


Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

1 Enunciado. Gestión de Inventario de Bibliotecas Públicas

Una red de bibliotecas públicas gestiona una gran variedad de libros y recursos digitales distribuidos en varias sucursales de una ciudad. Cada sucursal tiene un catálogo específico de libros y recursos, pero también permite que los usuarios realicen reservas y traslados de libros entre sucursales.

El sistema actual almacena información clave sobre cada libro, como su título, autor, ISBN (un identificador único), género y la fecha de adquisición. También se registran detalles sobre el estado actual de cada ejemplar (disponible, prestado, reservado o en mantenimiento), y cuántos ejemplares están disponibles en cada sucursal. En caso de tratarse de un recurso digital como eBooks o audiolibros, se deben registrar también detalles como el formato del archivo y un enlace para la descarga.

Además, los usuarios pueden reservar libros en línea y solicitar su traslado entre sucursales si el ejemplar deseado no se encuentra disponible en la biblioteca local. Los préstamos de estos recursos se registran con información sobre la fecha de préstamo y devolución, así como un historial de multas o penalizaciones en caso de retrasos en la devolución.

El sistema de usuarios también almacena el correo electrónico de cada persona registrada y sus preferencias de lectura, como los géneros que prefieren o la frecuencia de préstamos. Este historial permite a la biblioteca ofrecer recomendaciones personalizadas basadas en los intereses y hábitos de lectura del usuario.

Sin embargo, el sistema actual está limitado en cuanto a la gestión de grandes volúmenes de datos y en la eficiencia para realizar búsquedas complejas o gestionar inventarios a gran escala. Para solucionar esto, se ha decidido migrar a un sistema basado en una base de datos documental como mongoDB.

Se planea también implementar un nuevo sistema de gestión que permita ofrecer recomendaciones personalizadas, controlar mejor las reservas y traslados entre sucursales, y optimizar el manejo de inventario y multas que responden a los siguientes casos de uso.


Casos de Uso

1) Listado de libros disponibles en una sucursal

Este caso de uso tiene como objetivo proporcionar un listado completo de los libros disponibles en una sucursal específica, junto con detalles sobre su estado actual y la posibilidad de reserva.

Requisitos:

- Permitir la búsqueda de libros en una sucursal por título, autor o género.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

- Mostrar la información básica del libro (título, autor, ISBN, estado).
- Mostrar la cantidad de ejemplares disponibles y si se pueden reservar.
- Verificar que el ISBN sea único y el estado del libro solo pueda ser "Disponible", "Prestado", "Reservado" o "En Mantenimiento".

2) Gestión de reservas y traslados de libros entre sucursales

Este caso de uso busca optimizar la reserva de libros por parte de los usuarios, permitiendo el traslado de libros entre sucursales si no están disponibles en una sucursal determinada.

Requisitos:

- Permitir a los usuarios reservar libros en línea.
- Gestionar los traslados de libros entre sucursales, actualizando el estado de la reserva.
- Mostrar la disponibilidad de libros en diferentes sucursales para sugerir opciones alternativas.
- Verificar que el correo del usuario sea válido y que el libro reservado cumpla con los criterios de estado.

3) Gestión de recursos digitales

Este caso de uso se enfoca en la gestión de recursos digitales (eBooks, audiolibros), permitiendo su préstamo y seguimiento.

Requisitos:


- Permitir a los usuarios acceder y descargar recursos digitales con un límite de tiempo.
- Registrar las descargas y préstamos electrónicos de los usuarios.
- Ofrecer recomendaciones personalizadas de recursos digitales basadas en el historial de préstamos del usuario.
- En los casos de libros digitales, asegurar que se registre correctamente el formato del recurso y el enlace de descarga.

4) Gestión de multas y penalizaciones

Este caso de uso se enfoca en la gestión de multas por devoluciones tardías y el seguimiento de penalizaciones a los usuarios.

Requisitos:

- Registrar las multas generadas por retrasos en la devolución de libros.
- Notificar a los usuarios sobre las multas pendientes.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

- Permitir el pago de multas a través de la plataforma y llevar un seguimiento del historial de penalizaciones de cada usuario.
- Las multas deben ser valores positivos y su estado solo puede ser "Pagada" o "Pendiente".
- Asegurar que las fechas de préstamo y devolución sean correctas (la fecha de devolución debe ser mayor que la de préstamo).

5) Análisis del comportamiento de los usuarios

Este caso de uso tiene como objetivo analizar el comportamiento de los usuarios para ofrecer recomendaciones personalizadas de libros y otros recursos.

Requisitos:

- Recoger datos sobre los libros prestados, géneros favoritos y frecuencia de uso por cada usuario.
- Generar recomendaciones de libros o recursos digitales en función de las preferencias y hábitos de lectura de los usuarios.
- Evaluar los patrones de lectura para optimizar las compras y el inventario de libros.


2 Apartados a resolver

Realiza un modelado de agregados para implementar este dominio en MongoDB.

2a) Diseño conceptual (diagrama de clases de UML) siguiendo los casos de uso.

2b) Diseño de agregados indicando de cada agregado la raíz y perímetro.

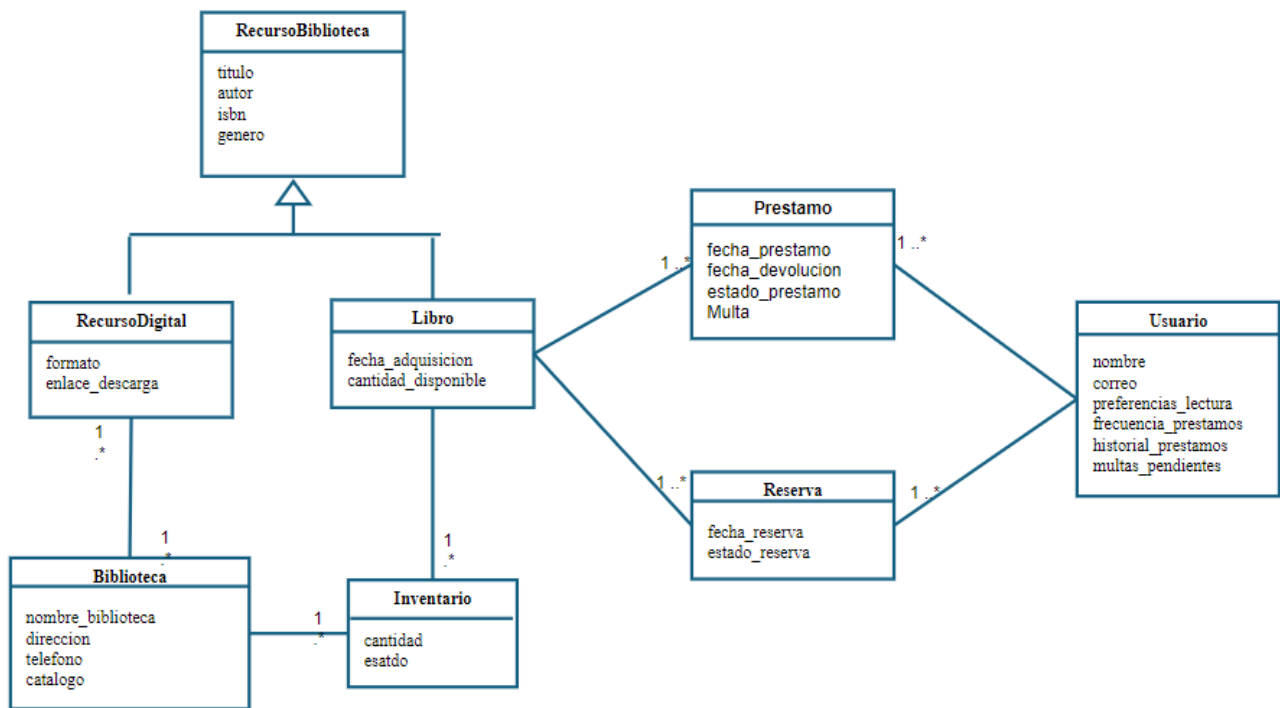
2c) Propón una estrategia de validación de esquema para las colecciones del sistema, asegurando que los datos cumplan con los requisitos definidos en los casos de uso y se haga un control de la integridad y consistencia de los datos.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

SOLUCIÓN:

SOLUCIÓN 2a) Diseño conceptual (diagrama de clases de UML) siguiendo los casos de uso.


Propuesta de diseño del sistema de partida sin tener en cuenta los casos de Uso (Modelado1)



En relación con el Caso de Uso (1), no es necesario introducir nuevas entidades ni atributos en el modelado actual. Todos los requisitos del Caso de Uso 1 están cubiertos con los elementos presentes en el Modelado1: la clase Libro con sus atributos, la clase Inventario que gestiona el estado y la cantidad de ejemplares, y las relaciones ya definidas entre las clases.

Se recomienda establecer índices en los campos título, autor y género para realizar búsquedas eficientes.

En relación con el Caso de Uso (2), de gestión de reservas y traslados de libros entre sucursales de las bibliotecas, para el Req1 ya existe una relación entre la clase Usuario y la clase Reserva, lo que permite gestionar las reservas de libros por parte de los usuarios. También, la clase Reserva está asociada a Libro, lo que permite reservar un libro específico.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	
--	---

Para el Req2, que se refiere a gestionar los traslados de libros entre sucursales actualizando el estado de la reserva, el Modelado1 ya contempla que un Libro está relacionado con múltiples bibliotecas a través de Inventario. Sin embargo, no se ha especificado el manejo del traslado entre bibliotecas. Podríamos agregar un atributo en Reserva para indicar si el libro ha sido trasladado de una biblioteca a otra, o un atributo de estado que registre si está en proceso de traslado.

Para el Req3, que consiste en poder mostrar la disponibilidad de libros en diferentes sucursales para sugerir alternativas, Inventario ya permite rastrear la cantidad de ejemplares disponibles por biblioteca.


Por último, para el Req4, de verificar que el correo del usuario sea válido y que el libro reservado cumpla con los criterios de estado, en el Modelado1, la clase Usuario ya tiene un atributo correo, y se puede extender la validación de este atributo (verificación de formato) en la lógica de la aplicación. En cuanto al estado del libro, la clase Inventario ya maneja el estado del libro (Disponible, Reservado, Prestado, etc.), lo cual cubre este requisito.

Sería bueno implementar índices en isbn, estado, biblioteca_id y correo para optimizar búsquedas y garantizar unicidad.

El Caso de Uso (3) indica que se van a querer gestionar Recursos Digitales, la información clave que se debe abordar son los préstamos de recursos digitales y el registro de información como el formato y el enlace de descarga. Para el Req1, los usuarios deben poder acceder a los recursos digitales (como eBooks o audiolibros) por un período de tiempo limitado. Esto implica que el sistema controle el tiempo durante el cual el usuario puede acceder al recurso antes de que expire su préstamo. Para cubrir este requisito, hay que extender el modelado. Se podría tener un PréstamoDigital (subclase de Préstamo) con un atributo tiempo limite, que indique la cantidad de días que un usuario puede tener el recurso digital descargado, y con el atributo fecha descarga. De esta manera, Usuario estará relacionado con PréstamoDigital, y PréstamoDigital con RecursoDigital.

Para cubrir el Req2, el sistema debe almacenar un historial de descargas para cada usuario, asociando a cada descarga el recurso digital correspondiente y el período de préstamo. Con la extensión anterior del Req1, ya lo tendremos recogido.

Para cubrir el Req3, el sistema debe ofrecer recomendaciones de recursos digitales, tales como eBooks o audiolibros, basadas en los géneros o tipos de recursos que el usuario ha descargado anteriormente. Con los atributos historial_prestamos y preferencias_lectura en Usuario, sirve para definir los géneros o tipos de recursos preferidos, y se puede relacionar indirectamente con los géneros de los RecursosDigitales. Así, basado en el análisis de los PréstamosDigitales previos, se pueden ofrecer recomendaciones de recursos del mismo género.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	
--	---

Por último, para cubrir el Req4, cada recurso digital debe tener su formato registrado (por ejemplo, PDF, MP3) junto con un enlace de descarga para que los usuarios puedan acceder al archivo digital. Esa información se tiene recogida.

Se sugieren índices sobre el correo del usuario, las fechas de préstamo y el formato de los recursos digitales para acelerar consultas relacionadas con préstamos, descargas y recomendaciones personalizadas.

En relación al Caso de Uso (4), de gestionar las multas y penalizaciones, para el Req1 de permitir registrar las multas generadas por retrasos en la devolución de libros, como Usuario ya tiene el atributo multas_pendientes, se puede actualizar ese atributo cuando se genera una multa en el Préstamo. El atributo multa en Préstamo es relevante para llevar un registro de cada multa específica, mientras que el atributo multas_pendientes en Usuario puede reflejar la suma total de multas pendientes.

Para el Req2 de poder notificar a los usuarios sobre las multas pendientes, como el Usuario tiene el atributo multas_pendientes, se puede usar ese valor para generar notificaciones automáticas.

Para el Req3 de permitir el pago de multas y llevar un seguimiento, la multa está almacenada en Préstamo, pero sería necesario añadir el atributo estado multa para reflejar si la multa está en estado "Pagada" o "Pendiente". Además, se puede actualizar el valor en multas_pendientes en Usuario cuando una multa sea pagada.


Para el Req4, las multas deben ser valores positivos y su estado solo puede ser "Pagada" o "Pendiente". Esto se puede controlar con la validación de esquema

Por último, para el Req5, que se debe asegurar que las fechas de préstamo y devolución sean correctas, se puede agregar validación de fechas también.

Se recomiendan índices en estado_multa para agilizar la búsqueda de multas pendientes o pagadas, y en fecha_devolucion para identificar devoluciones retrasadas. Un índice compuesto en usuario_id y estado_multa mejorará la eficiencia al consultar multas por usuario.

Por último, el **Caso de Uso (5)** es que se debe poder hacer análisis del comportamiento de los usuarios, y se deben cubrir los siguientes requisitos: para el Req1 de recoger datos sobre los libros prestados, géneros favoritos y frecuencia de uso por cada usuario, ya tenemos en Usuario los atributos historial_prestamos, preferencias_lectura, y frecuencia_prestamos, por lo que no se requiere una extensión.

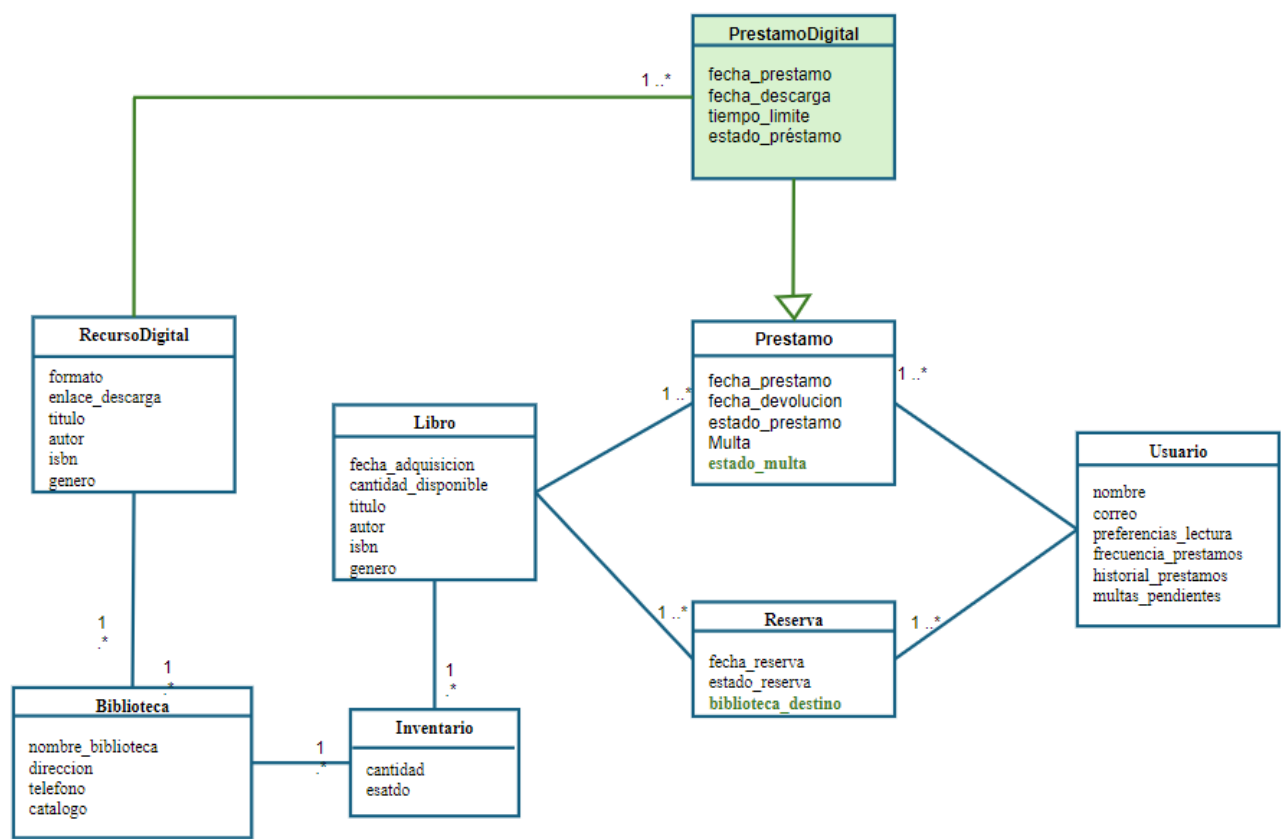
Para el Req2 de poder generar recomendaciones de libros o recursos digitales en función de las preferencias y hábitos de lectura de los usuarios, no es necesario recoger más información, pero puede ser recomendable añadir un atributo recomendaciones en Usuario para almacenar


Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

recomendaciones personalizadas. Aunque es posible derivar recomendaciones utilizando el historial_prestamos y las preferencias_lectura.

Para el Req3 de poder evaluar los patrones de lectura para optimizar las compras y el inventario de libros, no se requiere una modificación en el Modelado1, porque los datos del historial_prestamos y la frecuencia_prestamos en Usuario son suficientes para evaluar patrones y tomar decisiones sobre la compra de nuevos libros.

En este caso, se pueden usar índices en los atributos historial_prestamos y preferencias_lectura para optimizar las consultas relacionadas con los patrones de lectura y generar recomendaciones basadas en géneros y frecuencia de uso. Después de este análisis de la semántica de los casos de uso, se tiene el Modelado(2)



Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

SOLUCIÓN 2b) Diseño de agregados indicando de cada agregado la raíz y perímetro.

Es importante analizar si los casos de uso son solo de lectura, o hay de actualización, en este caso:

- Lectura: CS1, CS5 (principalmente).
- Lectura y actualización: CS2, CS3.
- Actualización: CS4.

En el diseño de agregados para el sistema de bibliotecas, se han definido tres agregados principales, cada uno optimizado para casos de uso específicos y con un equilibrio entre la eficiencia de las lecturas y las modificaciones.

Además, orientado a la implementación, es esencial determinar si las relaciones entre las entidades deben ser embebidas (dentro del mismo documento) o referenciadas (relacionadas entre documentos). El uso de una estrategia u otra depende principalmente de la frecuencia de lecturas y modificaciones, así como del tamaño y crecimiento de los datos.

A continuación, se describen los agregados con sus entidades comprendidas, la raíz del agregado, el perímetro de cada uno, los casos de uso que optimizan, y un análisis sobre cómo se comportan respecto a las lecturas y modificaciones.

1. AGREGADO_LibroInventario

Entidades (Clases) que comprende: Libro, Inventario, Biblioteca

Raíz: Libro


Perímetro: Este agregado se enfoca en gestionar el inventario de libros en las distintas sucursales de la red de bibliotecas. Cada Libro está asociado a una o más instancias de Inventario, que representan la cantidad y disponibilidad de ejemplares en cada Biblioteca.

Casos de Uso optimizados:

- CU1 (Listado de libros disponibles en una sucursal): Este agregado proporciona consultas rápidas para obtener información sobre la disponibilidad y el estado de los libros en distintas sucursales.
- CU2 (Gestión de reservas y traslados de libros entre sucursales): También soporta la gestión de traslados de libros entre sucursales, aunque las modificaciones son más costosas debido a la actualización de múltiples inventarios.

Embeber vs. Referencias:

- Lecturas frecuentes (CU1): Se ha optado por embeber los datos de inventario y disponibilidad directamente dentro del documento del Libro. Esto hace que las lecturas sobre la disponibilidad en sucursales sean rápidas y eficientes, dado que toda la

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	
--	---

información relacionada con un libro y su disponibilidad se encuentra dentro del mismo documento.

- **Modificaciones costosas (CU2):** Dado que un libro puede estar disponible en múltiples sucursales, embeber toda la información de inventario en el documento del libro puede ser costoso cuando se realizan actualizaciones frecuentes (como el traslado entre bibliotecas). En estos casos, se podría considerar referenciar el inventario con documentos separados para las sucursales si el volumen de actualizaciones crece.

2. AGREGADO_Usuario

Entidades (Clases) que comprende: Usuario, Reserva, Préstamo

Raíz: Usuario


Perímetro: Este agregado abarca todo el historial de préstamos, reservas y penalizaciones del usuario. Cada Usuario tiene múltiples Préstamos y Reservas, y estos préstamos incluyen información sobre multa y estado_multa cuando aplicable. Esta estructura permite un acceso rápido y eficiente a la información del usuario, sus préstamos, y cualquier reserva o penalización asociada.

Casos de Uso optimizados:

- **CU2 (Gestión de reservas y traslados de libros entre sucursales):** Gestiona las reservas que los usuarios realizan y permite el seguimiento del estado de estas.
- **CU3 (Gestión de recursos digitales):** Este agregado también permite registrar y gestionar los préstamos digitales asociados a cada usuario.
- **CU4 (Gestión de multas y penalizaciones):** Este agregado facilita el seguimiento de las multas relacionadas con los préstamos de los usuarios.
- **CU5 (Análisis del comportamiento de los usuarios):** Toda la información sobre los préstamos, reservas y multas está centralizada en este agregado, lo que permite realizar análisis detallados del comportamiento de los usuarios.

Embeber vs. Referencias:

- **Lecturas frecuentes (CU3, CU4, CU5):** Se ha decidido embeber el historial de préstamos y reservas dentro del documento del Usuario. Esta decisión permite lecturas eficientes cuando se necesita acceder al historial completo del usuario, incluyendo sus multas y reservas.
- **Modificaciones individuales (CU2, CU4):** En este caso, embeber es aún eficiente, ya que las modificaciones suelen ser a nivel de préstamo o reserva individuales.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	
--	---

Como supuesto, si las modificaciones en los préstamos y multas son relativamente pequeñas comparadas con las lecturas frecuentes sobre el historial del usuario, el embeber sigue siendo la opción preferida. Si, por otro lado, las modificaciones se vuelven frecuentes, se podría considerar referenciar los préstamos y reservas, especialmente si el número de estas crece significativamente con el tiempo.

Observación:

En este agregado, se podría utilizar una referencia con resumen del Libro dentro de la entidad Reserva. El resumen del libro podría incluir atributos como el título, ISBN y estado, en lugar de embeber toda la información del libro. Esto evita la duplicación excesiva y hace más eficiente la gestión de reservas en el AgregadoUsuario.

También, en el historial de préstamos dentro del AgregadoUsuario, en lugar de incluir toda la información del préstamo relacionado con una multa, se podría usar una referencia con resumen que incluya solo el importe de la multa y el estado ("Pagada" o "Pendiente"). Esto mejora la eficiencia y evita la duplicación de toda la información del préstamo.

3. AGREGADO_RecursoDigital

Entidades (Clases) que comprende: RecursoDigital, PréstamoDigital, Usuario

Raíz: RecursoDigital


Perímetro: Este agregado está optimizado para manejar los PréstamosDigitales y las descargas de RecursosDigitales (como eBooks y audiolibros). Cada RecursoDigital tiene múltiples préstamos asociados, y estos préstamos se registran de forma eficiente para el acceso y la gestión de las descargas sin afectar el resto de la información del usuario.

Casos de Uso Optimizados:

- CU3 (Gestión de recursos digitales): Este agregado está diseñado específicamente para manejar los préstamos y descargas de recursos digitales, y es ideal para consultas rápidas sobre el acceso a recursos como eBooks y audiolibros.

Embeber vs. Referencias:


- Lecturas frecuentes (CU3): En este caso, los préstamos de recursos digitales están embebidos dentro del documento de RecursoDigital para asegurar un acceso rápido a los datos relacionados con los préstamos y las descargas de estos recursos.
- Modificaciones independientes (CU3): Dado que las modificaciones sobre préstamos digitales son independientes de los datos generales del usuario, esta estructura de embeber los préstamos directamente en el recurso digital es eficiente.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

Como supuesto, si las lecturas sobre recursos digitales (como las descargas y acceso a libros electrónicos) son más comunes que las modificaciones, el embeber es la opción adecuada, manteniendo la eficiencia en la gestión de préstamos digitales.

Observación: se habría podido incluir un resumen del Usuario (con solo su nombre y correo electrónico) dentro de este agregado. Esto ayuda a no duplicar toda la información del usuario, pero sigue permitiendo un acceso rápido a los datos clave del usuario para los préstamos digitales.

A continuación se presenta una tabla resumen del diseño de agregados

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

AGREGADO	ENTIDADES (CLASES) QUE COMPRENDE	RAÍZ	PERÍMETRO	CASOS DE USO OPTIMIZADOS	OPTIMIZACIÓN LECTURAS VS. MODIFICACIONES
AGREGADO_LibroInventario	Libro, Inventario, Biblioteca	Libro	Inventario en sucursales y disponibilidad	CS1 (Lectura de libros disponibles en sucursales) CS2 (Traslado de libros entre sucursales)	Lecturas: Eficiente para consultas rápidas sobre disponibilidad y estado de libros (CU1). Modificaciones: Más costoso al actualizar múltiples sucursales (CU2).
AGREGADO_Usuario	Usuario, Reserva, Préstamo	Usuario	Historial de préstamos, reservas y multas	CS2 (Gestión de reservas y traslados) CS3 (Préstamos y descargas) CS4 (Gestión de multas) CS5 (Análisis de comportamiento de usuarios)	Lecturas: Eficiente para obtener todo el historial y preferencias de un usuario (CU3, CU4, CU5). Modificaciones: Fácil de actualizar un usuario individual (CU2, CU4).
AGREGADO_RecursoDigital	RecursoDigital, PréstamoDigital, Usuario	RecursoDigital	Préstamos y descargas de recursos digitales	CS3 (Gestión de recursos digitales)	Lecturas: Eficiente para acceso rápido a recursos digitales (CU3). Modificaciones: Permite gestionar préstamos digitales sin tocar datos generales del usuario (CU3).

Edición: 2024/2025

Arquitectura de Datos

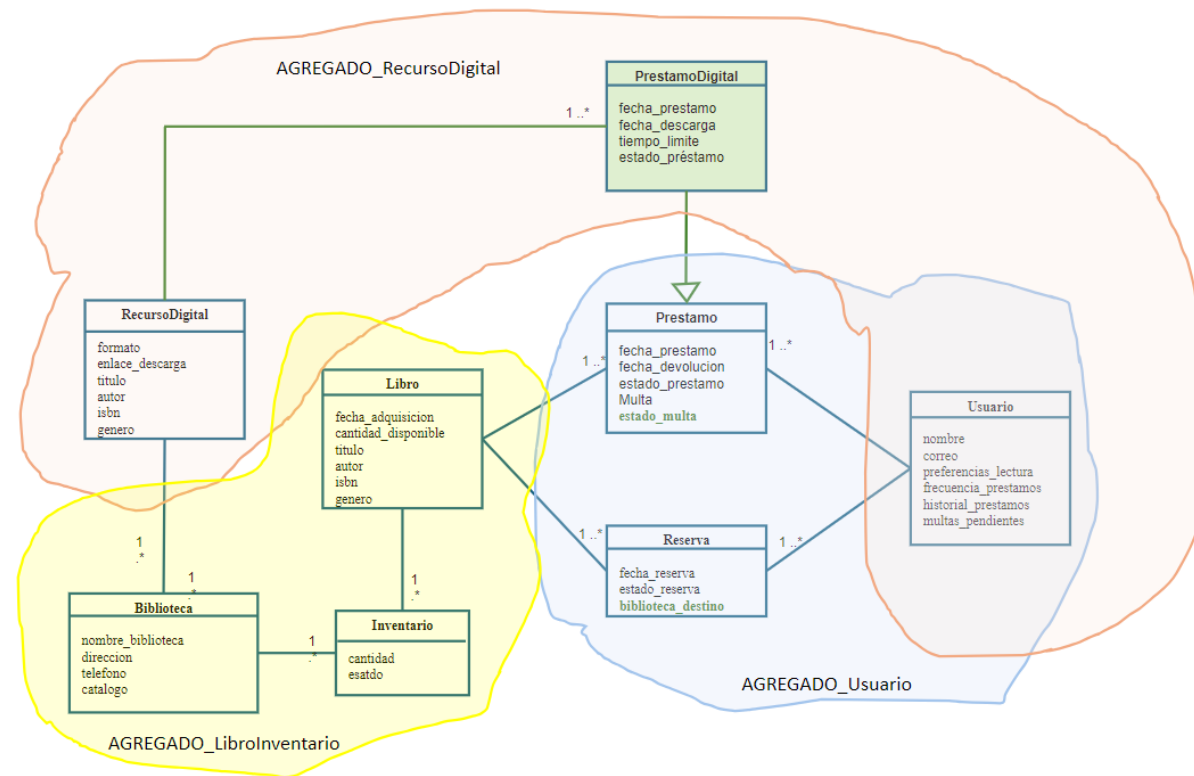
Grado en Ingeniería Informática


ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.

uc3m

Universidad
Carlos III
de Madrid

De manera grafica se puede ver así:



Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

En el diseño actual, se minimiza la redundancia en gran medida, pero siempre que se separan datos en distintos agregados, es posible que surja algún grado de redundancia para optimizar las consultas. Por ejemplo:


AGREGADO_Usuario contiene información sobre préstamos y reservas, mientras que AGREGADO_RecursoDigital también maneja préstamos digitales. Si bien son agregados separados, pueden tener información redundante en cuanto a los préstamos (préstamos físicos y digitales), pero están optimizados para diferentes tipos de consultas.

La redundancia aquí está controlada porque los préstamos digitales requieren un acceso más específico y rápido sin afectar el historial completo del usuario. Sin embargo, para gestionar los préstamos digitales eficientemente, algunos datos de los usuarios podrían duplicarse en el agregado de recursos digitales.

Documentos JSON

Un ejemplo de documento JSON del agregado AGREGADO_LibroInventario sería:

```
{
  "isbn": "123-456-789",
  "titulo": "Título del Libro",
  "autor": "Autor del Libro",
  "fecha_adquisicion": "2023-01-15",
  "inventario": [
    {
      "biblioteca": {
        "nombre_biblioteca": "Biblioteca Central",
        "direccion": "Calle Principal 123",
        "telefono": "123-456-789"
      },
      "cantidad_disponible": 5,
      "estado": "Disponible"
    },
    {
      "biblioteca": {
        "nombre_biblioteca": "Biblioteca Norte",
        "direccion": "Calle Norte 45",
        "telefono": "987-654-321"
      },
      "cantidad_disponible": 3,
      "estado": "Reservado"
    }
  ]
}
```

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--


```
}
  ]
}
```

Un ejemplo del agregado AGREGADO_Usuario sería:

```
{
  "nombre": "Juan Pérez",
  "correo": "juan.perez@example.com",
  "preferencias_lectura": ["Ciencia Ficción", "Fantasía"],
  "frecuencia_prestamos": 5,
  "historial_prestamos": [
    {
      "fecha_prestamo": "2023-09-01",
      "fecha_devolucion": "2023-09-15",
      "estado_prestamo": "Devuelto",
      "multa": 0,
      "estado_multa": "Pagada"
    },
    {
      "fecha_prestamo": "2023-10-01",
      "fecha_devolucion": "2023-10-10",
      "estado_prestamo": "Retrasado",
      "multa": 5.0,
      "estado_multa": "Pendiente"
    }
  ],
  "multas_pendientes": 5.0,
  "reservas": [
    {
      "fecha_reserva": "2023-11-01",
      "estado_reserva": "Activa",
      "biblioteca_destino": "Biblioteca Central"
    }
  ]
}
```

Se han incluido atributos adicionales como:


- `preferencias_lectura`: Los géneros que le gustan al usuario.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

- frecuencia_prestamos: El número de préstamos realizados por el usuario.
- historial_prestamos: Un historial de los préstamos previos del usuario.
- multas_pendientes es un valor agregado que refleja la suma total de las multas que aún no han sido pagadas por el usuario.

Un documento en JSON del agregado AGREGADO_RecursoDigital sería:

```
{
  "recurso_digital": {
    "formato": "PDF",
    "enlace_descarga": "https://example.com/ebook.pdf",
    "prestamos_digitales": [
      {
        "usuario": {
          "nombre": "Juan Perez",
          "correo": "juan.perez@example.com",
          "preferencias_lectura": ["Ficción", "Ciencia
ficción"],
          "frecuencia_prestamos": 12,
          "historial_prestamos": [
            {
              "fecha_prestamo": "2023-09-10",
              "fecha_devolucion": "2023-09-25",
              "estado_prestamo": "Devuelto",
              "multa": 0,
              "estado_multa": "Pagada"
            }
          ],
          "multas_pendientes": 0
        },
        "fecha_prestamo": "2024-09-10",
        "fecha_descarga": "2024-09-10",
        "tiempo_limite": 30,
        "estado_prestamo": "Activo"
      },
      {
        "usuario": {
          "nombre": "Ana Lopez",
          "correo": "ana.lopez@example.com",
```


Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

```


"preferencias_lectura": ["Historia", "Ciencia"],
"frecuencia_prestamos": 7,
"historial_prestamos": [
  {
    "fecha_prestamo": "2023-08-15",
    "fecha_devolucion": "2023-08-30",
    "estado_prestamo": "Devuelto",
    "multa": 2,
    "estado_multa": "Pendiente"
  }
],
"multas_pendientes": 2
},
"fecha_prestamo": "2024-09-05",
"fecha_descarga": "2024-09-05",
"tiempo_limite": 15,
"estado_prestamo": "Activo"
}
]
}

```

SOLUCIÓN 2c) Propón una estrategia de validación de esquema

La validación de esquema en MongoDB, junto con el uso de estrategias como referencias en lugar de incrustaciones, facilitan el mantenimiento de la consistencia e integridad, dependiendo de los volúmenes de lectura y escritura, y de las necesidades de consistencia, se podría ajustar.

- ISBN único para libros: Cada libro debe tener un ISBN único, lo que garantiza que no haya duplicados en el sistema.
- Formato de correo electrónico para usuarios: El correo electrónico de cada usuario debe tener un formato válido para garantizar la consistencia en los datos de contacto.
- Estados válidos para libros y préstamos: Los estados de los libros y préstamos deben limitarse a ciertos valores predefinidos.
- Multas positivas: Las multas deben ser siempre valores positivos.
- Fechas en préstamos: Las fechas de préstamo y devolución deben seguir un orden lógico. La fecha de devolución siempre debe ser posterior a la fecha de préstamo.
- Formato válido para recursos digitales: Los recursos digitales (como eBooks o audiolibros) deben tener un formato válido.

Edición: 2024/2025 Arquitectura de Datos Grado en Ingeniería Informática ASUNTO: Labo3.1.-Modelado de datos en mongoDB: Agregados y Casos de Uso.	 Universidad Carlos III de Madrid
--	--

- Reservas solo en libros disponibles: Los libros solo pueden ser reservados si su estado es Disponible o Reservado.
- Estado de los libros en inventario: Valida que el campo estado de los libros en Inventario solo pueda tomar los valores "Disponible", "Prestado", "Reservado", o "En Mantenimiento".
- Validación del traslado: asegurar que un libro solo pueda ser trasladado si está en estado Disponible.
- Enlace de descarga válido: Verificar que cada recurso digital tenga un enlace de descarga válido y activo.
- Validación de Multas: Verificar que el campo estado_multa solo acepte los valores "Pagada" o "Pendiente".
- Validación de historial de préstamos: Asegurar que el historial de préstamos tenga fechas válidas y consistentes.
- Validación de Recomendaciones: Si se incluyen recomendaciones, asegurar que estas contengan un ID de libro o recurso digital válido, así como su categoría de recomendación.