ISP-7	-
Como usuario registrado, quiero poder gestionar la información de mi cuenta	ISP-8	-
Como usuario logueado, quiero poder reportar a mi mascota perdida con información GPS	ISP-12	-
Como usuario logueado, quiero poder registrar a mis mascotas en el sistema	ISP-13	-
Como usuario logueado, quiero poder generar un reporte de mascota perdida automaticamente	ISP-14	-
Como usuario logueado, quiero poder generar un reporte de mascota perdida automatico y con alerta	ISP-15	-
Como usuario logueado, quiero poder editar una de mis publicaciones abiertas	ISP-16	-
Como usuario logueado, quiero poder agregar una foto en los reportes de mi mascota en el reporte de mascota perdida	ISP-17	-
Como usuario lgueado, quiero poder agregar tags al crear un reporte de mascota perdida	ISP-18	-
Como usuario logueado, quiero poder reportar una mascota vista y agregar ubicacion GPS	ISP-20	-
Como usuario logueado, quiero poder reportar una mascota vista y adjuntar una foto de la misma	ISP-21	-
Como usuario logueado, quiero poder cerrar un reporte de mascota vista que haya creado	ISP-22	-
Como usuario logueado, quiero poder agregar tags a un reporte de mascota vista	ISP-23	-
Como usuario logueado, quiero poder contactarme con la persona que reportó una mascota perdida	ISP-25	-
Como usuario logueado, quiero poder ver reportes de mascotas vistas similares a mi mascota perdida	ISP-26	-
Como usuario logueado, quiero recibir una notificación cuando haya una nueva coincidencia con mi mascota perdida	ISP-27	-
Como usuario logueado, quiero poder ignorar un reporte similar a mi mascota perdida	ISP-28	-
Como usuario logueado, quiero poder configurar mi rango de interés en mi cuenta	ISP-29	-
Como usuario logueado, quiero poder activar/desactivar las notificaciones del sistema	ISP-30	-

Gestión de riesgos

El uso de metodologías ágiles y sus cortas iteraciones reducen las probabilidades de que se presenten riesgos importantes que retrasen o incluso detengan el proyecto. Los riesgos que pueden aparecer en los distintos sprints solo podrán identificarse a la hora de planear dichos sprints, ya que es este el momento donde se tiene toda la información necesaria para realizar una correcta gestión de los riesgos.

Nos quedan entonces los riesgos a nivel del proyecto en general. Para poder gestionar estos riesgos es necesario identificarlos, medirlos, tomar acciones para mitigarlos y reducirlos en caso de que ocurran.

Riesgo	**Gravedad**	**Probabilidad**	**Acciones para mitigarlo**	**Plan de respuesta**
Un integrante del equipo abandona el proyecto	Alta	Baja	-	Habría que replanificar y ajustar el alcance para llegar a entregar en el plazo
Un integrante del equipo no puede trabajar por un tiempo	Media	Media	-	Habría que replanificar: se le pueden asignar sus stories al resto del equipo, hacer que trabaje más horas al reintegrarse o acortar el alcance
Poco conocimiento sobre tecnologías a utilizar	Alta	Media	Para mitigar este riesgo realizamos los spikes de los que ya hablamos	Se deberán realizar spikes que retrasaran la finalización de la user story. Probablemente se deba replanificar

Pérdida de información de la pc de un integrante del equipo	Baja	Alta	Para mitigar este riesgo utilizamos github, trabajando en ramas y haciendo commits y push diarios	Cómo utilizamos github, la pérdida de información sería poca y el dueño de la pc debería realizar el trabajo perdido nuevamente
Errores de estimación	Alta	Media	Es probable que suceda, para ello tenemos sesiones de retrospective, para evaluar qué se puede mejorar y reducir dicho riesgo en siguientes sprints	Se debería realizar una nueva estimación que tendrá como consecuencia una planificación que puede acortar el alcance del proyecto

La tabla tiene cinco columnas. La primera corresponde al riesgo en sí. La segunda es que tan grave serían las consecuencias de que ocurra el riesgo, y tiene los valores alta, media y baja. La tercera hace referencia a que tan probable es que ocurra realmente este riesgo, y tiene como valores alto, medio y bajo (siendo alto muy probable y bajo poco probable). En la cuarta columna aparecen las acciones que estaremos tomando para reducir las posibilidades de que el riesgo ocurra. Por último, mostramos cuales son las acciones que se deberían tomar en caso de que el riesgo ocurra.

Uso de tecnología

Para asegurar que el sistema cumpla con su objetivo de la manera más eficiente y efectiva posible, hemos decidido utilizar un conjunto de tecnologías y características de los dispositivos móviles, para así brindarle la mejor experiencia de uso a nuestros usuarios.
A continuación detallaremos las características que hemos de introducir al sistema y al proceso de desarrollo del mismo.

Repositorio en github

Dado que el equipo trabajará en el desarrollo del sistema de manera 100% remota, es de crucial importancia contar con un repositorio compartido en el cual podamos tener un manejo organizado de los módulos de código que vamos introduciendo al sistema, así como un control del versionado del mismo.

Para el desarrollo del proyecto estaremos utilizando un repositorio GIT y a su vez, para sacarle más provecho al uso del mismo, utilizaremos gitflow como flujo de trabajo.
Esto quiere decir que tendremos 2 ramas principales, master (donde estará nuestro código de producción, los releases del sistema) y develop (donde convergen todas las ramas de desarrollo).

Cada feature del sistema será desarrollada en una rama independiente, identificada con el prefijo '**feature/<nombre_de_rama>'.**
Cada corrección de bug que se haga en el sistema será desarrollada en una rama independiente con el prefijo **'bugfix/<nombre_de_rama>'.**

Los releases del sistema serán hechos en base a un merge de la rama develop hacia la rama master del repositorio.

Uso de cámara

Ya que identificamos que la captura de imagen de una mascota es de vital importancia para llevar a cabo el objetivo del sistema de la mejor manera, hemos decidido introducir el uso de la cámara del dispositivo. De esta forma, el usuario puede tener una experiencia más visual, lo cual le ayudará enormemente tanto a encontrar a su posible mascota perdida como a encontrar al dueño de una mascota perdida.

GPS

Al igual que el uso de la cámara, la ubicación GPS es de enorme ayuda a la hora de reportar una mascota perdida o encontrada; es por ello que también permitiremos el uso de esta herramienta en nuestro sistema. Las publicaciones realizadas en el sistema serán asociadas a una ubicación y en base a ella el sistema se encargará de generarle valor al usuario, por ejemplo alertando a los usuarios cercanos a la hora de reportar una mascota perdida o mostrando los eventos cercanos a la ubicación del usuario para facilitar la búsqueda de una mascota perdida.

Push notifications

Dado que una parte importante del sistema es reportar mascotas perdidas y que la rápida acción en el momento que una mascota se pierde aumenta considerablemente las probabilidades de que la misma sea encontrada, hemos decidido implementar un sistema de notificaciones a los usuarios cercanos a un reporte de mascota perdida en tiempo real. De esta manera se estaría alertando a los usuarios cercanos al suceso y así se aumentan las posibilidades de que la mascota sea encontrada.
También le llegará una notificación al usuario que haya perdido una mascota cuando se detecte una coincidencia.

Mapas

Para facilitar la visualización de ubicaciones que el sistema maneja, se implementará un mapa para que los usuarios puedan ver cómodamente dicha información.

Algoritmo de coincidencia

Para aumentar la efectividad del sistema, implementaremos un algoritmo que filtrará los reportes ingresados al sistema, y dado un usuario en específico que esté realizando una búsqueda de su mascota perdida, el sistema le proporcionará una lista de reportes ordenados por porcentaje de coincidencia para facilitar la búsqueda de dicha mascota.
La coincidencia entre reportes será calculada en base a tags que serán adjuntados en cada reporte guardado en el sistema.

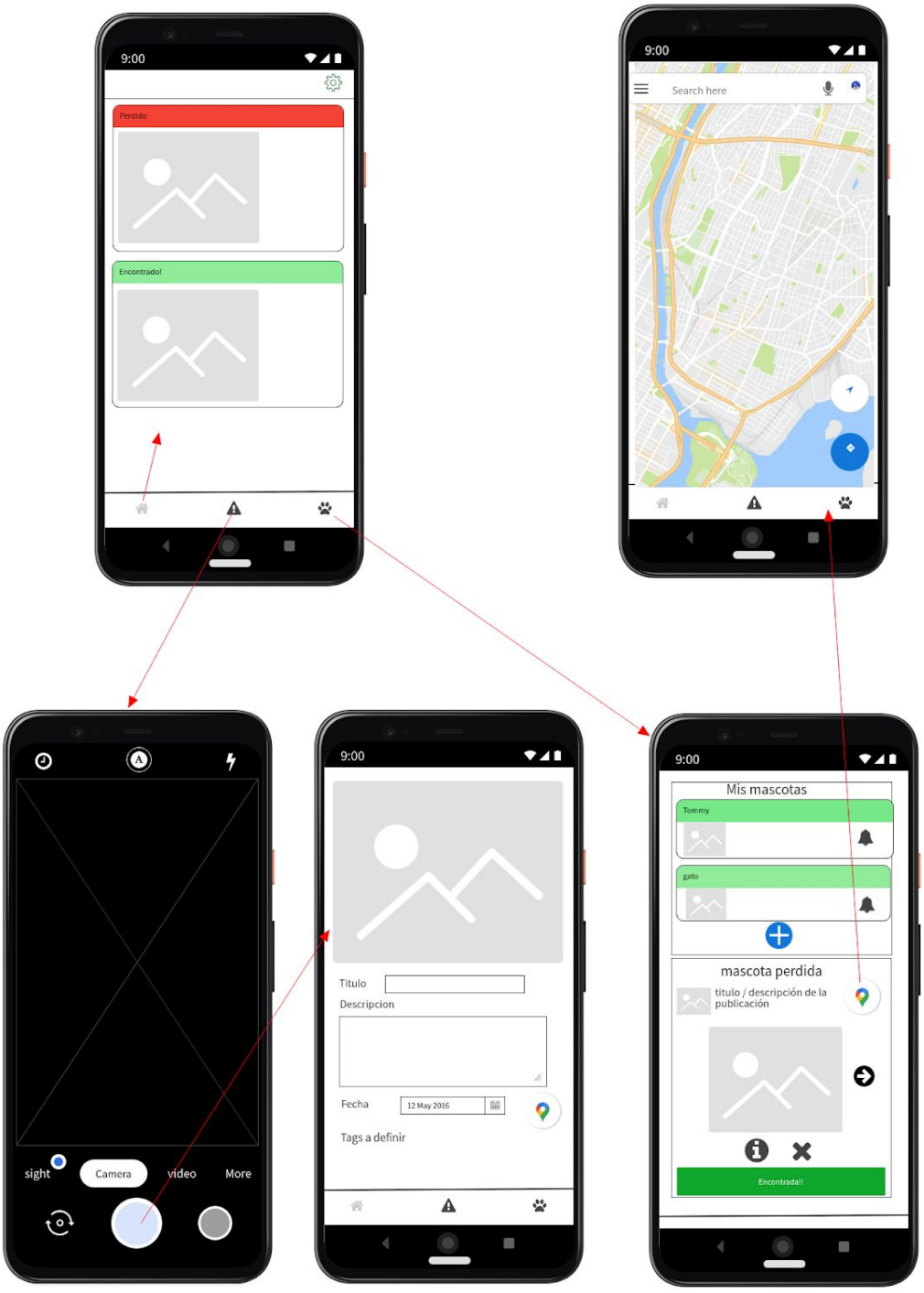
Gestión de calidad

Para garantizar la calidad del código del sistema, nos hemos planteado como meta tener una cobertura de pruebas mayor de 90% en el módulo de lógica de negocio, ya que identificamos que dicho módulo será el más importante dentro de nuestro sistema backend y el que más valor aportará.
Dado que el equipo enfocará su esfuerzo en desarrollar de la mejor manera posible todas las funcionalidades planteadas, se decidió que el único módulo en el cual se desarrollarán pruebas automáticas será el anteriormente mencionado, dado que los otros módulos no presentan una relación costo beneficio que el equipo considere aceptable.

Además, para verificar que las funcionalidades desarrolladas cumplan con los requerimientos definidos, el equipo estará realizando **cross testing**, esto es, los miembros del equipo estarán testeando el trabajo de sus compañeros.

En el aspecto de proceso de desarrollo, el equipo prestará especial atención a tener un proceso de calidad, dado que consideramos que un proceso de calidad da como resultado un producto de calidad. Para ello hemos de adoptar características de Scrum que identificamos como beneficiosas para el equipo y el proceso.
Por ejemplo, el equipo se aferrara a las ceremonias que la metodología plantea, ya sean **dailies**, **sprints** de 1 semana de duración, **plannings**, sesiones de **grooming**, **retrospectives** cada 2 **sprints**, sistema definido por un **backlog** con **user stories**, entre otras.
Basados en la experiencia personal de cada uno de los integrantes del equipo se ha decidido que adoptar dicha postura al seguir un proceso bien definido aportará un gran valor al desarrollo del sistema, en especial en la situación de desarrollo remoto en la que se llevará a cabo el proyecto.

Bocetos



La primera versión de la interfaz presentaba una complejidad mucho mayor a la hora de utilizar cada funcionalidad del sistema. Posteriormente, se busco reducir esta complejidad para aportar al usuario una navegabilidad más intuitiva. Esto se logró reduciendo la cantidad de botones y las posibles rutas que el usuario puede seguir.

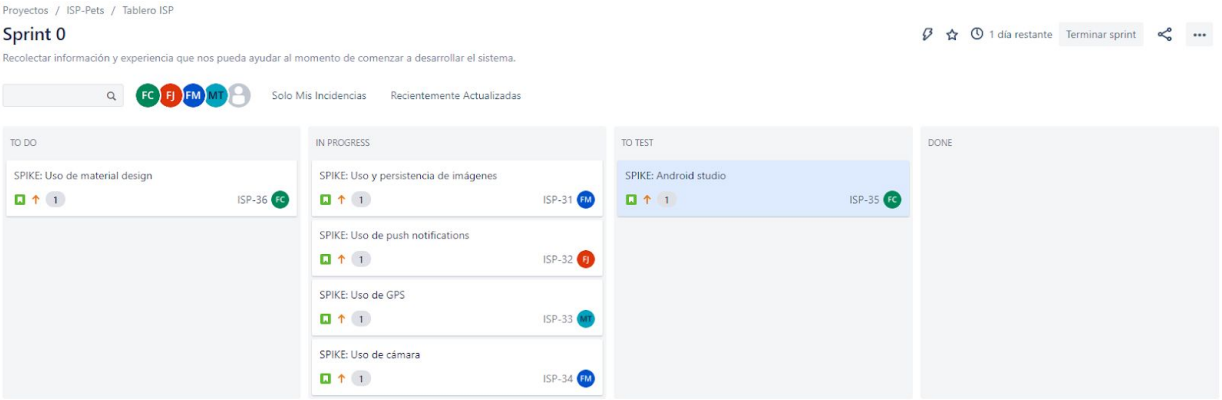
Sprint 0

Con el objetivo de aprovechar el tiempo con el cual se cuenta para realizar el proyecto y teniendo en cuenta que hay varios requerimientos de los cuales el equipo no tiene mucho conocimiento, hemos decidido realizar un primer Sprint 0, el cual solo cuenta con **User stories** de tipo **Spike**. Dichos **Spikes** fueron introducidos de manera tal, que el equipo pueda reducir la incertidumbre, y lograr así una mejor estimación de los requerimientos antes mencionados.
Al momento de la entrega de esta documentación, el Sprint 0 sigue en proceso, por lo cual no estamos en condiciones de mostrar todos los resultados del mismo. Aún así, hemos de adjuntar los resultados de los **Spikes** que han sido terminados antes de la fecha de finalización del Sprint 0.

Progreso actual del Spike de Android Studio
https://docs.google.com/document/d/1OgCDCr6Ic40zGfRuEc2P9QMB4V6YREJnhBSlkBLCXxA/edit?usp=sharing

Progreso actual del Spike Push Notifications
https://drive.google.com/file/d/1opBFOL6gcYdFu26BMVYkRX7pEMuIYLKL/view?usp=sharing

Progreso actual del Spike Cámara
https://docs.google.com/document/d/1QiuQyfMlS86cIJcTT-t_6Dp3bEMhnzsNPWj6gEEBF4g/edit?usp=sharing



Segunda entrega

Aplicación del proceso

En la primera parte del proyecto, se definieron ciertos procesos a los cuales nos aferramos para que el desarrollo del sistema sea lo más eficiente posible.
En esta sección veremos distintas evidencias de cómo dichos procesos fueron llevados a cabo y de la capacidad del equipo para pulir los mismos, de tal manera que el rendimiento sea cada vez mayor.

Como se definió previamente, el proyecto fue desarrollado utilizando la metodología de Scrum, cumpliendo con las ceremonias que la misma define, y ajustando detalles para que poder sacar el máximo provecho de dicha metodología dadas las circunstancias del proyecto y del equipo.

A continuación veremos evidencias de cómo la metodología fue utilizada, los detalles que se ajustaron para que el equipo tenga un mayor rendimiento y los resultados obtenidos.
Para facilitar el entendimiento de la información y exponer el crecimiento que tuvo el equipo con el tiempo, ordenaremos las evidencias en orden cronológico.

Sprint 0

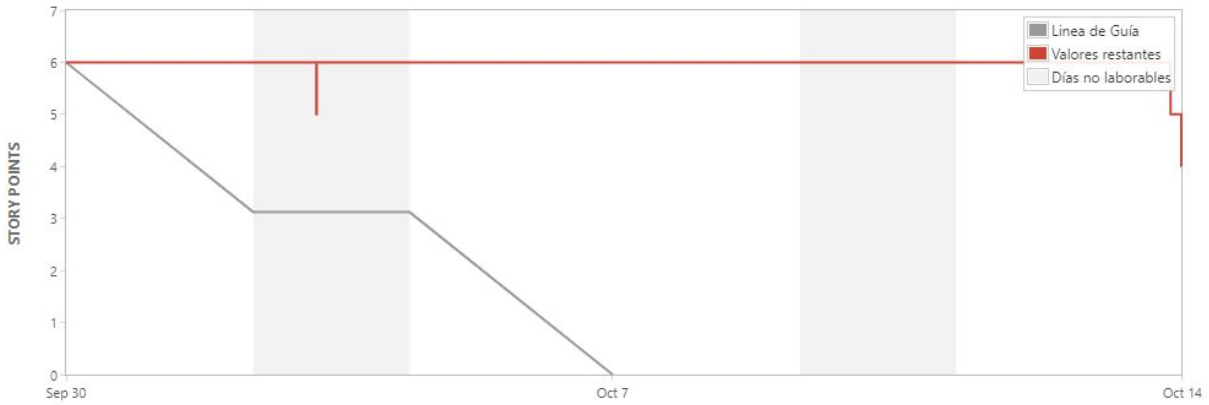
2 semanas de duración.
El objetivo de este sprint fue recolectar información y experiencia que pudiera facilitar el desarrollo del sistema.

Resultado

Sprint 0

Sprint cerrado, terminado por Mathias Talon 30/sep/20 2:24 PM - 14/oct/20 9:11 PM View linked pages

Recolectar información y experiencia que nos pueda ayudar al momento de comenzar a desarrollar el sistema.



Informe

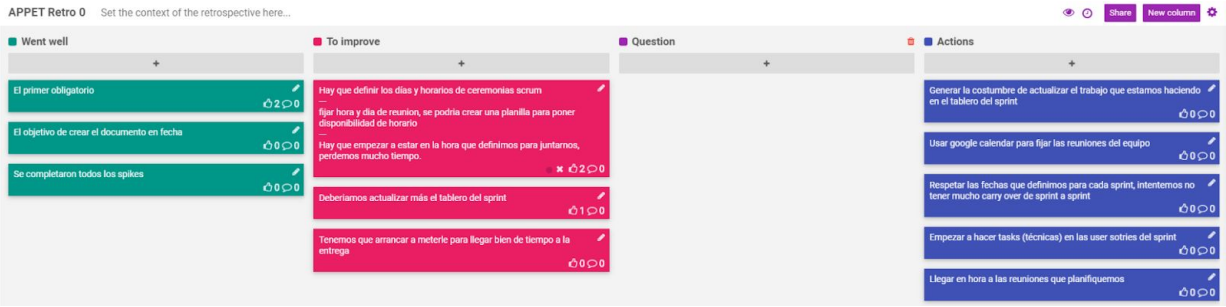
Incidencias terminadas — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (2)
ISP-33	SPIKE: Uso de GPS	Historia	Medium	FINALIZADA	1
ISP-35	SPIKE: Android studio	Historia	Medium	FINALIZADA	1

Incidencias Sin Completar — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (4)
ISP-31	SPIKE: Uso y persistencia de imágenes	Historia	Medium	TAREAS POR HACER	1
ISP-32	SPIKE: Uso de push notifications	Historia	Medium	EN CURSO	1
ISP-34	SPIKE: Uso de cámara	Historia	Medium	EN CURSO	1
ISP-36	SPIKE: Uso de material design	Historia	Medium	EN CURSO	1

Retro



Comentarios

Se puede ver claramente que el equipo aún estaba formándose, obteniendo resultados muy pobres, sin poder terminar la totalidad de los spikes definidos en un sprint de 2 semanas. Más allá de eso, se puede observar un interesante resultado en la reunión retrospectiva, un equipo con ganas de mejorar y generar buenas costumbres.

Sprint 1

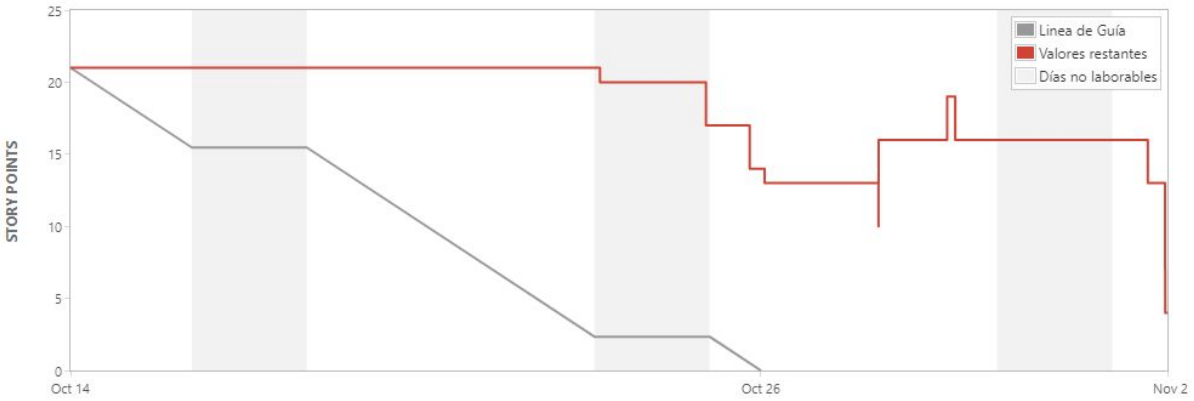
2 semanas de duración.
El objetivo para este sprint era terminar los spikes restantes y comenzar la construcción del sistema.
Para el final de este sprint se esperaba contar con un MVP del producto.
Para este sprint se decidió extender la duración a 2 semanas dado que el equipo identificó que esta primera parte podría presentar un desafío técnico que requeriría de dicho tiempo.

Resultado

Sprint 1

Sprint cerrado, terminado por Mathias Talon 14/oct/20 9:24 PM - 02/nov/20 11:04 PM View linked pages

Terminar los spikes que quedaron en progreso y producir el MVP del sistema.



Informe

Informe de estado

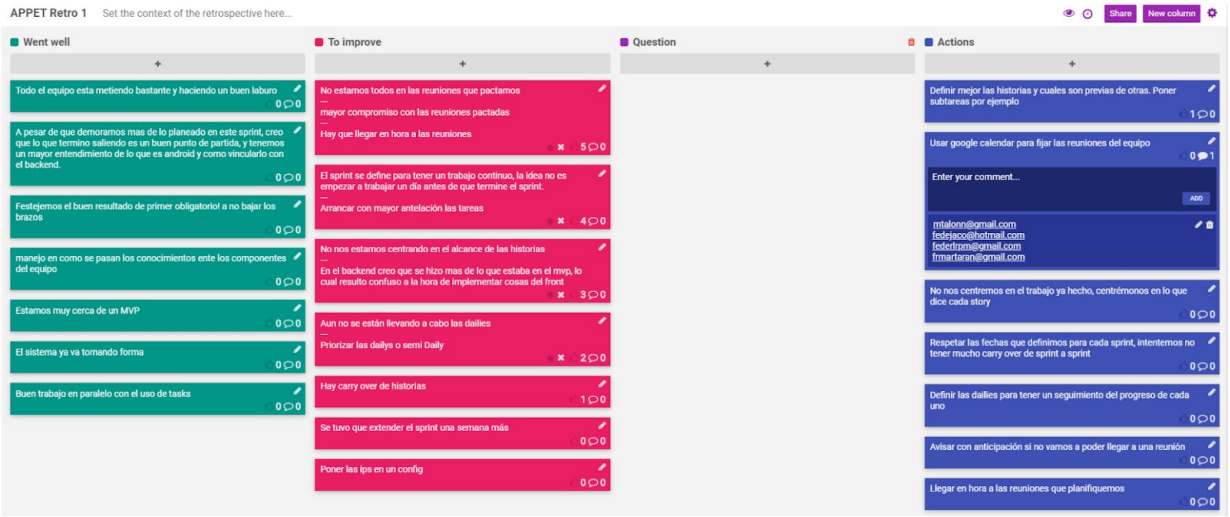
Incidencias terminadas — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (17)
ISP-5	Como usuario no registrado, quiero poder ver todos los eventos (reportes de mascota perdida/encontrada) del sistema	Historia	Medium	FINALIZADA	3
ISP-9	Como usuario registrado, quiero poder loguearme al sistema	Historia	Medium	FINALIZADA	3
ISP-10	Como usuario no registrado, quiero poder registrarme en el sistema	Historia	Medium	FINALIZADA	3
ISP-11	Como usuario logueado, quiero poder reportar a mi mascota perdida	Historia	Medium	FINALIZADA	3
ISP-19	Como usuario no registrado, quiero poder reportar una mascota vista	Historia	Medium	FINALIZADA	3
ISP-34	SPIKE: Uso de cámara	Historia	Medium	FINALIZADA	1
ISP-36	SPIKE: Uso de material design	Historia	Medium	FINALIZADA	1

Incidencias Sin Completar — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (4)
ISP-24	Como usuario logueado, quiero poder marcar como encontrada una mascota que reporté como perdida	Historia	Medium	TAREAS POR HACER	2
ISP-31	SPIKE: Uso y persistencia de imágenes	Historia	Medium	TAREAS POR HACER	1
ISP-32	SPIKE: Uso de push notifications	Historia	Medium	EN CURSO	1

Retro



Comentarios

Podemos observar que el equipo ya va tomando una velocidad de trabajo, aunque aún se sigue perdiendo mucho tiempo (podemos observar en la sección de “Resultado” que el equipo prácticamente desperdició una semana de trabajo).

De todos modos es importante destacar que el equipo supo identificar ese problema y se tomaron acciones para mejorar.

Además de lo antes mencionado, se puede observar que el equipo buscó mejorar no solo en los aspectos técnicos, sino que también se buscó continuamente mejorar el proceso.

Sprint 2

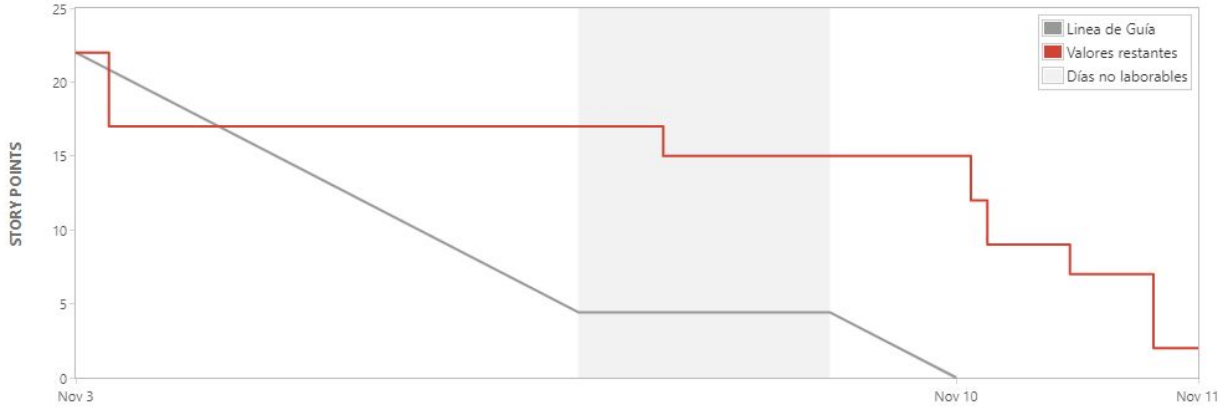
1 semana de duración.
El objetivo de este sprint fue terminar las historias del sprint anterior, para así lograr un MVP. Además se implementó la primer tecnología, el uso del GPS, lo cual fue un primer gran desafío para el equipo.

Resultado

Sprint 2

Sprint cerrado, terminado por Mathias Talon 03/nov/20 12:07 AM - 11/nov/20 10:16 PM View linked pages

Terminar el MVP y implementar la lógica con GPS.



Informe

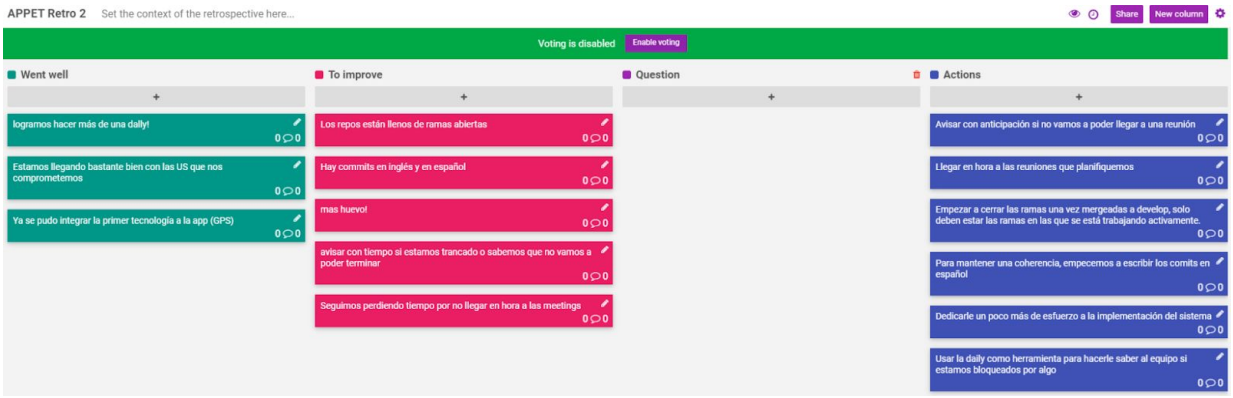
Incidencias terminadas — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (20)
ISP-6	Como usuario no registrado, quiero poder ver los eventos del sistema filtrados por ubicación	Historia	Medium	FINALIZADA	3
ISP-12	Como usuario logueado, quiero poder reportar a mi mascota perdida con información GPS	Historia	Medium	FINALIZADA	5
ISP-13	Como usuario logueado, quiero poder registrar a mis mascotas en el sistema	Historia	Medium	FINALIZADA	3
ISP-29	Como usuario logueado, quiero poder configurar mi rango de interés en mi cuenta	Historia	Medium	FINALIZADA	2
ISP-31	SPIKE: Uso y persistencia de imágenes	Historia	Medium	FINALIZADA	1
ISP-32	SPIKE: Uso de push notifications	Historia	Medium	FINALIZADA	1
ISP-38	Como usuario no registrado, quiero poder reportar una mascota vista y agregar ubicación GPS.	Historia	Medium	FINALIZADA	5

Incidencias Sin Completar — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (2)
ISP-24	Como usuario logueado, quiero poder marcar como encontrada una mascota que reporté como perdida	Historia	Medium	TAREAS POR HACER	2

Retro



Comentarios

El equipo empezó a agarrar confianza y los resultados empezaban a saltar a la vista, se empezó a cumplir casi con la totalidad de los puntos a los que se comprometía.
Importante destacar que aún se identificaban problemas en el proceso de desarrollo.

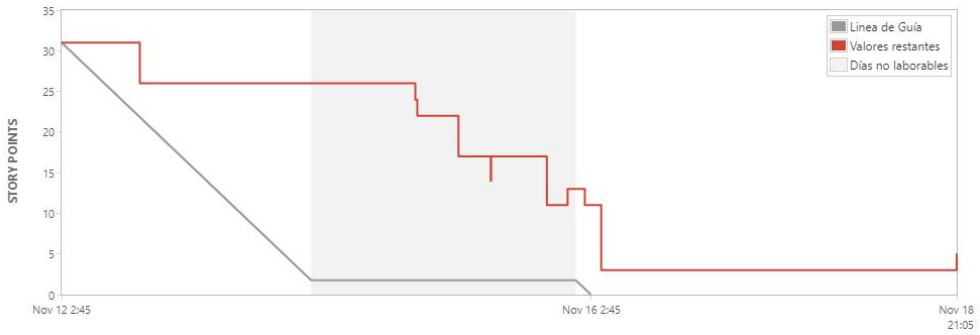
Sprint 3

1 semana de duración.
El objetivo de este sprint fue la implementación de la segunda nueva tecnología en el sistema, la cámara de fotos.
También se apuntaba a implementar toda la lógica relacionada a los tags de mascotas en el sistema y el algoritmo de coincidencia entre mascotas.

Resultado

Sprint 3

Sprint cerrado, terminado por Mathias Talon 12/nov/20 2:45 AM - 18/nov/20 9:05 PM View linked pages



Informe

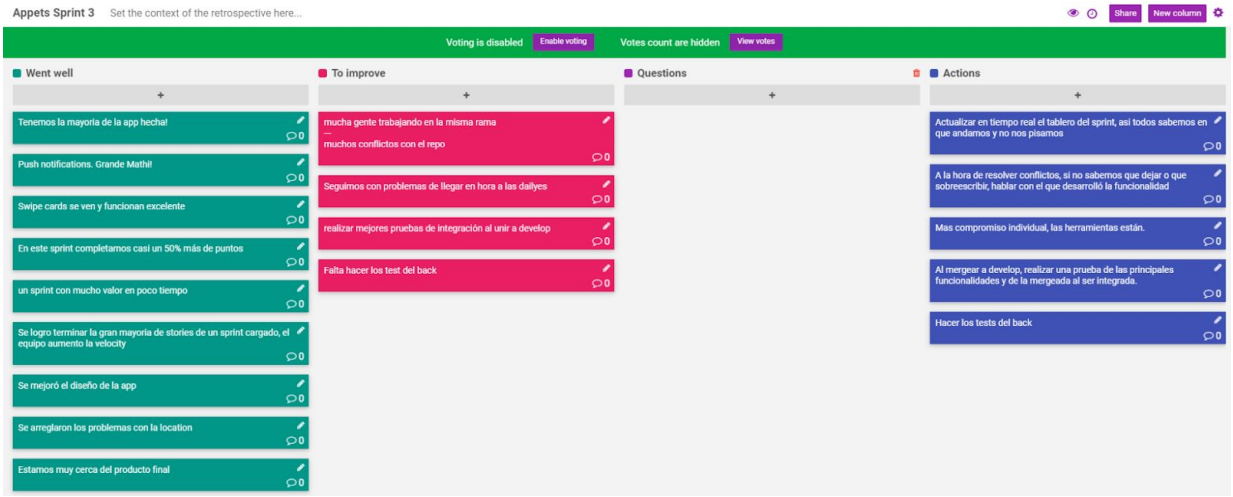
Incidencias terminadas — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (28)
ISP-17	Como usuario logueado, quiero poder agregar una foto en los reportes de mi mascota en el reporte de mascota perdida	Historia	Medium	FINALIZADA	5
ISP-18	Como usuario lgueado, quiero poder agregar tags al crear un reporte de mascota perdida	Historia	Medium	FINALIZADA	3
ISP-21	Como usuario no registrado, quiero poder reportar una mascota vista y adjuntar una foto de la misma	Historia	Medium	FINALIZADA	5
ISP-23	Como no registrado, quiero poder agregar tags a un reporte de mascota vista	Historia	Medium	FINALIZADA	3
ISP-24	Como usuario logueado, quiero poder marcar como encontrada una mascota que reporté como perdida	Historia	Medium	FINALIZADA	2
ISP-26	Como usuario logueado, quiero poder ver reportes de mascotas vistas similares a mi mascota perdida	Historia	Medium	FINALIZADA	8
ISP-91 *	Arreglo de bug en login	Error	Medium	FINALIZADA	-
ISP-93 *	Arreglo de bug en found	Error	Medium	FINALIZADA	-
ISP-94 *	Arreglo de bug al agregar mascota	Error	Medium	FINALIZADA	-
ISP-96 *	Arreglo de bug al Reportar Mascota perdida	Error	Medium	FINALIZADA	-
ISP-103 *	Mantener sesión al cerrar la app	Error	Medium	FINALIZADA	-
ISP-109 *	Como usuario logueado, quiero poder agregar una imagen al momento de registrar una mascota	Historia	Medium	FINALIZADA	2

Incidencias Sin Completar — Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (5)
ISP-25	Como usuario logueado, quiero poder contactarme con la persona que reportó una mascota perdida	Historia	Medium	EN CURSO	2
ISP-28	Como usuario logueado, quiero poder ignorar un reporte similar a mi mascota perdida	Historia	Medium	TAREAS POR HACER	3
ISP-92 *	Arreglo de bug al registrarse	Error	Medium	EN CURSO	-
ISP-98 *	Arreglo de bug UI	Error	Medium	TAREAS POR HACER	-
ISP-113 *	Arreglo de bug en Imagen en la creación de post	Error	Medium	TAREAS POR HACER	-

Retro



Comentarios

Un punto de quiebre para el equipo, se comenzó a trabajar bien en equipo, los integrantes empezamos a adaptarnos a trabajar juntos y esto se vio en un sprint con un excelente resultado, casi un 50% más productivo en comparación con el sprint anterior.

El producto ya casi terminado en sus funcionalidades clave, con la integración de todas las tecnologías acordadas en el alcance del mismo, dado que no sólo se cumplió con los objetivos del sprint sino que se adelantó trabajo (integración de push notifications al sistema).

Sprint 4

1 semana de duración.
El objetivo de este sprint fue centrarse en terminar las funcionalidades del sistema y solucionar bugs detectados, para obtener el producto más robusto y estable posible. Además si queda tiempo se utilizará en la mejora de aspectos de UI/UX.

Resultado

Informe

Incidencias terminadas

Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (19 → 21)
ISP-8	Como usuario registrado, quiero poder gestionar la información de mi cuenta (Mascotas)	Historia	Medium	FINALIZADA	3
ISP-15	Como usuario no registrado, quiero poder generar un reporte de mascota perdida y que alerte a los otros usuarios del sistema	Historia	Medium	FINALIZADA	8
ISP-16	Como usuario logueado, quiero poder editar una de mis publicaciones de mascota perdida abierta	Historia	Medium	FINALIZADA	3
ISP-25	Como usuario logueado, quiero poder contactarme con la persona que reportó una mascota perdida	Historia	Medium	FINALIZADA	2
ISP-28	Como usuario logueado, quiero poder ignorar un reporte similar a mi mascota perdida	Historia	Medium	FINALIZADA	3
ISP-92	Arreglo de bug al registrarse	Error	Medium	FINALIZADA	-
ISP-121 *	Mostrar mapa con la ubicación del reporte.	Historia	Medium	FINALIZADA	- → 2
ISP-126 *	Agregar numero de teléfono de contacto al crear una publicación	Error	Medium	FINALIZADA	-
ISP-127 *	Borrar el campo 'tipo' al crear mascota	Error	Medium	FINALIZADA	-
ISP-129 *	Poner descripcion por defecto al reportar mascota	Error	Medium	FINALIZADA	-
ISP-130 *	Al abrir la cámara al reportar una mascota vista la app se cae	Error	Medium	FINALIZADA	-
ISP-131 *	Siempre aparece un mensaje de acceso denegado cuando se pide la ubicación	Error	Medium	FINALIZADA	-

Incidencias Sin Completar

Ver en el navegador de incidencias

Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (12)
ISP-30	Como usuario no registrado, quiero poder activar/desactivar las notificaciones del sistema	Historia	Medium	EN CURSO	2
ISP-98	Arreglo de bug UI	Error	Medium	TAREAS POR HACER	-
ISP-113	Arreglo de bug en Imagen en la creación de post	Error	Medium	TAREAS POR HACER	-
ISP-116	Crear tests para las funciones del backend	Historia	Medium	TAREAS POR HACER	5
ISP-122 *	Crear documentación del sistema	Historia	Medium	EN CURSO	5
ISP-128 *	Refactorear la UI de las cartas del home screen	Error	Medium	EN CURSO	-

Comentarios

Se completaron las funcionalidades del sistema y hubo tiempo para realizar mejoras sobre el mismo.

El equipo está conforme con el resultado obtenido y el proceso con el cual se llevó a cabo la construcción del mismo.

Sprint 5

2 días de duración.
Este sprint se decidió llevar a cabo luego de una demo interna del sistema, en la cual el equipo detectó algunos cambios que se podrían realizar en el sistema para que sea más estable y mejorar la experiencia de usuario.

Dado que los cambios realizados en este sprint no fueron estimados, no aporta valor agregar un informe ni un reporte de velocidad.

VERSIONES

EPICS

Sprint 5 17 incidencias

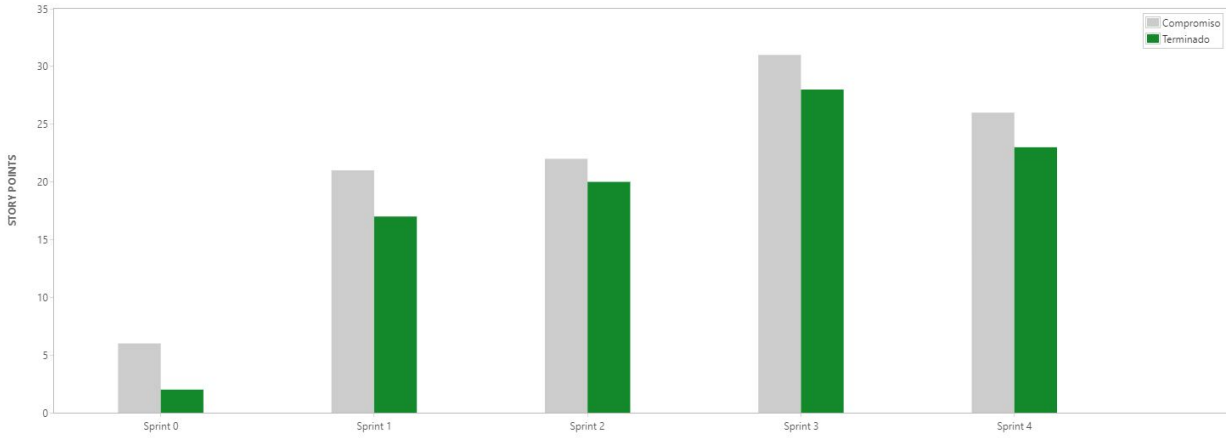
Arreglos y mejoras
25/nov/20 12:16 AM • 26/nov/20 12:16 AM

0 5 5 Páginas vinculadas 0

Arreglo de bug UI	~~ISP-98~~	
Crear documentación del sistema	ISP-122	5
Crear tests para las funciones del backend	~~ISP-116~~	5
Cambiar el color de fondo de mascotas perdidas (rojo más clarito)	~~ISP-132~~	
dejar los tags lindos y cambiar el icono del mapa (algo mas parecido a maps)	~~ISP-134~~	
poner telefono de contacto por defecto en creacion de mascota perdida	~~ISP-135~~	
poner tags en crear mascota y que se pongan automaticos en la publicacion de perdida	~~ISP-136~~	
sacar campo de coordenandas del formulario	~~ISP-133~~	
bug de llamar cuando es la ultima card del tinder	ISP-137	
accion para arriba del tinder deberia aparecer el icono de un telefono	~~ISP-138~~	
accion para abajo del tinder deberia aparecer el icono del mapa	~~ISP-139~~	
cambiar colores de las cards de mascota y perdida	ISP-140	
Agregar carteles de advertencia para determinadas acciones	~~ISP-141~~	
desloguear al cambiar datos del usuario	~~ISP-142~~	
cuando dejamos el color por defecto despues se muestra como negro	~~ISP-143~~	
loaders cuando se está pidiendo data del back (en home screen las listas de mascotas y perdidas)	~~ISP-144~~	
Agregar un botón que permita eliminar una de las mascotas del usuario	~~ISP-145~~	

Resumen de velocidad del equipo

Como vimos en la sección anterior, el equipo sprint a sprint fue evolucionando, así como el proceso aplicado que se fue puliendo cada vez más, logrando mejorar los resultados.
A continuación veremos el gráfico de velocidad del equipo en función del tiempo.



Sprint	Compromiso	Terminado
Sprint 0	6	2
Sprint 1	21	17
Sprint 2	22	20
Sprint 3	31	28
Sprint 4	26	23

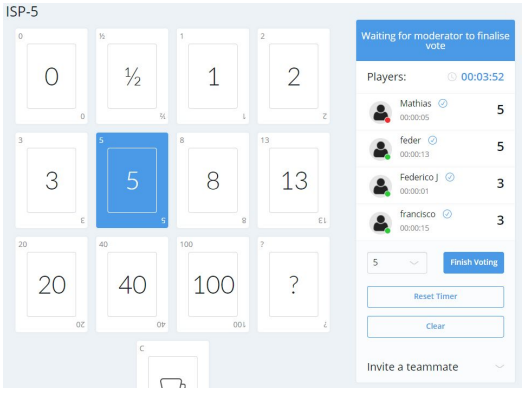
Nota: El sprint 4 contiene puntos relacionados con la creación de esta documentación (aún en progreso al momento de cerrar el sprint), por lo que se puede percibir una baja en la performance del equipo que no es del todo acertada.

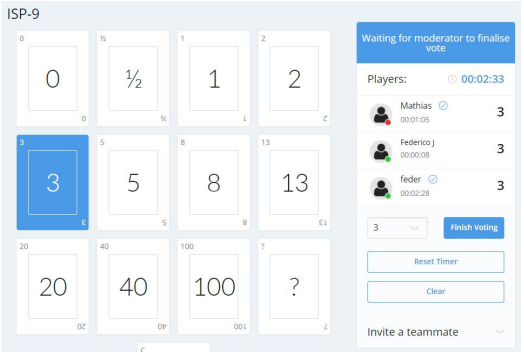
Salta a la vista la mejoría que hubo en el equipo desde el comienzo del proyecto, dejando en evidencia que el enfoque del equipo en mejorar continuamente el proceso de desarrollo se vió reflejado mediante una mejora en la productividad.

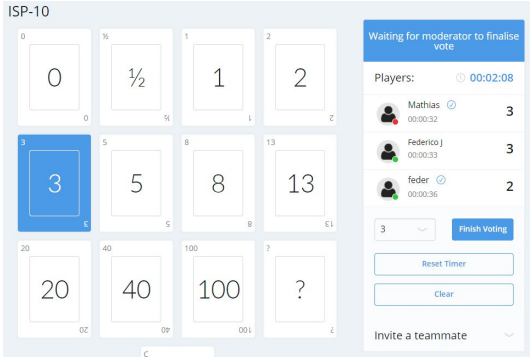
Estimaciones del equipo

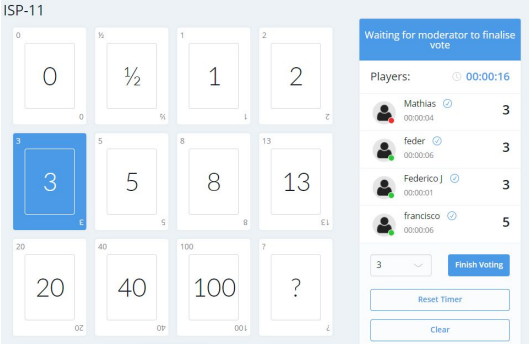
Como se había definido, para estimar las historias a realizar, el equipo participó de sesiones de estimación previas a la planning de cada sprint.
En base a las estimaciones dadas por el equipo y la velocity promedio, se decidió que alcance darle a cada sprint.
Para estimar las historias se utilizó la herramienta de **PlanItPoker.**
A continuación se adjuntan las evidencias de dichas sesiones.

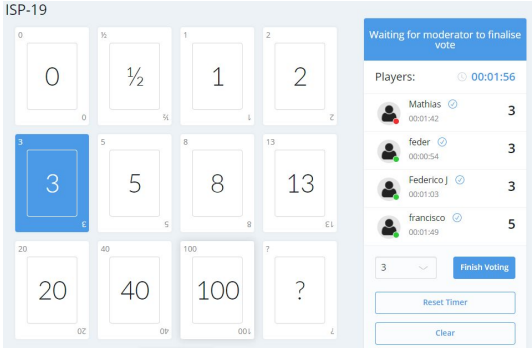
Nota: para el sprint 4 no se realizaron las sesiones de estimaciones dado que sin importar la estimación el equipo debía completar las historias para el final del sprint (los puntajes de las historias fueron puestos en base a historias similares hechas).
Sprint 1

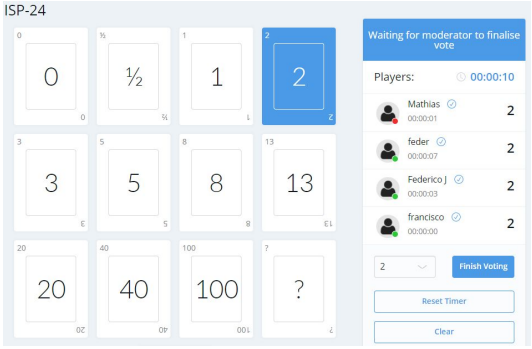




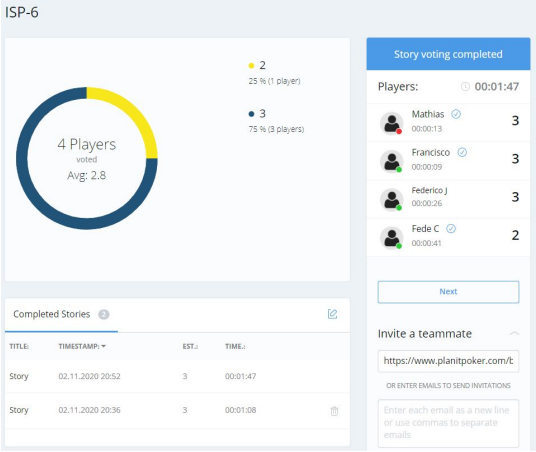


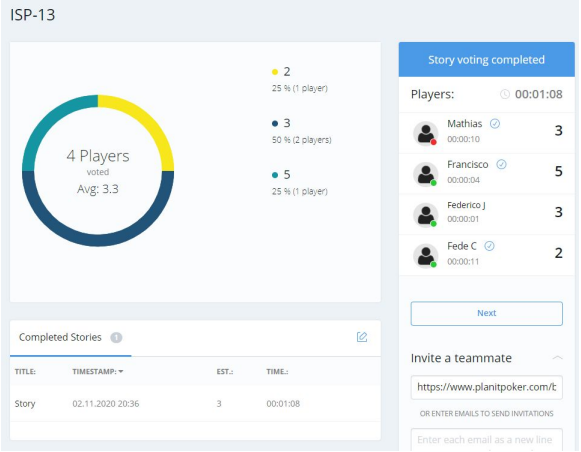


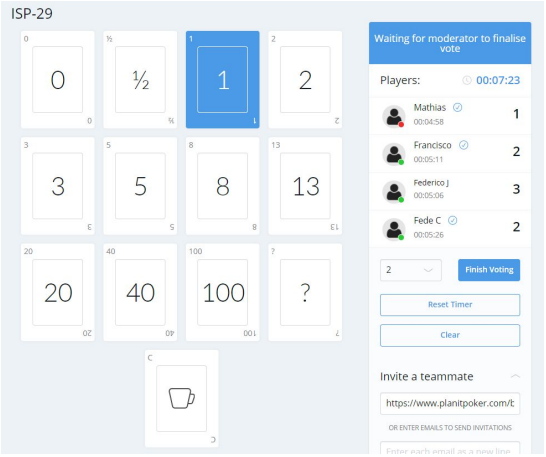




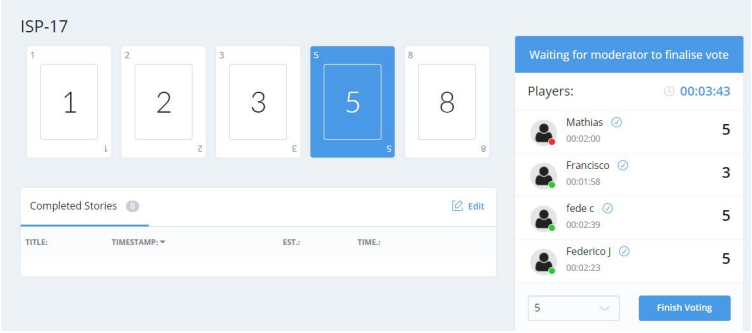
Sprint 2

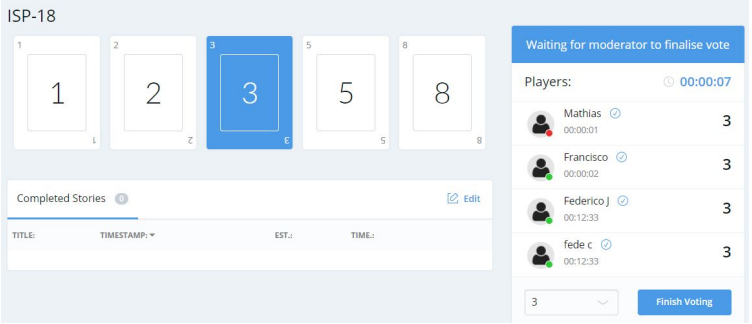

ISP-6
4 Players
voted
Avg: 2.8
2
25 % (1 player)
3
75 % (3 players)
Story voting completed
Players:
00:01:47
Mathias
00:00:13
3
Francisco
00:00:09
3
Federico J
00:00:26
3
Fede C
00:00:41
2
Next
Completed Stories
TITLE:
TIMESTAMP:
EST.:
TIME.:
Story
02.11.2020 20:52
3
00:01:47
Story
02.11.2020 20:36
3
00:01:08
Invite a teammate
https://www.planitpoker.com/t
OR ENTER EMAILS TO SEND INVITATIONS
Enter each email as a new line or use commas to separate emails

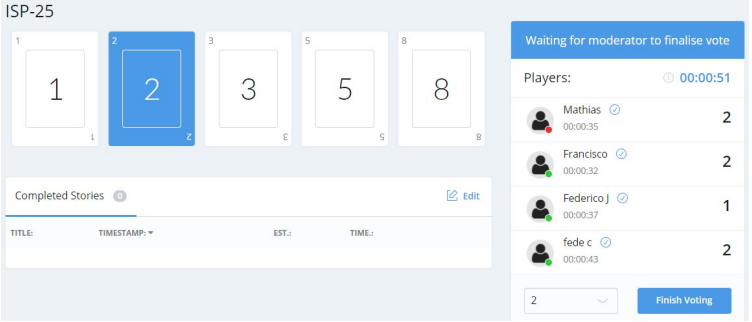

ISP-13
4 Players
voted
Avg: 3.3
2
25 % (1 player)
3
50 % (2 players)
5
25 % (1 player)
Story voting completed
Players:
00:01:08
Mathias
00:00:10
3
Francisco
00:00:04
5
Federico J
00:00:01
3
Fede C
00:00:11
2
Next
Completed Stories
TITLE:
TIMESTAMP:
EST.:
TIME.:
Story
02.11.2020 20:36
3
00:01:08
Invite a teammate
https://www.planitpoker.com/t
OR ENTER EMAILS TO SEND INVITATIONS
Enter each email as a new line

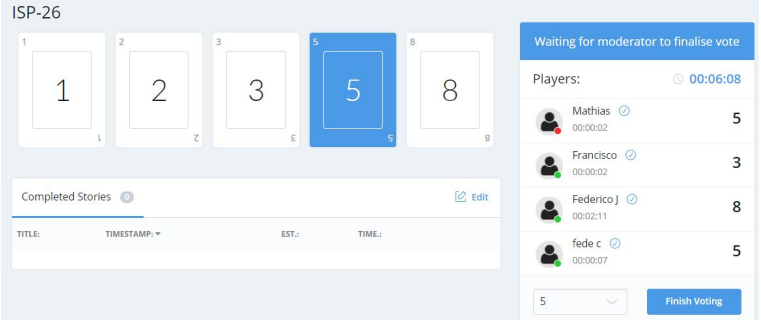

ISP-29
0
½
1
2
3
5
8
13
20
40
100
?
Waiting for moderator to finalise vote
Players:
00:07:23
Mathias
00:04:58
1
Francisco
00:05:11
2
Federico J
00:05:06
3
Fede C
00:05:26
2
2
Finish Voting
Reset Timer
Clear
Invite a teammate
https://www.planitpoker.com/t
OR ENTER EMAILS TO SEND INVITATIONS

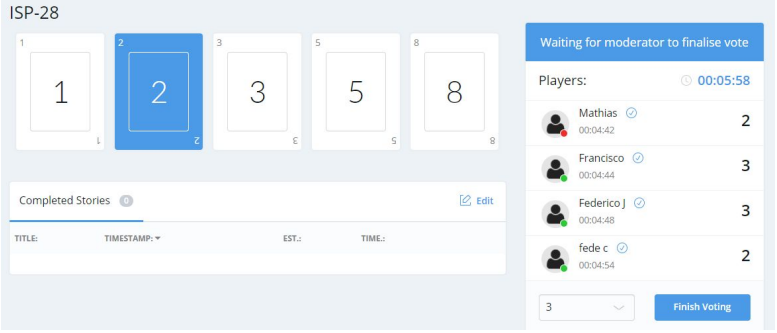
Sprint 3


ISP-17
1
2
3
5
8
Completed Stories
Edit
TITLE:
TIMESTAMP:
EST.:
TIME.:
Waiting for moderator to finalise vote
Players:
00:03:43
Mathias
00:02:00
5
Francisco
00:01:58
3
fede c
00:02:39
5
Federico J
00:02:23
5
5
Finish Voting


ISP-18
1
2
3
5
8
Completed Stories
Edit
TITLE:
TIMESTAMP:
EST.:
TIME.:
Waiting for moderator to finalise vote
Players:
00:00:07
Mathias
00:00:01
3
Francisco
00:00:02
3
Federico J
00:12:33
3
fede c
00:12:33
3
3
Finish Voting


ISP-25
1
2
3
5
8
Completed Stories
Edit
TITLE:
TIMESTAMP:
EST.:
TIME.:
Waiting for moderator to finalise vote
Players:
00:00:51
Mathias
00:00:35
2
Francisco
00:00:32
2
Federico J
00:00:37
1
fede c
00:00:43
2
2
Finish Voting

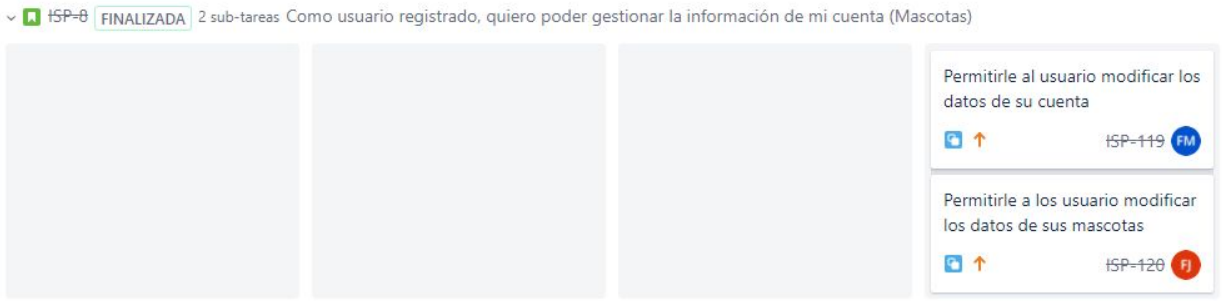

ISP-26
1
2
3
5
8
Completed Stories
Edit
TITLE:
TIMESTAMP:
EST.:
TIME.:
Waiting for moderator to finalise vote
Players:
00:06:08
Mathias
00:00:02
5
Francisco
00:00:02
3
Federico J
00:02:11
8
fede c
00:00:07
5
5
Finish Voting


ISP-28
1
2
3
5
8
Completed Stories
Edit
TITLE:
TIMESTAMP:
EST.:
TIME.:
Waiting for moderator to finalise vote
Players:
00:05:58
Mathias
00:04:42
2
Francisco
00:04:44
3
Federico J
00:04:48
3
fede c
00:04:54
2
3
Finish Voting

Tablero de sprint

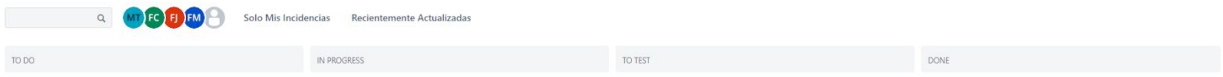
Como tablero de sprint se utilizó el de la herramienta **Jira.**
No hay mayores comentarios sobre el mismo, lo que creemos que cabe destacar es el uso de tasks en los tickets del tablero.
Esto se hizo dado que las user stories fueron realizadas con el objetivo de generar valor al cliente, pero a la hora de desarrollar una funcionalidad, la misma puede ser separada en múltiples tareas técnicas.
Con el objetivo de poder paralelizar trabajo dentro de una misma user story de manera organizada se optó por dividir los tickets del tablero en múltiples tareas independientes.

A continuación adjuntamos un ejemplo:



Como se puede observar, la user story ISP-8 fue dividida en 2 tareas, las cuales fueron llevadas a cabo por 2 miembros distintos en paralelo.

Además el tablero del sprint tuvo 4 columnas

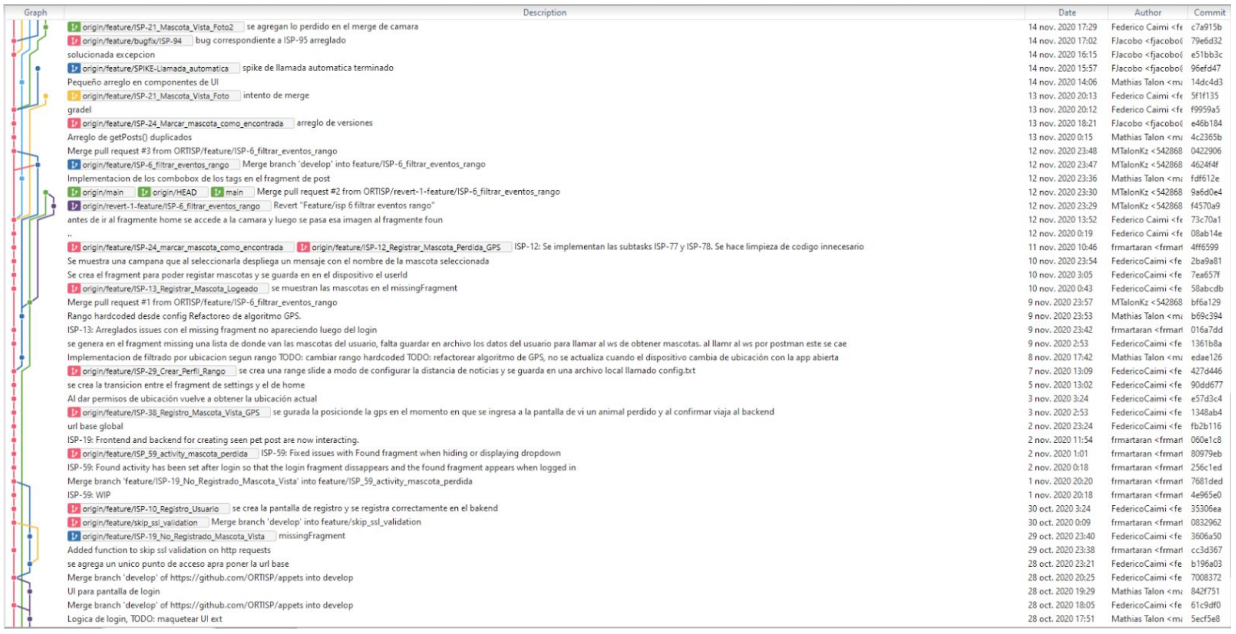


Tiene las 3 clásicas de Scrum (To do, In progress, Done) pero además agregamos la columna de To test, esto fue con el objetivo de asegurar que todas las funcionalidades implementadas fueron testeadas por otro integrante del equipo antes de ser consideradas como finalizadas.
Esto fue de gran ayuda, ya que en estas pruebas se detectaron muchos defectos temprano, lo que nos ahorró una gran cantidad de tiempo y esfuerzo en arreglos futuros.

Uso de los repositorios

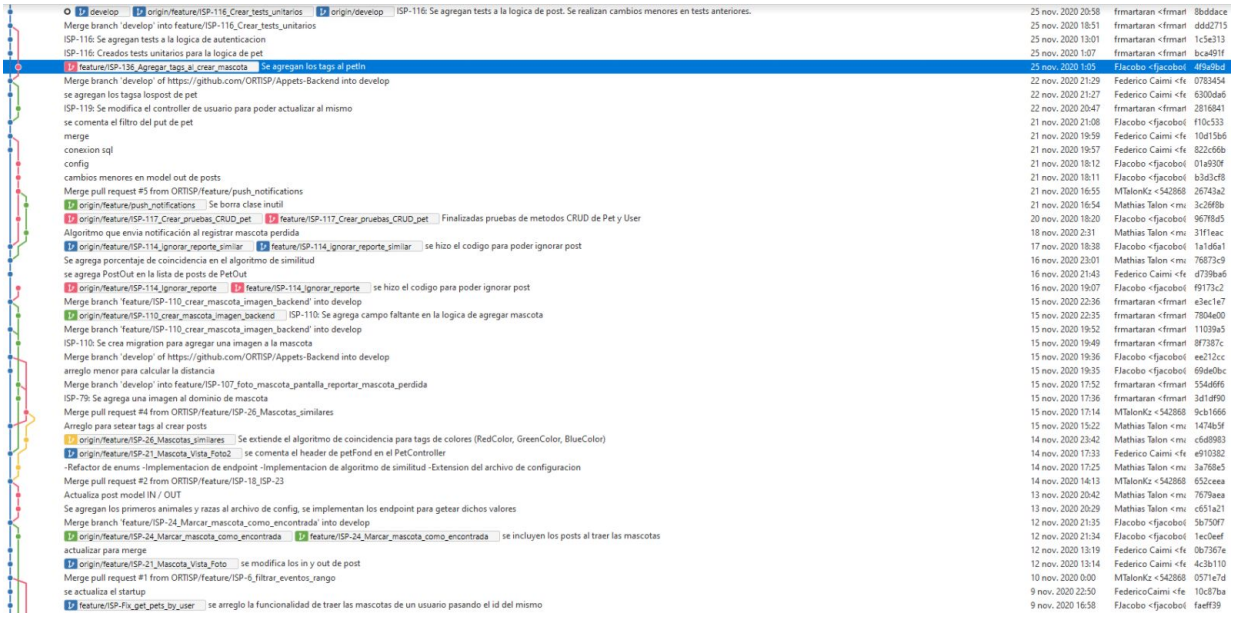
Para el desarrollo del proyecto se utilizaron 2 repositorios GitHub, uno para el sistema Frontend y otro para el Backend.
Ambos repositorios fueron utilizados bajo la metodología GitFlow y fueron de vital importancia dado que la completitud del proyecto fue desarrollado en modalidad online.
A continuación adjuntamos ejemplos del uso que se les dio a ambos repositorios.

Repositorio Frontend



Graph	Description	Date	Author	Commit
	origin/feature/ISP-21_Mascota_Vista_Foto2 se agregan lo perdido en el merge de camara	14 nov. 2020 17:29	Federico Caimi <fe	c7a915b
	origin/feature/bugfix/ISP-94 bug correspondiente a ISP-95 arreglado	14 nov. 2020 17:02	FJacobo <fjacobo(	79e6d32
	solucionada excepcion	14 nov. 2020 16:15	FJacobo <fjacobo(	e51bb3c
	origin/feature/SPIKE-Llamada_automatica spike de llamada automatica terminado	14 nov. 2020 15:57	FJacobo <fjacobo(	96efd47
	Pequeño arreglo en componentes de UI	14 nov. 2020 14:06	Mathias Talon <mı	14dc4d3
	origin/feature/ISP-21_Mascota_Vista_Foto intento de merge	13 nov. 2020 20:13	Federico Caimi <fe	5f1f135
	gradel	13 nov. 2020 20:12	Federico Caimi <fe	f9959a5
	origin/feature/ISP-24_Marcar_mascota_como_encontrada arreglo de versiones	13 nov. 2020 18:21	FJacobo <fjacobo(	e46b184
	Arreglo de getPosts() duplicados	13 nov. 2020 0:15	Mathias Talon <mı	4c2365b
	Merge pull request #3 from ORTISP/feature/ISP-6_filtrar_eventos_rango	12 nov. 2020 23:48	MTalonKz < 542868	0422906
	origin/feature/ISP-6_filtrar_eventos_rango Merge branch 'develop' into feature/ISP-6_filtrar_eventos_rango	12 nov. 2020 23:47	MTalonKz < 542868	462f4ff
	Implementacion de los combobox de los tags en el fragment de post	12 nov. 2020 23:36	Mathias Talon <mı	fdf612e
	origin/main origin/HEAD main Merge pull request #2 from ORTISP/revert-1-feature/ISP-6_filtrar_eventos_rango	12 nov. 2020 23:30	MTalonKz < 542868	9a6d0e4
	origin/revert-1-feature/ISP-6_filtrar_eventos_rango Revert "Feature/isp 6 filtrar eventos rango"	12 nov. 2020 23:29	MTalonKz < 542868	f4570a9
	antes de ir al fragmente home se accede a la camara y luego se pasa esa imagen al fragmente foun	12 nov. 2020 13:52	Federico Caimi <fe	73c70a1
	..	12 nov. 2020 0:19	Federico Caimi <fe	08ab14e
	origin/feature/ISP-24_Marcar_mascota_como_encontrada origin/feature/ISP-12_Registrar_Mascota_Perdida_GPS ISP-12: Se implementan las subtasks ISP-77 y ISP-78. Se hace limpieza de codigo innecesario	11 nov. 2020 10:46	frmartaran <frmart	4ff6599
	Se muestra una campana que al seleccionarla despliega un mensaje con el nombre de la mascota seleccionada	10 nov. 2020 23:54	FedericoCaimi <fe	2ba9a81
	Se crea el fragment para poder registar mascotas y se guarda en en el dispositivo el userId	10 nov. 2020 3:05	FedericoCaimi <fe	7ea657f
	origin/feature/ISP-13_Registrar_Mascota_Logeado se muestran las mascotas en el missingFragment	10 nov. 2020 0:43	FedericoCaimi <fe	58abcdb
	Merge pull request #1 from ORTISP/feature/ISP-6_filtrar_eventos_rango	9 nov. 2020 23:57	MTalonKz < 542868	bf6a129
	Rango hardcoded desde config Refactoreo de algoritmo GPS.	9 nov. 2020 23:53	Mathias Talon <mı	b69c394
	ISP-13: Arreglados issues con el missing fragment no apareciendo luego del login	9 nov. 2020 23:42	frmartaran <frmart	016a7dd
	se genera en el fragment missing una lista de donde van las mascotas del usuario, falta guardar en archivo los datos del usuario para llamar al ws de obtener mascotas. al llamr al ws por postman este se cae	9 nov. 2020 2:53	FedericoCaimi <fe	1361b8a
	Implementacion de filtrado por ubicacion segun rango TODO: cambiar rango hardcoded TODO: refactorear algoritmo de GPS, no se actualiza cuando el dispositivo cambia de ubicación con la app abierta	8 nov. 2020 17:42	Mathias Talon <mı	edae126
	origin/feature/ISP-29_Crear_Perfil_Rango se crea una range slide a modo de configurar la distancia de noticias y se guarda en una archivo local llamado config.txt	7 nov. 2020 13:09	FedericoCaimi <fe	427d446
	se crea la transicion entre el fragment de settings y el de home	5 nov. 2020 13:02	FedericoCaimi <fe	90dd677
	Al dar permisos de ubicación vuelve a obtener la ubicación actual	3 nov. 2020 3:24	FedericoCaimi <fe	e57d3c4
	origin/feature/ISP-38_Registro_Mascota_Vista_GPS se gurada la posicionde la gps en el momento en que se ingresa a la pantalla de vi un animal perdido y al confirmar viaja al backend	3 nov. 2020 2:53	FedericoCaimi <fe	1348ab4
	url base global	2 nov. 2020 23:24	FedericoCaimi <fe	fb2b116
	ISP-19: Frontend and backend for creating seen pet post are now interacting.	2 nov. 2020 11:54	frmartaran <frmart	060e1c8
	origin/feature/ISP_59_activity_mascota_perdida ISP-59: Fixed issues with Found fragment when hiding or displaying dropdown	2 nov. 2020 1:01	frmartaran <frmart	80979eb
	ISP-59: Found activity has been set after login so that the login fragment dissappears and the found fragment appears when logged in	2 nov. 2020 0:18	frmartaran <frmart	256c1ed
	Merge branch 'feature/ISP-19_No_Registrado_Mascota_Vista' into feature/ISP_59_activity_mascota_perdida	1 nov. 2020 20:20	frmartaran <frmart	7681ded
	ISP-59: WIP	1 nov. 2020 20:18	frmartaran <frmart	4e965e0
	origin/feature/ISP-10_Registro_Usuario se crea la pantalla de registro y se registra correctamente en el bakend	30 oct. 2020 3:24	FedericoCaimi <fe	35306ea
	origin/feature/skip_ssl_validation Merge branch 'develop' into feature/skip_ssl_validation	30 oct. 2020 0:09	frmartaran <frmart	0832962
	origin/feature/ISP-19_No_Registrado_Mascota_Vista missingFragment	29 oct. 2020 23:40	FedericoCaimi <fe	3606a50
	Added function to skip ssl validation on http requests	29 oct. 2020 23:38	frmartaran <frmart	cc3d367
	se agrega un unico punto de acceso apra poner la url base	28 oct. 2020 23:21	FedericoCaimi <fe	b196a03
	Merge branch 'develop' of https://github.com/ORTISP/appets into develop	28 oct. 2020 20:25	FedericoCaimi <fe	7008372
	UI para pantalla de login	28 oct. 2020 19:29	Mathias Talon <mı	842f751
	Merge branch 'develop' of https://github.com/ORTISP/appets into develop	28 oct. 2020 18:05	FedericoCaimi <fe	61c9df0
	Logica de login, TODO: maquetear UI ext	28 oct. 2020 17:51	Mathias Talon <mı	5ecf5e8

Repositorio Backend



	Date	Author	Commit
develop origin/feature/ISP-116_Crear_tests_unitarios origin/develop ISP-116: Se agregan tests a la logica de post. Se realizan cambios menores en tests anteriores.	25 nov. 2020 20:58	frmartaran <frmart	8bddace
Merge branch 'develop' into feature/ISP-116_Crear_tests_unitarios	25 nov. 2020 18:51	frmartaran <frmart	ddd2715
ISP-116: Se agregan tests a la logica de autenticacion	25 nov. 2020 13:01	frmartaran <frmart	1c5e313
ISP-116: Creados tests unitarios para la logica de pet	25 nov. 2020 1:07	frmartaran <frmart	bca491f
feature/ISP-136_Agregar_tags_al_crear_mascota Se agregan los tags al petIn	25 nov. 2020 1:05	FJacobo <fjacobo(	4f9e9bd
Merge branch 'develop' of https://github.com/ORTISP/Appets-Backend into develop	22 nov. 2020 21:29	Federico Caimi <fe	0783454
se agregan los tagsa lospost de pet	22 nov. 2020 21:27	Federico Caimi <fe	6300da6
ISP-119: Se modifica el controller de usuario para poder actualizar al mismo	22 nov. 2020 20:47	frmartaran <frmart	2816841
se comenta el filtro del put de pet	21 nov. 2020 21:08	FJacobo <fjacobo(	f10c533
merge	21 nov. 2020 19:59	Federico Caimi <fe	10d15b6
conexion sql	21 nov. 2020 19:57	Federico Caimi <fe	822c66b
config	21 nov. 2020 18:12	FJacobo <fjacobo(	01e930f
cambios menores en model out de posts	21 nov. 2020 18:11	FJacobo <fjacobo(	b3d3cf8
Merge pull request #5 from ORTISP/feature/push_notifications	21 nov. 2020 16:55	MTalonKz < 542868	26743a2
origin/feature/push_notifications Se borra clase inutil	21 nov. 2020 16:54	Mathias Talon <mı	3c26f8b
origin/feature/ISP-117_Crear_pruebas_CRUD_pet feature/ISP-117_Crear_pruebas_CRUD_pet Finalizadas pruebas de metodos CRUD de Pet y User	20 nov. 2020 18:20	FJacobo <fjacobo(	967f8d5
Algoritmo que envia notificación al registrar mascota perdida	18 nov. 2020 2:31	Mathias Talon <mı	31f1eac
origin/feature/ISP-114_ignorar_reporte_similar feature/ISP-114_ignorar_reporte_similar se hizo el codigo para poder ignorar post	17 nov. 2020 18:38	FJacobo <fjacobo(	1a1d6a1
Se agrega porcentaje de coincidencia en el algoritmo de similitud	16 nov. 2020 23:01	Mathias Talon <mı	76873c9
se agrega PostOut en la lista de posts de PetOut	16 nov. 2020 21:43	Federico Caimi <fe	d739ba6
origin/feature/ISP-114_ignorar_reporte feature/ISP-114_ignorar_reporte se hizo el codigo para poder ignorar post	16 nov. 2020 19:07	FJacobo <fjacobo(	f9173c2
Merge branch 'feature/ISP-110_crear_mascota_imagen_backend' into develop	15 nov. 2020 22:36	frmartaran <frmart	e3ec1e7
origin/feature/ISP-110_crear_mascota_imagen_backend ISP-110: Se agrega campo faltante en la logica de agregar mascota	15 nov. 2020 22:35	frmartaran <frmart	7804e00
Merge branch 'feature/ISP-110_crear_mascota_imagen_backend' into develop	15 nov. 2020 19:52	frmartaran <frmart	11039a5
ISP-110: Se crea migration para agregar una imagen a la mascota	15 nov. 2020 19:49	frmartaran <frmart	8f7387c
Merge branch 'develop' of https://github.com/ORTISP/Appets-Backend into develop	15 nov. 2020 19:36	FJacobo <fjacobo(	ee212cc
arreglo menor para calcular la distancia	15 nov. 2020 19:35	FJacobo <fjacobo(	69de0bc
Merge branch 'develop' into feature/ISP-107_foto_mascota_pantalla_reportar_mascota_perdida	15 nov. 2020 17:52	frmartaran <frmart	554d6f6
ISP-79: Se agrega una imagen al dominio de mascota	15 nov. 2020 17:36	frmartaran <frmart	3d1df90
Merge pull request #4 from ORTISP/feature/ISP-26_Mascotas_similares	15 nov. 2020 17:14	MTalonKz < 542868	9cb1666
Arreglo para setear tags al crear posts	15 nov. 2020 15:22	Mathias Talon <mı	1474b5f
origin/feature/ISP-26_Mascotas_similares Se extiende el algoritmo de coincidencia para tags de colores (RedColor, GreenColor, BlueColor)	14 nov. 2020 23:42	Mathias Talon <mı	c6d8983
origin/feature/ISP-21_Mascota_Vista_Foto2 se comenta el header de petFond en el PetController	14 nov. 2020 17:33	Federico Caimi <fe	e910382
-Refactor de enums -Implementacion de endpoint -Implementacion de algoritmo de similitud -Extension del archivo de configuracion	14 nov. 2020 17:25	Mathias Talon <mı	3a768e5
Merge pull request #2 from ORTISP/feature/ISP-18_ISP-23	14 nov. 2020 14:13	MTalonKz < 542868	652ceea
Actualiza post model IN / OUT	13 nov. 2020 20:42	Mathias Talon <mı	7679aea
Se agregan los primeros animales y razas al archivo de config, se implementan los endpoint para getear dichos valores	13 nov. 2020 20:29	Mathias Talon <mı	c651a21
Merge branch 'feature/ISP-24_Marcar_mascota_como_encontrada' into develop	12 nov. 2020 21:35	FJacobo <fjacobo(	5b750f7
origin/feature/ISP-24_Marcar_mascota_como_encontrada feature/ISP-24_Marcar_mascota_como_encontrada se incluyen los posts al traer las mascotas	12 nov. 2020 21:34	FJacobo <fjacobo(	1ec0eef
actualizar para merge	12 nov. 2020 13:19	Federico Caimi <fe	0b7367e
origin/feature/ISP-21_Mascota_Vista_Foto se modifica los in y out de post	12 nov. 2020 13:14	Federico Caimi <fe	4c3b110
Merge pull request #1 from ORTISP/feature/ISP-6_filtrar_eventos_rango	10 nov. 2020 0:00	MTalonKz < 542868	0571e7d
se actualiza el startup	9 nov. 2020 22:50	FedericoCaimi <fe	10c87ba
feature/ISP-Fix_get_pets_by_user se arreglo la funcionalidad de traer las mascotas de un usuario pasando el id del mismo	9 nov. 2020 16:58	FJacobo <fjacobo(	faeff39

Diseño y arquitectura del sistema

Backend

Para el sistema de backend de la aplicación se implementó una Web Api desarrollada en .NET core.
Para garantizar un sistema escalable y modificable se decidió implementar el sistema por capas, que se comunican entre ellas por medio de la técnica de inyección de dependencias.
Para la persistencia de datos fue utilizado Entity Framework en la modalidad de Code first, por lo que el esquema de la base de datos SQL fue construido a través del código de la solución.
Como se acordó en la primera entrega de este documento, la capa de lógica de negocios cuenta con una cobertura de pruebas mayor al 90%.

A continuación se mostrarán una serie de diagramas que facilitarán el entendimiento de la estructura del sistema.

Diagrama de componentes
En este diagrama veremos los distintos componentes del sistema representado por los archivos físicos a los que el mismo se traduce.

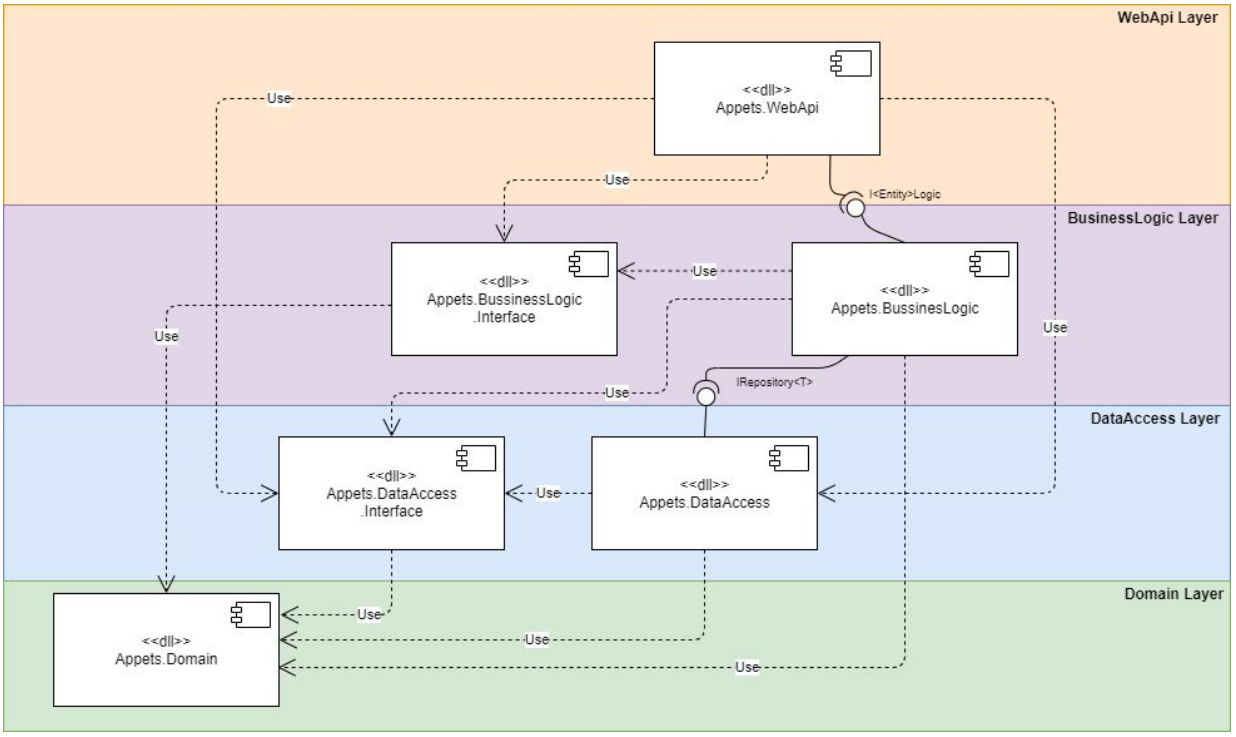
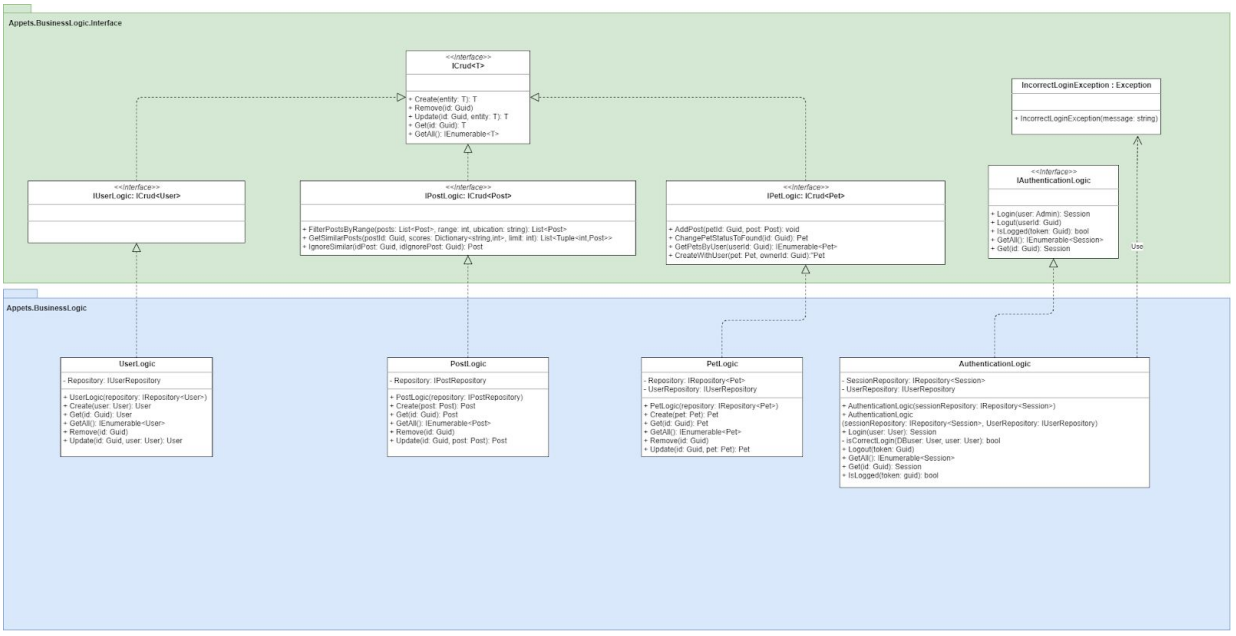


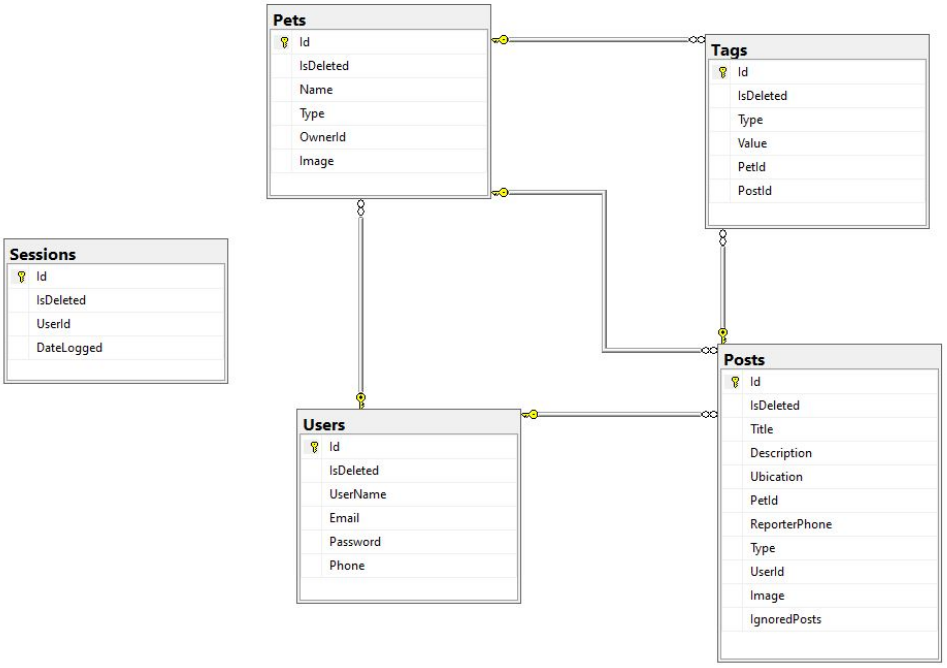
Diagrama de clases

En este caso veremos el diagrama de clases del paquete de lógica de negocio, ya que entendemos que es el más interesante de mostrar.



Esquema de base de datos

A continuación veremos la estructura de la base de datos, que fue generada a partir del código fuente utilizando la herramienta de Entity Framework.



Cobertura de código

Para asegurar la calidad del código el equipo definió la meta de conseguir una cobertura de pruebas en el código de lógica de negocio que sea superior a 90%.

Una vez finalizado el sistema podemos asegurar que el objetivo se cumplió y el equipo superó las expectativas, dado que el código ha quedado cubierto en un 98%.

A continuación se adjunta evidencia.

appets.businesslogic.dll	9	2.00%	442	98.00%
Appets.BusinessLogic	9	2.00%	442	98.00%
AuthenticationLogic	3	3.80%	76	96.20%
PetLogic	3	3.16%	92	96.84%
PetLogic.<>c__Displa...	1	12.50%	7	87.50%
PostLogic	2	0.95%	208	99.05%
PostLogic.<>c	0	0.00%	4	100.00%
PostLogic.<>c__Displ...	0	0.00%	5	100.00%
PostLogic.<>c__Displ...	0	0.00%	4	100.00%
UserLogic	0	0.00%	46	100.00%

Frontend

La implementación del sistema frontend fue realizada en Android studio, con código fuente basado en Java.

Nos basamos en una arquitectura basada en fragments, lo que implica una Activity principal de donde se llama a distintos fragmentos y estos se comunican entre sí a modo de componentes. Algunos de estos fragments fueron reutilizados para pantallas donde su funcionalidad era similar.

A nivel de experiencia de usuario utilizamos la librería de Material design que contiene una normativa de diseño enfocado en la visualización del sistema operativo Android.
Otra librería utilizada fue retrofit la cual brinda una forma muy sencilla de comunicación con el servidor.

Con el fin de poder enviar notificaciones se utilizó el servicio de firebase el cual genera un topic por el cual los dispositivos se registran y el backend envía notificaciones a los mismos. Logrando así una implementación de publish suscriber.

A modo de simplificar el desarrollo se utilizó aplicaciones de terceros como la cámara nativa y el google maps del dispositivo.

Manual de Instalación

Antes de comenzar a utilizar la aplicación, se deben de realizar los siguientes cambios de configuración:

- Para instalar la base de datos, abrimos la línea de comandos y nos cambiamos al directorio de la aplicación backend. Desde allí, nos cambiamos a la carpeta *Appets.WebApi,* y ejecutamos el siguiente comando:

```
dotnet ef database update -p ../Appets.DataAccess
```

- Para ejecutar la aplicación de backend, abrimos la línea de comandos y nos cambiamos al directorio de la aplicación backend. Luego ejecutamos lo siguiente:

```
dotnet run --urls {{url_de_pc}}
```

- Dentro del directorio de la aplicación de android, en el archivo *build.gradle* ubicado en `\*AndroidAppets\app*`, cambiamos la url de la configuración por la url correspondiente a la ip de nuestra pc.

```
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.1"

    defaultConfig {
        applicationId 'com.example.appet'
        minSdkVersion 16
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        buildConfigField "String", "API_URL", "\"https://192.168.251.1:44364/\""
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
}
```

- Si queremos cambiar los animales aceptados por el sistema, su raza o el puntaje, vamos al archivo *appsettings.json,* dentro de la carpeta *Appets.WebApi* que se encuentra en la solución del backend. En este archivo vamos a poder editar las secciones *TagAnimals, TagBreeds y TagScores* de acuerdo a los criterios que se consideren necesarios. Por defecto, el sistema soporta los siguientes animales, razas y puntajes:

```
"TagAnimals": ["Perro", "Gato", "Loro"],
"TagBreeds": {
  "Perro": ["Ovejero Aleman", "Chiuahua"],
  "Gato": ["Siamés", "Callejero"],
  "Loro": ["Tropical", "Pecho azul"]
},
"TagScores": {
  "Animal": 1000,
  "Breed": 500,
  "RedColor": 255,
  "GreenColor": 255,
  "BlueColor": 255
},
"LimitSimilarPost": 1600
```

La propiedad LimitSimilarPost define qué cantidad de puntos son necesarios para que el algoritmo de coincidencia entienda que una mascota sea similar a otra.
Por defecto el puntaje es de 1600 que se traduce a un 70% de coincidencia entre mascotas.

Aspectos a mejorar

Consideramos que finalizando este proyecto se obtuvo una aplicación completa y estable. Si bien estamos conformes con este resultado, somos conscientes de que hay aspectos de la aplicación que podrían ser mejorados para perfeccionar el producto, así como funcionalidades que se podrían agregar en un futuro.

Refactorizar código del frontend

El código final del frontend no quedó tan prolijo como nos hubiera gustado. Priorizamos el correcto funcionamiento de las funcionalidades del sistema sobre la prolijidad del desarrollo. Si a esto le sumamos que android studio es una tecnología nueva para nosotros que requirió mucho ensayo y error, obtenemos como resultado un código que no cumple correctamente varias de las buenas prácticas de clean code.
Hay funciones que hacen cosas muy similares dando como resultado código repetido.
También existen funciones con una cantidad de líneas de código excesiva.
Por último, hay nombres de variables y métodos que no son lo suficientemente nemotécnicos.

Guardar fotos en firebase

Actualmente, las fotos que se agregan a la aplicación se están guardando en la base de datos como un string. Este dato representa el BitMap correspondiente a la imagen transformado en string. Esta no es la mejor manera de guardar imágenes. Lo correcto sería utilizar un servicio de almacenamiento que permite guardar imágenes en su formato original y con una calidad mucho mayor. Para esto podríamos utilizar el servicio de almacenamiento en la nube que ofrece Firebase.

Agregar un logout

En la actualidad, la aplicación no permite que el usuario cierre la sesión. Igualmente, hay dos maneras de borrar los datos del usuario y quedar deslogueado. Una de estas es cambiar la contraseña en modificar datos del usuario. La otra manera es desinstalar la aplicación e instalarla nuevamente. Creemos que podría agregarle valor al usuario una funcionalidad que permita que este cierre su sesión actual quedando en estado deslogueado.

Desactivar notificaciones

No existe en la aplicación una manera de desactivar las notificaciones. Si bien todos los sistemas operativos tienen una manera de desactivar las notificaciones para aplicaciones específicas, creemos que, agregar una opción en la configuración que permita activar y desactivar las notificaciones de la aplicación, podría ser útil para el usuario.

Notificaciones automáticas filtradas por rango

El estado actual de la aplicación permite notificar a todos los usuarios del sistema cuando se crea un post de mascota perdida. No es útil para el usuario que se le notifique sobre el extravío de una mascota en otra parte del mundo ya que es imposible que encuentre a dicha mascota. Esto no le aporta valor al cliente por lo que se debería filtrar el envío de notificaciones push. La idea es que en un futuro se cambie el funcionamiento del sistema para que cuando se pierde una mascota solo se le notifique a los usuarios cercanos a la ubicación de la publicación.

Diseño UI / UX del sistema

En esta sección presentaremos capturas del sistema con un breve comentario de aspectos interesantes tanto de la perspectiva de UI como de UX.

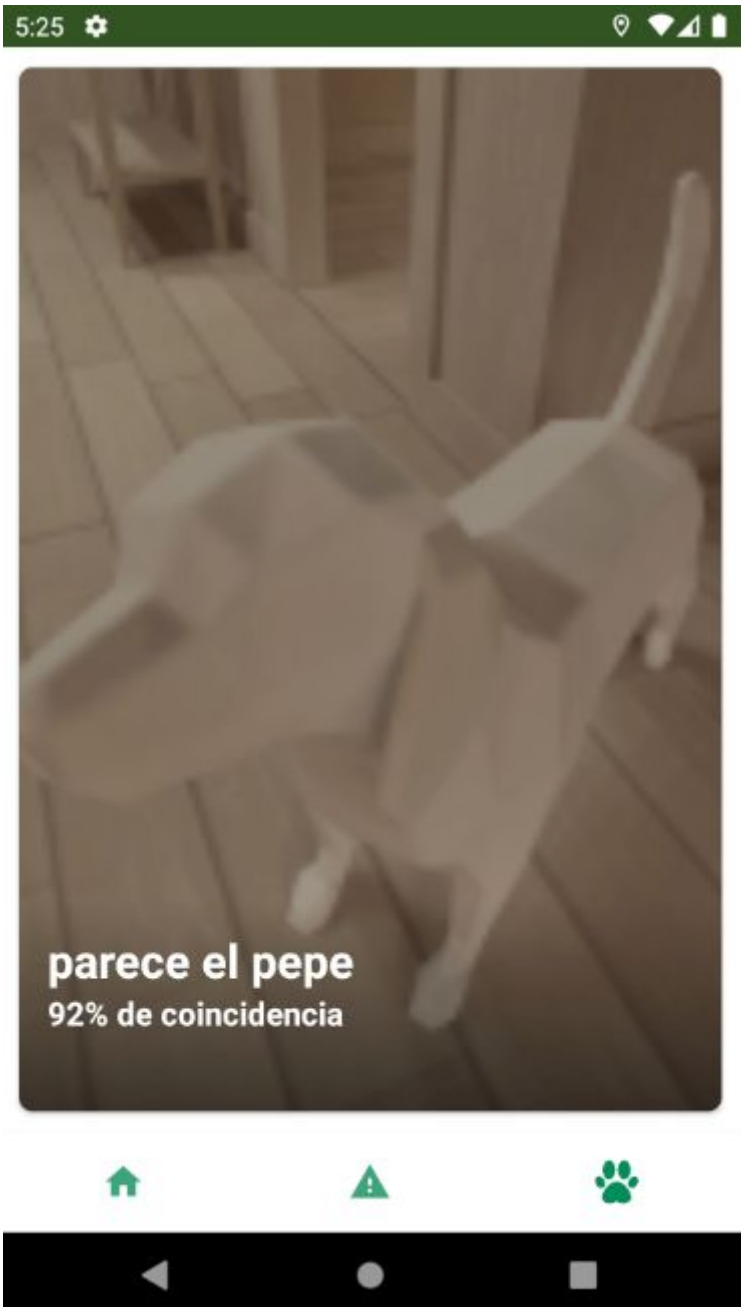
Pantalla principal

Se puso especial énfasis en lograr una interfaz sencilla e intuitiva.
Las opciones principales del sistema se ubicaron en la zona de la derecha/abajo dado que es la zona de más facilidad para llegar con el pulgar derecho de la mano.
Los eventos del sistema son mostrados en formatos de carta, las cuales presentan información relevante, de manera clara y son fáciles de identificar según su tipo.

Mis mascotas

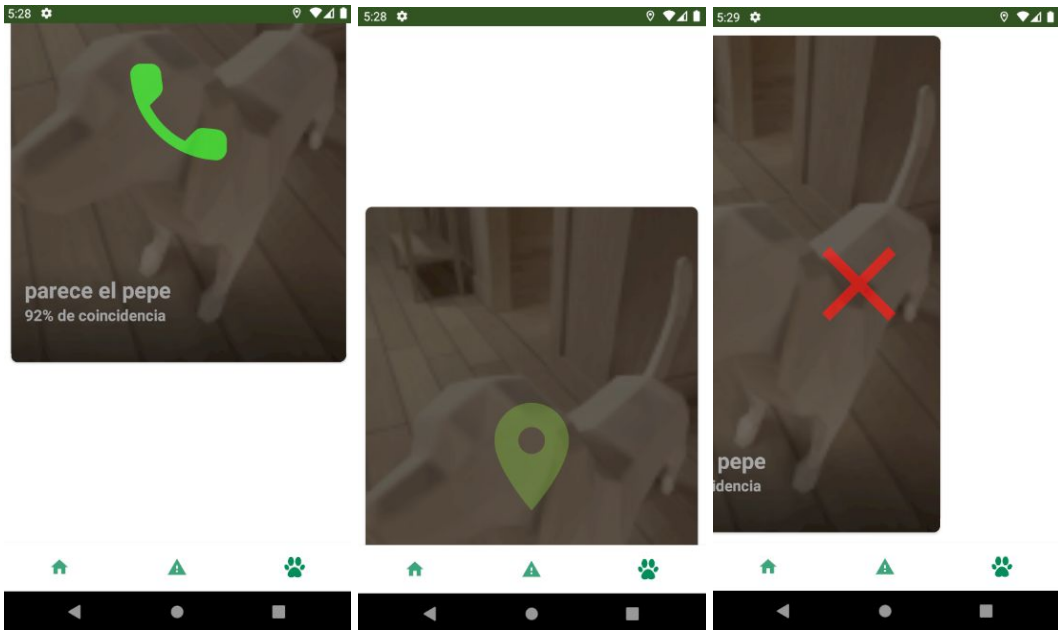
Se creó una pantalla en la cual se pudiese administrar las mascotas y mascotas perdidas del usuario de una manera sencilla e intuitiva.
Se decidió colocar iconos descriptivos para acceder a las principales funciones que se pueden realizar sobre cada ítem.

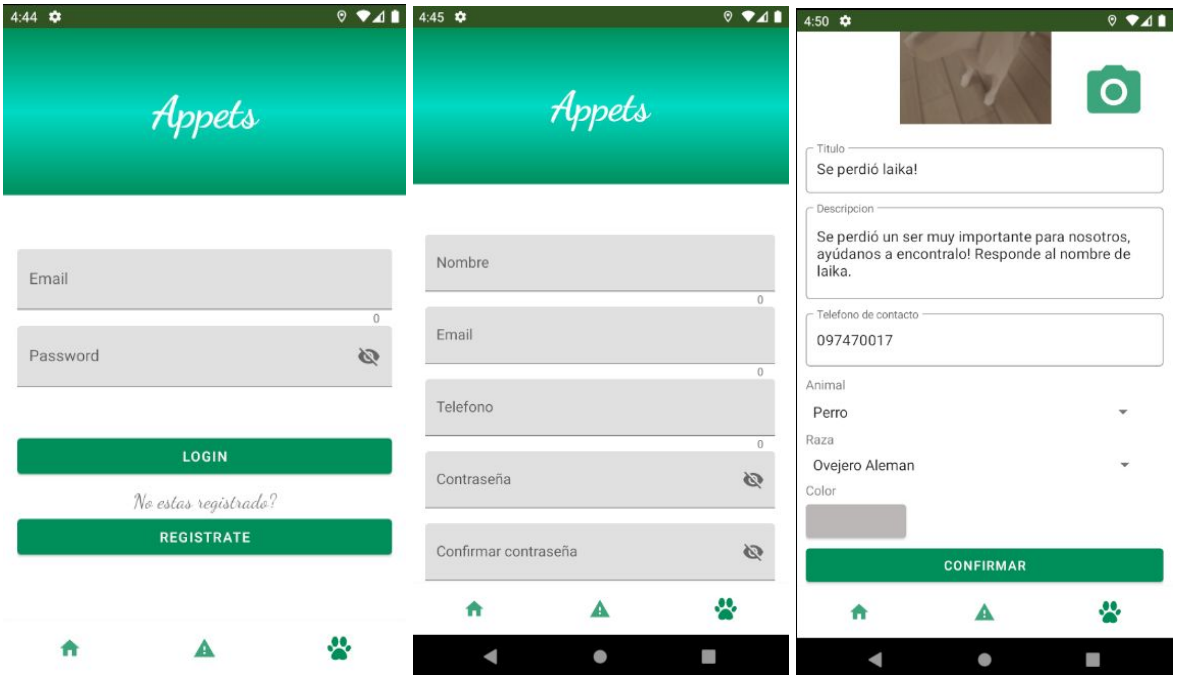
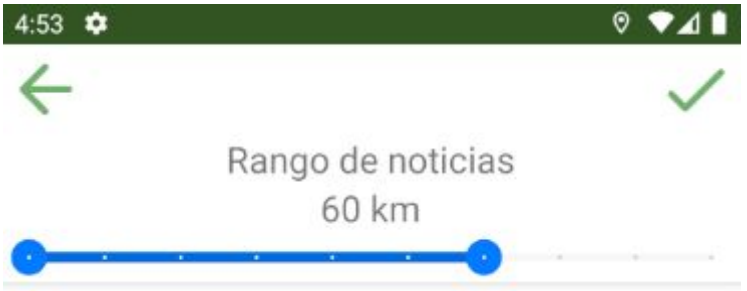
Mascotas similares



Creamos una pantalla para visualizar las mascotas similares a la mascota perdida del usuario que fueron vistas y que podrían ser la mascota buscada.
Se implementó utilizando tarjetas con movimiento y acciones asociadas a cada movimiento: ver ubicación, llamar al usuario que reportó la mascota vista, ignorar mascota, ver siguiente mascota.

A continuación mostramos capturas de las acciones mencionadas.



Formularios

Los formularios se diseñaron para ser claros e intuitivos de completar.
A continuación adjuntamos capturas de distintos formularios del sistema.