

# INF-239 Bases de Datos

## Tarea 2: PHP + MySQL

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

Abril, 2024

### 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 **15 de Mayo 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 encontrará en la sección Tareas.**

## 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.

### 3. Descripción del problema

La empresa Congresos Ltda luego de su prestigiosa implementación de la base de datos del sistema GESCON ha decidido continuar trabajando con usted pero ahora implementando el sistema con el cual podrán realizar ciertas actividades, entre ellas los casos de uso. 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 **Function**, 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, UPDATE Y DELETE de Autores y Revisores

- ☐ En la sección login se deberán poder registrar usuarios (Autores y Revisor), tener en cuenta que al momento de registrar cada tipo de usuario se debe realizar la asignación de Rol
- ☐ Se deberá crear una sección en la cual el usuario que este ingresado en el sistema pueda visualizar sus datos de usuario, modificarlos o eliminar el usuario si lo desea

#### ■ Página principal

- ☐ Se debe implementar una barra de búsqueda que permita encontrar artículos por un trozo de nombre (Ambos roles pueden usar esta función).
  - ☐ Una sección que permita visualizar todos los artículos evaluados por los revisores. Es decir, que se muestre el título, resumen, tópicos y autores (solo nombre para cada autor) (Ambos roles pueden usar esta función).
  - ☐ Una sección que permita visualizar los artículos que incluyan al usuario como autor. (Deben aparecer todos los artículos, no solo aquellos donde el usuario es el autor de contacto). **Además en esta sección deben realizarse las acciones conforme al caso de uso 2: Acceso al artículo, posterior al envío.**
  - ☐ Una sección que permita ingresar lo necesario para subir un artículo **conforme al caso de uso 1: Enviar artículo.**
  - ☐ **Exclusivamente para el jefe de comité del programa** Una sección para gestionar los datos de los revisores **conforme al caso de uso 3: gestión de revisores.**
  - ☐ **Exclusivamente para el jefe de comité del programa** Una sección para la asignación de artículos a revisores **conforme al caso de uso 4: Asignación de artículos a revisores.**
- **Búsqueda avanzada**
    - ☐ Se debe implementar la opción de ejecutar distintos filtros para filtrar la búsqueda. Estos son: Autor, Fecha de envío, Tópicos, Revisor. En los resultados de búsqueda deberá aparecer el nombre del artículo, su resumen y sus tópicos.
  - **CREATE, READ, UPDATE, DELETE Artículos**
    - ☐ Se solicita el desarrollo de un sistema CRUD (CREATE, READ, UPDATE, DELETE) para una página web de los artículos, con sus autores y revisores respectivos que generaron en la Tarea 1. Los autores deberán poder registrar, leer, actualizar y eliminar artículos, generados por ellos mismos, pero con la restricción de que los artículos en revisión no pueden ser modificados ni eliminados. Cada artículo tendrá campos como título, fecha de envío, resumen, tópicos y autores, y el primer autor registrado será el autor de contacto. Los revisores, por su parte, tendrán acceso a los artículos en revisión y podrán añadir reseñas, que incluirán observaciones y una calificación, las cuales pueden ser editadas o eliminadas.

## 4. Aclaraciones

1. En caso de que los enunciados mencionen algún tipo de envío por email usted suponga que esta implementación ya se encuentra realizada y solo deberá mostrar en la página una alerta con un mensaje que diga "CORREO ENVIADO"
2. En General se especifican implementaciones con los casos de uso y estas deben estar en su respectiva sección, cumpliendo todas las pruebas de aceptación .
3. Es obligatorio utilizar claves primarias y foráneas, existirá descuento en la pauta de no usarse.
4. Cualquier supuesto debe ser especificado de forma explícita en el README, habrá un descuento por cada supuesto no especificado.
5. Function, 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.

6. 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.
7. **Se recomienda iniciar con tiempo esta tarea.**
8. **No se dará mas plazo para esta entrega**

## 5. Anexos

Como complemento y para tener un mejor entendimiento de algunos puntos se anexan los siguientes casos de uso, junto a sus mockups y pruebas de aceptación. Estos sirven como apoyo para tener un mejor entendimiento del sistema GESCON.

1. Enviar Artículo
2. Acceso al artículo, posterior al envío
3. Gestión de Revisores
4. Asignación de artículos a revisores

### CASOS DE USO

#### 5.1. Enviar artículo

Quién realiza esta acción: Autor

Se introducen los datos del artículo (título, resumen, tópicos) y de los autores (nombre, e-mail). Uno de los autores debe indicarse como autor de contacto. El sistema confirma la correcta recepción del artículo enviando un e-mail al autor de contacto con un userid y password para que el autor pueda posteriormente acceder al artículo.

Hand-drawn mockup of the 'Enviar artículo' form. The form contains the following elements:

- Título:** A text input field with placeholder text 'Ta ...'.
- RESUMEN:** A text input field with placeholder text '... '.
- Autores:** A table with three columns: 'Nombre', 'e-mail', and 'Contacto'.

Nombre	e-mail	Contacto
N <sub>1</sub>	em <sub>1</sub>	<input type="radio"/>
N <sub>2</sub>	em <sub>2</sub>	<input checked="" type="radio"/>
⋮	⋮	⋮
- Tópicos del Artículo:** A list of topics with checkboxes.
  - Tópico<sub>1</sub> ☐
  - Tópico<sub>2</sub> ☒
  - Tópico<sub>3</sub> ☐
  - ⋮
- Enviar:** A button labeled 'Enviar'.

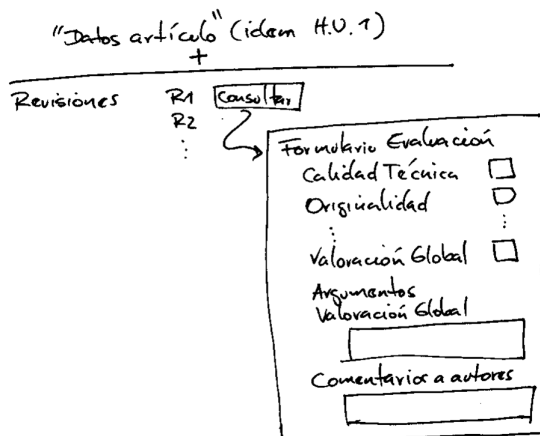
#### Pruebas de Aceptación

- Al enviar, debe existir título, al menos un autor (al menos uno marcado como de contacto). Al menos un tópico marcado.
- Al enviar, no debe existir el título de artículo para alguno de los autores
- Al enviar, no deben repetirse nombres de autores

## 5.2. Acceso al artículo, posterior al envío

Quién realiza esta acción: Autor

Posterior al momento del envío, el autor (previa autenticación) puede modificar los datos de su artículo. Todo esto sólo si no se ha sobrepasado la fecha límite de envío de trabajos. Cuando el artículo ha sido aceptado, estarán disponibles las revisiones para ser consultadas.



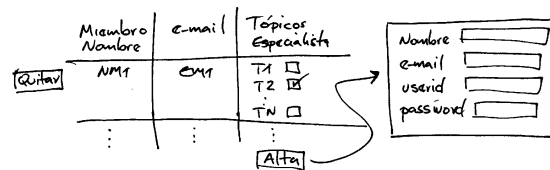
## Pruebas de Aceptación

- Cuando se han publicado los resultados de la revisión se podrá acceder en modo sólo consulta al Formulario de Evaluación

### 5.3. Gestión de Revisores

Quién realiza esta acción: Jefe Comité de Programa

Se podrán añadir, modificar, eliminar y consultar los Revisores (Miembros del Comité de Programa). Los datos asociados son: nombre, e-mail, tópicos en los cuales es especialista, userid y password. El sistema envía automáticamente un e-mail al miembro indicando su userid y password.



### Pruebas de Aceptación

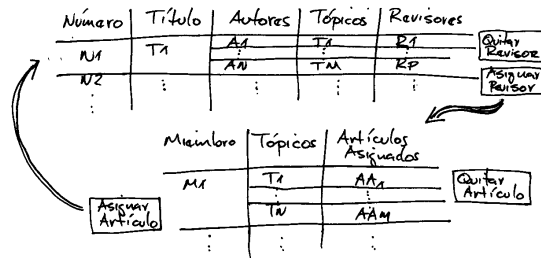
- Visualizar los revisores y poder editar sus datos
- Permitir añadir y eliminar revisores
- No permitir eliminar revisores que ya tengan asignado artículos para revisión
- Notificar por email los eventos de añadir o eliminar revisor a la persona correspondiente
- Cada revisor debe tener uno o más tópicos de interés
- Verificar que no existen revisores duplicados (por nombre y email)



## 5.4. Asignación de artículos a revisores

Quién realiza esta acción: Jefe Comité de Programa

A cada artículo se le asignan revisores entre los Miembros del comité de Programa. Debe intentarse hacer coincidir los tópicos del artículo con los tópicos en los cuales el revisor es especialista. No debe asignarse un artículo a un revisor si éste figura como autor. Sería interesante disponer de un mecanismo de asignación automática en base a tópicos y para que después se continuara manualmente (pero sin perder las asignaciones manuales si ya se han realizado). Hay que mostrar claramente aquellos artículos que tienen menos de dos revisores y también conocer cuántos artículos tiene asignado un revisor.



## Pruebas de Aceptación

- Asignar un artículo a un revisor
- Asignar un revisor a un artículo
- No permitir que un artículo se asigne a un revisor que sea autor del artículo
- Al asignar artículo-revisor avisar si los tópicos del artículo no coinciden con los del revisor
- Los artículos con menos de dos revisores aparecen resaltados
- Asignación automática de artículos con revisores cuando coincidan en cuanto a tópicos
- Re-asignación automática en cualquier momento sin perder las asignaciones manuales ya realizadas