

---

# AE U2.6 - DESPLIEGUE EN MONGODB + U2.7 ROOTER

---



Nombre y apellidos: Cristina Sandoval  
Laborde

Curso: 2ºDAM

Asignatura: Optativa GS

# Índice

1.Introduccion.....	1
2.Entidades principales .....	2
3.Relaciones en MongoDB.....	5
4.Implementación en MongoDB .....	6
5.Cambios generales al pasar de SQL a MongoDB .....	13

## 1.Introduccion

**StudyMatchKey** es una aplicación educativa diseñada para ayudar a los estudiantes de Bachillerato a descubrir y aplicar el método de estudio que mejor se adapta a sus hábitos y preferencias. A través de un test inicial, la app identifica el estilo de aprendizaje predominante del usuario y ofrece recomendaciones personalizadas, como el uso de mapas mentales, la técnica Pomodoro, flashcards o el método Feynman.

Además, la aplicación integra un calendario inteligente para organizar tareas y exámenes, un asistente paso a paso que guía en la aplicación de técnicas activas, y un sistema de gestión de materiales que centraliza apuntes y recursos. Todo ello se complementa con notificaciones automáticas y un enfoque en la privacidad y seguridad de los datos.

El objetivo principal de StudyMatchKey es **mejorar la organización, la motivación y el rendimiento académico**, reduciendo el estrés y el riesgo de fracaso escolar mediante un acompañamiento visual, práctico y adaptado a cada estudiante.

Para el desarrollo de **StudyMatchKey** se ha optado por utilizar **MongoDB**, una base de datos no relacional orientada a documentos. La elección se fundamenta en varias razones:

- **Flexibilidad en el almacenamiento de datos:** los perfiles de usuario, resultados de test y materiales de estudio no siguen una estructura fija. MongoDB permite guardar información en formato JSON dinámico, adaptándose fácilmente a cambios futuros sin necesidad de modificar esquemas rígidos.
- **Escalabilidad:** al tratarse de una aplicación que puede crecer en número de usuarios y volumen de datos (tareas, exámenes, materiales, notificaciones), MongoDB ofrece una arquitectura distribuida que facilita el escalado horizontal.
- **Integración con aplicaciones modernas:** al ser no relacional, se adapta mejor a entornos de desarrollo ágiles y aplicaciones móviles, donde los datos cambian con frecuencia y se necesitan respuestas inmediatas.
- **Gestión de datos heterogéneos:** StudyMatchKey maneja distintos tipos de información (test, calendario, materiales, notificaciones). Una base no relacional permite almacenar todo en colecciones relacionadas sin necesidad de normalizar excesivamente como en SQL.

## **2.Entidades principales**

### **Usuario**

Atributos:

- ID\_Usuario
- ID\_Entidad
- Nombre
- Fecha\_De\_Nacimiento
- Email
- Contraseña
- Fecha\_De\_Registro

Descripción: Representa a cada estudiante dentro de la aplicación. Incluye datos mínimos necesarios para identificar al usuario y proteger su cuenta.

### **Calendario**

Atributos:

- ID\_Calendario
- ID\_Usuario
- Fecha\_Evento
- Tipo\_De\_Evento
- Descripcion\_Evento

Descripción: Permite gestionar las tareas académicas del usuario y los recordatorios automáticos.

### **TecnicaEstudio**

Atributos:

- ID\_Tecnica
- Nombre\_Tecnica
- Descripcion\_TecnicaEstudio
- Tipo\_TecnicaEstudio

Descripción: Conjunto de técnicas sugeridas por la aplicación, útiles para generar recomendaciones personalizadas.

## **MetodoEstudio**

Atributos:

- ID\_Metodo
- Nombre\_Metodo
- Descripcion\_Metodo
- Categoria

Descripción: Conjunto de métodos sugeridos por la aplicación. Se asignan a estudiantes dependiendo de la asignatura y sus preferencias.

## **Recomendacion**

Atributos:

- ID\_Recomendacion
- ID\_Usuario
- ID\_Tecnica
- Fecha\_Recomendacion
- Efectividad

Descripción: Registra qué técnica se recomienda a cada usuario y cómo ha funcionado en la práctica.

## **ArchivoEstudio**

Atributos:

- ID\_Archivo
- ID\_Usuario
- Tipo\_De\_Archivo
- Nombre\_Archivo
- Ruta\_De\_Almacenamiento
- Fecha\_De\_Subida

Descripción: Material de estudio subido por el usuario. Se almacena en espacios seguros y cifrados.

## **MetricasEstudio**

Atributos:

- ID\_Metrica
- ID\_Usuario
- Horas\_De\_Estudio
- Tiempo\_De\_Concentración
- Uso\_De\_Tecnicas
- Fecha\_Registro

Descripción: Representa las estadísticas que usa la app para personalizar las recomendaciones y mostrar progreso.

### **Entidad / Organismo**

Atributos:

- ID\_Entidad
- Nombre\_Entidad
- Tipo
- Ubicacion

Descripción: Centro educativo, institución, organismo o academia a la que pertenece el estudiante. Sirve para organizar usuarios por colegios, institutos, universidad, etc.

### **Asignatura**

Atributos:

- ID\_Asignatura
- ID\_Entidad
- Nombre\_Asignatura

Descripción: Materias impartidas por la entidad educativa. Sirven para asociar usuarios con sus estudios y generar recomendaciones según el tipo de asignatura.

### 3.Relaciones en MongoDB

#### Usuario — Entidad

Tipo de relación: (N : 1)

##### Explicación (MongoDB):

- Muchos usuarios pueden pertenecer a la misma entidad.
- Cada usuario guarda **solo el ID de la entidad**.
- No se usa una "tabla intermedia" porque **MongoDB no lo necesita**.

#### Entidad — Asignatura

Tipo de relación: (1 : N)

(Una Entidad imparte muchas Asignaturas)

##### Explicación (MongoDB):

- Una entidad ofrece muchas asignaturas.
- Cada asignatura solo referencia la entidad.
- No se embeben asignaturas dentro de entidad porque pueden ser cientos.

#### Usuario — Asignatura

Tipo de relación: (N : M)

Un usuario puede matricularse en muchas asignaturas.

Una asignatura puede tener muchos usuarios matriculados.

En MongoDB se representa con **un array dentro del Usuario**

##### Explicación (MongoDB):

- Un usuario puede tener varias asignaturas → se ponen en un array.
- Una asignatura tiene varios usuarios → **MongoDB no necesita representarlo**.
- No se usa tabla intermedia

#### Asignatura — MetodoEstudio

Tipo de relación: (N : M)

La relación se gestiona **cuando el usuario recibe un método**, no antes.

### **Usuario — MetodoEstudio (por asignatura)**

Tipo de relación: ( N : M)

#### **Explicación (MongoDB):**

- Un usuario puede tener muchos métodos.
- Cada asignatura puede estar asociada a diferentes métodos dependiendo del perfil del usuario.
- No es que el método dependa de la asignatura, sino que la combinación Usuario + Asignatura determina qué métodos se recomiendan, es decir, el perfil del usuario dentro de esa asignatura concreta

### **Usuario — Calendario**

Tipo de relación: (1 : N)

#### **Explicación (MongoDB):**

- Un usuario puede tener muchos eventos en su calendario.
- Se usa referencia por ID.

## **4.Implementación en MongoDB**

**Colección Usuario:** relación con Entidad (ID\_Entidad) y el array con asignaturas (N:M)

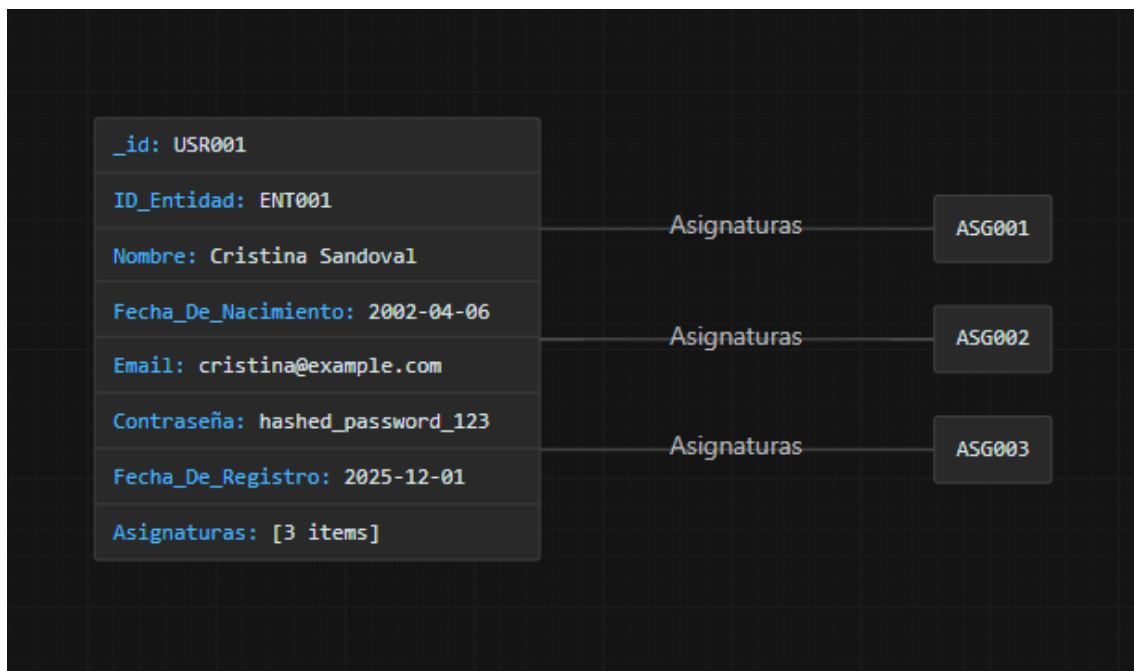
```
{
  "_id": "USR001",
  "ID_Entidad": "ENT001",
  "Nombre": "Cristina Sandoval",
  "Fecha_De_Nacimiento": "2002-04-06",
  "Email": "cristina@example.com",
  "Contraseña": "hashed_password_123",
  "Fecha_De_Registro": "2025-12-01",
  "Asignaturas": ["ASG001", "ASG002", "ASG003"]
}
```



```

{
  "_id": "USR001",
  "ID_Entidad": "ENT001",
  "Nombre": "Cristina Sandoval",
  "Fecha_De_Nacimiento": "2002-04-06",
  "Email": "cristina@example.com",
  "Contraseña": "hashed_password_123",
  "Fecha_De_Registro": "2025-12-01",
  "Asignaturas": [
    "ASG001",
    "ASG002",
    "ASG003"
  ]
}

```



### Colección: Entidad / Organismo

```

{
  "_id": "ENT001",
  "Nombre_Entidad": "IES Tecnológico Málaga",
  "Tipo": "Instituto",
  "Ubicacion": "Málaga, España"
}

```

```
{
  "_id": "ENT001",
  "Nombre_Entidad": "IES Tecnológico Málaga",
  "Tipo": "Instituto",
  "Ubicacion": "Málaga, España"
}
```

<code>_id</code>	ENT001
<code>Nombre_Entidad</code>	IES Tecnológico Málaga
<code>Tipo</code>	Instituto
<code>Ubicacion</code>	Málaga, España

### Colección: Asignatura

Relacionada con la entidad.

```
{
  "_id": "ASG001",
  "ID_Entidad": "ENT001",
  "Nombre_Asignatura": "Programación"
}
```

```
{
  "_id": "ASG001",
  "ID_Entidad": "ENT001",
  "Nombre_Asignatura": "Programación"
}
```

<code>_id</code>	ASG001
<code>ID_Entidad</code>	ENT001
<code>Nombre_Asignatura</code>	Programación

### Colección: Calendario

Eventos propios de cada usuario

```
{
  "_id": "CAL001",
  "ID_Usuario": "USR001",
  "Fecha_Evento": "2025-12-15",
  "Tipo_De_Evento": "Examen",
  "Descripcion_Evento": "Examen final de Programación"
}
```

```
{
  "_id": "CAL001",
  "ID_Usuario": "USR001",
  "Fecha_Evento": "2025-12-15",
  "Tipo_De_Evento": "Examen",
  "Descripcion_Evento": "Examen final de Programación"
}
```

_id:	CAL001
ID_Usuario:	USR001
Fecha_Evento:	2025-12-15
Tipo_De_Evento:	Examen
Descripcion_Evento:	Examen final de Programac...

### Colección: TecnicaEstudio

```
{
  "_id": "TEC001",
  "Nombre_Tecnica": "Pomodoro",
  "Descripcion_TecnicaEstudio": "Estudiar en intervalos de 25 minutos con 5 minutos de descanso.",
  "Tipo_TecnicaEstudio": "Concentración"
}
```

```
{
  "_id": "TEC001",
  "Nombre_Tecnica": "Pomodoro",
  "Descripcion_TecnicaEstudio": "Estudiar en intervalos de 25 minutos con 5 minutos de descanso.",
  "Tipo_TecnicaEstudio": "Concentración"
}
```

```
_id: TEC001
Nombre_Tecnica: Pomodoro
Descripcion_TecnicaEstudio: Estudiar en intervalos de 25 minu...
Tipo_TecnicaEstudio: Concentración
```

### Colección: MetodoEstudio

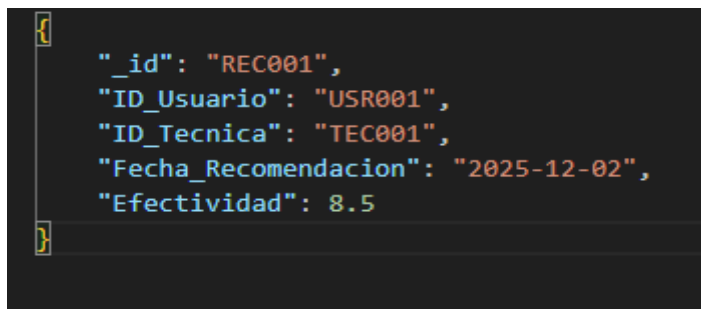
```
{
  "_id": "MET001",
  "Nombre_Metodo": "Mapas Mentales",
  "Descripcion_Metodo": "Organización gráfica del contenido.",
  "Categoria": "Visual"
}
```

```
{
  "_id": "MET001",
  "Nombre_Metodo": "Mapas Mentales",
  "Descripcion_Metodo": "Organización gráfica del contenido.",
  "Categoria": "Visual"
}
```

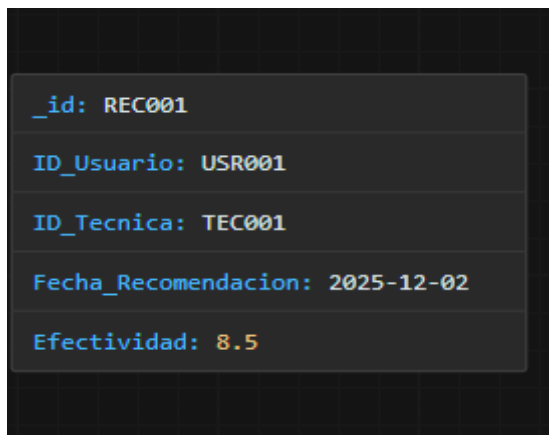
```
_id: MET001
Nombre_Metodo: Mapas Mentales
Descripcion_Metodo: Organización gráfica del ...
Categoria: Visual
```

### Colección: Recomendación

```
{
  "_id": "REC001",
  "ID_Usuario": "USR001",
  "ID_Tecnica": "TEC001",
  "Fecha_Recomendacion": "2025-12-02",
  "Efectividad": 8.5
}
```

A screenshot of a code editor with a dark background. It shows a JSON object with the following fields: "\_id": "REC001", "ID\_Usuario": "USR001", "ID\_Tecnica": "TEC001", "Fecha\_Recomendacion": "2025-12-02", and "Efectividad": 8.5. The text is color-coded: strings are in orange and the number is in green.

```
{
  "_id": "REC001",
  "ID_Usuario": "USR001",
  "ID_Tecnica": "TEC001",
  "Fecha_Recomendacion": "2025-12-02",
  "Efectividad": 8.5
}
```

A screenshot of a database interface showing a document model for the 'Recomendación' collection. The fields are listed in a table-like structure with labels and values.

_id	REC001
ID_Usuario	USR001
ID_Tecnica	TEC001
Fecha_Recomendacion	2025-12-02
Efectividad	8.5

### Colección: ArchivoEstudio

```
{
  "_id": "ARC001",
  "ID_Usuario": "USR001",
  "Tipo_De_Archivo": "PDF",
  "Nombre_Archivo": "apuntes_programacion.pdf",
  "Ruta_De_Almacenamiento": "/uploads/USR001/apuntes_programacion.pdf",
  "Fecha_De_Subida": "2025-12-01"
}
```

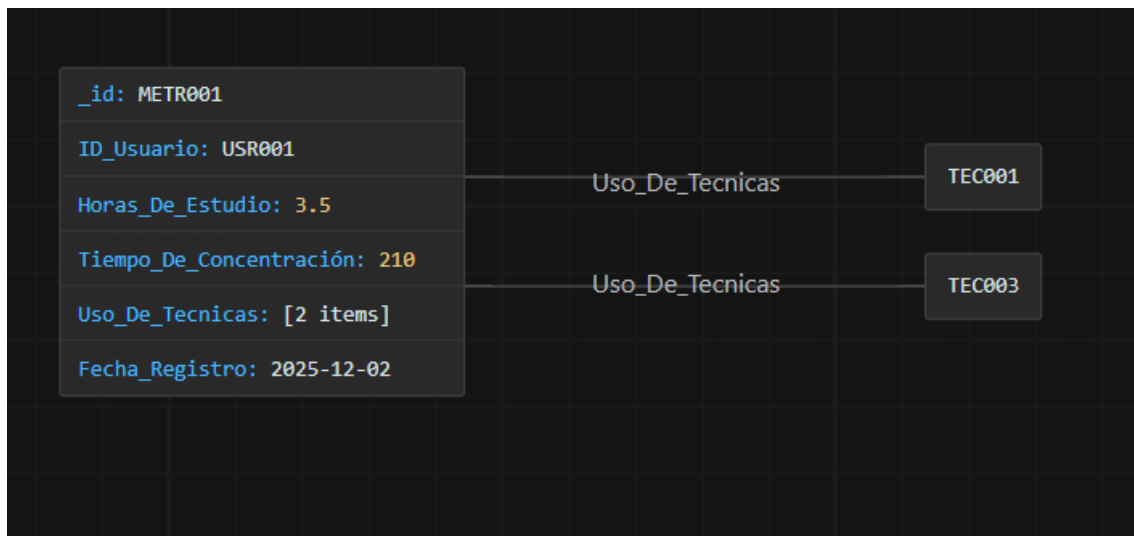
```
{
  "_id": "ARC001",
  "ID_Usuario": "USR001",
  "Tipo_De_Archivo": "PDF",
  "Nombre_Archivo": "apuntes_programacion.pdf",
  "Ruta_De_Almacenamiento": "/uploads/USR001/apuntes_programacion.pdf",
  "Fecha_De_Subida": "2025-12-01"
}
```

_id: ARC001
ID_Usuario: USR001
Tipo_De_Archivo: PDF
Nombre_Archivo: apuntes_programacion.pdf
Ruta_De_Almacenamiento: /uploads/USR001/apunt_
Fecha_De_Subida: 2025-12-01

### Colección: MetricasEstudio

```
{
  "_id": "METR001",
  "ID_Usuario": "USR001",
  "Horas_De_Estudio": 3.5,
  "Tiempo_De_Concentración": 210,
  "Uso_De_Tecnicas": ["TEC001", "TEC003"],
  "Fecha_Registro": "2025-12-02"
}
```

```
{
  "_id": "METR001",
  "ID_Usuario": "USR001",
  "Horas_De_Estudio": 3.5,
  "Tiempo_De_Concentración": 210,
  "Uso_De_Tecnicas": [
    "TEC001",
    "TEC003"
  ],
  "Fecha_Registro": "2025-12-02"
}
```



## 5. Cambios generales al pasar de SQL a MongoDB

En el modelo relacional original existían:

- Tablas independientes con claves primarias .
- Relaciones 1:N mediante claves foráneas.
- Relaciones N:M mediante tablas intermedias (Usuario\_Asignatura, Asignatura\_Metodo, etc.).
- Reglas de integridad referencial estrictas.

MongoDB trabaja con documentos y colecciones, por lo que fue necesario:

- Eliminar todas las tablas intermedias

Ya no son necesarias gracias a los arrays dentro de los documentos.

- Reemplazar claves foráneas por referencias simples (ID\_Entidad, ID\_Usuario, etc.)

Porque MongoDB no usa FOREIGN KEY ni JOIN.

- Mantener cada entidad como colección separada

Para asegurar independencia, escalabilidad y claridad del modelo.

### 5.1. Explicación de los cambios

#### ✓ Relación Usuario — Entidad (N:1)

##### Antes (SQL):

- Usuario tenía una clave foránea ID\_Entidad.

##### Ahora (MongoDB):

- Se mantiene igual, pero como referencia simple.
- No se necesitan restricciones ni validaciones automáticas.

##### Motivo del cambio:

MongoDB no usa claves foráneas; por ello, **se conserva solo el identificador como referencia.**

#### ✓ Relación Entidad — Asignatura (1:N)

##### Antes (SQL):

- Asignatura tenía ID\_Entidad como clave foránea.

##### Ahora (MongoDB):

- Exactamente igual: se mantiene un atributo ID\_Entidad dentro de cada asignatura.

##### Motivo:

Las asignaturas son muchas, por lo que **no deben ir dentro de una entidad.**

#### ✓ Relación Usuario — Asignatura (N:M)

Este es uno de los cambios más importantes.

##### Antes (SQL):

- Existía una tabla intermedia Usuario\_Asignatura.

##### Ahora (MongoDB):

- La tabla intermedia **desaparece.**
- En Usuario se añade un array:

"Asignaturas": ["ASG001", "ASG002", "ASG003"]



**Motivo:**

- MongoDB permite representar N:M con arrays.
- Se optimiza la operación principal: consultar asignaturas del usuario.

✓ **Relación Asignatura — MetodoEstudio (N:M)**

**Antes (SQL):**

- Tabla intermedia que conectaba asignaturas con métodos.

**Ahora (MongoDB):**

- **No se modela directamente.**
- Solo aparece la relación cuando un usuario recibe una recomendación.

**Motivo:**

La relación no es global, sino dependiente del perfil del usuario → por eso solo existe en la colección Recomendacion.

✓ **Relación Usuario — MétodoEstudio (N:M por asignatura)**

**Antes (SQL):**

- Requería tabla intermedia con usuario + asignatura + método.

**Ahora (MongoDB):**

- Se registra únicamente en la colección Recomendacion.
- No existe una tabla intermedia.

**Motivo:**

La recomendación depende del contexto del usuario(perfil), no de la asignatura global.

✓ **Relación Usuario — Calendario (1:N)**

**Antes (SQL):**

- Calendario tenía clave foránea a Usuario.

**Ahora (MongoDB):**

- Se mantiene igual, con ID\_Usuario.

**Motivo:**

Cada usuario puede tener muchos eventos y es mejor una colección independiente.

**Tablas eliminadas en la migración**

Las siguientes tablas del modelo relacional ya no existen en MongoDB:

<b>Tabla SQL</b>	<b>Motivo de eliminación</b>
<b>Usuario_Asignatura</b>	Sustituida por un array dentro de Usuario
<b>Asignatura_Metodo</b>	Relación implícita en Recomendacion
<b>Usuario_Metodo</b>	Ahora asociado mediante Recomendacion

Estas eliminaciones son una consecuencia directa de la flexibilidad del modelo documental.

Enlace a repositorio Optativa → <https://github.com/crisanlab94/2-Dam/tree/main/Optativa>