

PRÁCTICO 4

MVC

A partir del análisis de las clases teóricas y del material brindado, discuta y responda las siguientes consignas:

1. ¿Qué es MVC? ¿Qué ventajas tiene?
2. Indicar verdadero o falso, según corresponda. Justifique su respuesta.
 - a. En MVC, el modelo es el intermediario entre la vista y el controlador
 - b. En MVC, el controlador es el encargado de capturar los eventos del usuario
 - c. En MVC, las reglas de negocio se implementan en el modelo
3. ¿En qué casos no usaría MVC para desarrollar una aplicación web?

EJERCICIOS

4. Clone el repositorio [MOVIES](#) y realice un refactoring¹ para transformarlo en una solución MVC.
 - Importe la base de datos de películas que se encuentra en el archivo *db_movies.sql*. (Instrucciones en readme)
 - Identifique qué “partes” del código se corresponden al Modelo, Vista o Controlador y llévelos a los archivos correspondientes.
 - Puede empezar con archivos simples para cada componente MVC y después pasarlos a una solución utilizando POO.
 - No es necesario rutear la aplicación.
5. Determinar en qué componente del MVC anterior irían cada uno de los siguientes ítems:
 - a. Obtener de la base de datos las películas de *Disney*
 - b. Agregar sobre los listados un ícono de “estrella” en aquellas películas que superen el 90% de puntuación (*audience_score*).
 - c. Contar la cantidad de películas por cada género.
 - d. Verificar que el género solicitado exista en la base de datos.
 - e. Mostrar un formulario para ingresar una película al sistema.
6. Agregue a Movies, alguna de las “features” del punto 5. Respete el diseño MVC.

¹**Refactoring** es una técnica para reestructurar una porción de código, alterando su estructura interna sin generar cambios en su comportamiento exterior.

Guía de trabajo para el TPE

Se proponen los siguientes ejes para avanzar sobre la realización del Trabajo Práctico Especial.

- Crear tablas SQL con datos de test ✓
- Diseñar URLs + Ruteo ✓
- Crear página principal ✓
- Mostrar página con lista de ítems y categorías:
 - Diseñe la arquitectura del sitio usando el patrón MVC.
 - Cree las acciones necesarias en el router para ver la lista de ítems, y el detalle de cada uno.
 - Implementar las clases necesarias para completar el patrón MVC.

Ejercicios tipo parcial

Resolver, haciendo uso del **patrón MVC** y de la programación orientada a objetos.

1. El CLUB CIUDAD DE BOLIVAR desea desarrollar un sistema para administrar las habitaciones de su complejo habitacional. Para ello nos facilitan la siguiente tabla de su base de datos:

HABITACION(id: *int*; nro_habitacion: *int*, cant_camas: *int*; descripcion: *string*; ocupada: *boolean*)

Implemente una **solución MVC** para generar la siguiente funcionalidad. Además dibuje el diagrama de componentes MVC desarrollado.

- A. Obtener la lista de habitaciones, marcando en verde las que estén *libres*.
- B. Agregar una habitación al sistema. Se deben verificar los datos obligatorios. (solo el POST, no hacer el formulario).
- C. Obtener la capacidad hotelera disponible actualmente.

NOTA: cada requerimiento tiene que estar acompañado con la url de la tabla de ruteo correspondiente.

2. Nuestro cliente De Chiripa SA necesita un sistema de control de apuestas para su casino online. La empresa nos provee la siguiente base de datos:

Apuesta: (id: *int*, fecha: *datetime*, monto: *int*, id_juego: *int*,)

Juego: (id: *int*, nombre: *varchar(45)*, cantidad_jugadores: *int*, juego_de_cartas: *boolean*)

Implemente una **solución MVC** para generar la siguiente funcionalidad. Además dibuje el diagrama de componentes MVC desarrollado.

- Obtener y mostrar todos los **juegos** disponibles.
- Agregar o eliminar un **juego**.
- Obtener y mostrar la lista de **apuestas con su juego**.
- Agregar una **apuesta** a un **juego**.

Responda las siguientes preguntas:

- ¿Qué cambios realizaría para ordenar los listados por diferentes criterios?
- ¿En qué lugar del sistema verificaría que las apuestas estén dentro de dos valores mínimo y máximo permitidos? ¿Como haria para que estos valores fuesen modificables por un usuario administrador?

3. Una plataforma de streaming necesita implementar algunas características nuevas en su sistema de series online. Las serie están formadas por un conjunto de temporadas. Cada temporada tiene episodios. Cada episodio posee un título, una descripción, un flag indicando si el usuario ya vio el episodio y una calificación dada por el usuario (con valores de 1 a 5). Si el usuario no vio un episodio particular, la calificación dada será un valor negativo.

Las series poseen como atributos (además de los episodios correspondientes) un título, una descripción, un creador y un género.

- a. Determinar qué clase es responsable de responder los siguientes servicios:
- Ingresar la calificación de un episodio. Si el valor ingresado como calificación no es correcto imprimir un mensaje por pantalla y no cambiar el valor anterior.
 - Obtener el total episodios vistos de una temporada particular.
 - Obtener el promedio de las calificaciones dadas por el usuario para una serie.
- b. Implementar las clases del modelo.

Nota. Para calcular los promedios, tener sólo en cuenta los episodios vistos por el usuario.