Modelo lógico (entidades y relaciones)

Entidades principales

- **User**(userId, nombre, email, hashPassword, rol, afiliación, estado)
- **Project**(projectId, ownerId→User, título, áreaTemática, descripción, estado)
- **Membership**(projectId→Project, userId→User, rolProyecto, permisos) (controla compartir biblioteca/equipo)
- **Document**(docId, projectId→Project, ownerId→User, título, autores, año, doi/url, etiquetas[], archivoPDF{url,hash}, fuente{Scholar/PubMed/manual}, estadoIngesta)
- **Summary**(summaryId, docId→Document, modelo, idioma, tokensIn/Out, texto, calidadScore)
- **Suggestion**(suggestionId, projectId→Project, consulta, origenAPI, items[{titulo,doi,url,score}], ejecutadoPor, fecha)
- **Notification**(notifId, userId→User, tipo, payload, canal, leída)
- **Conversation**(convId, projectId?→Project, userId→User, modelo, propósito)
- **Message**(msgId, convId→Conversation, rol{user,assistant,system}, texto, tokens, referencias[docId])
- **SearchJob**(jobId, projectId→Project, query, fuentes, schedule, estado, últimoRun, n8nRunId)
- **MetricEvent**(eventId, userId, projectId?, tipo, metadata, ts)
- **AuditLog**(auditId, actorId→User, acción, recurso{tipo,id}, detalle, ts)

Relaciones (cardinalidad esencial)

- User 1..* Project (como owner) y N..M mediante **Membership**.
- Project 1..* Document; Document 1..* Summary.
- Project 1..* Suggestion y 1..* SearchJob.
- User 1..* Notification; User 1..* Conversation; Conversation 1..* Message.
- Todo genera **MetricEvent** y **AuditLog** (event sourcing ligero)

Modelo físico (MongoDB – colecciones, campos e índices)

1) users

```
{
  "_id": ObjectId, "email": String, "passwordHash": String,
  "name": String, "role": { "type": "enum", "values": ["admin","user"] },
  "affiliation": String, "status": { "type": "enum", "values": ["active","disabled"] },
  "createdAt": Date, "updatedAt": Date
}
```

2) projects

```
{
  "_id": ObjectId, "ownerId": ObjectId, // users._id
  "title": String, "area": String, "description": String,
  "status": { "type":"enum", "values":["draft","active","archived"] },
  "tags": [String], "createdAt": Date, "updatedAt": Date
}
```

3) memberships

```
{
  "_id": ObjectId, "projectId": ObjectId, "userId": ObjectId,
  "projectRole": { "type":"enum", "values":["viewer","editor","owner"] },
  "permissions": [String], // p.ej. ["read","write","share"]
  "createdAt": Date
}}
```

4) Documents

```
{
  "_id": ObjectId, "projectId": ObjectId, "ownerId": ObjectId,
  "title": String, "authors": [String], "year": Number,
  "doi": String, "url": String,
  "source": { "type":"enum", "values":["scholar","pubmed","manual"] },
  "keywords": [String],
  "pdf": { "url": String, "sha256": String, "sizeKB": Number },
  "ingestStatus": { "type":"enum", "values":["pending","ok","error"] },
  "createdAt": Date, "updatedAt": Date
}
```

5) summaries

```
{
  "_id": ObjectId, "docId": ObjectId, "model": String, "lang": String,
  "tokensIn": Number, "tokensOut": Number,
  "qualityScore": Number,
  "summary": String, "createdBy": ObjectId, "createdAt": Date
}
```

6) suggestions

```
{
  "_id": ObjectId, "projectId": ObjectId,
  "query": String, "source": [String], // scholar, pubmed
  "items": [{ "title": String, "doi": String, "url": String, "score": Number }],
  "executedBy": ObjectId, "createdAt": Date
}
```

7) notifications

```
{
  "_id": ObjectId, "userId": ObjectId, "type": String,
  "channel": { "type":"enum", "values":["inapp","email"] },
  "payload": Object, "read": Boolean, "createdAt": Date
}
```

8) notifications

```
{
  "_id": ObjectId, "userId": ObjectId, "projectId": ObjectId,
  "model": String, "purpose": String, "createdAt": Date
}
```

9) notifications

```
{
  "_id": ObjectId, "convId": ObjectId, "role": { "type":"enum", "values":["user","assistant","system"] },
  "text": String, "tokenUsage": { "in": Number, "out": Number },
  "refs": [ObjectId], // documents._id citados
  "createdAt": Date
}
```

10) searchJobs (automatización/n8n)

```
{
  "_id": ObjectId, "projectId": ObjectId, "query": String,
  "sources": [String], "schedule": String, // cron o RRULE
  "status": { "type":"enum", "values":["active","paused"] },
  "lastRunAt": Date, "n8nRunId": String, "createdBy": ObjectId
}
```

11) metricEvents (telemetría/productividad)

```
{
  "_id": ObjectId, "type": String, // "doc_read","summary_created","chat_msg"
  "userId": ObjectId, "projectId": ObjectId,
  "meta": Object, "ts": Date
}
```

12) auditLogs

```
{
  "_id": ObjectId, "actorId": ObjectId,
  "action": String, // CREATE_DOC, DELETE_SUMMARY...
  "resource": { "type": String, "id": ObjectId },
  "detail": Object, "ts": Date, "ip": String
}
```