**Examen Práctico de Laravel:**

**Gestión de Películas de Harry Potter y Directores de las mismas**



**“Las decisiones que tomamos muestran quiénes somos, mucho más que las habilidades que poseemos” (Albus Dumbledore)**

Contenido

[**Examen Práctico de Laravel:** 1](#_Toc188531725)

[**Gestión de Películas de Harry Potter y Directores de las mismas** 1](#_Toc188531726)

[**“Las decisiones que tomamos muestran quiénes somos, mucho más que las habilidades que poseemos” (Albus Dumbledore)** 1](#_Toc188531727)

[Objetivo: 2](#_Toc188531728)

[1. Instrucciones: 2](#_Toc188531729)

[2. Requisitos: 5](#_Toc188531730)

[2.1 Ejecutar migraciones 9](#_Toc188531731)

[3. Controlador y Rutas 10](#_Toc188531732)

[4. Vistas 13](#_Toc188531733)

[5. Validación 16](#_Toc188531734)

[6. Información para Población de las Tablas 17](#_Toc188531735)

[7. Bonus (Opcional) 18](#_Toc188531736)

[**Criterios de Evaluación:** 19](#_Toc188531737)

# Objetivo:

Crear una aplicación básica para gestionar un listado de películas de Harry Potter, donde cada película esté asociada a un director. Los usuarios podrán:

* Ver un listado de películas.
* Crear nuevas películas y asignarles un director.
* Editar y eliminar películas existentes.

# Instrucciones:

#### Crea un proyecto nuevo de Laravel utilizando:

1. laravel new gestion-peliculas

Texto

Descripción generada automáticamente

Aquí nos preguntara que bd por defecto queremos utilizar:

Texto

Descripción generada automáticamente

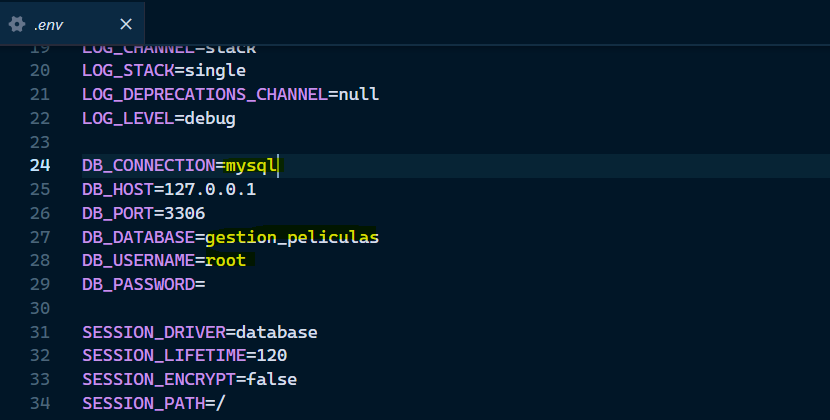
Le decimos que nos añada las tablas por defecto también:

Texto

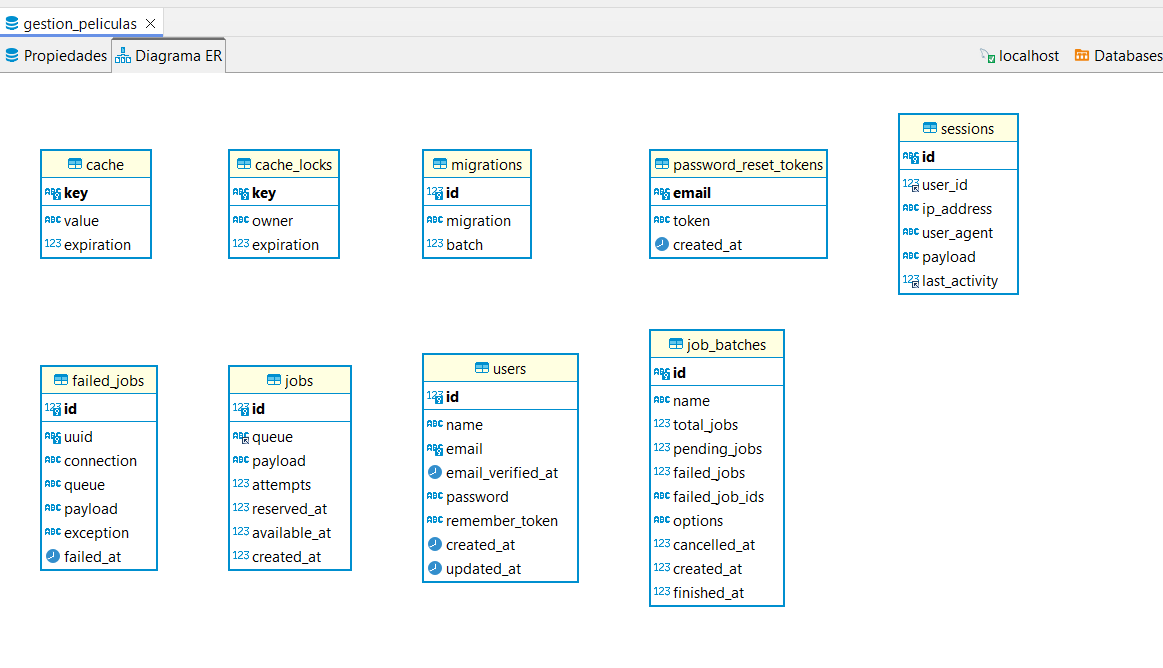
Descripción generada automáticamente

#### Configura la base de datos en el archivo. env para usar MySQL.

Una vez dentro del proyecto nos dirigimos a configurar el archivo .**env** de tal manera que configuramos los parámetros para el acceso a la conexión con la bd.



Una vez dentro el gestor podremos apreciar las tablas por defecto que tiene laravel:



1. Completa los pasos siguientes para desarrollar la aplicación.
2. Realiza capturas de pantalla de cada paso e inclúyelas en un documento PDF que entregarás junto con el código en un archivo comprimido (ZIP).

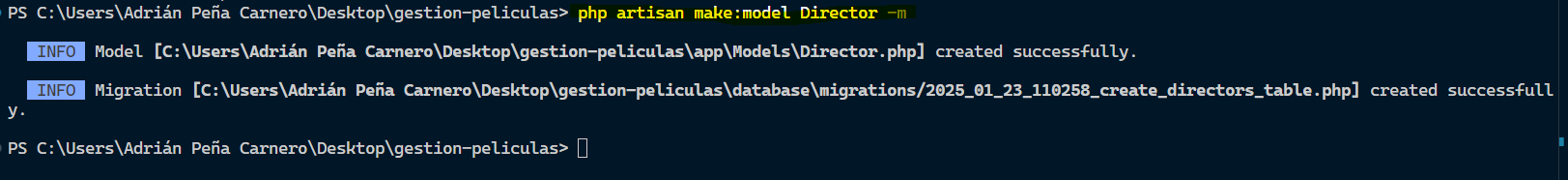
# 2. Requisitos:

#### 1. Modelos y Migraciones

Crea los modelos y sus migraciones correspondientes para las siguientes entidades:

1. **Director**:
   * **nombre** (string, requerido).
   * **fecha\_nacimiento** (date, requerido).
     + *Formato esperado para la fecha: YYYY-MM-DD (ejemplo: 1965-07-31).*

Con el comando php artisan make:model Director -m crearemos el modelo y la migración a la vez:

**

Editaremos la migración con los datos requeridos:

*Texto

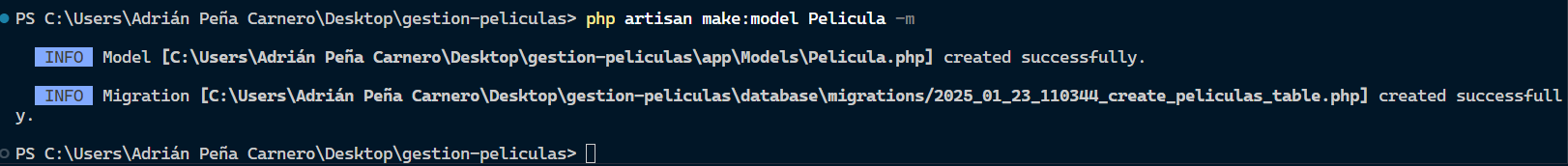
Descripción generada automáticamente*

Ahora pasaríamos a configurar el modelo con su propia relación:

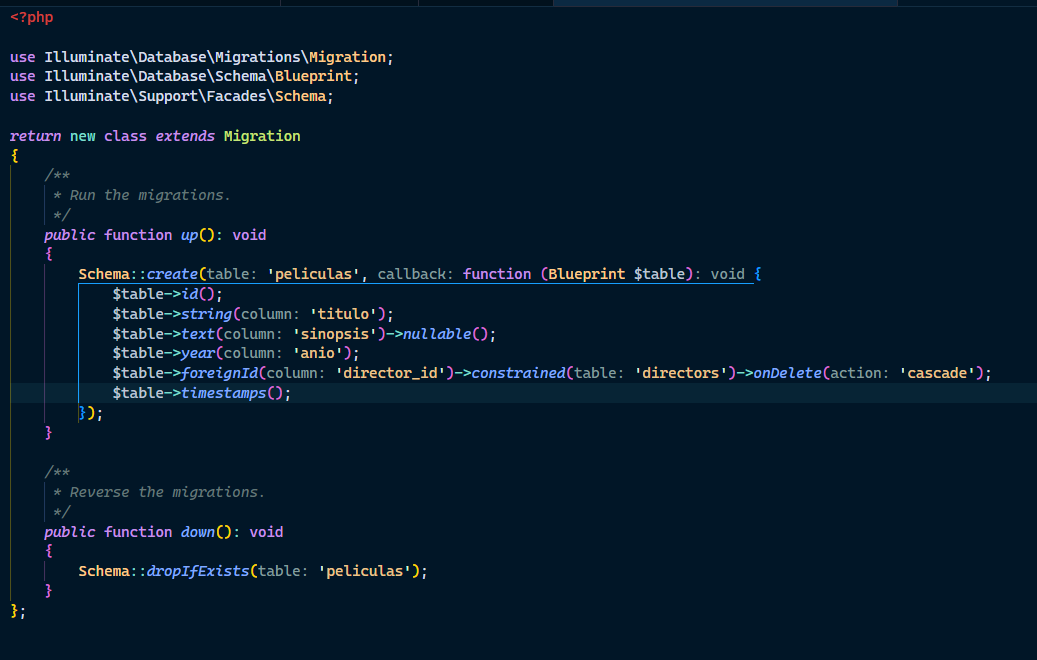
**

1. **Pelicula**:
   * **titulo** (string, requerido).
   * **sinopsis** (texto, opcional).
   * **anio** (entero, requerido, entre 2000 y 2025).
   * **director\_id** (entero, clave foránea que referencia al id de la tabla directors).

(Procedemos a realizar los pasos dichos anteriormente, pero con los parámetros requeridos en este caso)





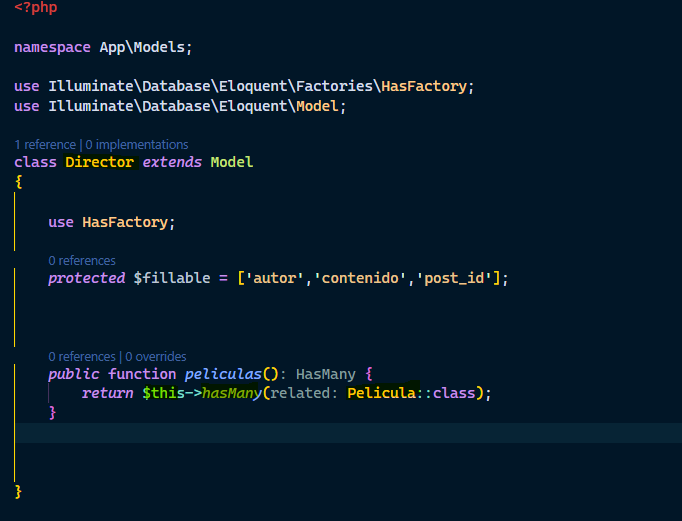


#### 2. Relaciones

Define las relaciones entre los modelos:

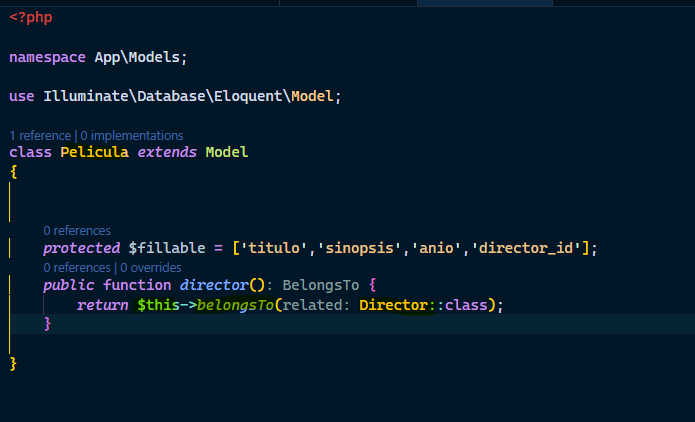
* **Director**: Un director puede haber dirigido muchas películas (relación uno a muchos).

Como podemos ver un director puede tener muchas películas y lo dejaríamos tal que así:



* **Pelicula**: Una película pertenece a un único director.

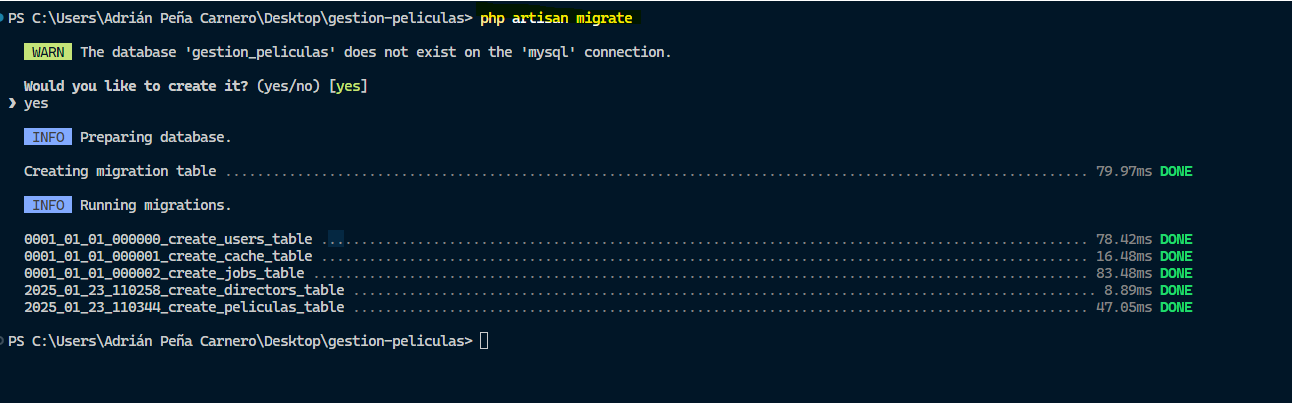
Una película pertenece a un único autor:

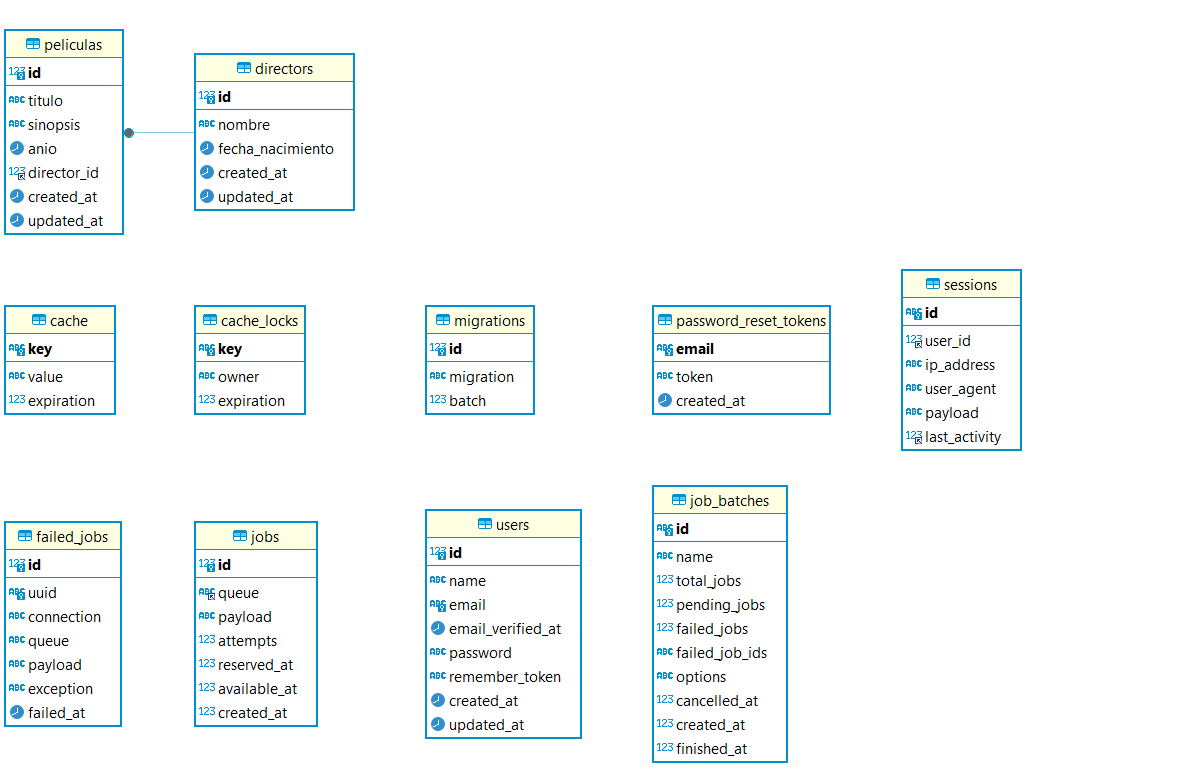


Configura estas relaciones en los modelos Director y Película

# 2.1 Ejecutar migraciones

Ahora ejecutaríamos las migraciones para tener todas las tablas de la bd:

****

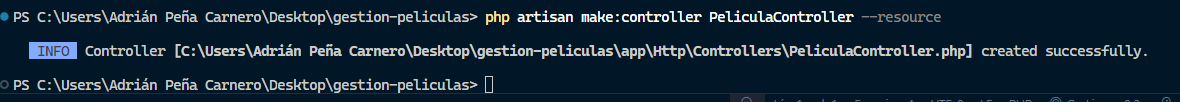
****

# 3. Controlador y Rutas

Crea un controlador llamado PeliculaController con las siguientes acciones:

1. **index()**: Devuelve una vista con el listado de películas y sus directores.
2. **create()**: Devuelve una vista con un formulario para crear una nueva película.
   * El formulario debe incluir:
     + Título.
     + Sinopsis.
     + Año (campo de tipo number, con valores entre 2000 y 2025).
     + Desplegable para seleccionar un director existente.
3. **store(Request $request)**: Almacena una nueva película en la base de datos.
4. **edit($id)**: Devuelve una vista con un formulario para editar una película existente.
   * El formulario debe incluir los datos actuales de la película y permitir cambiar el director.
5. **update(Request $request, $id)**: Actualiza los datos de una película en la base de datos.
6. **destroy($id)**: Elimina una película de la base de datos.

Configura las rutas necesarias en el archivo web.php



Texto

Descripción generada automáticamente

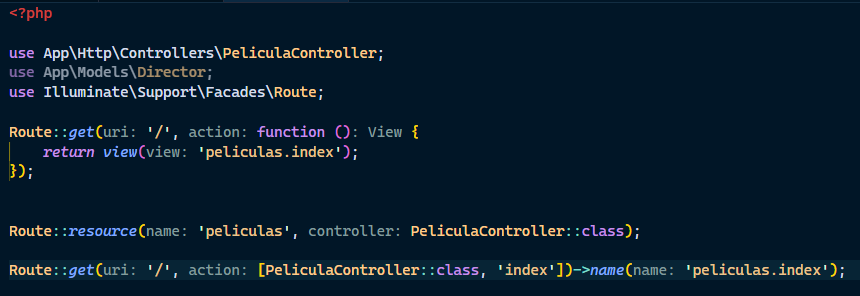
Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

#### Rutas

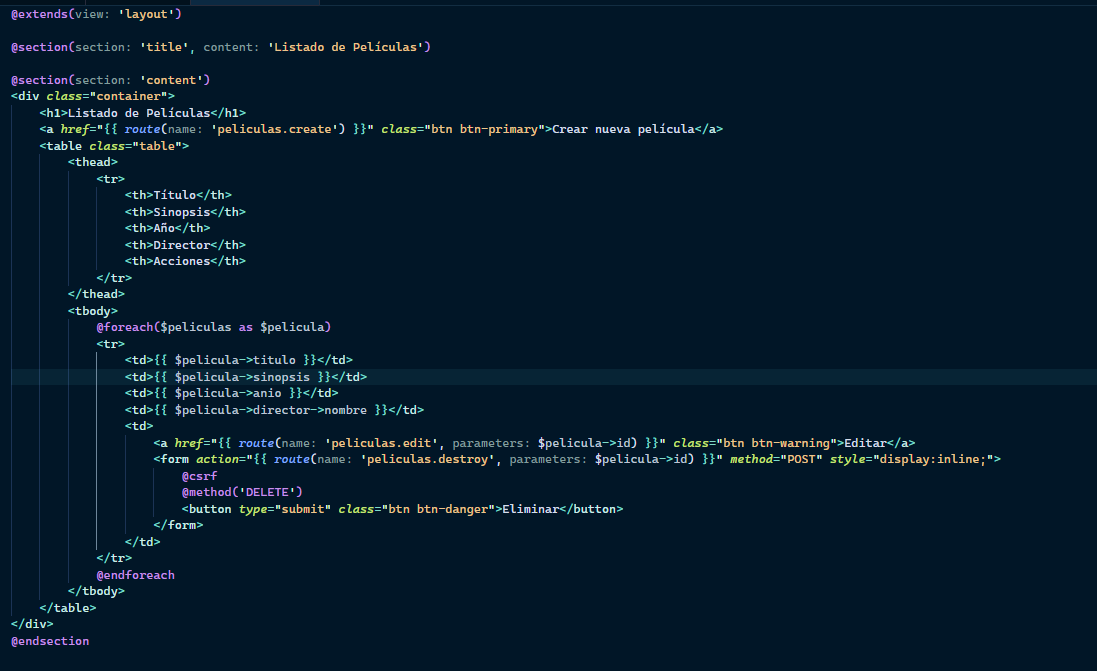


# 4. Vistas

Crea las vistas necesarias en la carpeta resources/views/peliculas:

1. **index.blade.php**:
   * Muestra un listado de películas con las siguientes columnas:
     + Título de la película.
     + Sinopsis.
     + Año.
     + Director.
     + Acciones (Editar y Eliminar).
   * Incluye un botón para "Crear nueva película" que enlace a peliculas.create.

Vista principal



1. **create.blade.php**:
   * Contiene un formulario para crear una nueva película con los campos:
     + **Título** (requerido).
     + **Sinopsis** (opcional).
     + **Año** (requerido, valores entre 2000 y 2025).
     + **Desplegable para seleccionar un director existente**.
     + **Botón "Guardar"**.

Vista creación:



Vista edición:

1. **edit.blade.php**:
   * Contiene un formulario similar al de creación, pero precargado con los datos de la película a editar.

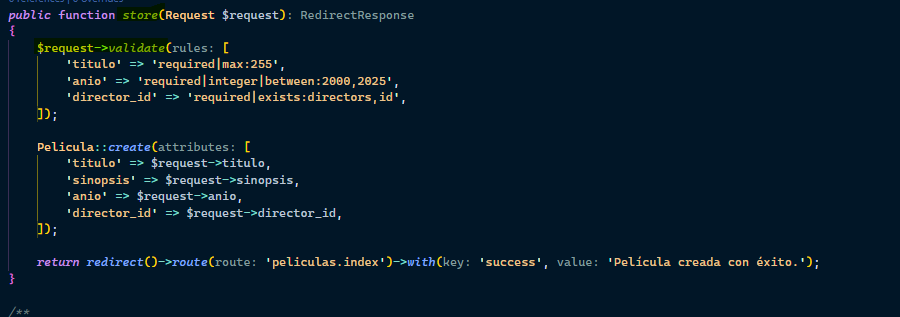


# 5. Validación

Implementa validaciones en los métodos store() y update():

* El campo titulo debe ser requerido y tener un máximo de 255 caracteres.
* El campo anio debe ser un número entero entre 2000 y 2025.
* El campo director\_id debe ser requerido y corresponder a un director existente.

Validaciones correspondientes:





# 6. Información para Población de las Tablas

Utiliza las siguientes entradas para rellenar las tablas:

**Directores**:

| **Nombre** | **Fecha de Nacimiento** |
| --- | --- |
| Chris Columbus | 1958-09-10 |
| Alfonso Cuarón | 1961-11-28 |
| Mike Newell | 1942-03-28 |
| David Yates | 1963-10-08 |

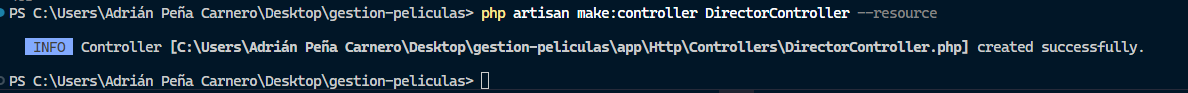
**Películas**:

| **Título** | **Sinopsis** | | **Año** | | **Director** | |
| --- | --- | --- | --- | --- | --- | --- |
| Harry Potter y la piedra filosofal | La primera aventura de Harry mientras descubre su magia. | | 2001 | | Chris Columbus | |
| Harry Potter y la cámara secreta | | Harry enfrenta el misterio de una cámara secreta en Hogwarts. | | 2002 | | Chris Columbus | |
| Harry Potter y el prisionero de Azkaban | | Harry descubre más sobre su pasado y a Sirius Black. | | 2004 | | Alfonso Cuarón | |
| Harry Potter y el cáliz de fuego | El torneo de los tres magos lleva a Voldemort de regreso. | | 2005 | | Mike Newell | |
| Los datos los he insertado con el DatabaseSeeder.php y así no tenia que ir 1 a 1 insertando con el gestor de bd: |  | |  | |  | |

# 7. Bonus (Opcional)

Si terminas antes de tiempo, añade alguna de las siguientes funcionalidades:

* Crea un controlador y vistas para gestionar los directores (listar, crear, editar y eliminar).



* Añade una funcionalidad para filtrar películas por director en la vista de listado.

**Criterios de Evaluación:**

| **Aspecto** | **Porcentaje** |
| --- | --- |
| Modelos y Migraciones | 20% |
| Relaciones y Eloquent | 15% |
| Controlador y Rutas | 25% |
| Vistas y Diseño Básico | 25% |
| Validación | 15% |
| Funcionalidad Extra (Bonus) | 10% (opcional) |

Ejemplo de ejecución:



