

Sistema de Publicación y Búsqueda de Mascotas

EnMA: Página de búsqueda Encuentra tu Mascota

Integrantes

- Alberto Sayos - 101050 - asayos@fi.uba.ar
- Agustina Arellano - 112043 - aarellano@fi.uba.ar
- Franco Agustín Romano - 111315 - fromano@fi.uba.ar
- Lautaro Bianchi - 108029 - labianchi@fi.uba.ar
- Franco Alejandro Lezcano - 111252 - flezcano@fi.uba.ar
- Matias Nicolas Cardoso - 112394 - mcardoso@fi.uba.ar
- Nahuel Mendoza - 111745 - nmendozab@fi.uba.ar

Resumen

Este trabajo describe el desarrollo de una página web donde las personas pueden publicar anuncios de mascotas que han encontrado. Los dueños de las mascotas pueden utilizar un buscador para verificar si su mascota está registrada en el sitio. Se crearán dos aplicaciones: una para el frontend, que manejará toda la parte visual, y otra para el backend. Utilizaremos Flask para renderizar las plantillas y comunicarnos con la base de datos, empleando diversas librerías, incluida la librería ‘requests’ para la comunicación entre ellas, pasando datos en formato JSON.

Abstract

This work describes the development of a website where people can post found pet advertisements. Pet owners can use a search engine to check if their pet is registered on the site. Two applications will be created: one for the frontend, handling the visual part, and another for the backend. We will use Flask to render templates and communicate with the database, employing various libraries, including the ‘requests’ library for communication between them, passing data in JSON format.

Palabras clave

Mascotas perdidas, Publicación de mascotas, Búsqueda de mascotas, Flask, Backend, Frontend, JSON, Requests

Keywords

Lost pets, Pet posting, Pet search, Flask, Backend, Frontend, JSON, Requests

1 Introducción

En este documento se presenta el desarrollo de una página web destinada a la publicación de anuncios de mascotas encontradas. El objetivo principal es facilitar la reunificación de mascotas perdidas con sus dueños mediante un sistema de publicación y búsqueda eficiente. A continuación, se detallan las partes que componen este trabajo, incluyendo la solución propuesta, las pruebas realizadas y el plan de actividades.

2 Solución propuesta

La solución propuesta consiste en una página web donde los usuarios pueden publicar anuncios de mascotas que han encontrado, proporcionando detalles como la especie, raza, ubicación y una descripción del animal. El sitio web también cuenta con un buscador que permite a los dueños de mascotas perdidas verificar si su mascota ha sido registrada en el sistema.

La estructura del sitio web incluye:

- Formularios de entrada de datos para publicar anuncios.
- Bases de datos para almacenar la información de las mascotas encontradas y perdidas.
- Una interfaz de usuario amigable.
- Un mapa proporcionado por la API de Google Maps JS para visualizar las ubicaciones de las mascotas.

Las tecnologías específicas utilizadas en el desarrollo del sistema son:

- **Flask:** Utilizado para renderizar plantillas y comunicar el frontend con la base de datos.
- **JWT (JSON Web Tokens):** Utilizado para la gestión de usuarios y autenticación.

Sistema de Publicación y Búsqueda de Mascotas

- **MySQL:** Base de datos utilizada para almacenar la información.
- **Variables de entorno:** Utilizadas para manejar configuraciones sensibles y credenciales.
- **API de Google Maps:** Utilizada para mostrar las ubicaciones de las mascotas en un mapa interactivo.
- **Requests:** Librería utilizada para la comunicación entre el backend y otras APIs, pasando datos en formato JSON.

El frontend del sitio web maneja la visualización y la interacción del usuario, mientras que el backend se encarga del procesamiento de datos, almacenamiento y lógica de negocio. Las APIs, tanto las proporcionadas por Google Maps como las creadas por nosotros, son llamadas desde Flask en el frontend para garantizar una comunicación fluida y eficiente entre las diferentes partes del sistema.

3 Pruebas y/o validación

Para garantizar la calidad del sitio web, se realizaron varios tipos de pruebas:

- **Pruebas manuales:** Se ejecutaba el código y se verificaba su funcionamiento de manera directa. En algunos casos, se utilizaban resultados previos para comparar y asegurar que los nuevos cambios no afectaran negativamente el sistema. Este método de prueba, aunque informal, permitió identificar y corregir errores rápidamente durante el desarrollo.
- **Pruebas con Postman:** El backend se probaba utilizando Postman, una herramienta que permite enviar solicitudes HTTP a la API y verificar las respuestas. Esto facilitó la validación de las rutas y la lógica del servidor, asegurando que los endpoints funcionaran correctamente y devolvieran los datos esperados.

Estas pruebas, tanto manuales como a través de Postman, aseguraron que el sistema funcionara según lo esperado y que las funcionalidades implementadas cumplieran con los requisitos del proyecto.

4 Plan de actividades

El proceso de construcción del software se organizó de la siguiente manera:

- **Mapa de Conexiones:** Inicialmente, se creó un mapa que detallaba cómo se conectarían las diferentes partes de la aplicación. Por ejemplo, se especificó cómo el frontend se conectaría a Flask para obtener información del mapa. Este mapa sirvió como base para la planificación de las tareas.
- **Generación de Tareas:** En base al mapa, se generaron una serie de tareas específicas, cada una con una función definida. Estas tareas se gestionaron en Trello, utilizando la metodología Scrum.
- **Uso de Scrum:** Se implementó un tablero de Trello con cinco columnas para gestionar el trabajo:
 1. Clasificación para el planning poker
 2. Asignación de tarea
 3. Codificación
 4. Testeo
 5. Realización (tareas completadas)
- **Reuniones Semanales:** Cada semana, antes de cada entrega parcial, se realizaban dos reuniones: una reunión larga para discutir en profundidad y otra al día siguiente para organizar las siguientes tareas. En estas reuniones, se discutía sobre la entrega de valor que se iba a realizar la próxima semana.

- **Gestión del Código en GitHub:** Cada miembro del equipo tenía su propia rama en GitHub. Subían su trabajo acompañado de un número () que coincidía con la tarea en Trello. Esto facilitaba la integración y el seguimiento del progreso de cada tarea.
- **Documentación:** Se generaron documentos iniciales que incluían un mapa con las especificaciones y un documento de Word con las primeras especificaciones del proyecto. Estos documentos se utilizaron como referencia durante el desarrollo.

Esta metodología permitió una gestión eficiente del proyecto, asegurando que todas las tareas fueran completadas a tiempo y cumplieran con los requisitos establecidos.

5 Hipótesis y Supuestos

Durante el desarrollo del proyecto, se tomaron las siguientes hipótesis y supuestos:

- **Adaptabilidad a dispositivos:** La página web está principalmente adaptada para computadoras de escritorio. No se garantiza la misma experiencia de usuario en dispositivos móviles o tablets.
- **Disponibilidad de la API de Google Maps:** Aunque es raro que la API de Google Maps falle, se asume que podría haber momentos en los que la API no esté disponible. Sin embargo, se confía en la robustez de Google Cloud para mantener una comunicación constante y sin saturaciones, gracias a la utilización de una clave de API.
- **Precisión de los datos ingresados por los usuarios:** Se asume que los datos ingresados por los usuarios serán precisos y completos. Los usuarios deben seleccionar entre perro, gato u otro, especificar la raza (dependiendo de la elección), y proporcionar detalles y ubicación (calle, altura y zona) según su criterio.