

Proyecto PETS

(Torneo Cepit)

INTEGRANTES:

Carratú, Alejandra
Gosparini, Celeste
Lastra, Juan Ignacio
Yakimiuk, Flavia Danila
Zorrilla, Jesús Emanuel

PROFESORES:

Beltracchi, Rodrigo
Bersano, Gustavo
Lasaga, Gonzalo

Tabla de contenido

Introducción.....	1
Objetivos	1
Alcance	2
Análisis de la competencia	2
Presentación de la aplicación	4
Desarrollo de la aplicación.....	5
Casos de uso	6
Metodología de trabajo	10
Inconvenientes	10
Desarrollos a futuro	11
Conclusiones.....	11
Anexos	13

Introducción

Existen una gran cantidad de mascotas perdidas y encontradas en la calles de nuestra ciudad. Se observa que la gente se organiza a través de múltiples grupos en Facebook para localizarlas, darles alojamiento temporal, brindarles asistencia médica, adoptarlas, etc. Además, como estos grupos son múltiples, la información se replica, ocasionando pérdida de tiempo tanto a quien lee como a quien publica.

Si bien Facebook puede servir, en principio, para poner en contacto a la gente, sus características no son las ideales para almacenar, ordenar, y recuperar datos de manera eficiente y clara. Por lo tanto, estas personas solidarias realizan grandes esfuerzos, en su tiempo libre, intentando recordar nombres, direcciones o zonas donde una mascota se perdió para lograr reencontrarla su dueño.

Debido a la gran cantidad de mascotas pérdidas por sus dueños o encontradas por otras personas, se decide crear este programa para unificar todas las búsquedas de manera eficiente, manteniendo un orden actualizado, mejorando la visualización, y diferenciando los distintos estados de las mascotas publicadas.

Se trata de un programa con base de datos que permite acceder o dar a conocer información relacionada con mascotas en estado de vulnerabilidad. El programa, por un lado, organiza la información según las características de tal estado: en adopción, perdido, encontrado, con necesidad de alojamiento, entre otras. Por otro lado, permite a los usuarios interactuar con la información según distintas modalidades: un usuario que solo observa, que participa o que se registra.

Objetivos

Nos enfocamos en los siguientes objetivos:

Objetivos del producto:

- Administrar la información de las mascotas de manera eficiente.
- Optimizar la búsqueda de mascotas y centralizar toda la información en un sólo lugar.
- Conectar personas y ONGs relacionadas con la protección de animales para que puedan ayudar a las mascotas en estado de vulnerabilidad.

Objetivos en relación al equipo:

- Fortalecer e integrar los conocimientos adquiridos en el curso.
- Afianzar habilidades de trabajo en equipo para la futura vida profesional.
- Contribuir a la solución de una problemática real.
- Desarrollar una aplicación completa desde la generación de la idea hasta el desarrollo del código.
- Competir sanamente con otras comisiones aprendiendo del intercambio.

Alcance

Este programa está dirigido a todas las personas, u ONGs de defensa y protección de animales, que tengan la necesidad de buscar un animal perdido o dar aviso de uno encontrado, adoptar, buscar tránsito o ayuda para una mascota, de forma simple, rápida y directa.

Análisis de la competencia

Hemos analizado otros programas que ofrecen un servicio parecido. Puntualizamos, a continuación las ventajas y desventajas que tales programas presentan:

Wizapet ubica en el mapa el lugar en donde se perdió o vio por última vez a la mascota ingresada y agrega una foto a la ubicación. Además permite agregar una recompensa por la ayuda, contactar al dueño de la mascota y enviar mensajes vía chat. Como desventaja se puede señalar que presenta problemas al registrarse por Facebook y para ingresar con el

usuario, además de que la interfaz es confusa y contiene muchos anuncios. Tiene unas 10.000 descargas lo que hace pensar que todavía le falta camino por recorrer.

Se *busCAN* puede configurar un radio de usuarios cercanos a la mascota perdida y notificar cuando la mascota fue encontrada. Tiene dos tipos de notificaciones: perdidos y encontrados. Los inconvenientes son que se produce error al registrarse, tiene sólo 500 descargas y eso es determinante ya que su funcionamiento depende de que muchos usuarios la usen.

Mascotas Perdidas si bien se ofrece como una app con múltiples idiomas (inglés, portugués, español), sin límites en las publicaciones, y con varias categorías (perdidos, rescatados, y adoptados), tiene los inconvenientes de no detectar el país de origen, está desactualizada, y sólo tiene 100 descargas con cuatro comentarios de usuarios, con juicios negativos.

WOF permite reportar mascotas perdidas, con urgencias o situación de maltrato, pero presenta el inconveniente de detenerse por momentos, de salir de la aplicación y fallar. Tiene excesiva publicidad y hace falta que mucha gente local la use para ser interesante.

Estoy Perdido – I'm Lost envía la foto y geolocalización de la mascota a Facebook por si alguien reconoce la mascota. Adicionalmente, las protectoras que reciban mascotas encontradas pueden enviar su foto y localización. Sin embargo, no permite subir fotos por celular, tiene sólo 1000 descargas, y no está actualizada.

Pet Finder SOS envía notificación SOS para todos los usuarios en los alrededores de la zona donde la mascota fue registrada como perdida y también a su dueño. Entre los aspectos negativos se destacan que solicita a los usuarios registrarse a través de Facebook, que tiene sólo 5000 descargas, poca gente opina sobre ella, se usa en Europa, y hace un año que no se actualiza.

Siguiendo Huellitas permite registrar animales en distintos estados, como perdidos, encontrados o en adopción, además de ingresar información relevante, acerca de las razas de perros, comportamiento, salud y cuidados, pero no deja filtrar mascotas por zona, y no aparecen en el mapa las distintas mascotas. Además, la aplicación es inestable o no permite su acceso. Es una creación argentina, que si bien fue pensada para latinoamérica, terminó usándose en CABA y alrededores. Su última versión es de hace 3 años.

Redmascotera es una red solidaria que se dedica a la recuperación de mascotas perdidas en su sitio web. Además brinda servicios de adopción, muestra los adoptados y los reencuentros. El sitio funciona bien pero exige registros con correo electrónico y nombre como datos obligatorios. Aunque con consentimiento del usuario, notamos que los datos sensibles como teléfono o dirección quedan expuestos. Nuevamente, aunque permite el uso filtrando por países y regiones o barrios, es otro programa usado mayormente por gente de un cierto lugar, en este caso CABA y GBA.

Se puede concluir que muchas aplicaciones no están mantenidas y han dejado de funcionar correctamente. Otras, prometen muchos servicios que luego no son capaces de cumplir. Por último, las que funcionan mejor, dependen de que muchos otros usuarios participen activamente para mantener la base de datos. Al ser aplicaciones tan globales, hay gente de distintas partes del mundo que publica y esa información no es útil para usuarios que viven en lugares alejados. Se desprende del relevamiento que los usuarios buscan programas sencillos que no impliquen registro, que brinden información actualizada y significativa relacionada con su zona de residencia.

Presentación de la aplicación

La pantalla principal ofrece las siguientes funciones:

- **Perdidos:** contiene publicaciones de mascotas que fueron perdidas por sus dueños.

- **Encontrados:** contiene publicaciones de mascotas que fueron encontradas por alguien.
- **Tránsito:** contiene publicaciones de mascotas que necesitan alojamiento mientras se busca a su dueño o esperan ser adoptadas, es decir, que buscan tránsito.
- **Ser Solidario:** contiene publicaciones de pedidos a la comunidad como pueden ser medicamentos, dinero para atención veterinaria, etc.
- **Adoptame:** contiene publicaciones de mascotas que esperan ser adoptadas.

Los tipos de usuarios que la aplicación prevé son:

- **Usuario visitante:** es aquel que baja el programa y explora la información.
- **Usuario participante:** es aquel que publica un aviso de mascota perdida o encontrada, dejando su correo electrónico para ser contactado.
- **Usuario registrado:** es aquel que deja su nombre, apellido, dirección, y correo electrónico porque desea hacer pedidos solidarios o de tránsito.

Además, el programa permite a los usuarios realizar las siguientes intervenciones:

- **Administración de usuarios:** las altas y modificaciones de usuarios son realizadas por ellos mismos. Los roles dependen de los servicios que quieran utilizar.
- **Administración de mascotas:** los usuarios participantes o registrados pueden dar de alta y modificar sus propias publicaciones de mascotas perdidas y encontradas.

El sistema gestiona las bajas de manera automática.

Desarrollo de la aplicación

El diseño del programa se realizó con el modelo Model-Control-View, separando lo que es estructura pura para el mapeo, la administración de objetos con sus funcionalidades, y la parte gráfica, respectivamente. Por tratarse de un sistema de mascotas, tenemos tres clases básicas: usuario (UsuarioDto), mascota (MascotaDto), y ficha (FichaDto). Además, un

administrador para cada una: UsuarioCtr, MascotaCtr, y FichaCtr. En esta primera versión, optamos por colores neutros como el gris y el blanco para las pantallas.

Los diagramas de clases, y de entidad-relación realizados se encuentran detallados en la segunda entrega.

Casos de uso

Nombre del CU	Cargar mascota
Actor Principal	Usuario (cualquiera)
Objetivo / Descripción Breve	Este CU tiene como objetivo introducir los datos de la mascota encontrada o perdida, creando la ficha correspondiente.
Flujo Básico	<p>Este CU comienza cuando un usuario ingresa al sistema y presiona “cargar mascota”.</p> <ol style="list-style-type: none"> 1. El sistema le pide un correo electrónico. 2. El usuario elige “Encontrados”, o “Perdidos” dentro de las opciones disponibles, y según sea el caso. 3. El usuario completa los datos restantes básicos: sexo, color, fecha en que la encontró, la zona, y tipo de mascota. 4. [INCLUDE] ingresar otros datos. 5. Si la mascota es “encontrada”, el sistema le asigna un nombre generado automáticamente. En cambio, si la mascota está “Perdida”, el usuario ingresará el nombre. 6. El usuario presiona en opción “guardar”. 7. El sistema guarda los datos. 8. El CU termina.
Flujos Alternativos	<p>A1. Cancelar</p> <p>A1.1 El usuario decide cancelar la carga.</p> <p>A1.2 El CU termina.</p>

Pre- y post-condiciones	[PRE] 1-El usuario ingresa al sistema. 2-El usuario tiene un correo electrónico. 3- La mascota no está previamente cargada en “encontrados”. [POST] El alta de la mascota encontrada está registrada.
Casos de usos incluidos	Ingresar otros datos

Nombre del CU	Ingresar otros datos
Actor Principal	Usuario (cualquiera)
Objetivo / Descripción Breve	Este CU tiene como objetivo introducir otros datos más específicos de acuerdo al tipo de mascota encontrada/perdida para crear la ficha correspondiente.
Flujo Básico	Este CU comienza cuando un usuario elige el tipo de mascota en “cargar mascota”. 1. El usuario completa otros datos específicos que le pide el sistema como raza, tamaño, color, etc. 2. El CU termina.
Pre- y post-condiciones	[PRE] 1-El usuario debe contar con los datos a ingresar. [POST] Los datos adicionales se muestran por pantalla junto a los datos de la ficha básica.
Casos de usos incluidos	Está incluido en los casos de uso “Cargar mascota encontrada” y en “Cargar mascota perdida”. Es insertado en el PUNTO POSTERIOR a elegir “tipo de mascota”.

Nombre del CU	Solicitar tránsito
Actor Principal	Usuario registrado
Objetivo / Descripción Breve	Este CU tiene como objetivo solicitar tránsito para una mascota previamente cargada en “encontrados”, modificando la ficha correspondiente.

Flujo Básico	<p>Este CU comienza cuando un usuario ingresa al sistema y presiona “modificar mascota”.</p> <ol style="list-style-type: none"> 1. El usuario cambia el estado a “tránsito” para solicitar tránsito. La mascota figurará en “encontrados” y en “tránsito”. 2. El usuario completará la descripción de la situación, si así lo desea. 3. El usuario presiona en opción “guardar”. 4. El sistema guarda los datos. 5. El CU termina.
Flujos Alternativos	<p>A1. Cancelar</p> <p>A1.1 El usuario decide cancelar la solicitud.</p> <p>A1.2 El CU termina.</p>
Pre- y post-condiciones	<p>[PRE]</p> <p>1-El usuario ingresa está registrado en el sistema.</p> <p>2-La mascota no está previamente cargada en “tránsito”.</p> <p>[POST]</p> <p>La modificación del estado de la mascota encontrada que ahora además está en busca de alojamiento.</p>

Nombre del CU	Cargar pedido solidario
Actor Principal	Usuario registrado
Objetivo / Descripción Breve	Este CU tiene como objetivo cargar un pedido solidario para una mascota previamente cargada en “encontrados” o en “adoptame”.
Flujo Básico	<p>Este CU comienza cuando un usuario ingresa al sistema y presiona “SerSolidario”.</p> <ol style="list-style-type: none"> 1. [INCLUDE] Mostrar mascotas. El sistema mostrará las mascotas perdidas y en adopción. 2. El usuario selecciona una. 3. El usuario completará la descripción de la situación en el campo descripción. 4. El usuario presiona en opción “guardar”. 5. El sistema guarda los datos. 6. El CU termina.

Flujos Alternativos	A1. Cancelar A1.1 El usuario decide cancelar la solicitud. A1.2 El CU termina.
Pre- y post-condiciones	[PRE] 1-El usuario ingresa está registrado en el sistema. 2-La mascota está previamente cargada en “perdidos” o “adoptame”. [POST] La modificación de la mascota encontrada o en adopción que ahora, además, tiene un pedido solidario.
Casos de usos incluidos	Mostrar mascotas

Nombre del CU	Mostrar mascotas
Actor Principal	Usuario
Objetivo / Descripción Breve	Este CU tiene como objetivo mostrar mascotas previamente cargada en “encontrados” o en “adoptame”.
Flujo Básico	Este CU comienza cuando un usuario ingresa al sistema y presiona “SerSolidario”. <ol style="list-style-type: none"> 1. El sistema mostrará las mascotas perdidas y en adopción. 2. El CU termina.
Flujos Alternativos	A1. Cancelar A1.1 El usuario decide cancelar la solicitud. A1.2 El CU termina.
Pre- y post-condiciones	[PRE] 1-La mascota está previamente cargada en “perdidos” o “adoptame”. [POST] Listado en pantalla de las mascotas encontradas o en adopción.

Metodología de trabajo

El programa se desarrolló de acuerdo al reglamento del torneo, utilizando el lenguaje de programación Java en el entorno de desarrollo libre NetBeans, con generación de interfaces en Swing y una base de datos MySQL que deberá estar alojada en un servidor en línea para favorecer su actualización inmediata a todos los usuarios.

El equipo de trabajo utilizó distintas dinámicas para organizar la tarea. Al comienzo, se realizó una lluvia de ideas (brainstorming) por Whatsapp para definir las primeras propuestas, luego se trabajó de manera remota para organizar toda esa información en un documento colaborativo. En una etapa posterior, y una vez elegido el proyecto, se hicieron reuniones en casas particulares para plasmar los primeros detalles, que se iban registrando en un documento colaborativo. Para avanzar en la idea, se propusieron charlas durante la clase, que también se iban registrando y charlas remotas por Discord o Hangouts. Los profesores, por su parte, registraban la participación de los alumnos en la planilla de “Daily Meetings” y nos mantenían informados a través de Slack. .

Inconvenientes

En relación al trabajo de desarrollo del programa, se presentaron serios inconvenientes técnicos en la conexión de la base de datos. Aún siguiendo los instructivos del curso, y con ayuda de los profesores, el 90% no logró conectar la base de datos a su computadora personal. Las causas, suponemos, se debieron a incompatibilidad con distintas configuraciones pre-existentes en los equipos, y también necesidad de actualizaciones de dependencias. Estos inconvenientes produjeron, en algunos integrantes, cierta frustración e incertidumbre al no poder aplicar lo aprendido.

En cuanto al trabajo en equipo se presentaron problemas por los distintos tiempos que manejamos cada uno de los integrantes. Hubo personas que venían de manera esporádica y otras que se vieron afectadas por problemas familiares, de salud, y/o de estudio.

A pesar de los imponderables, se intentó avanzar, con los escasos tiempos y recursos técnicos -ya que la sede no nos autorizó a trabajar fuera de horario- porque creemos que esta es la forma de aprender que más nos impulsa a crecer.

Desarrollos a futuro

Aunque consideramos que todavía estamos en plena etapa de desarrollo, apostamos por el crecimiento del programa a futuro, y nos proponemos ofrecer a los usuarios más funciones, tales como: mapas de perdidos y encontrados, chat entre usuarios, paleta de colores, y permitir realizar búsquedas comparativas, enviando un correo electrónico en caso de coincidencia a los usuarios interesados.

Conclusiones

Apuntamos a generar un amplio impacto social ayudando a esas personas que, de manera solidaria, brindan su tiempo para lograr dar con los dueños de las mascotas. La idea es que cualquier persona con interés en colaborar lo pueda hacer sencillamente brindando datos, un alojamiento temporal, o contribuyendo a solventar los gastos derivados de la manutención de mascotas perdidas o abandonadas. Creemos que la tecnología puede ser el puente para conectar a estas personas. Se podría difundir dentro de los grupos locales existentes en Facebook ya que cuanto más gente use el programa, mejor funcionará. Por todo lo expuesto, lo definimos como un software básicamente solidario, que procura conectar gente ayudando a organizar la información y la búsqueda de mascotas, tarea que, realizada de manera particular, requeriría de más esfuerzo y tiempo.

A nivel comercial, si bien se contempla la posibilidad de incluir espacios publicitarios para comercios afines que deseen auspiciar, consideramos que esto debe ser hecho con ciertos límites porque no es el fin del programa, y porque esto puede ocasionar molestias a los

usuarios. Lo primordial es que los usuarios usen el programa, y que si ven publicidad sea en la medida adecuada.

Queda mucho por hacer, somos conscientes de ello, y se nos siguen ocurriendo mejoras, hacia allá vamos...

Anexos

Aquí incluimos algunas partes del código de referencia.

Métodos para convertir imágenes:

```
public byte[] ImagenAByte(FileInputStream imagenMasco) {
    //convertir archivo de imagen a byte[]
    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    byte[] buf = new byte[1000000];
    try {
        for (int readNum; (readNum = imagenMasco.read(buf)) != -1;) {
            bos.write(buf, 0, readNum); //lee una cantidad de bytes buf y los guarda en bos
            System.out.println("read " + readNum + " bytes,");
        }
    } catch (IOException ex) {
        System.out.println("error de conversion archivo");
    }
    byte[] bytes = bos.toByteArray();//convierte a ByteArray
    return bytes;
}
```

```
public List<Image> byteToImage() {
    //genera una lista de imagenes a partir de la lista de mascotas
    System.out.println("pasó por byteToImage1");
    List<Image> listIm = new ArrayList<>();
    // listIm=null;
    Image fotoMasco;
    for (MascotaDto masco : listaMasco) {
        System.out.println("pasó por byteToImage2");
        System.out.println(masco.getNombImagen().toString());
        if (masco.getImagen() != null) {
            System.out.println("pasó por byteToImage3");
            fotoMasco = getToolkit().getImage(masco.getNombImagen()); //guarda imagen
            fotoMasco = fotoMasco.getScaledInstance(110, 110, Image.SCALE_DEFAULT); //escala la imagen a la ve
            listIm.add(fotoMasco);
            System.out.println("lista de fotos completa");
        } else {
            listIm = null;
            System.out.println("lista vacia de fotos");
        }
    }
    return listIm;
}
```

Botón para avanzar en el visor de imágenes.

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //siguiente
    if (fin + 8 <= listaFicha.size()) {
        /* Si el valor que esta en "fin" + 8 es igual o menor al largo de la lista de fichas*/
        fin = fin + 8;
        inicio = inicio + 8;
        this.refrescarGrilla();
    } else {
        if ((fin - listaFicha.size() <= 8) && (fin - listaFicha.size() > 0)) {
            /*verifico que el fin sea realmente el fin, y no pasarme ej: fin = 15 , ListaFicha = 16 */
            fin = fin + (fin - listaFicha.size());
            inicio = inicio + (fin - listaFicha.size());
            this.refrescarGrilla();
        } else {
            System.out.println("edod");
        }
    }
}
```

Método usado para mostrar las fotos en el visor.

```
private void refrescarGrilla() {  
    if (listaFicha.size() < 8) {  
        fin = listaFicha.size();  
    }  
    List<Image> listaFotos = byteToImage();  
    System.out.println("fin " + fin);  
    System.out.println("inicio" + inicio);  
    if (listaFotos != null) {  
  
        for (int i = inicio; i < fin; i++) {  
            System.out.println("esto es la I" + i);  
            switch (i) {  
                case 0:  
                    jLabel1.setIcon(new ImageIcon(listaFotos.get(i)));  
                    break;  
                case 1:  
                    jLabel2.setIcon(new ImageIcon(listaFotos.get(i)));  
                    break;  
                case 2:  
                    jLabel3.setIcon(new ImageIcon(listaFotos.get(i)));  
                    break;  
                case 3:  
                    jLabel4.setIcon(new ImageIcon(listaFotos.get(i)));  
                    break;  
                case 4:  
                    jLabel5.setIcon(new ImageIcon(listaFotos.get(i)));  
                    break;  
                case 5:  
                    jLabel6.setIcon(new ImageIcon(listaFotos.get(i)));  
                    break;  
            }  
        }  
    }  
}
```

Algunas pantallas:



Carga de fichas

Estado

Fecha

Usuario

Mascota

Descripción:

Enviar

Modificar

Cancelar

Carga de Mascotas

Nombre Id Mascota

Tipo

Color ☐ Blanco ☐ Negro ☐ Marron ☐ Gris

Sexo ☐ Macho ☐ Hembra

Ruta imagen

Foto Mascota

subir imagen

Enviar

Buscar

Modificar

Cancelar

Buscar Mascotas

Estado

Elegir

Tipo

Elegir

Color

☐ Blanco

☐ Negro

☐ Marron

☐ Gris

Sexo

☐ Macho

☐ Hembra

Descripción:

Enviar

Cancelar

Adminstración de usuario

Apellido y nombre:

apellido y nombre

contraseña:

contraseña

correo:

correo

dirección:

direccion

Telefono:

telefono

Guardar

Modificar

Cancelar