
 Universidad de los Andes Facultad de Ingeniería	Ingeniería de Sistemas y Computación Pregrado ISIS2304 – Sistemas Transaccionales	
---	--	---

PROYECTO DEL CURSO – ENTREGA 3 DEL PROYECTO

OBJETIVOS

- Desarrollar habilidades en el proceso de generación de un modelo de datos en bases de datos documentales.
- Incorporar la validación del esquema del modelo de datos en MongoDB.
- Implementar una aplicación informática de mediana complejidad que involucre bases de datos documentales en MongoDB.

HERRAMIENTAS

- Para la documentación del modelo conceptual pueden hacer un modelo UML o E/R. Para ello pueden usar las herramientas vistas en clase en las entregas anteriores.
- MongoDB Community Edition 7.0 disponible en la infraestructura computacional del Departamento (cada estudiante tiene una cuenta). También se puede utilizar la opción MongoDB Atlas, sin embargo, en este último caso cada grupo es responsable de crear las cuentas y configurar el acceso de los usuarios al clúster en el cual se cree la base de datos.
- MongoDB Compass. Puede seguir las instrucciones de descarga [aquí](#).
- Debe usar el IDE y framework de desarrollo son los utilizados en las entregas anteriores.

CASO DE ESTUDIO

La iteración se desarrolla utilizando el caso **Hotel de los Andes**, que se encuentra en Bloque Neón (BN) - Contenidos.

ACTIVIDADES A DESARROLLAR: DISEÑO Y CONSTRUCCIÓN DE LA APLICACIÓN

1. Revise el caso de estudio propuesto. Identifique los elementos fundamentales que hacen parte del negocio que se describe.
2. **(7%)** Análisis y modelo conceptual
 - a. Proponga un modelo conceptual en UML o E/R que describa las entidades del modelo de datos para la aplicación que se quiere desarrollar.
3. **(35%)** Diseño de la base de datos
 - a. **(10%)** Análisis de la carga de trabajo (workload). Para ello, presenten lo siguiente:
 - a. Identifiquen entidades y sus atributos
 - b. Cuantifiquen las entidades (cantidad de registros que tendría la BD para cada una de las entidades, pueden encontrar un aproximado en el enunciado).
 - c. Analicen las operaciones de lectura y escritura para cada entidad. Para ello utilicen una tabla como la del ejemplo del anexo A. Recuerden que este análisis sirve para saber que información se accederá de manera conjunta.
 - d. Cuantifiquen las operaciones de lectura y escritura para cada entidad. Para ello utilicen una tabla como la del ejemplo del anexo B.
 - b. **(15%)** Describan las entidades de datos y las relaciones entre ellas que corresponden al modelo conceptual UML propuesto. Para ello, presenten lo siguiente:
 - a. La lista de entidades con la descripción de cada una de ellas.
 - b. Las relaciones entre entidades y su cardinalidad (uno a uno, uno a muchos, o muchos a muchos).

- c. El análisis de selección de esquema de asociación (referenciado o embebido) para cada relación entre entidades. Para ello use la tabla de análisis vista en clase, la cual se retoma en el anexo C, junto con los resultados del análisis de la carga de trabajo (workload), descrita antes.
 - d. Una descripción gráfica usando Json de cada relación entre entidades en donde presente un ejemplo de datos junto con el esquema de asociación usado (referenciado o embebido). En el anexo D se muestra un ejemplo de lo que se requiere.
 - c. (10%) Cree en MongoDB las colecciones principales de su base, así como la validación de los esquemas. Puede usar Compass o Mongo Shell (Mongosh) para realizar este proceso. Guarde lo hecho en un archivo. Anexe a los entregables los archivos con los scripts utilizados.
4. **(58%)¹** Implemente los 7 requerimientos funcionales de CRUD (RF1 a RF7) y los 4 requerimientos funcionales de consulta (RFC1 a RFC3 + 1 RFC avanzado que escojan) solicitados en el documento marco, a través de una interfaz gráfica desde la aplicación. Evaluación detallada:
- a. 6% para cada requerimiento funcional de modificación (total RF 42%)
 - b. 4% para cada requerimiento funcional de consulta (total RFC 16%).
- Para la implementación pueden, si lo desean, reutilizar las clases que corresponden a la lógica y la conexión a la base de datos usadas en entregas pasadas, siempre y cuando se hagan las adaptaciones pertinentes para la implementación de los requerimientos de esta entrega.
5. **(5%)** Escenarios de prueba: realice las operaciones descritas abajo (ver escenarios de prueba) para poder realizar pruebas sobre los RF de CRUD (RF1-RF7) así como sobre los RF de Consulta (RFC1 a RFC3 + 1 RFC de consulta avanzado seleccionado) del documento marco del caso de estudio . Anexe los scripts SQL cuando sea necesario.

ESCENARIOS DE PRUEBA

Diseñe y desarrolle los escenarios de prueba que le permiten asegurar el correcto funcionamiento de la aplicación.

1. Para los RF CRUD (RF1-RF7), utilicen la interfaz gráfica de la aplicación desarrollada para crear, actualizar, borrar y presentar datos, que permitan probar cada uno de los RF del documento marco. No es necesario un script para esta parte.
2. Para los RF CRUD (RF1-RF7), para cada una de las entidades, trate de insertar a través de MongoDB Shell una tupla que no cumpla con su esquema de validación. Anexe el script a la entrega.
3. A través de la interfaz gráfica de la aplicación o de un script inserte suficientes datos para cada una de las entidad, de tal manera que las consultas de los RF de Consulta (RFC1 a RFC3 + 1 RFC de consulta avanzado seleccionado) puedan ser probadas. En caso de haber usado un script, anéxelo a la entrega.

ENTREGA

Fecha y hora límite: **3 de diciembre de 2023**

- En la actividad correspondiente en BN, envíe la URL del *commit* de la entrega en el repositorio GitHub asignado a su grupo.
- Contenido de la entrega (*commit* de GitHub):
 - Documento de informe, debe estar en formato *.pdf*. Debe incluir los modelos, tablas y diagramas solicitados, resultados logrados, resultados no logrados, balance del plan de pruebas. El nombre del archivo debe seguir el mismo formato de nombramiento del zip.
 - Proyecto Java con el código fuente y la documentación del proyecto de software.
 - La documentación del proyecto debe quedar en una carpeta *docs*, en la estructura principal de directorios del proyecto java.
 - En la carpeta *docs* del proyecto Java deben estar:
 - Los modelos desarrollados, tanto en el formato fuente como en *pdf*.

¹ Note que el porcentaje total es 105%

- La documentación de los requerimientos funcionales.
- Los archivos de scripts, sentencias y pruebas indicados en el enunciado (.sql).
- Archivo `readme.txt`, con todos los pasos a seguir para instalar la aplicación.
- El no seguimiento de los lineamientos de formato de entrega tiene una penalización de 0.5/5.0 puntos en la nota, por cada elemento faltante.

DOCUMENTACIÓN

1. **Cluster gratuito** en MongoDB Atlas: Puede seguir las instrucciones en el [siguiente enlace](#).
2. Descargar **MongoDB Compass** localmente en su computador. Puede seguir las instrucciones de descarga [aquí](#).
3. Creación de colecciones y esquemas de validación [manual de MongoDB](#).
4. Mongo University incluyendo,
 - Curso de diseño de BD, [Data Modeling for MongoDB](#)
 - Curso CRUD: [MongoDB CRUD Operations: MongoDB CRUD Operations: Insert and Find Documents](#).

ANEXO A – TABLA DE ANÁLISIS DE OPERACIONES DE LECTURA Y ESCRITURA

Entities	Operations	Information Needed	Type
Books*	Fetch book details	Book details + rating	Read
Reviews	Fetch 10 reviews for a book	Reviews + reviewer rating	Read
Users	Fetch user details	User details	Read
Authors, Books*	Fetch an author and their books	Book titles + author details	Read
Books*	Add/update book	Book details + stock level	Write
Printed Books	Sell copy of printed book	Stock level	Write
Users	Add/update user	User details	Write
Reviews	Add review	Review + book rating	Write
Printed Books	Fetch printed book titles where the stock level has fallen below 50	Book details + stock level	Read

ANEXO B – TABLA DE CUANTIFICACIÓN DE OPERACIONES DE LECTURA Y ESCRITURA

Entities	Operation	Information Needed	Type	Rate
Books*	Fetch book details	Book details + rating	Read	1000/sec
Authors, Books*	Fetch an author and their books	Book titles + author details	Read	50/sec
Print Books	Fetch printed book titles where the stock level has fallen below 50	Book details + stock level	Read	2/day
Books*	Add/update book	Book details + stock level	Write	10/hour
Print Books	Sell copy of printed book	Stock level	Write	5/sec
Reviews	Fetch 10 reviews for a book	Reviews + reviewer rating	Read	200/sec
Reviews	Add review	Review + book rating	Write	50/sec
Users	Fetch user details	User details	Read	5/min
Users	Add/update user	User details	Write	1/sec

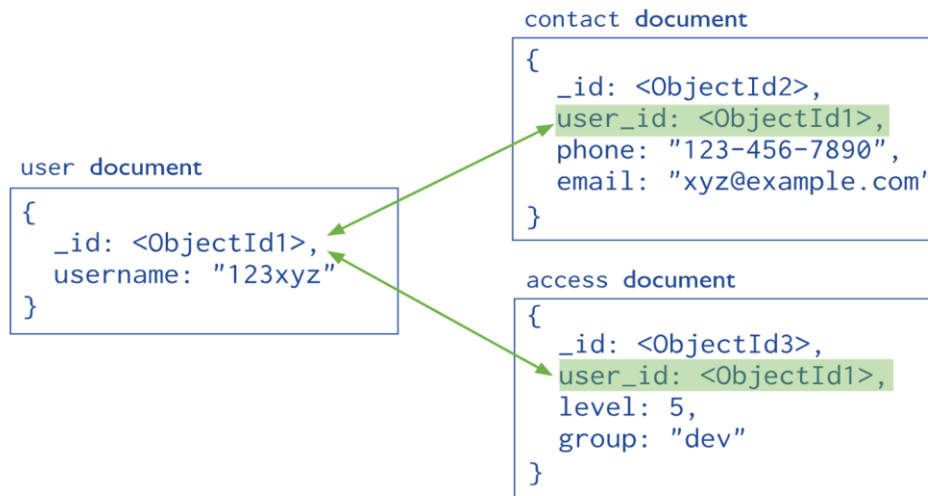
*eBooks, Audiobooks, and Printed Books

ANEXO C – TABLA DE ANÁLISIS DE ESQUEMA DE RELACIÓN ENTRE ENTIDADES

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces of information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information have a "has-a," "contains," or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	Yes	No
Archival	Should the pieces of information be archived at the same time?	Yes	No
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	No	Yes
Data Duplication	Would data duplication be too complicated to manage and undesired?	No	Yes
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	No	Yes
Document Growth	Would the embedded piece grow without bound?	No	Yes
Workload	Are the pieces of information written at different times in a write-heavy workload?	No	Yes
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

ANEXO D – EJEMPLOS GRÁFICOS DE ESQUEMAS EMBEBIDOS Y REFERENCIADOS

Ejemplo de esquema embebido:



Ejemplo de esquema referenciado:

```
{
  "_id": "book0003",
  "title": "MongoDB Data Modeling and Schema
Design",
  "authors": [
    "author0029",
    "author0073",
    "author0045"
  ]
}
```

```
[
  {
    "author_id": "author0029",
    "name": "Daniel Coupal"
  },
  {
    "author_id": "author0073",
    "name": "Pascal Desmarests"
  },
  {
    "author_id": "author0045",
    "name": "Steve Hoberman"
  }
]
```