

INF-239 Bases de Datos

Tarea 2: PHP + MySQL

Profesores: Ricardo Salas (ricardo.salas@usm.cl)
Mauricio Figueroa (mauricio.figueroac@usm.cl)
Ayudante Coordinador: Gonzalo Alarcón(gonzalo.alarconc@usm.cl)

Septiembre, 2025

1. Objetivo

Investigar y aplicar conceptos básicos de creación de aplicaciones web con bases de datos integradas, utilizando herramientas como PHP y MySQL. Además de la utilización de la normalización 3FN

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, utilizando PHP como herramienta de desarrollo y MySQL para gestionar la base de datos.
- La tarea debe ser entregada el **17 de Octubre hasta las 23:59 horas** vía aula.
- Se dará un bonus de 5 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.**
- Se dará un bonus de 5 puntos a quienes implementen en el sistema protección a inyecciones SQL. **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 **PDF**: Un archivo .pdf el cual contenga los pasos para la Normalización 3FN

2. Una carpeta **PHP**: Todos los archivos **.php** deben estar contenidos aquí. Puede crear los archivos que sean necesarios para resolver su tarea.
3. Una carpeta **BD**: Esta carpeta contiene la base de datos con el contenido del archivo que se le brindará para realizar la tarea.
4. 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.
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. 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**.
7. Es responsabilidad del equipo estar a la hora seleccionada, **no** deben esperar a que el ayudante lo contacte por Discord.
8. **Las consultas se deberán hacer mediante el foro de Aula el cual se encontrara en la sección Tareas, hasta 48 horas antes del plazo de entrega original**

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 Búsqueda pareja 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.
- 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.
- Si uno de los integrantes falta a la defensa tiene 3 días para contactarse con el ayudante, en caso contrario tendrá nota 0.

3. Descripción del problema

La empresa ZeroPressure luego de su prestigiosa implementación de la base de datos del sistema para su software de incidencias/funcionalidades ha decidido continuar trabajando con usted pero ahora implementando el sistema con el cual podrán realizar ciertas actividades. Como estudiantes de la UTFSM tienen grandes expectativas en su trabajo

El objetivo es realizar una página web con el lenguaje de programación php y como gestor de bases de datos MySQL.

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

■ Normalización

- Deben terminar/corregir el modelo creado a partir de la Tarea 1
- Deben convertir modelo a 3FN mostrando todos los pasos
- Los pasos pueden ser desarrollados a mano o digital, pero debe ser ordenado.
- Se debe basar en las vistas de usuario entregadas en los casos de uso (si su modelo ya cuenta con las vistas de usuario, utilizar el propio)

■ General

- ☐ Debe utilizar la base de datos Normalizada. Dispone de la herramienta diseñador en phpmyadmin (mysql) (no está permitido utilizar otra herramienta) o puede generarla con sentencias (Base de datos y tablas). No olvide las claves primarias, foráneas y eliminación en cascada, entre los otros conceptos aprendidos para modelo relacional.
- ☐ Se debe hacer uso de al menos un **Procedimiento almacenado** el cual deba ser funcional y acorde al contexto de la tarea
- ☐ Se debe hacer uso de al menos un **trigger**, este debe ser funcional y acorde al contexto de la tarea
- ☐ El sistema debe tener navegación entre secciones (recargar la página o retroceder con el navegador no es válido).
- ☐ Se debe hacer uso de al menos una **View**, este debe ser funcional y acorde al contexto de la tarea
- ☐ Se debe hacer uso de al menos una **Función SQL**, este debe ser funcional y acorde al contexto de la tarea

■ Login

- ☐ Crear un login para entrar a la plataforma, que involucre el registro de un usuario nuevo en la plataforma, y el eventual ingreso con su usuario y contraseña creada. Especificar validaciones para usuario y contraseña que levanten

■ CREATE, READ de Usuarios e Ingenieros

- ☐ En la sección login se deberán poder registrar usuarios (Usuario e Ingenieros), tener en cuenta que al momento de registrar cada tipo de usuario se debe realizar la asignación de Rol

■ **Página principal**

- ☐ Se debe implementar una barra de búsqueda que permita encontrar funcionalidades o errores por un trozo de nombre (Ambos roles pueden usar esta función).
- ☐ Una sección que permita visualizar todas las solicitudes de funcionalidad del sistemas. Es decir, que se muestre el título, ambiente, resumen, tópico, criterios de aceptación, solicitante y estado (Solo el Ingeniero puede acceder a esta sección).
- ☐ Una sección que permita visualizar todas las solicitudes de gestión de errores del sistemas. Es decir, que se muestre el título, descripción, fecha de publicación, tópico, autor de creación de la incidencia, estado. (Solo el Ingeniero puede acceder a esta sección).
- ☐ Una sección que permita visualizar las solicitudes de funcionalidad que incluyan al usuario como autor. (Solo Usuarios pueden revisar esta sección)
- ☐ Una sección que permita visualizar las solicitudes de gestión de error que incluyan al usuario como autor. (Solo Usuarios pueden revisar esta sección)
- ☐ Una sección que permita visualizar todas las solicitudes de funcionalidad y gestión de errores que estén asignadas al ingeniero. (Solo Ingenieros pueden revisar esta sección)
- ☐ Una sección que permita crear solicitudes de funcionalidad y gestión de errores con sus respectivos campos

■ **Búsqueda avanzada**

- ☐ Se debe implementar la opción de ejecutar distintos filtros para filtrar la búsqueda. Estos son: Fecha de envío, Tópico, Ambiente, Estado. En los resultados de búsqueda deberá aparecer el nombre de la solicitud, su resumen/descripción y su tópico.

■ **CREATE, READ, UPDATE, DELETE Solicitudes de Funcionalidad**

- ☐ Se solicita el desarrollo de un sistema CRUD (CREATE, READ, UPDATE, DELETE) de las solicitudes de funcionalidad, con los campos mencionados en la Tarea 1. Los usuarios deberán poder registrar, leer, actualizar y eliminar las solicitudes de funcionalidad, generados por ellos mismos, pero con la restricción de que los artículos " En Progreso " no pueden ser modificados ni eliminados. Los ingenieros, por su parte, tendrán acceso a las solicitudes en revisión asignados a ellos y podrán añadir reseñas, que incluirán observaciones, las cuales pueden ser editadas o eliminadas.

■ **CREATE, READ, UPDATE, DELETE Solicitudes de Gestión de Errores**

- ☐ Se solicita el desarrollo de un sistema CRUD (CREATE, READ, UPDATE, DELETE) de las solicitudes de gestión de errores, con los campos mencionados en la Tarea 1. Los usuarios deberán poder registrar, leer, actualizar y eliminar las solicitudes de gestión de error, generados por ellos mismos, pero con la restricción de que los artículos " En Progreso " no pueden ser modificados ni eliminados. Los ingenieros, por su parte, tendrán acceso a las solicitudes de gestión de error asignados a ellos y podrán añadir reseñas, que incluirán observaciones, las cuales pueden ser editadas o eliminadas.

■ **Asignación automática de solicitudes**

- ☐ Se solicita implementar la asignación automática de solicitudes mediante tópicos, de la misma forma que se menciona en la tarea 1.

4. Aclaraciones

1. Es obligatorio utilizar claves primarias y foráneas, existirá descuento en la pauta de no usarse.
2. Cualquier supuesto debe ser especificado de forma explícita en el README, habrá un descuento por cada supuesto no especificado.
3. Función SQL, trigger, view y procedimiento almacenado deben ser incluidos en algunos de los puntos solicitados en la tarea, de aplicarse en algo no solicitado anteriormente tendrán únicamente la mitad del puntaje.
4. Obligatoriamente deben existir suficientes datos de prueba para probar cada inciso de la tarea. Por ejemplo, si hipotéticamente se les solicitase un top 10 no coloque 10 datos en su BD, en cambio, coloque unos 12 para que así el ayudante compruebe que efectivamente se realiza una selección correctamente.
5. **Se recomienda iniciar con tiempo esta tarea.**