

**Prof. Lorena Zúñiga**

**Proyecto #2. Valor 12.50% de la nota final**

**II semestre 2025**

## **Bases de datos NoSQL**

### **Descripción**

Para el segundo proyecto del curso se requiere implementar un sistema sencillo que permita la gestión de una competencia de talentos, donde el público puede votar por sus participantes favoritos. El sistema debe tener dos módulos o perfiles dentro de la misma aplicación.

#### ***Módulo para votación pública***

Es mediante este módulo que el público puede votar por su favorito(a), se debe presentar una lista con los datos de los participantes en el concurso: nombre completo, categoría en la que participa, una fotografía. No se debe mostrar la cantidad de votos acumulados por los concursantes.

Debe existir una forma de seleccionar a la persona concursante y como mínimo un botón para votar.

Un usuario o persona del público puede votar por todos los participantes si así lo desea. Lo único que no está permitido es votar más de una vez por la misma persona concursante, en ningún momento. Por ejemplo: el usuario Juan votó por la concursante Ana, debido a eso ya no podrá registrar más votos para Ana.

#### ***Módulo para gestionar el concurso***

Este módulo lo utilizará la persona que administra el concurso y debe contar con las siguientes funcionalidades:

- Cargar base de datos: mediante esta función se debe cargar desde un archivo json los datos de los participantes en el concurso. En la carga inicial de los datos, todos los participantes tienen la cantidad de votos acumulados en cero.

- Agregar participante: esta función permitirá incluir los datos (nombre completo, categoría, fotografía) de un(a) nuevo(a) participante. La cantidad de votos inicial será cero. Tome en cuenta que los datos de los participantes agregados mediante esta opción deben quedar registrados en la base de datos en MongoDB, y cuando reciban su primer voto deberán quedar registrados en la base de datos en Redis.
- Votación en tiempo real: esta función le permite a la persona que administra ver en pantalla el total de votos registrados (sumatoria de todos los votos), así como el nombre, la foto de cada concursante y la cantidad de votos que tiene. Cada vez que alguien del público vota, se debe actualizar en esta pantalla, en tiempo real, la cantidad de votos del concursante que corresponda, así como el total de votos recibidos.

Adicionalmente, para el perfil del administrador se deben tener las siguientes consultas:

- Top 3 participantes con más votos: muestra en orden descendente por cantidad de votos recibidos el nombre, la categoría y la foto de los tres concursantes con más votos recibidos.
- Votos por categoría: se debe mostrar el total de votos recibidos detallado por categoría de talento. Por ejemplo: baile 100 votos, canto: 95 votos, etc.
- Participantes sin votos: el nombre, la categoría y la foto de los concursantes que aún tienen cero votos.

Dado que se cuenta con dos tipos de usuario (público y administrador), es necesario implementar una función de log in que habilite sólo las funcionalidades que correspondan según el perfil de ingreso.

## Condiciones del proyecto

- Para este sistema, se requiere el uso de bases de datos de documentos (MongoDB) y bases de datos key-value (Redis). En la base de datos de documentos, como mínimo se deben tener las siguientes colecciones:
  - Una colección con los datos de los concursantes (nombre, categoría, fotografía y total de votos acumulados)
  - Una colección que registre la fecha y la hora en que cada participante recibe un voto. Esto para tener un registro de los votos emitidos con fecha y hora.
  - Una colección que represente los usuarios del sistema, ya que se requiere un perfil para la persona que administra y otro para el público que vota.
- En cuanto a la base de datos key-value, esta es la que almacenará la información de los votos totales por concursante que se deben mostrar en el panel o módulo del administrador.
- El sistema o aplicación puede ser desarrollado en cualquier lenguaje de programación a elección del equipo de trabajo.
- Una lista inicial de participantes o concursantes se proveerá mediante el archivo concursantes.json. No se proveen las fotografías, sino que estas deben ser buscadas por el equipo de trabajo, ya sea en bancos de imágenes gratuitas o bien generadas mediante algún servicio gratuito de Inteligencia Artificial. Para los concursantes que se añadan mediante la función “Agregar participante”, deben tener una fotografía o imagen genérica.
- La aplicación debe utilizar obligatoriamente bases de datos orientadas a documentos y bases de datos key-value. Específicamente MongoDB y Redis. **Si utiliza otro tipo de base de datos no se revisará el proyecto y automáticamente la nota asignada será cero.**

- Los resultados de las consultas y todos los resultados que se muestren en el módulo del administrador, deberán verse en la aplicación web. Es decir, no se permite presentar datos en formato JSON, csv, ni ningún otro similar. Debe presentarse de una forma que sea amigable con el usuario.
- Si los resultados de las consultas no se muestran en la interfaz de la aplicación, se perderá el 50% del puntaje de la(s) consulta(s) que presente el problema mencionado.
- No se permite agregar otras funcionalidades distintas a las solicitadas.
- Bajo ninguna circunstancia se recibirán proyectos fuera de la fecha/hora indicadas, tampoco se recibirán por correo electrónico ni otro medio que no sea el TECDigital, a menos que la plataforma presente un problema de funcionamiento comprobable.
- Todo el proyecto debe ser 100% original del grupo de trabajo.
- Toda copia o plagio tratará según lo establecido en el Reglamento de Enseñanza Aprendizaje.
- Para la revisión, el grupo de trabajo deberá contar con los equipos y software necesarios para poder mostrar el sistema en funcionamiento.
- Posteriormente la profesora publicará las citas de revisión en las fechas y horas disponibles, a fin de que los grupos de trabajo se registren.

- Todos los miembros del grupo de trabajo deberán estar presentes en la revisión, quien no asista perderá por completo la nota del proyecto.
- Se habilitará un foro para plantear dudas generales del proyecto. Si tiene una duda general, debe enviarla por ese medio. El foro cierra dos días antes de la entrega del proyecto.

**Por entregar:**

- Documentación:
  - Pruebas de funcionalidad

Pruebas de funcionalidad de todos y cada uno de los requerimientos descritos en este enunciado. Todo lo anterior ejecutado desde la aplicación web. Debe describir en qué consiste la prueba y agregar screenshots para demostrar la funcionalidad. Para las pruebas de las consultas, los resultados deben verse mediante la interfaz de la aplicación web (no consultas directas a la base de datos). Para las pruebas que impliquen inserción o modificación de los datos originales, deben tener evidencia de registro y/o modificación en la base de datos.

**Importante:** Toda funcionalidad que no quede debidamente documentada en la sección de pruebas de funcionalidad quedará automáticamente excluida de la revisión y el grupo de trabajo perderá los puntos del rubro correspondiente.

El archivo con la documentación debe tener como nombre el nombre de los integrantes del grupo. Se penalizará con 5 puntos el incumplimiento de este punto.

**Fecha de entrega:** a más tardar el 21 de noviembre de 2025, a las 23:00, vía TECDigital.

**Forma de trabajo:** grupos previamente definidos por la profesora.

## Evaluación

Rubro	Puntaje
<b>Aspectos generales</b>	<b>25</b>
Colecciones mínimas requeridas en MongoDB	10
Uso correcto de base de datos key-value	10
Manejo de perfiles de usuario (público y admin.)	5
<b>Funciones para votación</b>	<b>18</b>
Presentación la lista completa de concursantes, incluyendo fotografía	8
Registro de votos correcto (tanto en MongoDB como en Redis)	10
<b>Funcionalidades para los(as) administradores(as)</b>	<b>50</b>
Carga inicial de la base de datos	5
Agregar participante nuevo	5
Votación en tiempo real	10
Consulta top 3 participantes	10
Consulta de votos por categoría	10
Consulta de participantes sin voto	10
<b>Pruebas de funcionalidad</b>	<b>1</b>
<b>Interfaz gráfica (ordenada, clara)</b>	<b>6</b>
<b>Total</b>	<b>100</b>