

## RedES: Cache, Sesiones y HelpDesk

Siguiendo el desarrollo de la red social se procederá con la fase de diseño e implementación de cache, sesiones y HelpDesk. Toda la funcionalidad nueva desarrollada en esta fase hará uso de la base de datos Redis. Utilizar una base de datos Redis diferente para cache y otra para sesiones y HelpDesk.

### Cache

Se deberá actualizar la clase Model implementada en la fase I para incluir la funcionalidad de cache en los métodos que corresponda. Cualquier documento almacenado en la BD Mongo que sea accedido a través de los modelos deberá ser almacenado en cache por un tiempo máximo de 24 horas. Este plazo deberá ser actualizado cada vez que se acceda o actualice dicha entrada. Si un objeto se encuentra almacenado en cache, este deberá ser consultado a través de la cache y no de la base de datos MongoDB.

En cualquier caso, el espacio en memoria principal utilizado por la base de datos no deberá superar los 150MB. En caso de que se supere dicho uso, se deberá eliminar de forma automática aquellos datos almacenados en cache cuyo tiempo restante en cache sea menor.

Para simplificar la práctica, se creará un nuevo método de consulta llamado `find_by_id` que solo se podrá realizar consultas por la id de la persona y utilizará el método `findOne` para mongoDB. Esa misma id será la utilizada para almacenar el documento en cache.

### Sesiones

Diseñar e implementar las funciones necesarias (solo API) para el control de acceso al sistema mediante usuario y contraseña. Los datos asociados al control de acceso serán almacenados en base de datos. Se almacenará de cada usuario su nombre completo, nombre de usuario, contraseña (puede ser guardado en plano), privilegios (un valor entero positivo aleatorio) y un token de sesión en caso de tener una sesión activa actualmente. Las sesiones tendrán una duración de 1 mes. Una vez pasado ese mes, la sesión debe ser eliminada automáticamente y será necesario generar un token nuevo la siguiente vez que se haga *log in*.

Se deberá proporcionar:

- un método de acceso y otro de actualización de los datos del usuario;
- un método para hacer *log in* y generar una nueva sesión. Devolverá el valor del campo privilegios junto al token asignado en caso de *log in* satisfactorio y -1 en caso contrario;
- un método para hacer *log in* mediante un token de sesión Devolverá el valor del campo privilegios asignados en caso de *log in* satisfactorio y -1 en caso contrario;

## HelpDesk

Diseñar e implementar las funciones necesarias (solo API) para la gestión de peticiones de ayuda de los usuarios de la red social. Esta API debe incluir dos funciones:

- Función de petición de ayuda con prioridad: esta función debe registrar una petición de ayuda de un usuario determinado (guardar la id del usuario) junto con un valor que determine la prioridad de esta petición, siendo los valores pequeños peticiones de baja prioridad y los valores grandes, peticiones de alta prioridad.
- Función de atención a usuarios: esta función debe obtener la petición de mayor prioridad y borrarla del registro de peticiones. Se devolverá el id de usuario almacenado. En caso de no existir peticiones pendientes, deberá quedarse en espera hasta que llegue una nueva petición.

## Requisitos de la práctica

- Cache (4 puntos)
- Sesiones (3 puntos)
- Empaquetado (3 puntos)

## Normativa de realización, entrega y evaluación de la práctica:

- La práctica se realizará y entregará en grupos de hasta dos integrantes.
- La práctica se realizará en python y haciendo uso de Redis.
- La práctica deberá ir acompañada de las pruebas necesarias para comprobar el buen funcionamiento de la funcionalidad que se pide. Estas comprobaciones se realizarán en el main de ODM.py
- La entrega se compondrá de un único fichero ZIP, que contendrá el directorio del proyecto con los archivos ODM.py, models.yml y un archivo nombre\_colleccion.json por cada colección en la base de datos que contengan un volcado de la base de datos generada.
- El directorio y fichero zip será renombrado con el número de la práctica P#, seguido del número del grupo G#, y finalmente seguido con el nombre y el primer apellido de los alumnos integrantes del grupo, separados mediante guiones bajos '\_'

Ejemplo: *P2\_G25\_Quijote\_de\_la\_Mancha-Sancho\_Panza.zip*

- Se considerará suspensa toda práctica cuyo fichero comprimido no contenga los ficheros fuente.
- Las prácticas entregadas fuera de plazo serán calificadas sobre 9. Por cada día de retraso en la entrega se reducirá el rango de calificación en 0,2 puntos.
- Cualquier sospecha de COPIA entre dos o más prácticas o de código obtenido en internet derivará en la calificación de 0 para todos los alumnos involucrados en la evaluación en curso y la siguiente. En caso de que el alumno tenga duda de que el código pueda ser susceptible de ser entendido como copia, consultar con el profesor antes de la entrega.