

BASE DE DATOS DISCORD

PROYECTO

Base de Datos



Jose Fortunati Ruiz
Grado Superior DAW
IES ALIXAR
Fecha

ÍNDICE

1.Descripción	3
2.Modelo Entidad Relación	4
3.Modelo Relacional	8
Paso a Tablas (WORKBENCH)	9
4.Creación de Tablas y Carga Masiva de Datos	9
5.Consultas	10
Consultas Multitabla	10
Subconsultas	11
4.Enlace GitHub	12
5.CONCLUSIÓN	12

CONTENIDO

1.Descripción

Este proyecto está dirigido para la creación de una base de datos funcional con la libertad de elección del tema sobre el que se creará dicha base de datos.

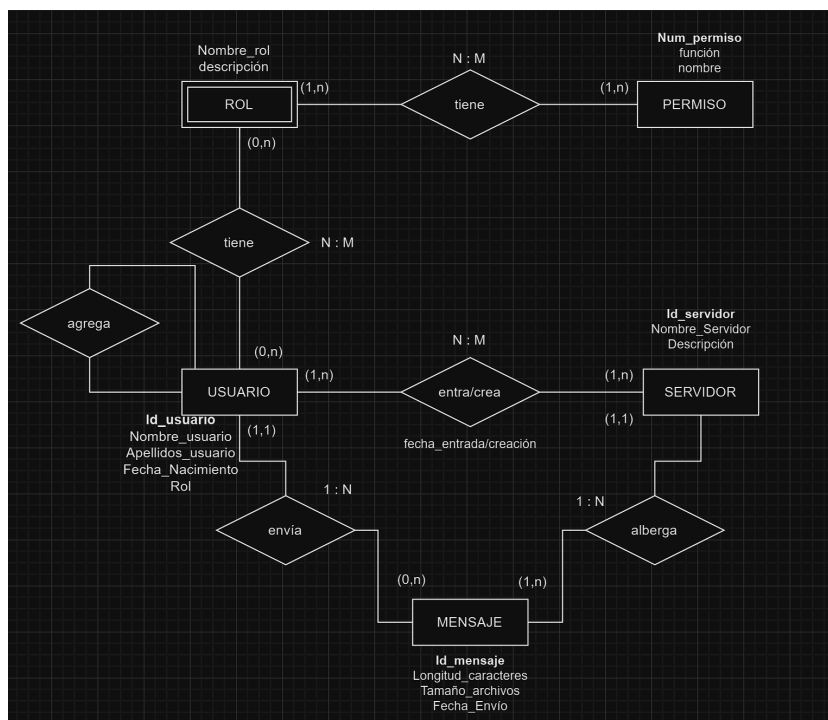
En mi caso he elegido crear la base de datos en torno a la famosa aplicación de comunicación Discord. Creo que es una aplicación muy completa que cumple con todas las necesidades de los usuarios como un apartado de mensajería individual o la creación de grupos o servidores en los que pueden entrar varios usuarios. Además de mensajes, esta aplicación también permite hacer llamadas entre usuarios para que se comuniquen entre ellos.

Si bien es cierto que esta aplicación está más dirigida para un sector específico como es el de los videojuegos debido a su simplicidad, esta se puede usar también para otros ambientes como el laboral.

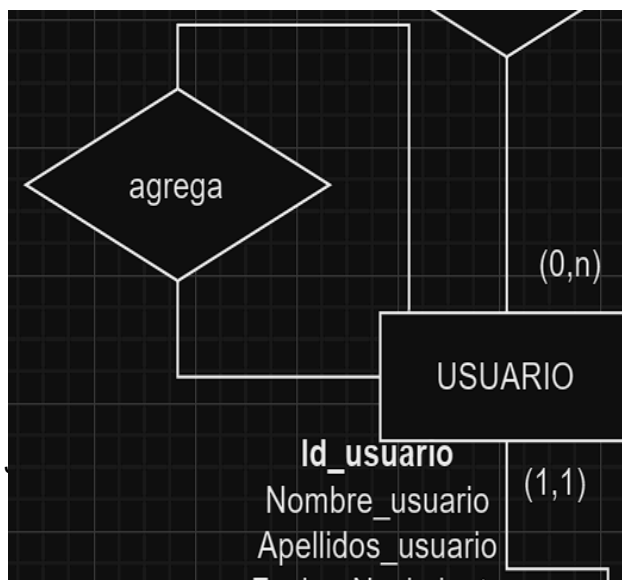
Su antecesor más conocido podría ser Skype, una aplicación que de hecho sigue en funcionamiento pero que se ha visto totalmente eclipsada por Discord debido a la gran cantidad de mejoras que tiene respecto a la anterior mencionada.

A lo largo de este proyecto veremos un Modelo Entidad Relación en el que se pueden ver las entidades sobre las que se pretende almacenar información, el Modelo Relacional y el paso a tablas en Workbench para ver las tablas de forma gráfica y la generación del script de la base de datos

2. Modelo Entidad Relación



En este Modelo Entidad Relación se presentan todas las entidades sobre las que se quiere almacenar la información además de las relaciones entre estas y los campos que posee cada entidad.

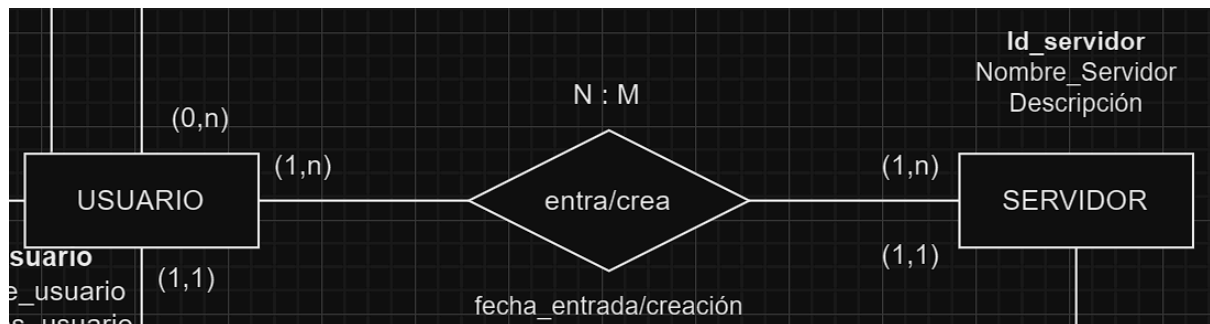


En Discord, cada cuenta que se crea se registra como un nuevo usuario al cual se le asigna un identificador en forma de código recogido como la ID del usuario.

Por otra parte, para la creación de una cuenta se necesitan ciertos datos como el nombre, los apellidos o la fecha de nacimiento del usuario.

Los usuarios pueden enviar solicitudes de amistad para así agregarse y poder enviarse mensajes.

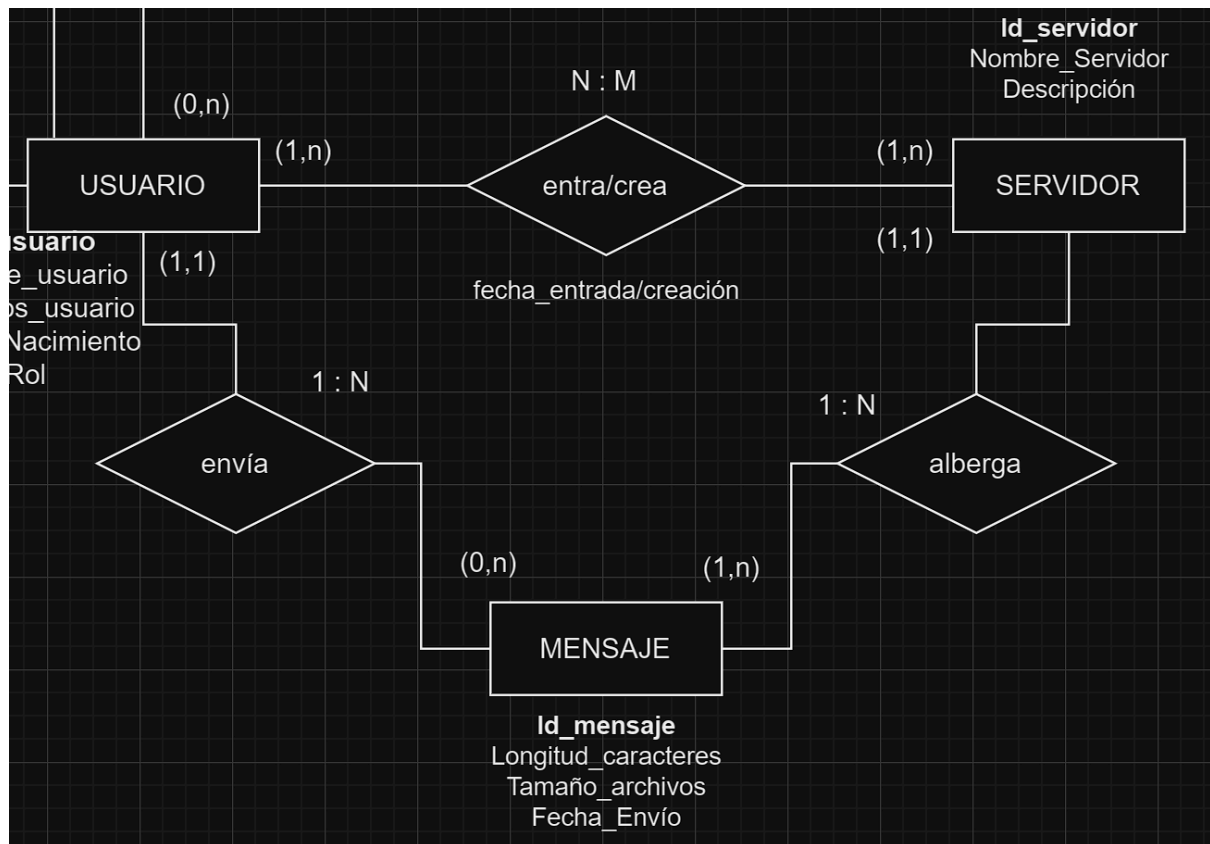
Por último, cada usuario tiene un rol en cada servidor pero ya entraremos en detalles después.



Cada usuario puede entrar a un servidor ya creado por otro usuario o puede crear un servidor propio. Si un usuario entra a un servidor se guardará la fecha de entrada de dicho usuario a ese servidor mientras que si es un usuario el que crea el servidor, se almacenará la fecha de creación del servidor.

De cada servidor se guarda su nombre y una pequeña descripción sobre éste. Además, Discord proporciona un código para cada servidor para así poder identificarlos siendo esto el ID del servidor.

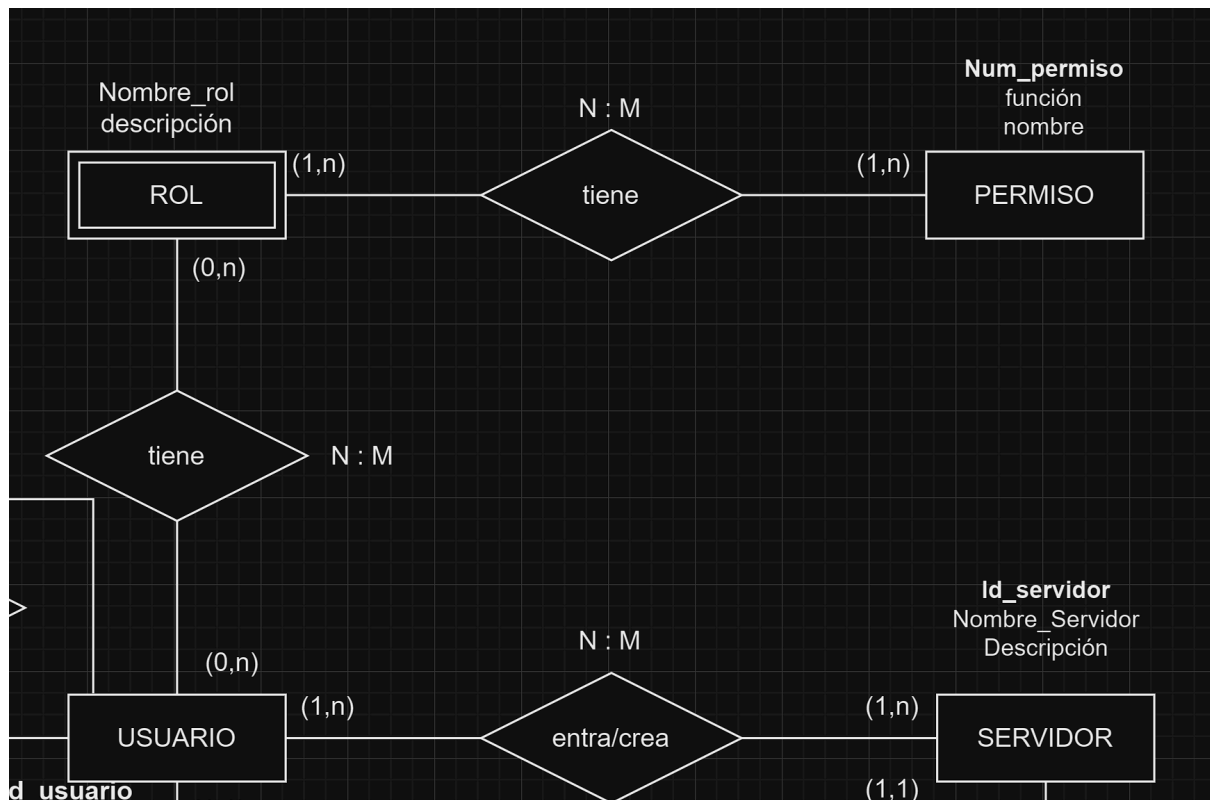
Como ya mencioné antes, cada usuario tiene un rol en función del servidor. El usuario que crea el servidor tendrá el rol de creador del servidor y podrá proporcionar un rol a cada usuario que entre en el servidor determinando con este las funciones que los usuarios desempeñan dentro del servidor. Los permisos y los roles se verán con mayor detalle después.



La principal función por la que se caracteriza discord es por su velocidad en el envío de mensajes y la facilidad que tienen los usuarios para interactuar con estos.

Los mensajes son enviados por los usuarios y de estos se guarda la longitud de los caracteres que incluye los mensajes, el tamaño de los archivos enviados en los mensajes y la fecha de envío del mensaje. Cada mensaje es único por tanto cada uno de estos tiene un código que lo identifica siendo este el ID del mensaje.

Cada mensaje se almacena en un servidor. Las conversaciones usuario-usuario se almacenan en un servidor propio de Discord al que solo pueden acceder los 2 usuarios involucrados en la conversación, si se trata de un servidor en el que hay más personas pues se alberga en un servidor ya creado por el usuario.



Por último, vamos a hablar de los roles que los usuarios pueden poseer y para qué sirven cada uno de estos.

Cada rol tiene una descripción que indica las características de dicho rol y las funciones que puede desempeñar el usuario que lo posea, además de por supuesto tener un nombre de rol que puede ser creado por el usuario. Este nombre es de libre elección por parte del usuario por lo que puede repetirse en más de un usuario o diferentes roles pueden tener los mismos permisos.

De cada permiso se quiere guardar la función que cumple cada uno de estos , el nombre que recibe y el número del permiso que funciona como identificador del permiso.

3. Modelo Relacional

USUARIO : (**ID_USUARIO** , Nombre_usuario , Apellidos_usuario , Fecha_Nacimiento , Rol)

- **ID_USUARIO (PK)**

SERVIDOR : (**ID_SERVIDOR** , Nombre_servidor , Descripción)

- **ID_SERVIDOR (PK)**

MENSAJE : (**ID_MENSAJE** , Longitud_caracateres , Tamaño_archivos , Fecha_envío)

- **ID_MENSAJE (PK)**

ROL : (Nombre_rol , descripción , **ID_USUARIO** , **NUM_PERMISO**)

- **ID_USUARIO (FK** referencia a **TABLA USUARIO)**
- **NUM_PERMISO (FK** referencia a **TABLA PERMISO)**

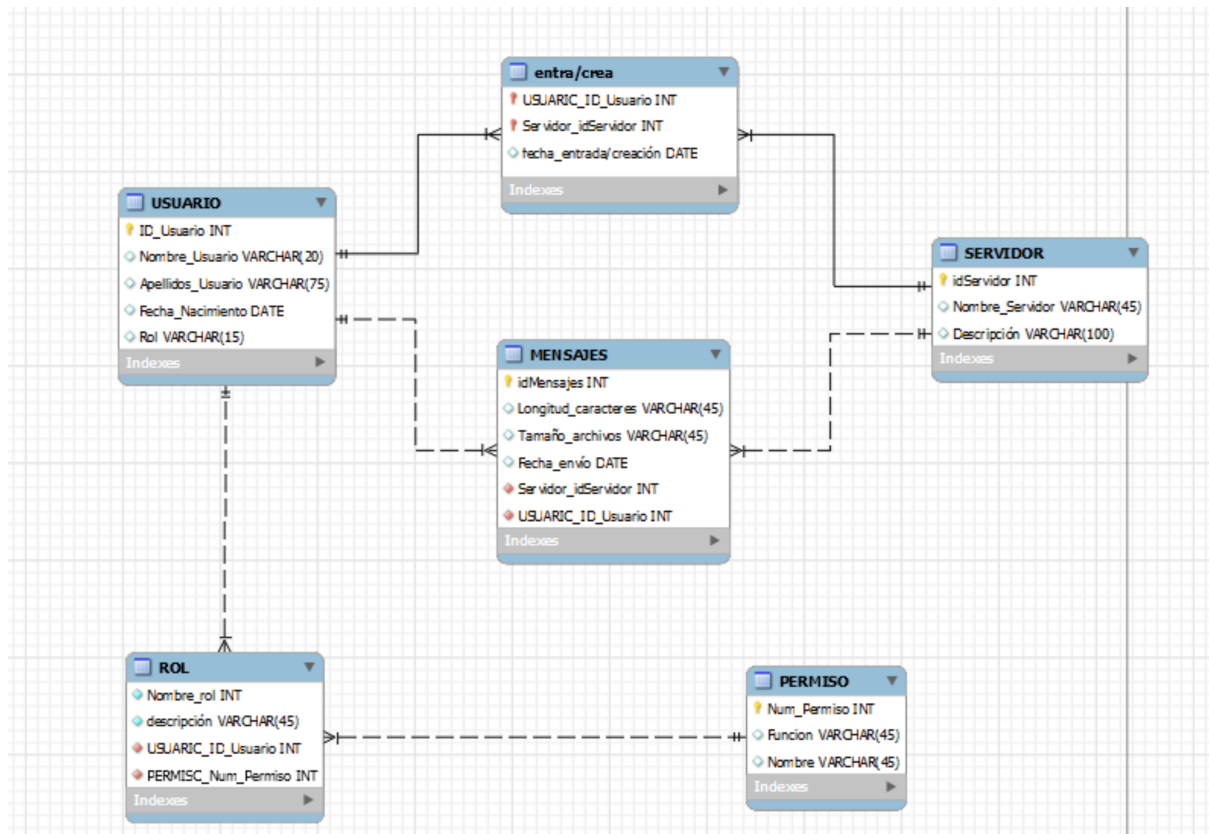
PERMISO : (**NUM_PERMISO** , Función , nombre)

- **NUM_PERMISO (PK)**

LLAMADA : (**ID_Llamada** , Fecha_inicio , Fecha_fin)

- **ID_LLAMADA (PK)**

Paso a Tablas (WORKBENCH)



4. Creación de Tablas y Carga Masiva de Datos

Para la creación de las tablas he tenido que hacer a mano el script debido a que el proyecto se ha visto sujeto a cambios. En el tiempo que hemos tenido para poblar la base de datos he visto errores en el proyecto que me han permitido optimizar el proyecto por lo que he generado desde cero la creación de las tablas.

Realizar la carga de datos no ha sido una tarea fácil, he tenido errores de formato que me han hecho tener que volver a generar los datos pero también me ha servido para aprender a ver como funciona el formato SQL.

En mi caso, la carga de datos la he hecho con formato SQL ya que la página que he utilizado para la generación de datos, que es Mockaroo, permitía generar código SQL que luego podía ejecutar como un script.

Gracias a lo estudiado durante el curso he sabido desde un principio que estaba haciendo y salvo pequeños errores en el formato de los datos que he sabido solucionar por mi cuenta, he conocido a detalle cómo funciona la creación de tablas y la carga de datos en SQL.

Los scripts de generación de tablas y de carga de datos están en mi github(Enlace más adelante)

5.Consultas

Gracias a lo que hemos estudiado podemos hacer consultas a nuestra base de datos para así trabajar con ella y hacer de ésta una base de datos funcional.

En mi caso he generado 5 consultas multitabla y 4 subconsultas.

Consultas Multitabla

1.Listar los usuarios que enviaron mensajes en un servidor específico junto con el nombre del servidor y el rol que poseen:

```
SELECT u.nombre_usuario, s.nombre_servidor, r.nombre_rol
FROM usuarios u
INNER JOIN mensaje m ON u.id_usuario = m.id_usuarioEnvia
INNER JOIN servidor s ON m.id_servidorEnviado = s.id_Servidor
INNER JOIN rol r ON u.id_rolposee = r.id_rol
WHERE s.nombre_servidor = 'NombreServidorEspecifico';
```

Grilla		ABC nombre_usuario ▼	ABC nombre_servidor ▼	ABC nombre_rol ▼
	1	Oscar	Servidor_A	Usuario

2. Mostrar todos los mensajes enviados por un usuario en particular junto con su longitud y tamaño de archivos:

```
SELECT m.id_mensaje, m.Longitud_Caracteres, m.Tamano_Archivos
FROM mensaje m
INNER JOIN usuarios u ON m.id_usuarioEnvia = u.id_usuario
WHERE u.nombre_usuario = 'Oscar';
```

	id_mensaje	Longitud_Caracteres	Tamano_Archivos
1	37	120	5120
2	87	100	2048
3	50	140	1024
4	100	150	4096

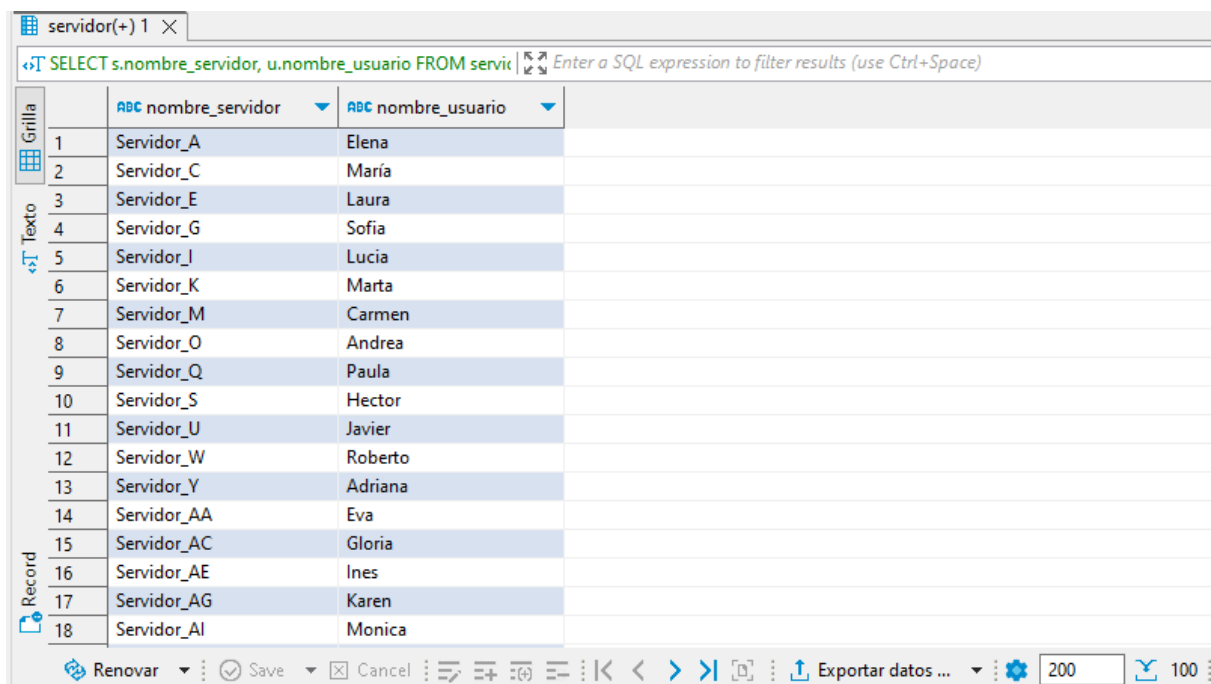
3. Listar todos los usuarios que poseen un cierto permiso junto con el nombre del permiso y el rol que poseen:

```
SELECT u.nombre_usuario, p.Nombre_Permiso, r.nombre_rol
FROM usuarios u
INNER JOIN rol r ON u.id_rolposee = r.id_rol
INNER JOIN permiso p ON r.id_rol = p.Num_permiso;
```

	nombre_usuario	Nombre_Permiso	nombre_rol
1	Elena	Ver canal	Administrador
2	María	Ver canal	Administrador
3	Laura	Ver canal	Administrador
4	Sofía	Ver canal	Administrador
5	Lucía	Ver canal	Administrador
6	Marta	Ver canal	Administrador
7	Carmen	Ver canal	Administrador
8	Andrea	Ver canal	Administrador
9	Paula	Ver canal	Administrador
10	Hector	Ver canal	Administrador
11	Javier	Ver canal	Administrador
12	Roberto	Ver canal	Administrador
13	Adriana	Ver canal	Administrador
14	Eva	Ver canal	Administrador
15	Gloria	Ver canal	Administrador
16	Ines	Ver canal	Administrador
17	Karen	Ver canal	Administrador
18	Monica	Ver canal	Administrador

4. Mostrar los servidores creados por usuarios que tengan el rol administrador:

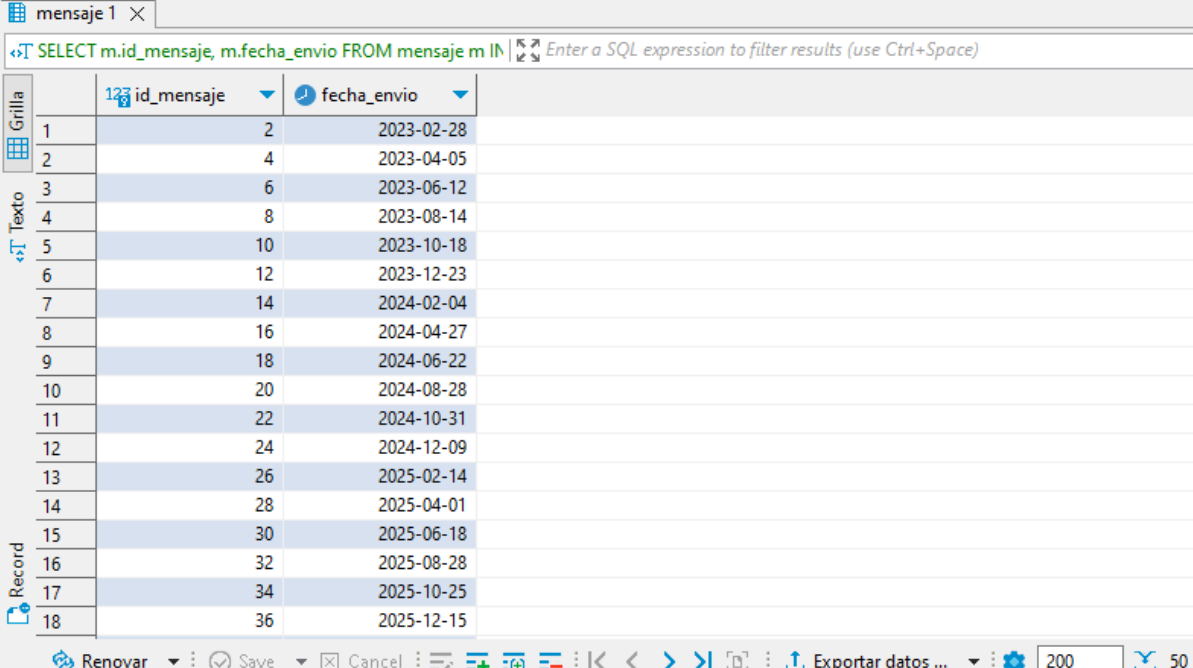
```
SELECT s.nombre_servidor, u.nombre_usuario  
FROM servidor s  
INNER JOIN usuarios u ON s.id_usuarioCreador = u.id_usuario  
INNER JOIN rol r ON u.id_rolposee = r.id_rol  
WHERE r.nombre_rol = 'Administrador';
```



	nombre_servidor	nombre_usuario
1	Servidor_A	Elena
2	Servidor_C	Maria
3	Servidor_E	Laura
4	Servidor_G	Sofia
5	Servidor_I	Lucia
6	Servidor_K	Marta
7	Servidor_M	Carmen
8	Servidor_O	Andrea
9	Servidor_Q	Paula
10	Servidor_S	Hector
11	Servidor_U	Javier
12	Servidor_W	Roberto
13	Servidor_Y	Adriana
14	Servidor_AA	Eva
15	Servidor_AC	Gloria
16	Servidor_AE	Ines
17	Servidor_AG	Karen
18	Servidor_AI	Monica

5. Listar los mensajes enviados en servidores creados por usuarios que tengan el rol "administrador":

```
SELECT m.id_mensaje, m.fecha_envio
FROM mensaje m
INNER JOIN servidor s ON m.id_servidorEnviado = s.id_Servidor
INNER JOIN usuarios u ON s.id_usuarioCreador = u.id_usuario
INNER JOIN rol r ON u.id_rolposee = r.id_rol
WHERE r.nombre_rol = 'Administrador';
```

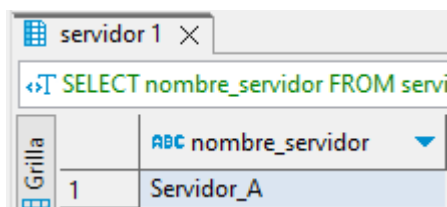


	id_mensaje	fecha_envio
1	2	2023-02-28
2	4	2023-04-05
3	6	2023-06-12
4	8	2023-08-14
5	10	2023-10-18
6	12	2023-12-23
7	14	2024-02-04
8	16	2024-04-27
9	18	2024-06-22
10	20	2024-08-28
11	22	2024-10-31
12	24	2024-12-09
13	26	2025-02-14
14	28	2025-04-01
15	30	2025-06-18
16	32	2025-08-28
17	34	2025-10-25
18	36	2025-12-15

Subconsultas

1. Obtener el nombre del servidor donde se envió el mensaje más reciente:

```
SELECT nombre_servidor  
FROM servidor  
WHERE id_Servidor = (SELECT id_servidorEnviado FROM mensaje ORDER BY  
fecha_envio DESC LIMIT 1);
```

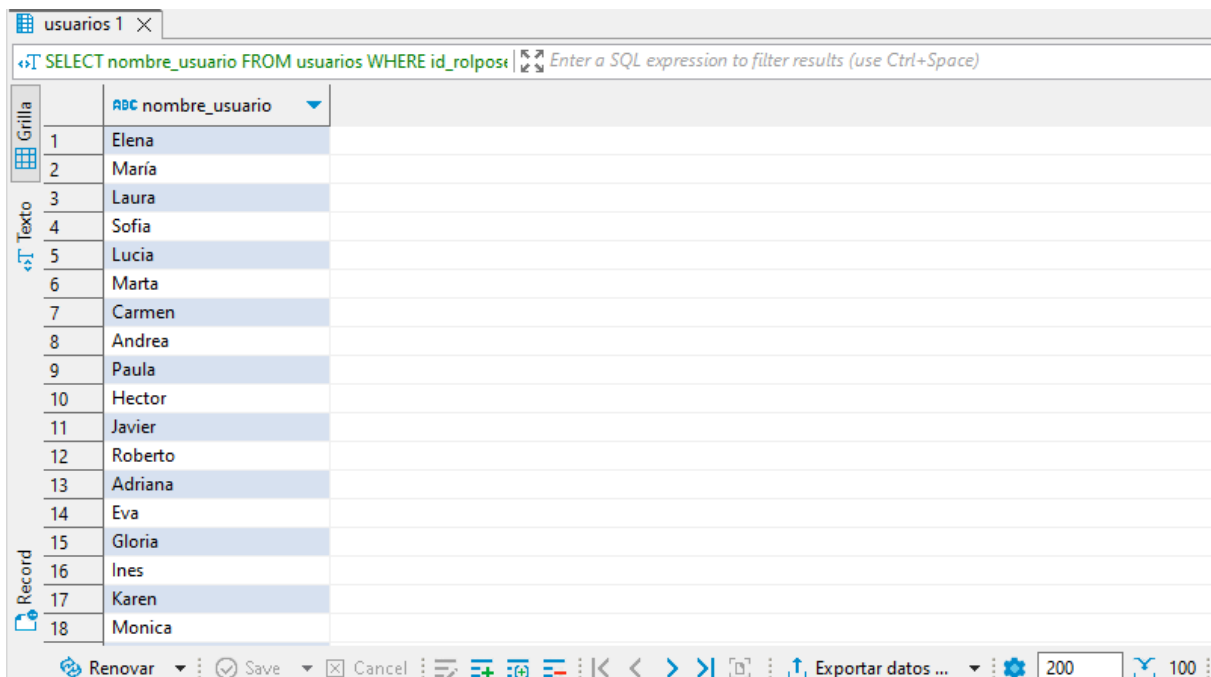


The screenshot shows a database window titled 'servidor 1'. The SQL query entered is 'SELECT nombre_servidor FROM servi'. The result is displayed in a table with one row containing the value 'Servidor_A'.

	ABC nombre_servidor
1	Servidor_A

2. Listar todos los usuarios que tienen un rol cuyo nombre contiene la palabra "admin":

```
SELECT nombre_usuario  
FROM usuarios  
WHERE id_rolposee IN (SELECT id_rol FROM rol WHERE nombre_rol LIKE '%admin%');
```

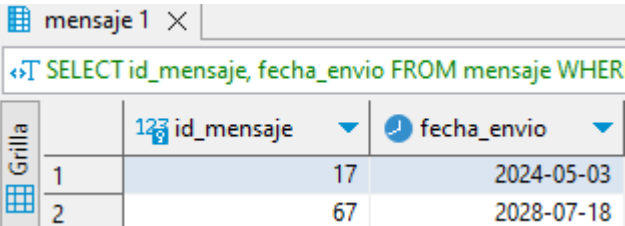


The screenshot shows a database window titled 'usuarios 1'. The SQL query entered is 'SELECT nombre_usuario FROM usuarios WHERE id_rolposee'. The result is displayed in a table with 18 rows, each containing a user name. The table is sorted alphabetically by name.

	ABC nombre_usuario
1	Elena
2	María
3	Laura
4	Sofía
5	Lucía
6	Marta
7	Carmen
8	Andrea
9	Paula
10	Hector
11	Javier
12	Roberto
13	Adriana
14	Eva
15	Gloria
16	Ines
17	Karen
18	Monica

3. Mostrar todos los mensajes enviados en servidores creados por el usuario más joven:

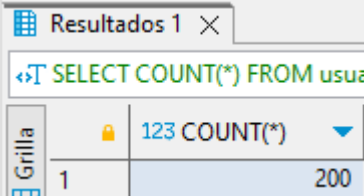
```
SELECT id_mensaje, fecha_envio
FROM mensaje
WHERE id_servidorEnviado IN (SELECT id_Servidor FROM servidor WHERE
id_usuarioCreador = (SELECT id_usuario FROM usuarios ORDER BY fecha_nacimiento
ASC LIMIT 1));
```



	id_mensaje	fecha_envio
1	17	2024-05-03
2	67	2028-07-18

4. Contar cuántos usuarios tienen el permiso número 3:

```
SELECT COUNT(*)
FROM usuarios
WHERE id_rolposee IN (SELECT id_rol FROM rol WHERE id_rol IN (SELECT id_rol FROM
permiso WHERE Num_permiso = 3));
```



	COUNT(*)
1	200

Gracias a las consultas podemos obtener información sobre la base de datos y ver los resultados de estas. No me ha supuesto mucho trabajo generar las bases de datos, el mayor problema ha sido plantear desde cero la consulta y determinar qué información quería recoger con la consulta.

6. Enlace GitHub

[jforru2508/ProyectoBDD: Proyecto Base De Datos \(github.com\)](https://github.com/jforru2508/ProyectoBDD)

7.CONCLUSIÓN

Con este trabajo he aprendido a saber cómo podría ser la gestión de información de una aplicación de mensajería y he conseguido comprender mejor el funcionamiento de la aplicación Workbench.

Por otra parte, encuentro este tipo de proyectos como algo muy importante en nuestra formación ya que es lo más próximo a lo que nos pueden llegar a pedir en el futuro en nuestros trabajos.

