

CC5002 – Desarrollo de Aplicaciones WEB

TAREA 3

Profesor: José Urzúa

Fecha de entrega: viernes 24 de octubre del 2025

Descripción

El desarrollo de su Tarea 2 fue muy bien recibido y ahora es necesario completar las funcionalidades relacionadas a las estadísticas descritas en la Tarea 1. Para esto, debe complementar su proyecto hecho en Flask en la Tarea 2 para incluir la funcionalidad:

- Estadísticas: Debe desplegar los 3 gráficos mencionados anteriormente:
 - (1 punto) El primero es un gráfico de líneas que informa la cantidad de avisos de adopción agregados por día. En el eje X muestra los días y en el eje Y muestra la cantidad de avisos.
 - (1 punto) El segundo gráfico es un gráfico de torta que muestra el total de avisos de adopción por tipo de mascota: perro o gato.
 - (1 punto) El tercer gráfico es uno de barras que muestra dos barras por cada punto del eje X. El eje X son los meses y para cada mes muestra una barra con la cantidad de avisos de adopción de gatos y otra barra con los que corresponden a avisos de adopción de perros. El eje Y indica la cantidad.

Al final de esta pantalla debe agregar un enlace que permite volver a la portada del sistema. Debe usar Javascript (XHR o fetch) para generar los gráficos en el lado del cliente, llamando a una URL del lado del servidor que obtiene la información desde la base de datos.

Además, en esta tarea debe agregar una nueva funcionalidad que permite ingresar y visualizar comentarios a los avisos de adopción que se han ingresado al sistema. Para esto deberá implementar:

- Agregar comentario (2 puntos): Cuando se visualiza la información de un aviso de adopción, se debe incluir un formulario para agregar un nuevo comentario. Este formulario debe solicitar nombre (caja de texto, obligatorio con largo mínimo 3 y máximo 80) y texto del comentario (Área de texto, 4 filas y 50 columnas, dato obligatorio con al menos 5 caracteres). Junto con esta información debe agregar un botón con el texto “Agregar comentario”, al hacer clic debe validar que la información del nuevo comentario cumpla con las reglas, si las cumple, se debe hacer submit a una URL que es atendida por Flask. En esta URL se debe validar los datos del comentario en el lado del servidor y además debe insertar en la base de datos en la tabla comentario. Si no se cumplen las validaciones debe informar al usuario manteniendo visible el formulario para que lo pueda corregir.
- Listado de comentarios asociados a un aviso de adopción (1 punto): Cuando se consulta la información de un aviso de adopción, debe obtener los comentarios

asociados a dicho aviso desde la base de datos y los debe incluir como un listado con la fecha, nombre del comentarista y texto del comentario.

Las funcionalidades para agregar comentario y listado de comentarios las debe implementar usando llamadas asíncronas desde el lado del cliente. Esto lo puede hacer directamente usando Javascript con el objeto XHR o usando la función fetch o alguna biblioteca Javascript que usted decida.

Consideraciones

- Si necesita algún apoyo para la interacción con la base de datos no dude en comunicarse con el cuerpo docente por medio del foro de u-cursos. Use la misma base de datos de la Tarea 2 pero ahora debe agregar la tabla “comentario” usando el script adjunto al enunciado `tabla-comentario.sql`
- Debe usar HTML5 y CSS3. No se complique con el diseño gráfico. Puede usar plantillas e imágenes para las cuales tenga derecho de uso.
- Implemente su aplicación usando Python con Framework Flask y considere entradas de texto “maliciosas” en los formularios y parámetros en las URL.
- Use XHR o llamadas con fetch para implementar los gráficos en el lado del cliente. Puede usar bibliotecas externas. En el lado del servidor debe programar en Python y el framework Flask.
- Para la generación de gráficos puede usar las bibliotecas vistas en clases: Flot <https://www.flotcharts.org/> o Highcharts <https://www.highcharts.com/blog/products/highcharts/> u otras para la cual tenga derechos de uso y la mencione en el README.
- En la evaluación se utilizará el validador de HTML (<http://validator.w3.org/>) y CSS (<http://jigsaw.w3.org/css-validator/>) de W3C. Se descontarán 0.1 puntos por cada error de HTML5 o CSS mostrado por los validadores.
- Pruebe que sus archivos funcionen bien en distintos browser y en distintas resoluciones de pantalla.
- Deben entregar su tarea mediante un repositorio de [GitHub](#). Para ello, deben crear un usuario en GitHub (si aún no tienen uno) y configurar un repositorio **público** con el siguiente esquema de nombre: **desarrollo_web_nombre_apellido**.
- En el repositorio, deberán crear una rama llamada **"Tarea 3"**, donde subirán todos los archivos correspondientes a la entrega. Además, dentro de esta rama, el repositorio debe incluir un archivo **README.md**, en el cual explique algún detalle o decisión que usted tomó y que se deba tener en cuenta para la corrección.
- Para hacer entrega de su tarea por u-cursos, deberán escribir en los comentarios de la tarea el URL de su repositorio, y además **DEBE ENVIAR** los archivos relevantes de la tarea en formato .zip a modo de respaldo en caso de problemas con su repositorio.