

# CC5002 - Tarea 2

Profesor: José Urzúa

Fecha de Entrega: viernes 14 de mayo

## Descripción

¡Felicitaciones! El prototipo desarrollado en su tarea 1 generó mucho interés y ahora tiene todo el apoyo para comenzar con la implementación de la aplicación web. En esta tarea, usted deberá usar el lenguaje de programación Python junto con una base de datos MySQL para implementar las funcionalidades definidas en la tarea 1.

1. **Portada:** Muestra un mensaje de bienvenida al sistema, con un menú que contiene las siguientes opciones: Informar avistamiento, Ver listado de avistamientos y Estadísticas. Además, incluye un listado con información de los últimos 5 avistamientos informados de acuerdo al formato definido en la Tarea 1, pero ahora debe obtener la información desde la base de datos.
2. **Informar avistamiento:** Debe solicitar los mismos datos definidos en la tarea 1, pero debe asegurarse que en el formulario los datos “sector”, “nombre”, “email”, “número de celular”, “día y hora” sean elementos html **input de tipo de texto**. El formulario debe mantener el mismo comportamiento definido en la tarea 1, el cual permite agregar más de 1 archivo por avistamiento y más de un avistamiento en el mismo sector. Cuando el usuario hace clic en el botón “Enviar información de avistamiento” y luego de confirmar en el mensaje “¿Está seguro que desea enviar esta información?” se debe hacer submit a un CGI que **valida los datos en el lado del servidor** y debe insertar los registros en la base de datos en las tablas avistamiento y detalle\_avistamiento. En el caso de las fotos, debe insertar en la tabla foto la información del archivo y el contenido del archivo lo debe dejar en un directorio en el servidor, considerando que posteriormente los deberá usar para mostrar los listados de información e información detallada de un avistamiento. Si todo salió bien, debe volver a la portada informando un mensaje de éxito, en caso contrario debe mostrar los errores y mantenerse en el formulario de informar avistamiento.
3. **Listado de Avistamientos:** Debe obtener la información desde la base de datos y desplegarla de la misma forma indicada en la Tarea 1 considerando como máximo 5 filas de información y en caso de haber más de 5 filas en la base de datos se debe mostrar la información por páginas, permitiendo avanzar y retroceder según corresponda. Al hacer click sobre alguna fila del listado debe obtener desde la base de datos toda la información correspondiente al avistamiento y desplegarla como se indicó

en la tarea 1, considerando al final un botón o enlace que permite volver al listado de avistamientos y otro que permite volver a la portada.

4. **Estadísticas:** Esta funcionalidad no será implementada en esta tarea y la debe mantener tal cual como la desarrolló en la tarea 1.

## Consideraciones

- Debe usar HTML5 y CSS3. No se complique con el diseño gráfico.
- Implemente sus CGI en Python versión 3.8 y considere entradas de texto e información maliciosa en formularios.
- Debe crear la estructura de la base de datos con el archivo adjunto "tarea2.sql", luego carga la información de las comunas con `region-comuna.sql`. Para para la interacción con la base de datos puede usar las sentencias SQL de "sentencias.sql" que incluye ejemplos. Si necesita otras sentencias SQL no dude en pedir las por medio del foro de u-cursos al cuerpo docente.
- Para el acceso a la base de datos utilice las siguientes credenciales: host: localhost, puerto: 3306, nombre base datos: tarea2, username: cc5002 y password: programacionweb.
- Puede usar plantillas para las cuales tenga derecho de uso.
- Respete los nombres de campos html, de otro modo se considerará el elemento como no presente y descontará puntaje.
- Utilice imágenes para las cuales tenga derechos de uso.
- 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.
- Cada una de las validaciones deben ser hechas en JavaScript. El uso del atributo `required` de los formularios, NO cuenta como validación.
- Recuerde que DEBEN subir sus tareas al servidor anakena, en caso de no hacerlo, serán evaluados con la nota mínima. De igual forma deben hacer entrega de la tarea por medio de u-cursos esto también es OBLIGATORIO.