

# **Instituto Tecnológico de Costa Rica**

## **Ingeniería en Computación**

### **Programación Orientada a Objetos**

**Profesor: Adriana Álvarez**

**Proyecto: Dogs Lovers**

## **Descripción del proyecto**

Usted pertenece a un grupo de voluntarios que defienden los derechos de los animales.

El sistema debe cumplir con los siguientes requisitos:

1. Registro de mascotas encontradas o perdidas.
  - a. El registro de mascotas debe incluir al menos los siguientes datos:
    1. Tipo de mascota.
    2. Raza (con un listado de razas, incluyendo la raza única).
    3. Nombre de la mascota.
    4. Chip de identificación.
    5. Color.
    6. Foto.
    7. Teléfono de contacto (de la persona que lo encuentra o pierde).
    8. Correo de contacto (de la persona que lo encuentra o pierde).
    9. Si está extraviado o si fue encontrado.
    10. Lugar donde fue encontrado o donde se perdió.
    11. Fecha de extravío o de encontrado.
    12. Monto de recompensa (si aplica) y moneda (colones, dólares).
    13. Notas: Campo de descripción libre.
  - b. El sistema debe permitir al usuario hacer la publicación en su usuario de Facebook y Twitter de la mascota encontrada o perdida.
2. Edición: El sistema permite editar y actualizar todos los datos.
3. Los registros de las mascotas pueden tener algunos estados para identificarlos. Por ejemplo:
  - a. Encontrado
  - b. Perdido

- c. Adoptado
  - d. Otros. Identifique los posibles estados.
- 4. Consultas.
  - a. El sistema debe permitir consultar por cualquier campo de registro o combinación de campos. Por ejemplo:
    - i. Por lugar.
    - ii. Por chip.
    - iii. Por estado.
    - iv. Por raza.
    - v. Otros.
  - b. La consulta de mascotas en cualquiera de sus estados debe mostrarse ordenado por fecha (desde el más reciente hasta el más antiguo).
- 5. Casas cuna.
  - a. El sistema debe permitir registrar los datos de las personas que quieran ser casas cuna con condiciones para ser casa cuna como por ejemplo el tipo de mascota, el tamaño, si requiere donación para alimentos, entre otros.
  - b. El sistema debe permitir que una persona que reporta un perro encontrado pueda contactar a una de las personas que quieren ser casas cuna.
- 6. Adopciones.
  - a. El sistema debe permitir registrar adopciones de mascotas con los datos del adoptante.
    - i. Debe permitir registrar fotos de la persona adoptante y la mascota.
    - ii. Debe permitir registrar fotos del nuevo dueño mostrando la nueva vida de la mascota.
    - iii. El sistema debe permitir al usuario hacer la publicación en su usuario de Facebook y Twitter de las fotos de la adopción.
- 7. Calificaciones de adoptantes.
  - a. La persona que da una mascota en adopción puede poner calificaciones (estrellas del 1 al 5) y notas sobre los nuevos dueños.
- 8. Lista negra de adoptantes o personas.
  - a. El sistema debe permitir reportar a un adoptante o a una persona en una lista negra.
  - b. El sistema permite generar un reporte de lista negra de adoptantes o personas. Al presionar click sobre una persona se debe mostrar la calificación y sus notas.
- 9. Donaciones.
  - a. El sistema debe permitir donar una recompensa en caso que la persona que encuentra una mascota perdida no quiera recibirla. La donación se haría a la asociación de rescate animal que la persona indique.
  - b. Debe existir una consulta que muestre los datos de las donaciones con filtros de los datos de las personas que hacen la donación y a cuál asociación la donan así como el monto.
  - c. Debe existir una consulta que muestre el monto total de las donaciones agrupado por asociación beneficiada.

- 10.Registro de asociaciones de empresas que trabajan en pro del bienestar animal a efectos de reportarle las donaciones.
- 11.Procesos.
  - a. El sistema debe ejecutar un proceso que haga un match entre las mascotas reportadas como encontradas y las reportadas como perdidas. El match debe ser según las características de las mascotas y debe enviar correos electrónicos a las personas involucradas. En un único correo debe enviar los datos de los matchs incluyendo las fotos.
  - b. El sistema debe generar un reporte de los match a demanda.
  - c. El proceso también puede enviar mensajes de texto al celular a la persona que reporta la mascota como encontrada o perdida en caso de un match. Hint: utilice el servicio de [www.navegalo.com](http://www.navegalo.com) que es gratis aunque tiene publicidad.
  - d. Los mensajes tanto de correo como de texto deben ser configurables según el tipo de mensaje.
- 12.El sistema debe manejar un nivel de seguridad con registro de usuario y contraseña.

## **Otras consideraciones**

1. La aplicación debe ser gráfica.
2. La aplicación puede ser web (extra +1).
3. La aplicación debe tener un logo creado por el equipo. No puede ser un logo copiado de internet o de otras persona y debe ser original. Puede solicitar ayuda a un estudiante de diseño.
4. El código debe trabajarse desde un programa de control de versiones y se deberá demostrar en la defensa la cantidad de commits realizados por cada miembro del equipo.
5. Para aplicar a los puntos extras, la aplicación debe tener el resto de la funcionalidad completa.
6. El equipo puede estar conformado por grupos de un número máximo según le indique el profesor.
7. La entrega oficial se debe hacer en un zip en el Tec Digital antes de la fecha y hora ahí indicada. No se aceptarán trabajos posteriores a esa fecha y hora. En caso de problemas de conexión, se debe aplicar el plan B de la presentación de Reglas del Juego.
8. La entrega debe contener los archivos fuente y el ejecutable de la aplicación (si aplica).
9. Cualquier sospecha de copia anulará el trabajo y se procederá con el trámite administrativo.

## **Documentación**

1. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba con los cuales se probará el programa. Debe contener un id como enumerador del caso de prueba, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente). Además de una tabla pivote con la cantidad de casos de prueba en cada estado y el total.
2. Manual de usuario con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación. Si utiliza un wiki para la documentación, esta debe accederse desde la aplicación.
3. El código debe estar documentado internamente según Clean Code. Las clases o métodos importantes debe ir documentados con descripción, autor, fecha de creación.

## **Entregables por Etapas:**

### **Entregable #1 (Proyecto #2):**

4. Documento con los siguientes diagramas UML:
  - a. Diagrama de clases
  - b. Diagrama de casos de uso
  - c. Diagrama de estados
  - d. Diagrama de paquetes
  - e. Diagrama de actividad (por calles)
5. Los diagramas se entregan en UML y en .pdf o imagen .png.
6. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba con los cuales se probará el programa. Debe contener un id, la descripción del caso de prueba, el resultado esperado y el estado "Pendiente".

### **Entregable #2 (Proyecto #3):**

1. La aplicación funcional.
2. Documento de matriz de casos de prueba (en Excel) con al menos 50 casos de prueba con los cuales se probará el programa. Debe contener un id, la descripción del caso de prueba, el resultado esperado y el estado (Certificado, Con Error, Pendiente).
3. Manual de usuario en una herramienta Wiki (Wikia, Google docs) con la descripción del propósito y uso del sistema que incluya imágenes significativas de la aplicación.

4. El código debe estar documentado internamente según Clean Code. Las clases o métodos importantes debe ir documentados con descripción, autor, fecha de creación.

#### Referencias:

<http://www.perdidosyencontrados.com/>

Grupo de facebook de Zagatec:

[https://www.facebook.com/ZaguaTEC?ref=br\\_tf](https://www.facebook.com/ZaguaTEC?ref=br_tf)

Grupo de facebook de Asociación Animales de Asís:

<https://www.facebook.com/animalesdeasis>