

Desarrollar una aplicación web en PHP usando Laravel. Finalidad gestionar las películas, proyecciones y entradas de un cine.

Las películas se proyectan en proyecciones que se realizan en salas, y los espectadores compran entradas para una determinada proyección.

La base de datos se implantara en PSQL y deberá contar, al menos, con las siguientes tablas y columnas:

- películas(id, titulo)

- salas(id, numero)

- proyecciones(id, pelicula_id, fecha_hora, sala_id)

(La sala_id es el identificador de la sala donde se proyecta esa proyección. La pelicula_id es el

identificador de la película que se proyecta durante esa proyección. Además, hay una restricción

UNIQUE múltiple sobre la agrupación de columnas pelicula_id + fecha_hora + sala_id)

- entradas(id, proyeccion_id)

(La proyeccion_id es el identificador de la proyección para la que es esa entrada).

1. (2pts) Crear las tablas con las migraciones correspondientes (una migración por cada tabla) en el orden correcto, así como los modelos y relaciones entre ellos. (Indicación: Tener en cuenta que el singular de “proyecciones” es “proyeccion”, no “proyeccions”. Consultar posibles argumentos en el método constrained().)

php artisan make:model -mcr Pelicula
configuras la migracion

php artisan make:model -mcr Sala
configuras la migracion

php artisan make:model -mcr Proyecciones
configuras la migracion, pero aquí cambias el nombre del archivo de proyeccions a proyecciones

En la que pide unique, no hay que poner unique a cada una sino hacer un unique de las 3 a la vez sería así:

table->unique(['pelicula_id', 'fecha_hora', 'sala_id'])

php artisan make:model -mcr Entradas
configuras la migracion

Al finalizar haces un php artisan migrate

Ahora hacemos las relaciones entre las tablas (esto es base de datos puros, modelo E/R)

Luego hacemos los modelos en base a las relaciones.
(Esto ultimo esta en el git de ricardo, echarle un vistazo)

2. (2pts) CRUD de películas con acciones index, create, store, show, edit, update y destroy.

Aquí nos vamos a los controllers también poner en el web.php las rutas que creamos, aquí coges la vista de categorías y la modificas para este ejercicio, así como con los helpers (las funciones que ayudan, aunque es opcional).

3. (3pts) En la ruta GET peliculas/id (siendo id el identificador de la película), mostrar el título de la película y el número total de entradas que se han comprado para esa película, teniendo en cuenta todas las proyecciones que existan para esa película, ya sean pasadas o futuras.

Dicha ruta es la del show que hicimos antes

Es irte al controller y con las relaciones mostrar lo que pide

Para sacar eso usa un foreach

```
$total = 0;
foreach ($pelicula->proyecciones as $proyeccion){
    $total += $proyeccion->entradas->count();
}
```

luego esto lo metes en la vista:

4. (3pts) Impedir cambiar o borrar una película si ya se han comprado entradas para cualquier proyección de esa película, ya sean pasadas o futuras.

Es hacer una función que mire si la película tiene entradas y tire un mensaje de error.