

INF-239 Bases de Datos

Tarea 2: PHP + MySQL/PostgreSQL

Profesores: Marcelo Mendoza - Rodrigo Olavarria
Ayudantes de Tareas: Debora Alayo - Diego Beltrán - Isidora Ubilla

11 de junio de 2022

1. Objetivo

- **Primera parte:** Analizar y comprender un problema real, modelando correctamente su contexto en un modelo de datos consistente, en base a los métodos estudiados en clases.
- **Segunda parte:** Investigar y aplicar conceptos básicos de creación de aplicaciones web con bases de datos integradas, utilizando herramientas como PHP y MySQL/PostgreSQL.

2. Especificaciones y reglas

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

2.1. Primera parte

- Se debe entregar un informe vía Aula, en formato .pdf y debe llevar el nombre: MODELO-ROL.pdf, en la sección Tareas, a más tardar el día **Miércoles 22 de Junio, hasta las 23:55 horas**, existirá un descuento de 10 puntos dentro de la primera hora de atraso (primeros 5 minutos no cuentan) y 20 puntos por cada hora siguiente de atraso (considerando los 10 puntos iniciales).
- Se responderán dudas de enunciado por aula y discord a las preguntas hechas hasta 2 días previos a la entrega es decir hasta el 20 de junio a las 23:55.
- El informe no tiene un formato en específico, este debe incluir lo siguiente:
 1. Descripción breve del problema, incluyendo los supuestos que realizó para la tarea.
 2. Modelo de datos conceptual, expresado en un diagrama de clases UML, utilizando una herramienta CASE, como starUML o Draw.io.
 3. Modelo de datos relacional obtenido utilizando el proceso de 8 pasos vistos en clases. También expresado en un diagrama de clases UML.

2.2. Segunda parte

- Se debe desarrollar una aplicación web básica, teniendo libertad en elegir como implementar lo que se les solicita, utilizando PHP como herramienta de desarrollo.
- **La base será asignada eventualmente por aula.**
- La parte 2 debe ser entregada el **18 de Julio hasta las 23:55 horas**, existiendo una tolerancia de 10 minutos de atraso posterior a esto no se podrán realizar más entregas.
- Se dará un bonus de 20 puntos a quienes desarrollen frontend (diseño de la página). Ejemplo: CSS, Bootstrap.

2.2.1. Sobre la entrega

La tarea debe ser entregada como un archivo comprimido de la forma T2-ROL.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 **CrunchyUSM**.
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.
5. Se debe incluir un archivo **README**, puede ser **.txt** o **.md**, el cual debe contener el nombre, rol del alumno 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 de enunciado por aula a las preguntas hechas hasta 2 días previos a la entrega es decir hasta el sábado 9 de julio a las 23:55 (no se considera la semana de vacaciones).
7. Por respeto, están prohibidos los spoilers. De existir alguno, habrá un descuento de 50 puntos.
8. Esta parte de la 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.**

2.3. Reglas

Las siguientes reglas corren para ambas partes de la tarea.

- 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**.

3. Descripción del problema

CrunchyUSM es una plataforma de streaming de anime de alto prestigio, donde se puede encontrar un catálogo enorme de animaciones y películas. Los usuarios pueden registrarse en el sistema para acceder a diversas funcionalidades, como por ejemplo crear una lista de animes favoritos, seguir a sus amigos y ver sus listas de anime favoritos, comentar y dar estrellas (del 1 al 5) a los animes y películas, pueden acceder a un menú donde pueden visualizar animes y películas según géneros, multi-géneros (ejemplo: shounen + aventura), orden alfabético, novedades (ordenados según tiempo) y los más populares (puntuación media) en la plataforma, además tienen un historial de los animes vistos ordenados de manera cronológica. Además, CrunchyUSM cuenta con un súper-usuario, quien es el encargado de subir los anime y películas con toda la información referente a ellos.

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

■ General

- ☐ Se deben crear índices secundarios para acelerar algunas consultas frecuentes.
- ☐ Se debe hacer uso de **triggers útiles**.
- ☐ Se debe hacer uso de **vistas útiles**.
- ☐ 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.

■ Página principal

- ☐ Se debe implementar una barra de búsqueda, esta permite buscar directamente por nombre de animes, películas y a otros usuarios. Los resultados de la búsqueda deben tener la opción de llevar al anime o película.
- ☐ Se debe implementar un menú donde se pueda visualizar animes y películas según géneros, multi-géneros (ejemplo: shounen + aventura), orden alfabético, novedades (ordenados según tiempo) y los más populares (puntuación media) en la plataforma.
- ☐ Se debe mostrar el Top de los 5 anime con mayor popularidad (según puntuación media).
- ☐ Se debe mostrar el Top de los 5 animes con menor popularidad (según puntuación media).
- ☐ Se debe mostrar el Top de los 5 animes con mayor cantidad de comentarios.

■ Perfil

- ☐ **CRUD del usuario:** Eliminar cuenta, crear cuenta, editar información del perfil y visualizar la información del usuario en el perfil.
- ☐ **CRUD para las lista de animes favoritos:** Crear lista de animes, agregar animes a la lista, eliminar animes de la lista, dar una calificación de 1 a 5 estrellas y poder actualizarlas si se cambia de opinión.
- ☐ Se debe mostrar la información básica del perfil, como la cantidad de seguidores que tiene el usuario, el nombre de usuario, la cantidad de anime visto y se debe poder acceder a la lista de animes favoritos.

- ☐ Se debe poder ver todas las personas a las que se sigue, con la opción de ir al perfil de cualquiera de ellas.
- ☐ Se debe poder acceder al historial de

■ Súper-usuario

- ☐ **CRUD de los animes:** Agregar animes y películas a la plataforma junto con toda su información (descripción, cantidad de capítulos, puntuación media inicial de 0 estrellas y etiquetas de género), actualizar la cantidad de capítulos disponibles en la plataforma, eliminar animes o películas.
- ☐ En cuanto al Create, se debe poder subir una imagen en formato .jpg, en representación del anime o película, este debe como mínimo tener un título y un color.

■ Anime

- ☐ CRUD para comentarios.
- ☐ Se debe mostrar la imagen relacionada al anime.
- ☐ Se debe mostrar la puntuación media, la cantidad de comentarios y capítulos o duración en el caso de ser película, la descripción y las etiquetas de género.
- ☐ Para simular la visualización de un anime por parte del usuario, se debe agregar un botón que permita distinguir si el usuario ha visto o no el anime.

4. Imágenes referenciales

A continuación, se muestran imágenes de referencia para que puedan tener una idea de la información que se les pide:



Figura 1: Vista de búsqueda de anime en específico.

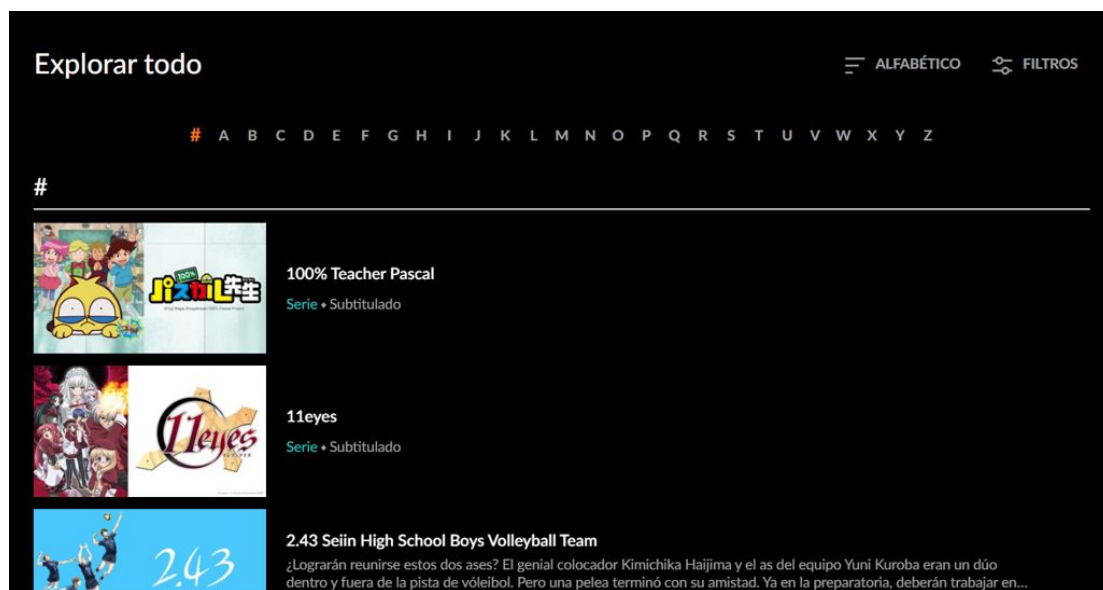


Figura 2: Visualización de anime por orden alfabético.

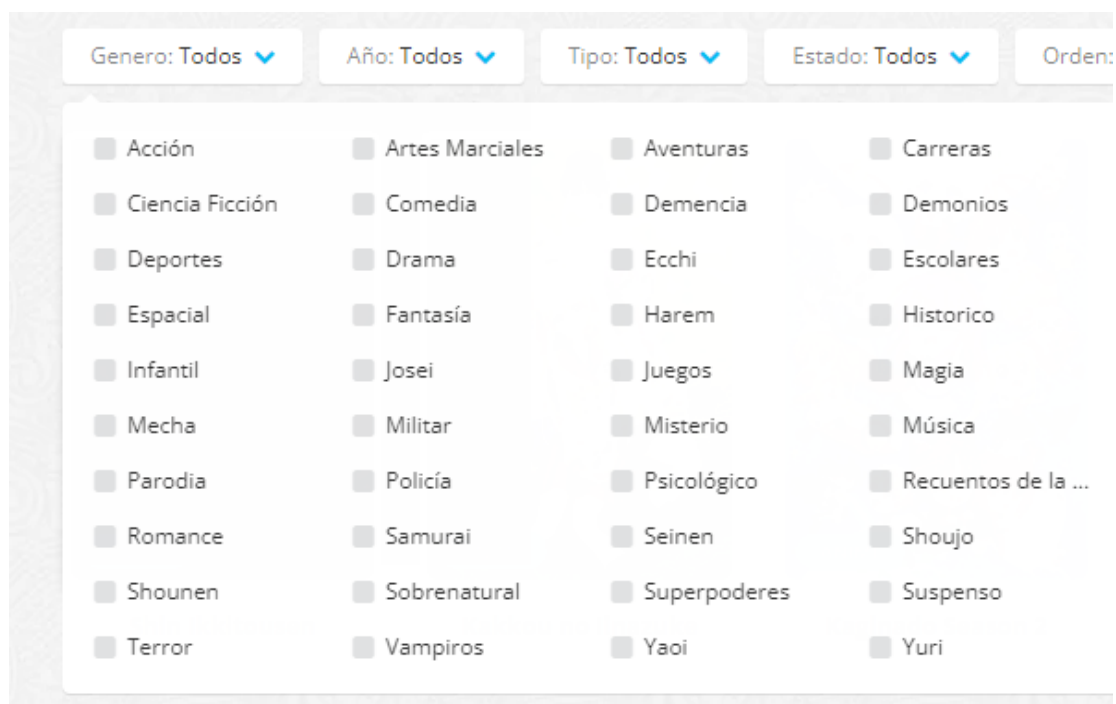


Figura 3: Visualización de anime por multi-género.



Figura 4: Visualización de menú para búsquedas por popularidad, género, novedad y orden alfabético.

5. Aclaraciones

1. Aunque se hable de anime todo el tiempo, **no se debe subir ni reproducir un anime real**, solamente se debe subir una imagen .JPG en representación del archivo multimedia. En caso de querer usar videos, se puede hacer pero no es obligación ni afectará la nota.
2. Implementar **READ** significa que se debe poder ver las entidades mencionadas.
3. Toda persona que quiera dar puntuación, comentar o seguir a otros debe iniciar sesión.
4. Un anime puede tener múltiples géneros.
5. Una persona puede hacer cuantos comentarios quiera a un anime, también puede editar y eliminar.
6. No se puede dar puntuaciones distintas al mismo tiempo. Pero sí actualizar la puntuación.
7. Los usuarios pueden dejar de seguir a otro usuario.