

# Bases de Datos 2024

## Parcial II: NoSQL

Juan Cabral, Sergio Canchi, Tadeo Cocucci,  
Ramiro Demasi, Guido Ivetta

### Contexto

La base de datos **university** contiene la siguiente colección:

Colección	Descripción
grades	Contiene información sobre las calificaciones de los alumnos en una clase determinada, incluidas las puntuaciones obtenidas en diferentes evaluaciones.

Para cargar la base de datos **university** ejecutar los siguientes comandos:

**Instructivo para los que usan sus computadoras personales:**

```
$ $ tar xzvf parcial2.tar.gz
$ $ mongorestore --host <host> --drop --gzip --db university university/
```

\$

**Instructivo para los que usan las computadoras del laboratorio:**

```
$ tar xzvf parcial2.tar.gz
$ docker run --name mongo-parcial2
    -p 27017:27017 -v mongod.conf:/etc/mongod.conf
    -v ./university:/data/university -d mongo:8
$ docker exec -it mongo-parcial2 mongorestore
    --drop --gzip --db university /data/university/
// (Opcional) para ejecutar el cliente
$ mongoshdocker exec -it mongo-parcial2 mongosh
```

# Consignas

1. Buscar los documentos donde el alumno tiene:
  - (i) un puntaje mayor o igual a 80 en "exam" o bien un puntaje mayor o igual a 90 en "quiz" y
  - (ii) un puntaje mayor o igual a 60 en todos los "homework" (en otras palabras no tiene un puntaje menor a 60 en algun "homework")Se debe mostrar todos los campos excepto el `_id`, ordenados por el id de la clase y id del alumno en orden descendente y ascendente respectivamente..
2. Calcular el puntaje mínimo, promedio, y máximo que obtuvo el alumno en las clases 20, 220, 420. El resultado debe mostrar además el id de la clase y el id del alumno, ordenados por alumno y clase en orden ascendentes.
3. Para cada clase listar el puntaje máximo de las evaluaciones de tipo "exam" y el puntaje máximo de las evaluaciones de tipo "quiz". Listar en orden ascendente por el id de la clase. HINT: El [operador \\$filter](#) puede ser de utilidad.
4. Crear una vista "top10students" que liste los 10 estudiantes con los mejores promedios.
5. Actualizar los documentos de la clase 339, agregando dos nuevos campos: el campo "score\_avg" que almacena el puntaje promedio y el campo "letter" que tiene el valor "NA" si el puntaje promedio está entre [0, 60), el valor "A" si el puntaje promedio está entre [60, 80) y el valor "P" si el puntaje promedio está entre [80, 100]. HINTS: (i) para actualizar se puede usar pipeline de agregación. (ii) El operador `$cond` o `$switch` pueden ser de utilidad.
6. (a) Especificar reglas de validación en la colección *grades* para todos sus campos y subdocumentos anidados. Inferir los tipos y otras restricciones que considere adecuados para especificar las reglas a partir de los documentos de la colección.  
(b) Testear la regla de validación generando dos casos de fallas en la regla de validación y un caso de éxito en la regla de validación. Aclarar en la entrega cuales son los casos y por qué fallan y cuales cumplen la regla de validación. Los casos no deben ser triviales, es decir los ejemplos deben contener todos los campos..

## Puntos a tener en cuenta

- Algunas consultas pueden resolverse con `find` y `aggregate`. En estos casos se puede resolver eligiendo alguno de estos métodos.
- Resolver las consultas sin vistas salvo que se lo pida explícitamente.
- Mostrar únicamente los campos pedidos en la consigna.
- Se piden que los campos que se devuelven sean valores escalares a menos que se pida los valores de los campos devueltos podrán ser documentos anidados, arreglos de escalares o arreglos de documentos.
- Buscar hacer la consulta de la forma más sencilla y eficiente posible.
- Se evaluará el correcto formato de las soluciones:
  - El código entregado debe ser legible.
  - Utilizar indentación de espacios de manera uniforme.

### Detalle Importante

El alumno puede usar sus notas personales e internet, pero queda prohibido utilizar modelos generativos (Chat-GPT, LLAMA, etc) y comunicarse con otras personas oral o digitalmente.

Romper alguna de estas dos restricciones invalida el examen

## Entrega

- Se entregará un archivo **`soluciones.js`** con las soluciones de los 6 ejercicios. Separar las soluciones mediante comentarios de Javascript (`/* */` o `//`).
- La entrega se hará mediante el [Aula Virtual](#) en el [correspondiente apartado](#).
  - Tendrán hasta las 17:00 para que se considere una entrega completa. La recomendación es empezar a subir el archivo a las 16:30 para evitar cualquier eventualidad.