



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICO

PROGRAMACIÓN WEB 2

1er Avance

Equipo

Melany Arleth Jiménez Gómez 1904119

Denisse Alejandra Cardoza Pezina 1896868

Mario Asael Martínez Bocanegra 2031920

12 de Septiembre del 2023

ÍNDICE

Contenido

ÍNDICE	1
Diseño de Base de Datos	2
1. Principales características y funcionalidades	2
2. Modelo de Datos	3
3. Documentación de Esquemas	4
4. Conclusión	6

Diseño de Base de Datos

La aplicación es una plataforma de comunicación en línea que permite a los usuarios registrarse, conectarse con amigos y participar en conversaciones individuales o grupales. Los usuarios pueden enviar mensajes de texto, realizar videollamadas, compartir archivos y crear grupos o subgrupos de chat. La aplicación también permite enviar invitaciones de amistad a otros usuarios.

1. Principales características y funcionalidades

Usuarios: Los usuarios se registran con su correo electrónico y contraseña. Pueden crear perfiles, agregar amigos y administrar su lista de contactos.

Conversaciones: Los usuarios pueden iniciar conversaciones individuales o grupales. Las conversaciones almacenan mensajes enviados entre usuarios y grupos.

Grupos y Subgrupos: Los grupos son espacios de chat donde múltiples usuarios pueden participar. Los subgrupos son conversaciones dentro de grupos más grandes.

Mensajes: Los usuarios pueden enviar mensajes de texto y archivos adjuntos en sus conversaciones. Los mensajes pueden ser de tipo directo (entre dos usuarios) o de grupo.

Videollamadas y Llamadas: Los usuarios pueden realizar videollamadas y llamadas de voz en las conversaciones individuales o grupales.

Invitaciones de Amigos: Los usuarios pueden enviar y recibir invitaciones de amistad.

2. Modelo de Datos

El modelo de datos se compone de las siguientes colecciones en MongoDB:

users (usuarios):

Campos: _id, mail, username, password, Friends

groups (grupos):

Campos: _id, name, participants, messages, subgroups, isPublic

subgroups (subgrupos):

Campos: _id, name, participants, group

messages (mensajes):

Campos: _id, author, content, date, type, contentType, filename, isEncrypted

conversations (conversaciones):

Campos: _id, participants, messages

friendinvitations (invitaciones de amigos):

Campos: _id, receiver, sender

Este modelo de datos y descripción forman la base para desarrollar una aplicación de mensajería y comunicación similar a Discord utilizando MongoDB como base de datos para almacenar y gestionar la información de los usuarios y sus interacciones.

3. Documentación de Esquemas

Usuarios (users):

- **_id**: Identificador único del usuario ([ObjectId](#)).
- **mail**: Correo electrónico del usuario ([String](#)).
- **username**: Nombre del usuario ([String](#)).
- **password**: Contraseña del usuario ([Hash de contraseña](#), [String](#)).
- **friends**: Lista de amigos del usuario ([Array de ObjectId](#)).

Grupos (groups):

- **_id**: Identificador único del grupo ([ObjectId](#)).
- **name**: Nombre del grupo ([String](#)).
- **participants**: Lista de participantes en el grupo ([Array de ObjectId](#)).
- **messages**: Lista de mensajes en el grupo ([Array de ObjectId](#)).
- **subgroups**: Lista de subgrupos dentro del grupo ([Array de ObjectId](#)).
- **isPublic**: Indicador de si el grupo es público o privado ([Boolean](#)).

Subgrupos (subgroups):

- **_id**: Identificador único del subgrupo ([ObjectId](#)).
- **name**: Nombre del subgrupo ([String](#)).
- **participants**: Lista de participantes en el subgrupo ([Array de ObjectId](#)).
- **group**: Identificador del grupo al que pertenece el subgrupo ([ObjectId](#)).

Ejemplo de usuario

```
{
  "_id": ObjectId(' '),
  "mail": "d@gmail.com",
  "username": "Denisse45",
  "password":
"$2a$10$E33AX4Pz5U",
  "friends": [
    ObjectId(' '),
    ObjectId(' ')
  ]
}
```

Ejemplo de grupo

```
{
  "_id": ObjectId(' '),
  "name": "Crazy girls",
  "participants": [
    ObjectId(' '),
    ObjectId(' ')
  ],
  "messages": [
    ObjectId(' '),
    ObjectId(' ')
  ],
  "subgroups": [
    ObjectId(' '),
    ObjectId(' ')
  ],
  "isPublic": false
}
```

Ejemplo de subgrupo

```
{
  "_id": ObjectId(' '),
  "name": "Prueba 2",
  "participants": [
    ObjectId(' '),
    ObjectId(' ')
  ],
  "group": ObjectId(' ')
}
```

Mensajes (messages):

- **_id**: Identificador único del mensaje ([ObjectId](#)).
- **author**: Identificador único del autor del mensaje ([ObjectId](#)).
- **content**: Contenido del mensaje ([String](#)).
- **date**: Fecha y hora del mensaje ([Fecha y hora](#)).
- **type**: Tipo de mensaje ([String](#), puede ser "DIRECT" o "GRUPO").
- **contentType**: Tipo de contenido del mensaje ([String](#), por ejemplo, "text").
- **filename**: Nombre de archivo si el mensaje contiene un archivo adjunto ([String](#)).
- **isEncrypted**: Indicador de si el mensaje está encriptado ([Boolean](#)).

Ejemplo de Mensajes

```
{
  "_id": ObjectId(' '),
  "author": ObjectId(' '),
  "content": "U2FsdGVkX1 ",

  "date": "2023-05-
28T12:20:54.591+00:00",

  "type": "DIRECT",
  "contentType": "text",
  "filename": " ",
  "isEncrypted": true
}
```

Conversaciones (conversations):

- **_id**: Identificador único de la conversación ([ObjectId](#)).
- **participants**: Lista de participantes en la conversación ([Array de ObjectId](#)).
- **messages**: Lista de mensajes en la conversación ([Array de ObjectId](#)).

Ejemplo de conversación

```
{
  "_id": ObjectId(' '),

  "participants": [
    ObjectId(' '),
    ObjectId(' ')
  ],
  "messages": [
    ObjectId(' ')
  ]
}
```

Invitaciones de Amigos (friendinvitations):

- **_id**: Identificador único de la invitación ([ObjectId](#)).
- **Receiver**: Identificador único del receptor de la invitación ([ObjectId](#)).
- **Sender**: Identificador único del que envía la invitación ([ObjectId](#)).

Ejemplo de invitación

```
{
  "_id": ObjectId(' '),

  "Receiver": ObjectId(' '),

  "Sender": ObjectId(' '),

}
```

4. Conclusión

En resumen, hemos creado una documentación completa para la base de datos de una aplicación de mensajería, similar a Discord, que utiliza MongoDB. Esta documentación desempeña un papel fundamental en el desarrollo y mantenimiento de la aplicación porque ayuda a los miembros del equipo a entender cómo funciona la base de datos. Además, contribuye a que la aplicación sea eficiente y cumpla con las necesidades y expectativas de los usuarios.