

## CC5002 – Desarrollo de Aplicaciones WEB

### TAREA 3

Profesor: José Urzúa

*Fecha de entrega: viernes 31 de mayo del 2024*

### Descripción

El desarrollo de su tarea 2 fue muy bien recibido y ahora es necesario completar las funcionalidades relacionadas a los pedidos y de paso agregar una funcionalidad enfocada en mejorar la visualización de datos del sistema. En primer lugar se deben desarrollar las funcionalidades asociadas a los pedidos:

- Agregar pedido (2 puntos): debe mostrar el mismo formulario desarrollado en la Tarea 1, mantener todas las validaciones realizadas con Javascript, pero al momento de presionar el botón “Agregar Pedido” y presionar el botón de confirmación, se debe hacer submit a una URL que es atendida por Flask y que debe validar los datos en el lado del servidor e insertar un registro en la base de datos en las tablas “pedido” y “pedido\_fruta\_verdura”. Si todo resulta bien, debe volver a la página de inicio informando un mensaje apropiado. En caso de errores de validación, debe mantener visible el formulario con los mensajes de error correspondientes.
- Ver Pedidos (1 punto): debe obtener el listado de pedidos registrados desde la base de datos y los debe desplegar tal como se indicó en la tarea 1. Se deben mostrar grupos de 5 filas, si hay más de 5 pedidos registrados en la base de datos, los debe mostrar por página permitiendo avanzar y retroceder según corresponda.
- Información Pedido (1 punto): Si el usuario hace click sobre una fila del listado de pedidos, se debe mostrar la información del pedido tal como se pidió en la tarea 1, pero ahora obteniendo desde la base de datos la información del pedido correspondiente.

La nueva funcionalidad corresponde a una nueva página que contiene 2 gráficos, uno para los productos (1 punto) y otro para los pedidos (1 punto). Para los productos debe mostrar el total de productos por tipo de fruta o verdura. Para el caso de los pedidos, debe mostrar el total por comuna. Los gráficos pueden ser de barras o circular. Debe usar Javascript (AJAX) 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.

### Consideraciones

- Si necesita sentencias SQL no dude en pedir las al cuerpo docente por medio del foro de u-cursos.
- 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 AJAX 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/>.
- 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 hacer entrega de la tarea por medio de u-cursos, incluyendo un archivo README en el cual explique algún detalle necesario para la evaluación.