**Tarea PC3**

Veremos los enunciados de las preguntas:

Texto

Descripción generada automáticamente

Comprobamos los resultados en consola:

Texto

Descripción generada automáticamente

Comprobamos que el resultado es falso, es se debido a que la clase Movie valida algunos campos del objeto Movie, instanciamos pero la consola nos alerta que **m no es valido, porque no tiene titulo, y porque la ‘release\_date’ debe ser en 1930 o después;**

Explica el siguiente código:

Texto

Descripción generada automáticamente

**index:** Obtiene todos los registros de la tabla Movie y los asigna a la variable de instancia @movies. Esto se utiliza generalmente para mostrar una lista de películas en la interfaz de usuario.

**show:** Recupera el ID de la película desde la ruta URI, luego busca y asigna la película correspondiente a la variable de instancia @movie. La vista por defecto renderizada será app/views/movies/show.html.haml.

new: Inicializa una nueva instancia de Movie y la asigna a la variable de instancia @movie. Esto se usa para mostrar el formulario para crear una nueva película.

**create:** Intenta crear una nueva película con los parámetros proporcionados (movie\_params). Si la creación es exitosa, redirige a la ruta de películas (movies\_path) con un mensaje de éxito. Si hay errores, muestra un mensaje de error y vuelve a renderizar la vista '**new'.**

**edit:** Encuentra y asigna a la variable de instancia @**movie** la película que corresponde al ID proporcionado en los parámetros. Esto se usa para mostrar el formulario de edición de una película existente.

**update:** Encuentra la película correspondiente al ID proporcionado y intenta actualizar sus atributos con los parámetros proporcionados (movie\_params). Si la actualización es exitosa, redirige a la ruta de la película actualizada con un mensaje de éxito. Si hay errores, muestra un mensaje de error y vuelve a renderizar la vista **'edit'.**

**destroy:** Encuentra la película correspondiente al ID proporcionado y la elimina de la base de datos. Luego, redirige a la ruta de películas con un mensaje de éxito.

**movie\_params:** Método privado utilizado para filtrar y permitir solo los parámetros específicos necesarios para crear o actualizar una película. En este caso, permite solo los parámetros **title, rating y release\_date** dentro de los parámetros **movie** proporcionados en la solicitud.

Texto

Descripción generada automáticamente

Comprobamos:

Texto

Descripción generada automáticamente

Usamos el método **create!** para validar los datos que consignamos antes de insertar la tupla en la base de datos, por ello usa **transaction**; si la validación falla, manda una excepción.

**SSO y autenticación de terceros**

Generamos el modelo moviegoers y una migración

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Actualizamos routes.db

Texto

Descripción generada automáticamente

Y creamos su controlador sessions\_controller:

Texto

Descripción generada automáticamente

**Claves Foráneas:**

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

**SELECT reviews**.\*: Selecciona todas las columnas de la tabla reviews.

**FROM movies JOIN reviews ON movies.id=reviews.movie\_**id: Realiza una operación de JOIN entre las tablas movies y reviews utilizando la condición de igualdad **movies.id = reviews.movie\_id**.

**WHERE movies.id = 41**: Filtra las filas para seleccionar solo aquellas donde el id de la tabla movies es igual a 41.

La consulta busca y selecciona todas las revisiones asociadas a una película específica identificada por el id 41 en la tabla movies.

Ahora creamos la migración y la tabla reviews:

Texto

Descripción generada automáticamente

También creamos un nuevo modelo review.rb

Texto

Descripción generada automáticamente

Inseramos la línea de codigo:

Texto

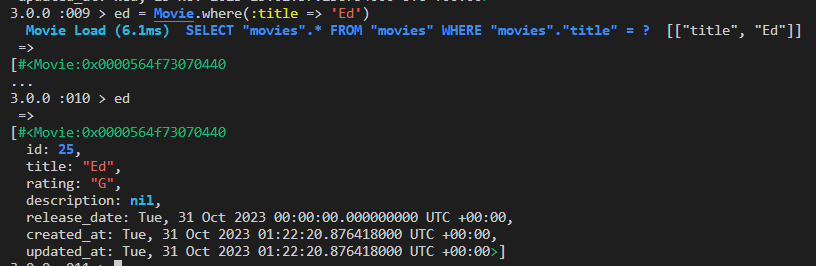
Descripción generada automáticamente

Pregunta:

Texto

Descripción generada automáticamente

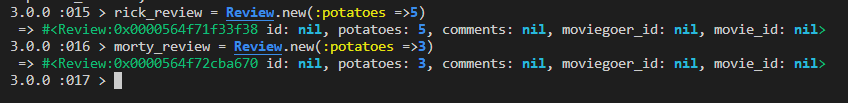
Guardamos la película Ed en la variable ed:



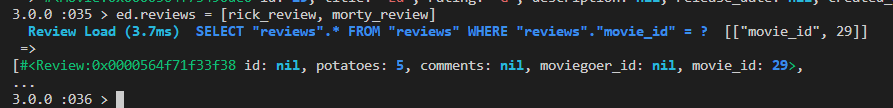
Creamos las variables Rick y morty que van a guardar los registros con id 1 y 2 respectivamenteTexto

Descripción generada automáticamente

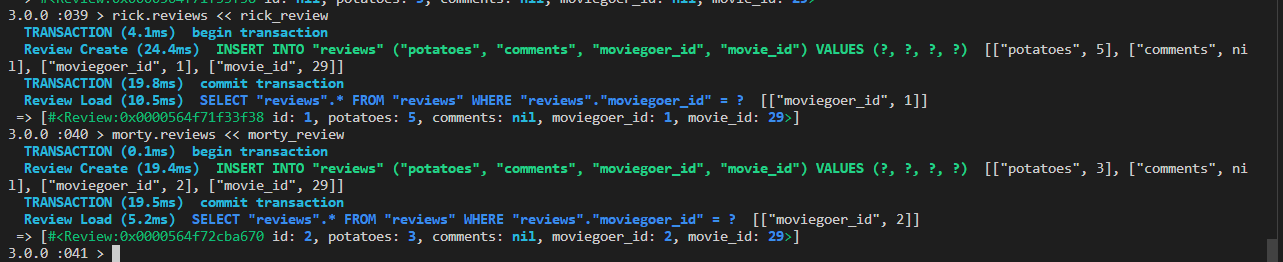
Creamos las variables rick\_review y morty\_review con la calificación de cada uno



Ahora agregamos estas reviews recién creadas a la pelicula Ed cuya referencia esta guardada en ed



Por ultimo, guardamos el Rick\_review dentro de las reviews de Rick, similarmente con Morty



Y ahora podemos ver quien hizo las reviews de la película Ed:

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Respuesta:

La consulta está buscando todas las películas que han sido revisadas por el moviegoer con el id 1, y selecciona todas las columnas de esas películas.