



Arquitectura de Servicios

# 2do Avance del proyecto

**Docente:** Roberto Suarez Zinzun

**Alumno:**

Alfaro Herrera Jesús Antonio  
Cruz Hernández Juan Manuel  
Piñones Ramos Said Rafael  
Ramos Arévalos Ricardo

**Fecha y Lugar:**

14 de marzo de 2025.  
Zamora, Mich

## INDICE

<b>2do Avance del proyecto .....</b>	<b>3</b>
Diseño del Modelo de Datos.....	3
Script de BD .....	4
Carga inicial de datos .....	5
Definición de los requerimientos de servicios .....	14

## 2do Avance del proyecto

### 1. Diseño del Modelo de Datos

Para el proyecto Elephocus, se ha optado por utilizar una base de NoSQL, y para ello utilizaremos MongoDB. Esto permitirá una mayor flexibilidad al proyecto, por su capacidad de escalar verticalmente. Como resultado también, de ya tener mejor dominio y concepto de este tipo de base de datos, por su manejo de documentos en formato JSON, además de que el tipo de proyecto está en su mayoría enfocado, para ser una app móvil.

#### Modelo de Base de Datos (NoSQL - MongoDB)

Para el almacenamiento de datos se han definido las siguientes colecciones:

- Colección de Usuarios (usuarios)

```
{
  "_id": ObjectId(),
  "nombre": String,
  "correo": String,
  "password": String,
  "edad": Number,
  "nivel_academico": String,
  "pais_region": String,
  "estatus": String,
  "tipo_usuario": String
}
```

- Colección de Temas (temas)

```
{
  "_id": ObjectId(),
  "nombre": String,
  "descripcion": String
}
```

- Colección de Flashcards (flashcards)

```
{
  "_id": ObjectId(),
  "pregunta": String,
  "respuesta": String,
  "id_tema": ObjectId(),
}
```

```
{  
  "autor": ObjectId("idUsuario"),  
  "fecha_creacion": ISODate()  
}
```

- Colección de Temas\_Flashcards (temas\_flashcards)

```
{  
  "_id": ObjectId(),  
  "id_tema": ObjectId(),  
  "id_flashcard": ObjectId()  
}
```

## 2. Script de BD

```
use Elephocus
```

- Colección “usuarios”

```
db.createCollection("usuarios");  
{  
  "_id": ObjectId(),  
  "nombre": String,  
  "correo": String,  
  "password": String,  
  "edad": Number,  
  "nivel_academico": String,  
  "pais_region": String,  
  "estatus": String,  
  "tipo_usuario": String  
}
```

- Colección “temas”

```
db.createCollection("temas");  
{  
  "_id": ObjectId(),  
  "nombre": String,  
  "descripcion": String  
}
```

#### - Colección “flashcards”

```
db.createCollection("flashcards");
{
  "_id": ObjectId(),
  "pregunta": String,
  "respuesta": String,
  "id_tema": ObjectId(),
  "autor": ObjectId("idUsuario"),
  "fecha_creacion": ISODate()
}
```

#### - Colección “temas\_flashcards”

```
db.createCollection("temas_flashcards");
{
  "_id": ObjectId(),
  "id_tema": ObjectId(),
  "id_flashcard": ObjectId()
}
```

### 3. Carga inicial de datos

En este apartado se detalla el proceso de creación, configuración y carga de datos en la base de datos Elephocus utilizando MongoDB. Se establecieron validaciones para garantizar la integridad de los datos y se realizó la inserción de documentos en cada colección.

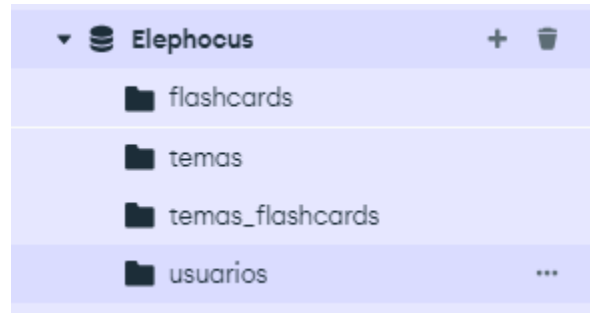
#### 1. Creación de la base de datos

Para iniciar, se creó la base de datos Elephocus en MongoDB Compass o mediante la shell de MongoDB con el siguiente comando:

use Elephocus;

A continuación, se generaron las siguientes colecciones:

- usuarios
- temas
- flashcards
- temas\_flashcards



## 2. Validaciones en cada colección

Para asegurar que los datos ingresados cumplan con una estructura definida, se implementaron validaciones con JSON Schema en MongoDB Compass.

## 2.1 Validaciones para la colección usuarios

Documents 0 Aggregations Schema Indexes 1 **Validation**

Validation Action ⓘ Error Validation Level ⓘ Strict

```
1 {
2   $jsonSchema: {
3     bsonType: 'object',
4     required: [
5       'nombre',
6       'correo',
7       'password',
8       'edad',
9       'nivel_academico',
10      'pais_region',
11      'estatus',
12      'tipo_usuario'
13    ],
14    properties: {
15      nombre: {
16        bsonType: 'string',
17        description: 'El nombre del usuario debe ser una cadena de texto.'
18      },
19      correo: {
20        bsonType: 'string',
21        pattern: '^\\S+@\\S+\\.\\S+$',
22        description: 'Debe ser un correo electrónico válido.'
23      },
24      password: {
25        bsonType: 'string',
26        minLength: 8,
27        description: 'La contraseña debe tener al menos 8 caracteres.'
28      },
29      edad: {
30        bsonType: 'int',
31        minimum: 13,
32        description: 'Debe ser un número entero mayor o igual a 13.'
33      },
34      nivel_academico: {
35        bsonType: 'string',
36        enum: [
37          'Primaria',
38          'Secundaria',
39          'Preparatoria',
40          'Universidad',
41          'Posgrado'
42        ],
43        description: 'Debe ser un nivel académico válido.'
44      },
45      pais_region: {
46        bsonType: 'string',
47        description: 'Debe ser una cadena representando el país o región.'
48      },
49      estatus: {
50        bsonType: 'string',
51        enum: [
52          'activo',
53          'inactivo'
54        ],
55        description: 'Debe ser \'activo\' o \'inactivo\'.'
56      },
57      tipo_usuario: {
58        bsonType: 'string',
59        enum: [
60          'estudiante',
61          'profesor',
62          'administrador'
63        ],
64        description: 'Debe ser un tipo de usuario válido.'
65      }
66    }
67  }
68 }
```

## 2.2 Validaciones para la colección temas

Documents 0 Aggregations Schema Indexes 1 **Validation**

Validation Action ⓘ Error ▼

Validation Level ⓘ Strict ▼

```
1 {
2   $jsonSchema: {
3     bsonType: 'object',
4     required: [
5       'nombre',
6       'descripcion'
7     ],
8     properties: {
9       nombre: {
10        bsonType: 'string',
11        description: 'El nombre del tema debe ser una cadena de texto.'
12      },
13      descripcion: {
14        bsonType: 'string',
15        description: 'Debe ser una breve descripción del tema.'
16      }
17    }
18  }
19 }
```

## 2.3 Validaciones para la colección flashcards

Documents 0 Aggregations Schema Indexes 1 **Validation**

Validation Action ⓘ Error ▼

Validation Level ⓘ Strict ▼

```
1 {
2   $jsonSchema: {
3     bsonType: 'object',
4     required: [
5       'pregunta',
6       'respuesta',
7       'id_tema',
8       'autor',
9       'fecha_creacion'
10    ],
11    properties: {
12      pregunta: {
13        bsonType: 'string',
14        description: 'Debe ser una pregunta en formato de cadena de texto.'
15      },
16      respuesta: {
17        bsonType: 'string',
18        description: 'Debe ser una respuesta en formato de cadena de texto.'
19      },
20      id_tema: {
21        bsonType: 'objectId',
22        description: 'Debe ser una referencia válida a la colección de temas.'
23      },
24      autor: {
25        bsonType: 'objectId',
26        description: 'Debe ser una referencia válida al usuario creador.'
27      },
28      fecha_creacion: {
29        bsonType: 'date',
30        description: 'Debe ser una fecha en formato ISODate.'
31      }
32    }
33  }
34 }
```



## 2.4 Validaciones para la colección temas\_flashcards

Documents 0

Aggregations

Schema

Indexes 1

**Validation**

Validation Action ⓘ

Error ▼

Validation Level ⓘ

Strict ▼

```
1  {
2    $jsonSchema: {
3      bsonType: 'object',
4      required: [
5        'id_tema',
6        'id_flashcard'
7      ],
8      properties: {
9        id_tema: {
10         bsonType: 'objectId',
11         description: 'Debe ser una referencia válida a la colección de temas.'
12       },
13       id_flashcard: {
14         bsonType: 'objectId',
15         description: 'Debe ser una referencia válida a la colección de flashcards.'
16       }
17     }
18   }
19 }
```

### 3. Carga de datos

Para poblar la base de datos, se realizó la inserción de 8 documentos en cada colección.

#### 3.1 Inserción de datos en usuarios

Se insertaron registros con diferentes perfiles, asegurando que cumplan con los requisitos establecidos en las validaciones.

```
> db.usuarios.insertMany([
  {
    _id: ObjectId("640000000000000000000001"),
    nombre: "Juan Pérez",
    correo: "juan.perez@example.com",
    password: "password123",
    edad: 25,
    nivel_academico: "Universidad",
    pais_region: "México",
    estatus: "activo",
    tipo_usuario: "estudiante"
  },
  {
    _id: ObjectId("640000000000000000000002"),
    nombre: "María García",
    correo: "maria.garcia@example.com",
    password: "securepass456",
    edad: 30,
    nivel_academico: "Posgrado",
    pais_region: "Argentina",
    estatus: "activo",
    tipo_usuario: "profesor"
  },
  {
    _id: ObjectId("640000000000000000000003"),
    nombre: "Carlos López",
    correo: "carlos.lopez@example.com",
    password: "mypass789",
    edad: 22,
    nivel_academico: "Preparatoria",
    pais_region: "Chile",
    estatus: "activo",
    tipo_usuario: "estudiante"
  },
  {
    _id: ObjectId("640000000000000000000004"),
    nombre: "Ana Martínez",
    correo: "ana.martinez@example.com",
    password: "anapassword",
    edad: 28,
    nivel_academico: "Universidad",
    pais_region: "Colombia",
    estatus: "activo",
    tipo_usuario: "profesor"
  },
  {
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('640000000000000000000001'),
    '1': ObjectId('640000000000000000000002'),
    '2': ObjectId('640000000000000000000003'),
    '3': ObjectId('640000000000000000000004'),
    '4': ObjectId('640000000000000000000005'),
    '5': ObjectId('640000000000000000000006'),
    '6': ObjectId('640000000000000000000007'),
    '7': ObjectId('640000000000000000000008')
  }
}
```

## 3.2 Inserción de datos en temas

Se incluyeron diferentes temas educativos como Matemáticas, Historia, Ciencias, Literatura, entre otros.

```
> db.temas.insertMany([
  {
    _id: ObjectId("650000000000000000000001"),
    nombre: "Matemáticas",
    descripción: "Conceptos básicos y avanzados de matemáticas."
  },
  {
    _id: ObjectId("650000000000000000000002"),
    nombre: "Historia",
    descripción: "Eventos históricos y contextos importantes."
  },
  {
    _id: ObjectId("650000000000000000000003"),
    nombre: "Ciencias",
    descripción: "Fundamentos de biología, química y física."
  },
  {
    _id: ObjectId("650000000000000000000004"),
    nombre: "Literatura",
    descripción: "Análisis de obras literarias y textos."
  },
  {
    _id: ObjectId("650000000000000000000005"),
    nombre: "Geografía",
    descripción: "Conocimientos sobre países, ciudades y territorios."
  },
  {
    _id: ObjectId("650000000000000000000006"),
    nombre: "Arte",
    descripción: "Historia del arte y técnicas artísticas."
  },
  {
    _id: ObjectId("650000000000000000000007"),
    nombre: "Programación",
    descripción: "Conceptos y lenguajes de programación."
  },
  {
    _id: ObjectId("650000000000000000000008"),
    nombre: "Física",
    descripción: "Leyes y principios de la física clásica y moderna."
  }
]);
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('650000000000000000000001'),
    '1': ObjectId('650000000000000000000002'),
    '2': ObjectId('650000000000000000000003'),
    '3': ObjectId('650000000000000000000004'),
    '4': ObjectId('650000000000000000000005'),
    '5': ObjectId('650000000000000000000006'),
    '6': ObjectId('650000000000000000000007'),
    '7': ObjectId('650000000000000000000008')
  }
}
```

### 3.3 Inserción de datos en flashcards

Cada flashcard está asociada a un tema y a un usuario creador. Se añadieron preguntas de distintos temas educativos.

```
> db.flashcards.insertMany([
  {
    _id: ObjectId("660000000000000000000001"),
    pregunta: "¿Cuánto es 2+2?",
    respuesta: "4",
    id_tema: ObjectId("650000000000000000000001"), // Matemáticas
    autor: ObjectId("640000000000000000000001"), // Juan Pérez
    fecha_creacion: ISODate("2025-03-14T10:00:00Z")
  },
  {
    _id: ObjectId("660000000000000000000002"),
    pregunta: "¿Quién fue Simón Bolívar?",
    respuesta: "Líder de la independencia en América Latina.",
    id_tema: ObjectId("650000000000000000000002"), // Historia
    autor: ObjectId("640000000000000000000002"), // María García
    fecha_creacion: ISODate("2025-03-14T11:00:00Z")
  },
  {
    _id: ObjectId("660000000000000000000003"),
    pregunta: "¿Qué es la fotosíntesis?",
    respuesta: "Proceso mediante el cual las plantas producen energía.",
    id_tema: ObjectId("650000000000000000000003"), // Ciencias
    autor: ObjectId("640000000000000000000003"), // Carlos López
    fecha_creacion: ISODate("2025-03-14T12:00:00Z")
  },
  {
    _id: ObjectId("660000000000000000000004"),
    pregunta: "¿Quién escribió 'Cien Años de Soledad'?",
    respuesta: "Gabriel García Márquez",
    id_tema: ObjectId("650000000000000000000004"), // Literatura
    autor: ObjectId("640000000000000000000004"), // Ana Martínez
    fecha_creacion: ISODate("2025-03-14T13:00:00Z")
  },
  {
    _id: ObjectId("660000000000000000000005"),
    pregunta: "¿Cuál es la capital de España?",
    respuesta: "Madrid",
    id_tema: ObjectId("650000000000000000000005"), // Geografía
    autor: ObjectId("640000000000000000000005"), // Luis González
    fecha_creacion: ISODate("2025-03-14T14:00:00Z")
  },
  {
    _id: ObjectId("660000000000000000000006"),
    pregunta: "¿Qué es el cubismo?",
    respuesta: "Movimiento artístico del siglo XX.",
    id_tema: ObjectId("650000000000000000000006"), // Arte
    autor: ObjectId("640000000000000000000006"), // Sofía Rodríguez
    fecha_creacion: ISODate("2025-03-14T15:00:00Z")
  },
])
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('660000000000000000000001'),
    '1': ObjectId('660000000000000000000002'),
    '2': ObjectId('660000000000000000000003'),
    '3': ObjectId('660000000000000000000004'),
    '4': ObjectId('660000000000000000000005'),
    '5': ObjectId('660000000000000000000006'),
    '6': ObjectId('660000000000000000000007'),
    '7': ObjectId('660000000000000000000008')
  }
}
```

### 3.4 Inserción de datos en temas\_flashcards

Esta colección funciona como una tabla relacional que vincula temas con flashcards específicas.

```
> db.temas_flashcards.insertMany([
  {
    _id: ObjectId("670000000000000000000001"),
    id_tema: ObjectId("650000000000000000000001"),
    id_flashcard: ObjectId("660000000000000000000001")
  },
  {
    _id: ObjectId("670000000000000000000002"),
    id_tema: ObjectId("650000000000000000000002"),
    id_flashcard: ObjectId("660000000000000000000002")
  },
  {
    _id: ObjectId("670000000000000000000003"),
    id_tema: ObjectId("650000000000000000000003"),
    id_flashcard: ObjectId("660000000000000000000003")
  },
  {
    _id: ObjectId("670000000000000000000004"),
    id_tema: ObjectId("650000000000000000000004"),
    id_flashcard: ObjectId("660000000000000000000004")
  },
  {
    _id: ObjectId("670000000000000000000005"),
    id_tema: ObjectId("650000000000000000000005"),
    id_flashcard: ObjectId("660000000000000000000005")
  },
  {
    _id: ObjectId("670000000000000000000006"),
    id_tema: ObjectId("650000000000000000000006"),
    id_flashcard: ObjectId("660000000000000000000006")
  },
  {
    _id: ObjectId("670000000000000000000007"),
    id_tema: ObjectId("650000000000000000000007"),
    id_flashcard: ObjectId("660000000000000000000007")
  },
  {
    _id: ObjectId("670000000000000000000008"),
    id_tema: ObjectId("650000000000000000000008"),
    id_flashcard: ObjectId("660000000000000000000008")
  }
]);
```

```
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('670000000000000000000001'),
    '1': ObjectId('670000000000000000000002'),
    '2': ObjectId('670000000000000000000003'),
    '3': ObjectId('670000000000000000000004'),
    '4': ObjectId('670000000000000000000005'),
    '5': ObjectId('670000000000000000000006'),
    '6': ObjectId('670000000000000000000007'),
    '7': ObjectId('670000000000000000000008')
  }
}
```

## 4. Definición de los requerimientos de servicios

### *Servicio de Gestión de Usuarios*

**Tipo de servicio:** Entidad

**Responsable:** Cruz Hernández Juan Manuel

**Operaciones expuestas:**

- Crear Usuario
- Obtener Usuarios
- Actualizar Usuario
- Eliminar Usuario

Crear Usuario	
<b>Actor(es)</b>	Usuarios nuevos, Administrador
<b>URL</b>	/usuarios/crear
<b>Método HTTP</b>	POST
<b>Lógica de Negocio</b>	Registrar un nuevo usuario a la base de datos.
<b>Entrada</b>	<pre>{   "nombre": "string",   "email": "string",   "password": "string",   "edad": "integer",   "grado": "string" }</pre>
<b>Salida</b>	<pre>{   "_id": "ObjectId",   "nombre": "string",   "email": "string",   "edad": "integer",   "grado": "string" }</pre>

Obtener Usuarios	
<b>Actor(es)</b>	Usuarios nuevos, Administrador
<b>URL</b>	/usuarios/(id)
<b>Método HTTP</b>	GET
<b>Lógica de Negocio</b>	Consultar la información de un usuario mediante su _id.
<b>Entrada</b>	<pre>{   "_id": "ObjectId" }</pre>
<b>Salida</b>	<pre>{   "_id": "ObjectId",</pre>

	<pre> "nombre": "string", "email": "string", "edad": "integer", "grado": "string" } </pre>
--	--

Actualizar Usuario	
<b>Actor(es)</b>	Usuarios, Administrador
<b>URL</b>	/usuarios/(id)
<b>Método HTTP</b>	PUT
<b>Lógica de Negocio</b>	Permitir la modificación de los datos del usuario. Se validará que el correo electrónico no este en uso por otro usuario de la aplicación.
<b>Entrada</b>	<pre> {   "nombre": "string",   "correo": "string",   "edad": "integer",   "grado": "string" } </pre>
<b>Salida</b>	<pre> {   "mensaje": "Usuario actualizado correctamente",   "_id": "ObjectId",   "nombre": "string",   "correo": "string",   "edad": "integer",   "grado": "string" } </pre>

Eliminar Usuario	
<b>Actor(es)</b>	Usuarios, Administrador
<b>URL</b>	/usuarios/(id)
<b>Método HTTP</b>	DELETE
<b>Lógica de Negocio</b>	Eliminar un usuario de la base de datos. Se validará que el usuario exista antes de su eliminación.
<b>Entrada</b>	<pre> {   "_id": "ObjectId" } </pre>
<b>Salida</b>	<pre> {   "mensaje": "Usuario eliminado correctamente" } </pre>

## Servicio de Gestión de Flashcards

**Tipo de servicio:** Entidad

**Responsable:** Ramos Arévalos Ricardo

**Operaciones expuestas:**

- Crear Flashcard
- Obtener Flashcards
- Actualizar Flashcard
- Eliminar Flashcard

Crear Flashcard	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/flashcards/crear
<b>Método HTTP</b>	POST
<b>Lógica de Negocio</b>	Permite a los usuarios crear una nueva flashcard dentro de un temario específico.
<b>Entrada</b>	<pre>{   "id_temario": "ObjectId",   "pregunta": "string",   "respuesta": "string" }</pre>
<b>Salida</b>	<pre>{   "_id": "ObjectId",   "id_temario": "ObjectId",   "pregunta": "string",   "respuesta": "string" }</pre>

Obtener Flashcards	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/flashcards/(id_temario)
<b>Método HTTP</b>	GET
<b>Lógica de Negocio</b>	Obtiene todas las flashcards asociadas a un temario específico.
<b>Entrada</b>	<pre>{   "id_temario": "ObjectId" }</pre>
<b>Salida</b>	<pre>[   {     "_id": "ObjectId",     "id_temario": "ObjectId",     "pregunta": "string",     "respuesta": "string"   } ]</pre>



	} ]
--	--------

Actualizar Flashcard	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/flashcards/(id)
<b>Método HTTP</b>	PUT
<b>Lógica de Negocio</b>	Permite modificar el contenido de una flashcard específica.
<b>Entrada</b>	{ "pregunta": "string", "respuesta": "string" }
<b>Salida</b>	{ "mensaje": "Flashcard actualizada correctamente", "_id": "ObjectId", "pregunta": "string", "respuesta": "string" }

Eliminar Flashcard	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/flashcards/(id)
<b>Método HTTP</b>	DELETE
<b>Lógica de Negocio</b>	Elimina una flashcard específica. Se validará que la flashcard exista antes de su eliminación.
<b>Entrada</b>	{ "_id": "ObjectId" }
<b>Salida</b>	{ "mensaje": "Flashcard eliminada correctamente" }

## Servicio de Gestión de Temarios

**Tipo de servicio:** Entidad

**Responsable:** Alfaro Herrera Jesús Antonio

**Operaciones expuestas:**

- Crear Temario

- Obtener Temarios
- Actualizar Temario
- Eliminar Temario

Crear Temario	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/ temarios /crear
<b>Método HTTP</b>	POST
<b>Lógica de Negocio</b>	Permite a los usuarios crear un nuevo temario.
<b>Entrada</b>	{ <div>"nombre": "string",  "descripcion": "string"</div> }
<b>Salida</b>	{ <div>"_id": "ObjectId",  "nombre": "string",  "descripcion": "string"</div> }

Obtener Temarios	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/ temarios /(id)
<b>Método HTTP</b>	GET
<b>Lógica de Negocio</b>	Obtiene la información de un temario específico.
<b>Entrada</b>	{ <div>"_id": "ObjectId"</div> }
<b>Salida</b>	{ <div>"_id": "ObjectId",  "nombre": "string",  "descripcion": "string"</div> }

Actualizar Temario	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/ temarios /(id)
<b>Método HTTP</b>	PUT
<b>Lógica de Negocio</b>	Permite modificar la información de un temario.
<b>Entrada</b>	{ <div>"nombre": "string",  "descripcion": "string"</div> }

	}
<b>Salida</b>	{ "mensaje": "Temario actualizado correctamente" }

Eliminar Temario	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/ temarios /(id)
<b>Método HTTP</b>	DELETE
<b>Lógica de Negocio</b>	Elimina un temario específico.
<b>Entrada</b>	{ "_id": "ObjectId" }
<b>Salida</b>	{ "mensaje": "Temario eliminado correctamente" }

## Servicio de Autenticación

**Tipo de servicio:** Entidad

**Responsable:** Ramos Arévalos Ricardo

**Operaciones expuestas:**

- Iniciar Sesión
- Cerrar Sesión

Iniciar Sesión	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/auth/login
<b>Método HTTP</b>	POST
<b>Lógica de Negocio</b>	Permite a los usuarios autenticarse en la aplicación mediante sus credenciales. Si las credenciales son correctas, se genera un token de autenticación.
<b>Entrada</b>	{ "email": "string", "password": "string" }
<b>Salida</b>	{ "token": "string", "usuario": {

	<pre>"_id": "ObjectId", "nombre": "string", "email": "string" } }</pre>
Cerrar Sesión	
<b>Actor(es)</b>	Usuario autenticado
<b>URL</b>	/auth/logout
<b>Método HTTP</b>	POST
<b>Lógica de Negocio</b>	Revoca el token de autenticación para cerrar sesión de manera segura.
<b>Entrada</b>	<pre>{     "token": "string" }</pre>
<b>Salida</b>	<pre>{     "mensaje": "Sesión cerrada correctamente" }</pre>

## Servicio de Gestión de Categorías

**Tipo de servicio:** Entidad

**Responsable:** Piñones Ramos Said Rafael

**Operaciones expuestas:**

- Crear Categoría
- Obtener Categorías

Crear Categoría	
<b>Actor(es)</b>	Administrador
<b>URL</b>	/categorias/crear
<b>Método HTTP</b>	POST
<b>Lógica de Negocio</b>	Permite registrar una nueva categoría para organizar temarios o flashcards.
<b>Entrada</b>	<pre>{     "nombre": "string",     "descripcion": "string" }</pre>
<b>Salida</b>	<pre>{     "_id": "ObjectId",     "nombre": "string",     "descripcion": "string" }</pre>

Obtener Categorías	
<b>Actor(es)</b>	Usuario registrado
<b>URL</b>	/categorias
<b>Método HTTP</b>	GET
<b>Lógica de Negocio</b>	Retorna la lista de categorías disponibles en la aplicación.
<b>Entrada</b>	NA
<b>Salida</b>	<pre>[   {     "_id": "ObjectId",     "nombre": "string",     "descripcion": "string"   } ]</pre>