

# BASE DE DATOS DISCORD

## PROYECTO

Base de Datos

---



Jose Fortunati Ruiz  
Grado Superior DAW  
IES ALIXAR  
12/02/24

# ÍNDICE

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

# 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.

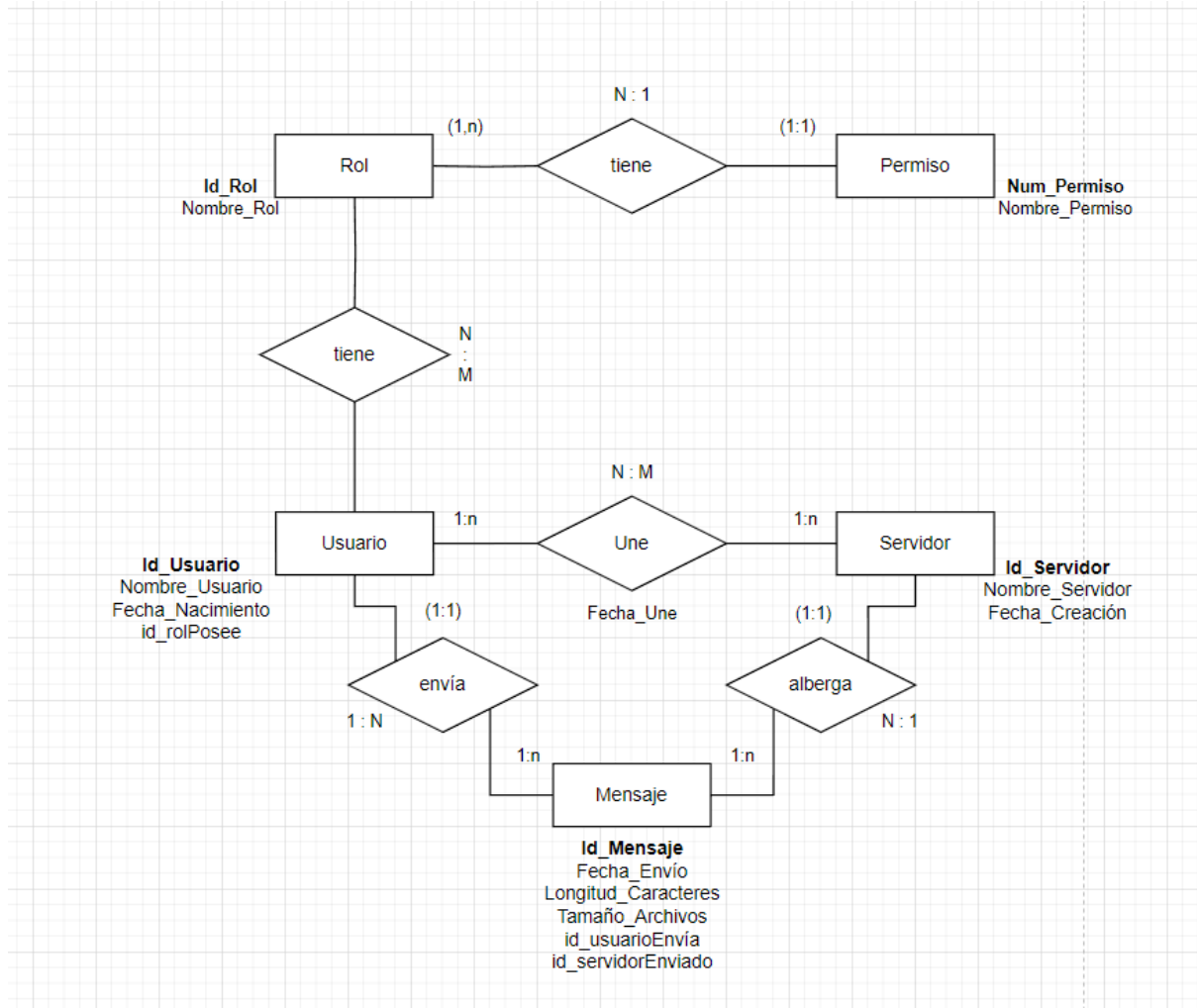
Si bien es cierto que esta aplicación está más dirigida para un sector específico como es el de los videojuegos debido a su simplicidad, esta se puede usar también para otros ambientes como el laboral.

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

PROYECTO DISCORD  
BASE DE DATOS || 2023-2024  
VERSIÓN II

## 2.Modelo Entidad Relación



### 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)**

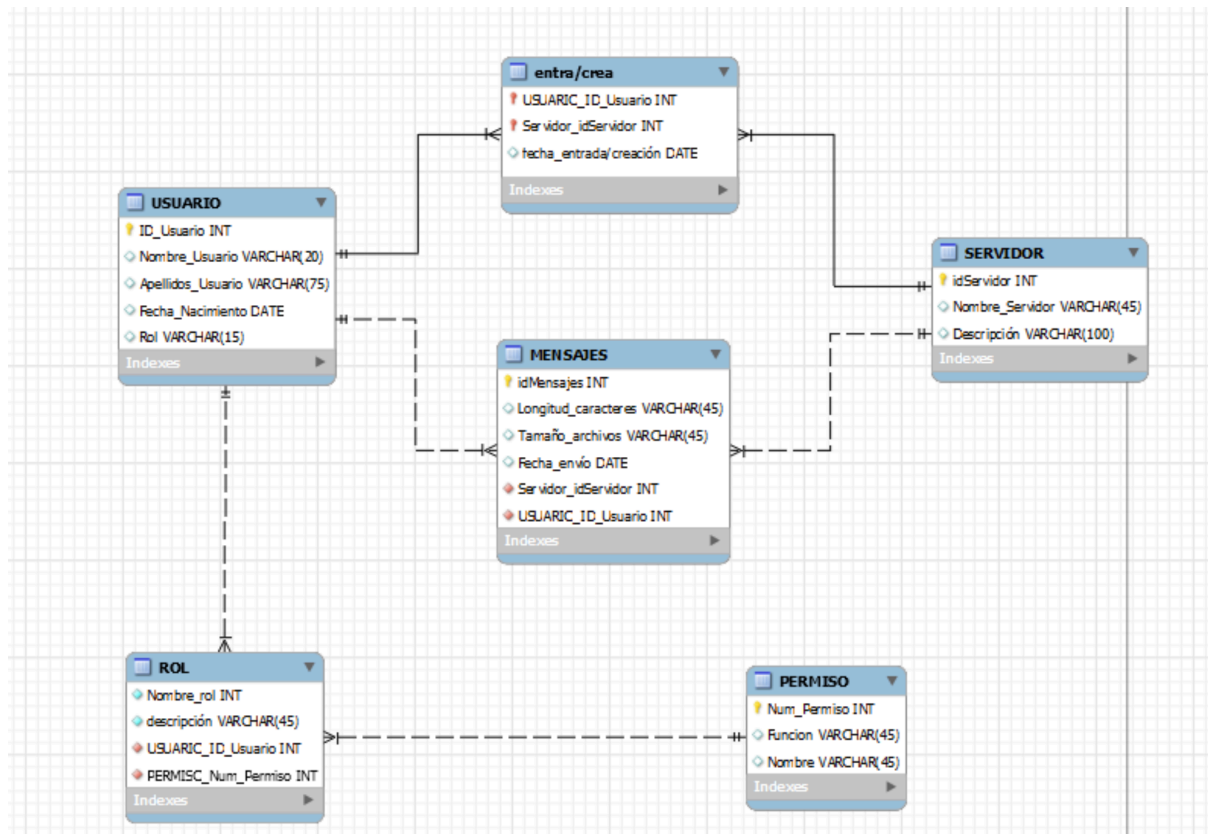
LLAMADA : ( **ID\_Llamada** , Fecha\_inicio , Fecha\_fin )

- **ID\_LLAMADA (PK)**

PROYECTO DISCORD  
BASE DE DATOS || 2023-2024  
VERSIÓN II

---

## 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 = 'Servidor_A';
```

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

PROYECTO DISCORD  
BASE DE DATOS || 2023-2024  
VERSIÓN II

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';
```

SQL Query: `SELECT m.id_mensaje, m.Longitud_Caracteres, m.Tamano_Archivos`

	id_mensaje	Longitud_Caracteres	Tamano_Archivos
1	37	120	5120
2	50	140	1024
3	87	100	2048
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;
```

SQL Query: `SELECT u.nombre_usuario, p.Nombre_Permiso, r.nombre_rol`

	nombre_usuario	Nombre_Permiso	nombre_rol
984	Sofía	Gestionar canal	Usuario
985	Victoria	Gestionar canal	Usuario
986	Gabriela	Gestionar canal	Usuario
987	Constanza	Gestionar canal	Usuario
988	Carla	Gestionar canal	Usuario
989	Rocío	Gestionar canal	Usuario
990	María	Gestionar canal	Usuario
991	Luz	Gestionar canal	Usuario
992	Laura	Gestionar canal	Usuario
993	Julieta	Gestionar canal	Usuario
994	Martina	Gestionar canal	Usuario
995	Valentina	Gestionar canal	Usuario
996	Paula	Gestionar canal	Usuario
997	Natalia	Gestionar canal	Usuario
998	Catalina	Gestionar canal	Usuario
999	Sofía	Gestionar canal	Usuario
1000	Antonella	Gestionar canal	Usuario

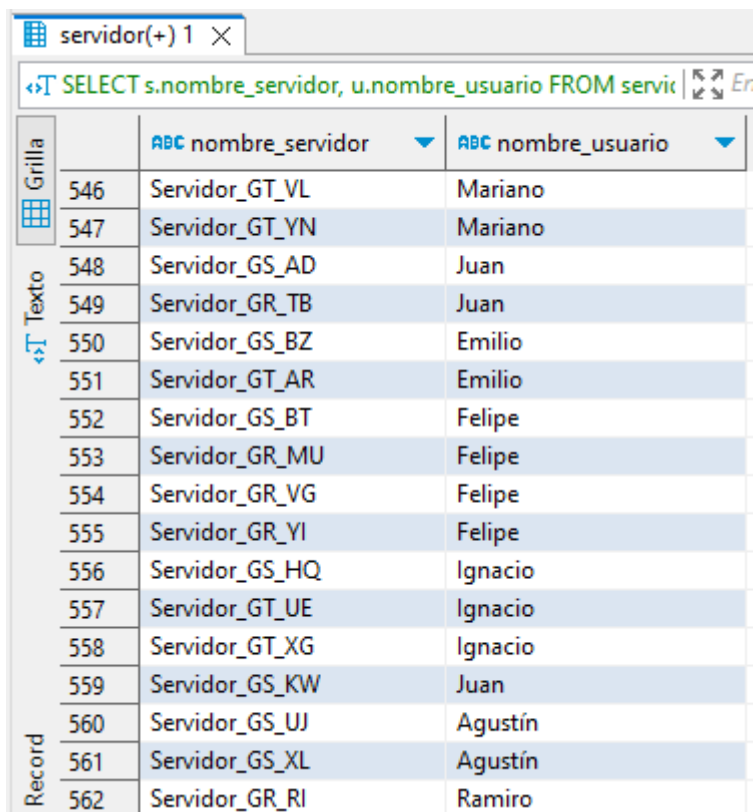


PROYECTO DISCORD  
BASE DE DATOS || 2023-2024  
VERSIÓN II

---

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';
```



Grilla	nombre_servidor	nombre_usuario
546	Servidor_GT_VL	Mariano
547	Servidor_GT_YN	Mariano
548	Servidor_GS_AD	Juan
549	Servidor_GR_TB	Juan
550	Servidor_GS_BZ	Emilio
551	Servidor_GT_AR	Emilio
552	Servidor_GS_BT	Felipe
553	Servidor_GR_MU	Felipe
554	Servidor_GR_VG	Felipe
555	Servidor_GR_YI	Felipe
556	Servidor_GS_HQ	Ignacio
557	Servidor_GT_UE	Ignacio
558	Servidor_GT_XG	Ignacio
559	Servidor_GS_KW	Juan
560	Servidor_GS_UJ	Agustín
561	Servidor_GS_XL	Agustín
562	Servidor_GR_RI	Ramiro

PROYECTO DISCORD  
BASE DE DATOS || 2023-2024  
VERSIÓN II

---

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';
```

mensaje 1 X			
<> SELECT m.id_mensaje, m.fecha_envio FROM mensaje r			
Grilla		id_mensaje	fecha_envio
	34	68	2028-08-03
Texto	35	70	2028-10-20
	36	72	2028-12-04
<>	37	74	2029-02-17
	38	76	2029-04-05
Record	39	78	2029-06-18
	40	80	2029-08-28
	41	82	2029-10-25
	42	84	2029-12-15
	43	86	2030-02-22
	44	88	2030-04-07
	45	90	2030-06-09
	46	92	2030-08-03
	47	94	2030-10-20
	48	96	2030-12-04
	49	98	2031-02-17
	50	100	2031-04-05

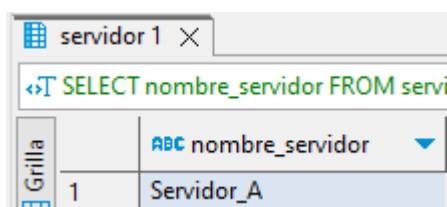
PROYECTO DISCORD  
BASE DE DATOS || 2023-2024  
VERSIÓN II

---

## 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);
```

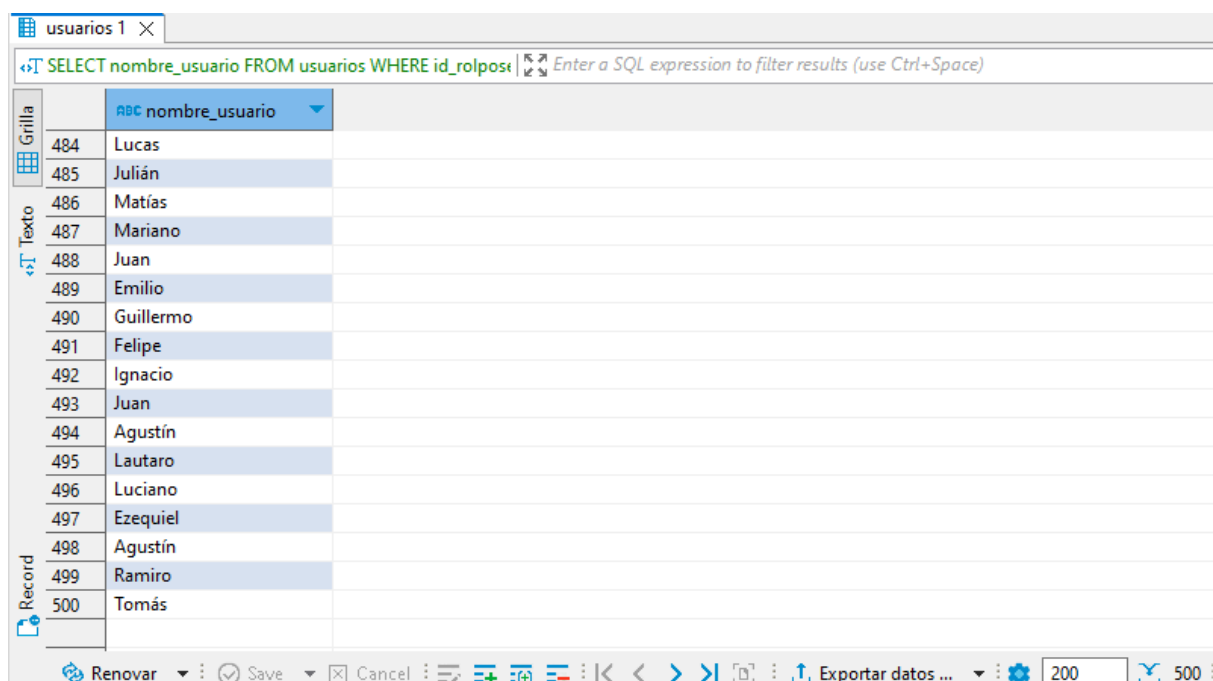


The screenshot shows a database window titled 'servidor 1'. The SQL query entered is 'SELECT nombre\_servidor FROM servi'. The result is displayed in a table with one row containing the value 'Servidor\_A'.

Grilla	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%');
```

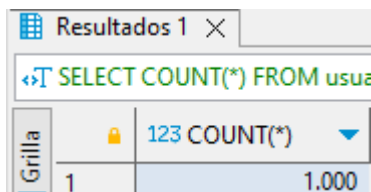


The screenshot shows a database window titled 'usuarios 1'. The SQL query entered is 'SELECT nombre\_usuario FROM usuarios WHERE id\_rolposee'. The result is displayed in a table with 16 rows, each containing a user's name.

Grilla	ABC nombre_usuario
484	Lucas
485	Julian
486	Matias
487	Mariano
488	Juan
489	Emilio
490	Guillermo
491	Felipe
492	Ignacio
493	Juan
494	Agustin
495	Lautaro
496	Luciano
497	Ezequiel
498	Agustin
499	Ramiro
500	Tomás

3. 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));
```



Grilla	1	COUNT(*)
	1	123

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.