

INF-239 Bases de Datos

Tarea 2: PHP + MySQL

Profesores: Rodrigo Olavarria - Sebastián Gallardo

Ayudante de Cátedra: Tomás Guttman

Ayudantes de Tareas: Debora Alayo - Daniela Sánchez - Sebastián Castro

21 de octubre de 2022

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.
- La tarea debe ser entregada el **3 de diciembre 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 15 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.

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 **Blockbusm**.
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**, 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 jueves 1 de diciembre a las 23:55.
7. Por respeto, están prohibidos los spoilers. De existir alguno, habrá un descuento de 50 puntos.
8. 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**.
9. **Las instancias de consulta son los laboratorios**, el foro de consultas de aula está destinado a consultas puntuales, prefiera su uso por sobre el correo electrónico.

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.
- La tarea fue subida con tiempo para que pudieran trabajar adecuadamente, y la entrega esta programada para finales de semestre, por lo que **es imposible negociar un cambio de fecha**.
- 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 que falle la ejecución de algún comando, no se asignará puntaje a éste.
- Existe la posibilidad de que a su defensa, asista su profesor.
- 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 prestigiosa empresa creadora de Multiusm, busca aventurarse en el rubro del servicio de arriendo de películas por medio de la plataforma Blockbusm. Dicha plataforma pretende ofrecer a sus clientes un amplio catálogo de películas para rentar por un precio muy accesible.

En Blockbusm los usuarios pueden registrarse en el sistema para acceder a diversas funcionalidades además de rentar películas, como por ejemplo crear una lista de películas favoritas, agregar a su wishlist las películas que en un futuro les gustaría rentar, también pueden escribir una reseña y dar calificación (del 1 al 5) para una película en específico, entre otras funciones en la página.

Los usuarios cuentan con un nombre, saldo, número de seguidores, número de cuentas seguidas y una descripción personal. Las películas cuentan con géneros, descripción, ejemplares disponibles, ejemplares totales, público (G, PG o R), duración, precio, reparto, calificación media según el sitio, veces rentada en el sitio, calificación media según usmtomatoes y una imagen que representa el contenido multimedia. Utilizando su conocimiento en BD, se le solicita crear una versión de Blockbusm que cumpla con las exigencias de la empresa.

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) 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.
- ☐ Se debe hacer uso de al menos un **Procedimiento almacenado** y debe ser funcional
- ☐ Se debe hacer uso de al menos un **trigger** y debe ser funcional
- ☐ Se debe hacer uso de al menos una **función** y debe ser funcional
- ☐ Se debe hacer uso de al menos una **View** y debe ser funcional
- ☐ El sistema debe tener navegación entre secciones (recargar la página o retroceder con el navegador no es válido).
- ☐ Implementar sistema de usuarios.
- ☐ El sistema debe mostrar el nombre de usuario que tiene la sesión activa en todo momento.
- ☐ El sistema debe mostrar el logo de la plataforma en la parte superior de la pantalla en todo momento. (puede diseñarlo usted o utilizar una imagen de internet)
- ☐ Cada usuario debe poder rentar y devolver películas de acuerdo al saldo que disponga en su cuenta. (las películas no tienen límite de tiempo en que deben ser devueltas).
- ☐ Los usuarios pueden seguir a otros usuarios. Esto implica que si el usuario 1 sigue al usuario 2, al usuario 1 le aparecerán las reseñas del usuario 2 en su página de inicio.
- ☐ Se debe mostrar la cantidad de veces que la película fue rentada en el pasado.
- ☐ NOTA 1: cada reseña está asociada a una calificación pero no toda calificación está asociada a una reseña. Lo anterior implica que un usuario puede dejar su calificación con o sin reseña en una película.
- ☐ NOTA 2: Con respecto a las películas usted deberá ingresarlas a la base de datos junto con la información necesaria (puede ser ficticia).

■ Página principal

- ☐ Se debe implementar una barra de búsqueda, esta permite buscar directamente por nombre de películas y a otros usuarios. Incluyendo fracción de nombres (por ejemplo, buscar la palabra Iron en películas y que aparezca iron man 1, iron man 2, Iron sky, etc). Los resultados de la búsqueda deben tener la opción de llevar a la película o usuario.
- ☐ Mostrar top 5 mejores películas según la calificación de usmtomatoes. Se debe mostrar imagen, nombre y calificación actual.
- ☐ Mostrar top 5 peores películas según la calificación de usmtomatoes. Se debe mostrar imagen, nombre y calificación actual.
- ☐ Se debe mostrar el Top 5 de películas con mayor cantidad de reseñas. Se debe mostrar imagen, nombre y cantidad de reseñas actual.
- ☐ Mostrar top 5 de películas más rentadas según cada público (G, PG o R).
- ☐ Una sección con las películas de las cuales quedan pocas unidades para poder rentar. Debe mostrar la portada de la película, una breve descripción y cuántas unidades quedan para rentar (todas aquellas menores a 3 unidades).
- ☐ Una sección que muestre la información de la sesión de usuario actual y permita llevar al perfil completo del usuario
- ☐ Una sección donde según los usuarios a los que siga la sesión actual, se muestren las reseñas de estos según orden cronológico (más recientes primero).

■ Perfil

- ☐ **CRUD del usuario:** Eliminar cuenta, crear cuenta, editar información del perfil y visualizar la información del usuario en el perfil.
- ☐ **CRUD para la wishlist:** Crear lista de películas, agregar películas a la lista, eliminar películas de la lista.
- ☐ Se debe mostrar la información básica del perfil, como la cantidad de seguidores que tiene el usuario, el nombre de usuario, la cantidad de películas rentadas históricamente y actualmente y se debe poder acceder a la lista de películas favoritas.
- ☐ Se debe poder ver todas las personas a las que se sigue, con la opción de ir al perfil de cualquiera de ellas.
- ☐ Se deben mostrar todas las reseñas hechas por el usuario hasta la actualidad en orden cronológico.

■ Películas

- ☐ CRUD (Eliminar, editar, añadir y visualizar) de reseña. Solo el usuario que creó la reseña puede editarla o eliminarla.
- ☐ Se debe mostrar la imagen relacionada a la película.
- ☐ Se debe mostrar la puntuación media de los clientes y del sitio, duración de la película, el precio de la película, la descripción, las etiquetas de género, la cantidad disponible de la película, la cantidad total de la película, la clasificación por edades y lo mencionado en items anteriores.
- ☐ Para simular la renta de la película, se debe implementar un botón que permita rentar la película si es que hay disponibles, luego debe descontar el valor del saldo del usuario y además debe implementar un botón para devolver la película si es que se rentó.
- ☐ Implementar forma para que el cliente que la haya rentado puede darle una puntuación y reseña opcional. Si se da una reseña, debe haber una puntuación asociada.
- ☐ Mostrar en cada película la cantidad histórica de veces que ha sido rentada

4. Aclaraciones

1. Puede asumir que cada usuario tendrá una cantidad suficiente de dinero para arrendar una gran cantidad de películas, es decir, no es necesario que implemente una manera de cargar dinero, pero sí debe existir la transacción de arriendo, es decir, se le descuenta de su saldo.
2. Las películas pueden ser añadidas a la base de datos directamente (deben mostrarse en la página para arrendarlas), no es necesario crear un usuario especial con poderes de admin para agregar contenido, pero puede hacerlo si le acomoda más.
3. Aunque se hable de película todo el tiempo, **no se debe subir ni reproducir una película real**, solamente se debe subir una imagen .JPG en representación de la película. Recuerde que es una página de arriendo de películas no de visualización.
4. Implementar **READ** significa que se debe poder ver las entidades mencionadas.
5. Toda persona que quiera dar calificación, reseñar o seguir a otros debe iniciar sesión.
6. No se puede dar puntuaciones distintas al mismo tiempo. Pero sí actualizar la puntuación y editar una reseña.
7. Los usuarios pueden dejar de seguir a otro usuario.