



MongoDB

GESTIÓN DE LA INFORMACIÓN EN LA WEB

PRÁCTICA 3

Grupo 04

Claudia Gil Navarro

Ángel Luis Ortiz Folgado

Óscar Eduardo Pérez la Madrid

Esteban Vargas Rastrollo

MongoDB

Definición del esquema implícito

Dentro de nuestra base de datos **GIW_grupo04** hemos incluido las siguientes colecciones:

- ✓ teatros: en esta colección se incluye lo siguiente:
 - id: identificador único del teatro, necesario para muchas de las consultas que se hacen a la base de datos (fichero *teatro2.php*)
 - nombre_teatro
 - nombre_obra
 - descripción
 - sesion1
 - sesion2
 - sesion3
 - nume_filas
 - nume_asientos

- ✓ usuarios: contiene los campos requeridos para registrar y trabajar con clientes y administradores de la página:
 - usuario
 - nombre
 - apellido
 - dni: identificador único del usuario (no se puede modificar una vez registrado)
 - correo
 - password
 - rol: cliente/administrador

- ✓ entradas: contiene:
 - id: identificador de la entrada
 - user_dni: dni del usuario que ha comprado la entrada
 - id_teatro
 - sesion
 - fila
 - asiento
 - dia: fecha de compra de la entrada

fila y *asiento* son los parámetros necesarios para recorrer la matriz *ixj* que forma el patio de butacas.

- ✓ valoraciones:
 - valoración: recoge la valoración, en estrellas, de la obra.
 - comentarios
 - id_teatro
 - nombre_obra
 - fecha
 - user_dni

- ✓ contTeatros: esta colección lleva el contador de los teatros registrados para poder autoincrementar su identificador:
 - name: parámetro utilizado para poder realizar las consultas *find()*
 - num_Teatros: almacena el resultado de la consulta, que es el siguiente índice de teatros a usar

- ✓ contEntradas: similar a la anterior, lleva el contador de las entradas compradas para poder incrementar automáticamente su id:
 - name: encargado de la búsqueda *find()*
 - num_Entradas: almacena el resultado de la consulta, que es el siguiente índice de entradas que se va a usar

Ejemplos de cada agregado

- ✓ usuarios:
 - `db.usuarios.insert({"usuario":"esteban","nombre":"Esteban","apellidos":"Vargas","dni":"123789456","correo":"esteban.mail","password":"esteban","rol":"user"})`
- ✓ teatros:
 - `db.teatros.insert({"id":"1","nombre_teatro":"Canal","nombre_obra":"Miles Gloriosus","descripcion":"La famosa obra de Plauto","sesion1":"16:00","sesion2":"19:00","sesion3":"22:00","nume_filas":"10","nume_asientos":"10"})`
- ✓ entradas:
 - `db.entradas.insert({"id":"1","user_dni":"123789456","Id_teatro":"1","sesion":"16:00","fila":"1","asiento":"3","dia":"2014-5-5"})`
- ✓ valoraciones
 - `db.valoraciones.insert({"valoracion":"2","comentario":"Mediocre","id_teatro":"1","nombre_obra":"Miles Gloriosus","fecha":"2014-5-5","user_dni":"123789456"})`
- ✓ contEntradas
 - `db.contEntradas.insert({"name":"contE","num_Entradas":"1"})`
- ✓ contTeatros
 - `db.contTeatros.insert({"name":"contT","num_Teatros":"1"})`

Conclusión y razonamiento

En cuanto a porqué elegimos estos agregados y no otros, primero pensamos que era intuitivo y eficiente distinguir claramente entre usuarios, teatros y entradas, ya que incluyen documentos que no tienen relación directa entre sí (el DNI de un usuario no tiene nada que ver con el identificador de un teatro, por ejemplo).

Posteriormente vimos que no era suficiente con estas colecciones, y al adaptar el código de la práctica anterior en SQL a Mongo, decidimos que era más sencillo y cómodo para nosotros que los identificadores se incrementaran sin necesidad de tener que hacerlo manualmente cada vez.

Índices

Hemos implementado los índices en las colecciones de teatros y usuarios.

El motivo es que la mayor parte de las búsquedas las realizamos mediante el dni de los usuarios y el id de los teatros, por lo que así conseguimos una mayor eficiencia en las mismas.

Los índices creados son:

- `db.usuarios.ensureIndex({dni:1})`
- `db.teatros.ensureIndex({id:1})`