

Trabajo Práctico

Introducción al Desarrollo de Software

Informe

Nombre del grupo: Codekies

Integrantes:

- Nicolás Escobar.
- Rolando Mamani.
- Inti Nahuel Quispe.
- Diego Nadares.
- Franco Arias.
- Sebastian Aznarez.
- Malena Mena.

Tutor:

- Franco Daniel Capra.

Fechas de entrega:

- Jueves 28/11/2024.
- Martes 3/12/2024.

Resumen:

Nuestro proyecto es una página web creada principalmente con el framework de Python "Flask". La página permite a los usuarios de Argentina publicar sus mascotas perdidas, o mascotas que hayan encontrado sin dueño, como también ver las mascotas publicadas por otras personas. La misma cuenta con una aplicación para celulares Android, con la cual el usuario puede publicar una mascota perdida/encontrada.

Introducción:

Se nos pidió crear un sitio web que permita publicar información sobre mascotas perdidas y también tener un buscador de mascotas por características, raza, color, condición, ubicación. El usuario debe ser capaz de agregar información que se pueda considerar relevante en la descripción de la mascota.

Solución propuesta:

Proponemos una página web enlazada a una base de datos, en la cual se cargan las mascotas publicadas por los usuarios, la información de los usuarios (nombre de usuario, contraseña, dirección de correo electrónico, etc) y la referencia del animal con el usuario que lo publicó asignando identificadores a ambos para poder acceder a la información luego. Creamos varios templates con distintas funcionalidades, uno de ellos, el de login, permite al usuario iniciar sesión en caso de ya haber registrado una cuenta, y también permite registrar una si no lo había hecho previamente. Otro template es el formulario para añadir una mascota, para acceder a él se deberá estar logueado y deberán completar ciertos campos para poder mandar información de una mascota perdida correctamente. En otro se muestran todas las mascotas que están cargadas en la base de datos, con una pequeña descripción de ellas, con un botón a otra página en la que muestre más detalles de la misma y otro botón para reclamarlas. Cuando alguien reclama una mascota, le llega un mail al correo electrónico del usuario que realizó la publicación, con los datos del usuario que la reclamó.

La página permite también ver todas las mascotas cargadas y filtrar por zona, tipo de mascota, estado (perdido o encontrado), raza, color, descripción y fecha en la que se encontró o perdió la mascota.

A su vez, creamos una aplicación móvil para android que permite iniciar sesión y cargar una mascota al sitio, agregando los datos correspondientes. La misma fue desarrollada utilizando el framework Kivy de Python. Este framework utiliza programación orientada a objetos. La aplicación cuenta con 4 vistas, cada una representada por una clase, las cuales son:

- Login: permite iniciar sesión al usuario y acceder al formulario para publicar una mascota. Si el usuario o contraseña no están registrados en la base de datos, se muestra un aviso en pantalla.

- Main: muestra el formulario y al presionar el botón “Añadir” envía la solicitud a la API, y carga la mascota añadida a la base de datos. Si falta completar alguno de los campos obligatorios, se muestra un aviso en pantalla.
- Importar Imagen: se muestra cuando el usuario presiona el botón para subir la imagen de la mascota. Esta vista permite elegir un archivo del dispositivo del usuario y guardar la ruta del archivo para subirla junto con el formulario a la base de datos.
- Mostrar Mapa: se muestra cuando el usuario presiona el botón para añadir ubicación. Esta vista muestra un mapa y una barra para buscar la ubicación deseada, la cual tiene la función de completarse según lo que ingrese el usuario, y muestra un marcador en la ubicación seleccionada por el usuario. Una vez aceptada la ubicación, se guardan los datos de la dirección para ser enviados a la base de datos junto con el formulario.

Las vistas son desplegadas a través de la clase MyApp, la cual muestra las vistas cargadas en una lista, con la funcionalidad WindowManager del framework.

Tecnología utilizada:

Utilizamos editores de código como Visual Studio Code, Trello para administrar el backlog, GitHub para conectarnos entre nosotros y poder realizar commits al proyecto, Whimsical para realizar el mockup, lenguajes de programación como Python y Javascript, lenguaje de marcado HTML, lenguaje de hojas de estilo CSS, los frameworks Flask y Kivy de Python, Postman, ambientes virtuales, maquinas virtuales, Freehtml5 para utilizar un template prearmado (<https://freehtml5.co/neat-free-html5-bootstrap-website-template-for-portfolio/>), Geoapify, Open Street Map, Leaflet (apis que nos permitieron cargar un mapa en la pagina web para pedirle al usuario que corrobore que la direccion era la correcta), instalamos la dependencia requests, una api para enviar correo con la informacion de un usuario (sendgrid), persistimos los datos con el motor MYSQL, y ademas usamos docker para correr algunas aplicaciones como por ejemplo MYSQL.

Conclusión final:

Con este trabajo práctico aprendimos nuevas herramientas tanto para desarrollar proyectos web como para trabajar en equipo. Desarrollamos mucho la comunicación y logramos repartirnos las tareas de manera equitativa. Utilizamos métodos de agilidad, como Scrum, para la resolución del proyecto. Esto implicaba reuniones diarias, cortas para ver como vamos cada uno. También nos preparábamos para un sprint asignando tareas a los miembros del equipo, hacíamos un planning poker para asignarle una dificultad a cada una de las tareas del backlog y la implementación de Trello con un tablero kanban que facilitó la resolución y asignación de tareas.