

INF-239 Bases de Datos

Tarea 2: PHP + MySQL

Profesores: Ricardo Salas - Rodrigo Olavarria
Ayudante de Cátedra: Beatriz Vásquez
Ayudantes de Tareas: Débora Alayo- Daniela Sánchez - Isidora Ubilla

7 de octubre de 2023

1. Objetivo

Investigar y aplicar conceptos básicos de creación de aplicaciones web con bases de datos integradas, utilizando herramientas como PHP y MySQL.

2. Especificaciones y reglas

El desarrollo de esta tarea debe cumplir las siguientes especificaciones, de lo contrario existirá un descuento en la nota final:

- Se debe desarrollar una aplicación web básica en parejas, teniendo libertad en elegir como implementar lo que se les solicita, utilizando PHP como herramienta de desarrollo y MySQL para gestionar la base de datos.
- La tarea debe ser entregada el **10 de noviembre hasta las 23:55 horas** vía aula, existiendo una tolerancia de 10 minutos de atraso, posterior a esto no se podrán realizar más entregas.
- Se dará un bonus de 10 puntos a quienes desarrollen frontend (diseño de la página). Ejemplo: Bootstrap. No habrá descuento para quienes no desarrollen frontend, pero la información que se debe mostrar en la página debe verse de manera clara y no desordenada. En caso de desorden sí habrá descuento. **Los puntos solo se darán a alumnos cuya nota sea menor a 100, siendo el tope nota 100. Los puntos no son acumulables ni transferibles a otras evaluaciones, sólo validos en esta tarea.**

2.0.1. Sobre la entrega

La tarea debe ser entregada como un archivo comprimido de la forma T2-ROL1-ROL2.zip y debe contener los siguientes archivos:

1. Una carpeta **PHP**: Todos los archivos **.php** deben estar contenidos aquí. Puede crear los archivos que sean necesarios para resolver su tarea.
2. Una carpeta **BD**: Esta carpeta contiene la base de datos que se debe exportar en formato **.sql**, el nombre de este archivo debe ser **Prestigetravels**.

3. Si necesita tener más archivos, por ejemplo, imágenes que adornen el sitio o archivos .js o .css, estos deben estar ordenados dentro de carpetas que contengan archivos del mismo estilo.
4. **La tarea debe traer datos de prueba creados en la base de datos:** Deben ser datos suficientes para probar todas las características solicitadas en la tarea, por ejemplo, si va a mostrar un top 10, no coloque solo 10 datos, en cambio, coloque al menos un par mas para ver que efectivamente se realiza una selección entre los datos disponibles.
Si no existen datos almacenados se realizará un descuento de 30 puntos en su nota final. Este descuento será gradualmente menor si hay datos de prueba pero insuficientes para probar todas las funcionalidades pedidas.
5. Se debe incluir un archivo **README**, puede ser .txt o .md, el cual debe contener nombres, rol de alumnos y las instrucciones para la correcta ejecución de su programa. Ante cualquier duda sobre algo que no aparezca especificado en la tarea **y no afecte las reglas de esta**, puede asumir lo que estime conveniente, pero debe ser especificado en el README, **en caso de asumir eventos y no especificar habrá descuento.**
6. Se responderán dudas por aula a las preguntas hechas hasta 2 días previos a la entrega es decir hasta el miercoles 8 de noviembre a las 23:55.
7. Esta tarea debe ser defendida para obtener la nota, para esto se habilitará un documento donde podrán inscribir cuándo realizar su defensa según los horarios disponibles del ayudante que les revisará. **Quienes inscriban un horario y no asistan tendrán 0 como nota de esta entrega.**
8. Es responsabilidad del equipo estar a la hora seleccionada, no deben esperar a que el ayudante lo contacte por Discord.
9. Para las consultas prefiera el uso del foro en aula por sobre el correo electrónico, su duda puede ser útil para alguien mas.

2.1. Reglas

- **Debe** realizarse en parejas, no se aceptarán tareas individuales.
- Las copias serán evaluadas con nota 0 y se informará a las respectivas autoridades.
- Consultas vía la plataforma oficial AULA.
- En el foro de consultas podrán buscar pareja quienes no tengan, esto es exclusiva responsabilidad del estudiante.
- En caso de problemas con su pareja podrán contactar al profesor explicando su situación.
- Solo un alumno debe realizar la entrega.
- En caso de que falle la ejecución de algún comando, no se asignará puntaje a éste.
- Existe la posibilidad de que a su defensa, asista su profesor y realice preguntas.
- En la Defensa se seleccionarán consultas SQL al azar, el equipo debe explicar que hace la consulta y explicar la razón de realizarlo de esa forma.
- Es responsabilidad del estudiante inscribir un horario de defensa y estar presente en la fecha y hora elegida.

- Cada grupo tendrá un horario definido para su defensa, en caso de atraso contarán con un tiempo menor para presentar su trabajo.
- La información respecto a la defensa será eventualmente publicada en Aula, esto considera el detalle sobre los descuentos. Es su obligación estar atento a esta información y cumplir con lo establecido allí.

3. Descripción del problema

El casino de una prestigiosa universidad ultimamente ha notado son cada vez más los alumnos que solicitan conocer las recetas de sus platillos, en especial del budin de lechuga asada que se sirvió hace 3 meses. El equipo del casino, orgulloso del buen recibimiento de sus recetas exóticas y comprometido con el aseguramiento de la calidad ha decidido iniciar un proyecto llamado "Sabor USM".

El proyecto consiste en una página web donde la comunidad universitaria puede ver la minuta del casino, ver y calificar recetas, consultar la cantidad de almuerzos disponibles, entre otras funciones.

Los usuarios cuentan con un nombre, un correo, una contraseña, número de almuerzos disponibles y fecha y hora del último inicio de sesión.

Las recetas cuentan con una foto ilustrativa, nombre, tipo de platillo (entrada, postre o plato de fondo), listado de ingredientes junto con la cantidad de cada uno, tiempo de preparación, instrucciones, promedio de calificaciones, una sección de reseñas y además indican si el platillo es: apto para diabeticos, apto para intolerantes a la lactosa, si tiene gluten y si es apto para veganos.

OBSERVACIÓN 1: Un almuerzo es considerado como una entrada, un plato de fondo y un postre. Nótese que la votación es solo para decidir el plato de fondo, nada mas. También, los miembros de la comunidad universitaria compran N almuerzos, los cuales disminuyen en 1 por cada consumo.

Para fines de mejoramiento del servicio del casino, la página debe ser capaz de mostrar el top 10 mejores recetas y el top 10 peores recetas, ambos tops según un parámetro que debe ser ingresado en la página, el cual puede ser:

- Tipo de platillo: (entrada, postre o plato de fondo)
- Ingrediente: por ejemplo, los tops de las recetas que contengan arroz
- Etiquetas: (apto para diabéticos, apto para intolerantes a la lactosa, si tiene gluten y si es apto para veganos)
- Promedio de calificaciones del sitio

OBSERVACIÓN 2: El parámetro es único y no se le solicitará combinar criterios para generar los tops, por ejemplo, no se le solicitarán tops de recetas con arroz y veganas.

También, debido al extenso historial de innovación de la universidad, se ha decidido incorporar una sección en la página donde los usuarios pueden votar semanalmente por el plato de fondo que se servirá el viernes.

Utilizando su conocimiento en BD, el equipo del casino le solicita crear una versión de "Sabor Usm" que cumpla con las exigencias de la universidad.

Para ayudarles a organizar todo el trabajo por hacer, se le deja la siguiente To-Do list:

■ **General**

- ☐ Debe diseñar su base de datos (modelo relacional), para la cual deberá utilizar los conceptos aprendidos en clases. Dispone de la herramienta diseñador en phpmyadmin (mysql) (no esta permitido utilizar otra herramienta) o puede generarla con sentencias. No olvide las claves primarias, foráneas y eliminación en cascada, entre los otros conceptos aprendidos para modelo relacional. Este modelo debe revisarlo e ir iterando sus mejoras conforme lo considere necesario para cumplir con lo expuesto en la tarea.
- ☐ Debido a múltiples quejas de alumnos con respecto a la velocidad de otros sitios de la universidad, para el cruce de tablas es **OBLIGATORIO** utilizar joins, el uso de subconsultas es poco eficiente.
- ☐ De los siguientes componentes: procedimiento almacenado, trigger, view y function se debe incluir al menos uno de cada uno, deben ser funcionales y útiles de acuerdo al contexto de la tarea.
- ☐ El sistema debe tener navegación entre secciones (recargar la página o retroceder con el navegador no es válido).
- ☐ Implementar sistema de usuarios, por motivos de privacidad, se debe iniciar sesión para ver el resto de la página.
- ☐ El sistema debe mostrar el nombre de usuario que tiene la sesión activa en todo momento y poder llevar a una página con su información.
- ☐ Cada usuario puede visualizar las recetas de la página, calificar y/o escribir una reseña.
- ☐ Cada usuario tiene una lista de recetas favoritas donde puede añadir cuantas desee.
- ☐ Se debe ingresar la información necesaria sobre las recetas (puede ser ficticia).

■ **Página principal**

- ☐ Se debe implementar una barra de búsqueda, esta debe permitir encontrar recetas por nombre, ingredientes y/o por filtros. Estos filtros corresponden a tipo de platillo, si es apto para diabeticos, celíacos y/o intolerantes a la lactosa.
- ☐ En otra sección, se deben mostrar las 10 mejores recetas y las 10 peores recetas considerando las calificaciones de los usuarios del sitio.
- ☐ Una sección que muestre la información de la sesión de usuario actual y permita acceder a la lista de favoritos.
- ☐ Una sección que dirija al apartado de votación semanal

■ **Recetas favoritas**

- ☐ Se deben poder visualizar los elementos que han sido agregados a favoritos, indicando nombre, imagen y calificación promedio, cada elemento debe poder llevar a la página de la receta.
- ☐ **CRD favoritos:** Crear lista de favoritos, agregar y eliminar elementos de la lista.

■ **Perfil**

- ☐ **CRUD del usuario:** Crear cuenta, eliminar cuenta, editar información del perfil y visualizar la información del usuario. .

- ☐ Se debe mostrar la información básica del perfil, esto es, nombre de usuario, correo, almuerzos disponibles y fecha y hora del último inicio de sesión. Además, se debe poder acceder desde el perfil a favoritos.
- ☐ Se deben mostrar todas las calificaciones y reseñas hechas por el usuario hasta la actualidad en orden cronológico.

■ Recetas

- ☐ **CRUD de la calificación** (Eliminar, editar, añadir y visualizar). Solo el usuario creador de la calificación puede editar y eliminar aquellas que realizó.
- ☐ **CRUD de la reseña** (Eliminar, editar, añadir y visualizar). Solo el usuario creador puede editar y eliminar.
- ☐ Se debe mostrar la imagen relacionada a la receta.
- ☐ Se debe mostrar el nombre, ingredientes, tiempo de preparación y lo indicado en la descripción del problema.
- ☐ Para agregar la receta a favoritos, se debe implementar un botón que lo permita.

■ Votación

- ☐ **CRD del voto** (Eliminar, añadir y visualizar) la elección del usuario así como también debe visualizarse la cantidad de votaciones para cada opción.
- ☐ La cantidad de opciones a elegir por cada votación es 3, estas se seleccionan al azar cada semana.
- ☐ Cada usuario puede votar una única vez por una sola opción y puede modificar su elección.
- ☐ Cada viernes a las 15:00 las opciones cambian para la siguiente semana. Considere el efecto cascada para reiniciar la votación.

4. Aclaraciones

1. Las recetas pueden ser añadidas a la base de datos directamente, no es necesario crear un usuario especial con privilegios de administrador, pero puede hacerlo si le acomoda más.
2. Considere el efecto cascada, es decir, a las acciones que se deben propagar automáticamente a otras tablas de la base de datos cuando se realiza una acción en una tabla específica, como por ejemplo, eliminar una cuenta en la plataforma debe eliminar calificaciones y voto (si existe).
3. Está permitido el uso de la librería time, schedule y datetime. Hint: Se sugiere recordar operaciones sobre strings para modelar los ingredientes.
4. Un usuario puede calificar una receta, pudiendo o no dejar una reseña. No puede dejar un reseña sin haber calificado antes.
5. Las calificaciones son de 1 a 5, siendo 5 muy buena y 1 muy mala.
6. Se recomienda revisar los laboratorios disponibles en aula, pueden ser útiles para comenzar.