

## Diseño de la Base de Datos

### Base de Datos Principal

Se comienza el análisis de la Base de Datos que pueda resolver la problemática planteada, se decide inicialmente trabajar en la Base de Datos principal, donde se crean 3 tablas:

- **Roles:** La tabla donde se manejan los roles que existen en el programa, estos roles son: **Administrador** y **Operador**, los cuales se describen en la definición del proyecto. Esta tabla posee dos campos:
  1. **id:** Un campo autoincremental que sirve como llave primaria en esta tabla y con la que se relaciona la tabla **User** que se describe más adelante con la tabla **Roles**.
  2. **str\_name:** Un campo de tipo TEXT que contiene el nombre de cada rol
- **User:** La tabla donde se almacena la información de cada usuario registrado en el programa, esta tabla posee los siguientes campos:
  1. **id:** Un campo autoincremental que sirve como llave primaria en esta tabla y con la que se relaciona la tabla **Drawings** descrita más adelante con la tabla **User**.
  2. **id\_role:** Un campo de tipo INT, utilizado como llave foránea, que hace referencia al rol que posee dicho usuario en el programa.
  3. **str\_userName:** Un campo de tipo VARCHAR el cual contiene en nombre de usuario que sirve para autenticarse. Ya que éste campo debe ser único, se utiliza el tipo de dato VARCHAR, ya que se encuentra un problema con el tipo de dato TEXT al declarar que éste debe ser único.

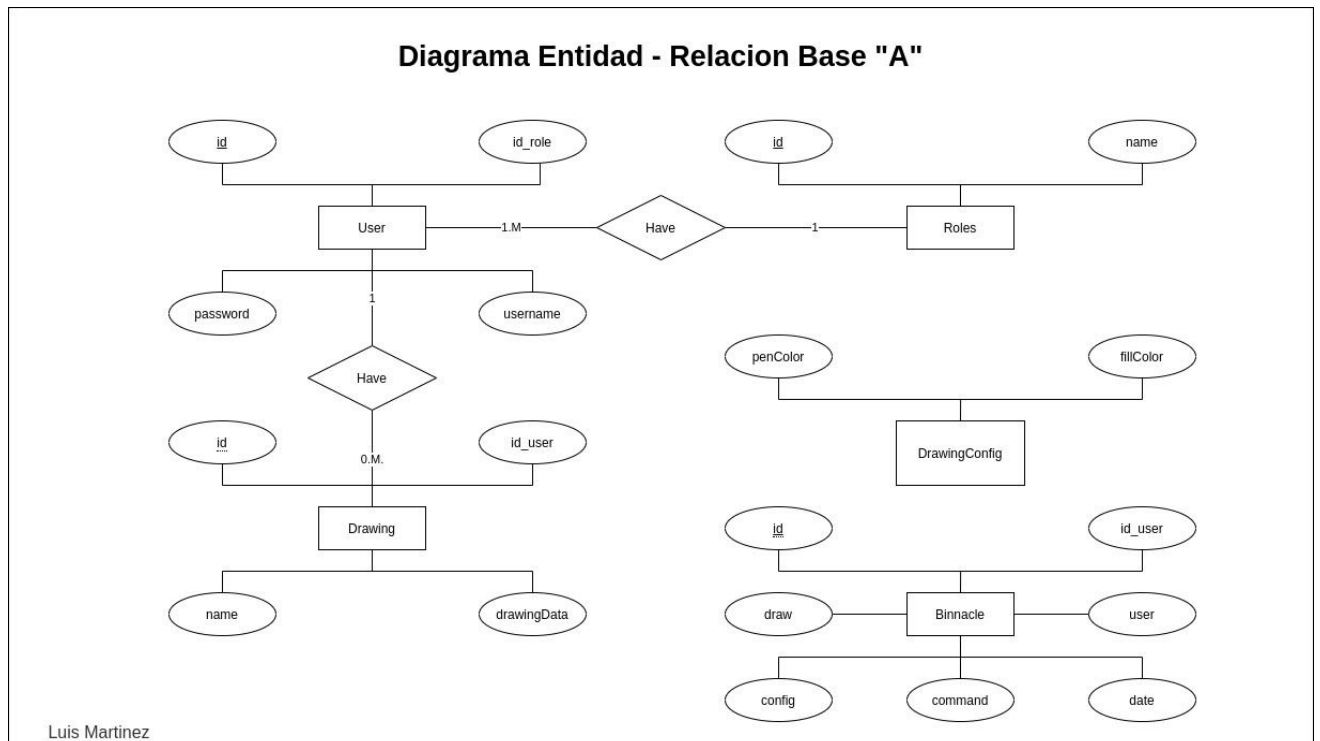
4. **str\_password:** Un campo de tipo TEXT donde se almacena la contraseña que el usuario debe utilizar para poder autenticarse y, de ésta manera, poder ingresar al programa.
- **Drawing:** Es la tabla donde se guarda la información referente a un dibujo guardado, esta tabla posee los siguientes campos:
    1. **id:** Un campo autoincremental que sirve como llave primaria en esta tabla, para poder identificar cada dibujo, a pesar que existan dos dibujos con el mismo nombre.
    2. **id\_user:** Un campo de tipo INT, utilizado como llave foránea, el cual hace referencia al usuario propietario del dibujo guardado.
    3. **str\_name:** Un campo de tipo TEXT donde se almacena el nombre con el cual se guarda el dibujo en el programa.
    4. **jso\_drawingData:** Un campo de tipo JSON donde se guarda la información necesaria para poder cargar el dibujo en el programa, una vez que este se guarda.

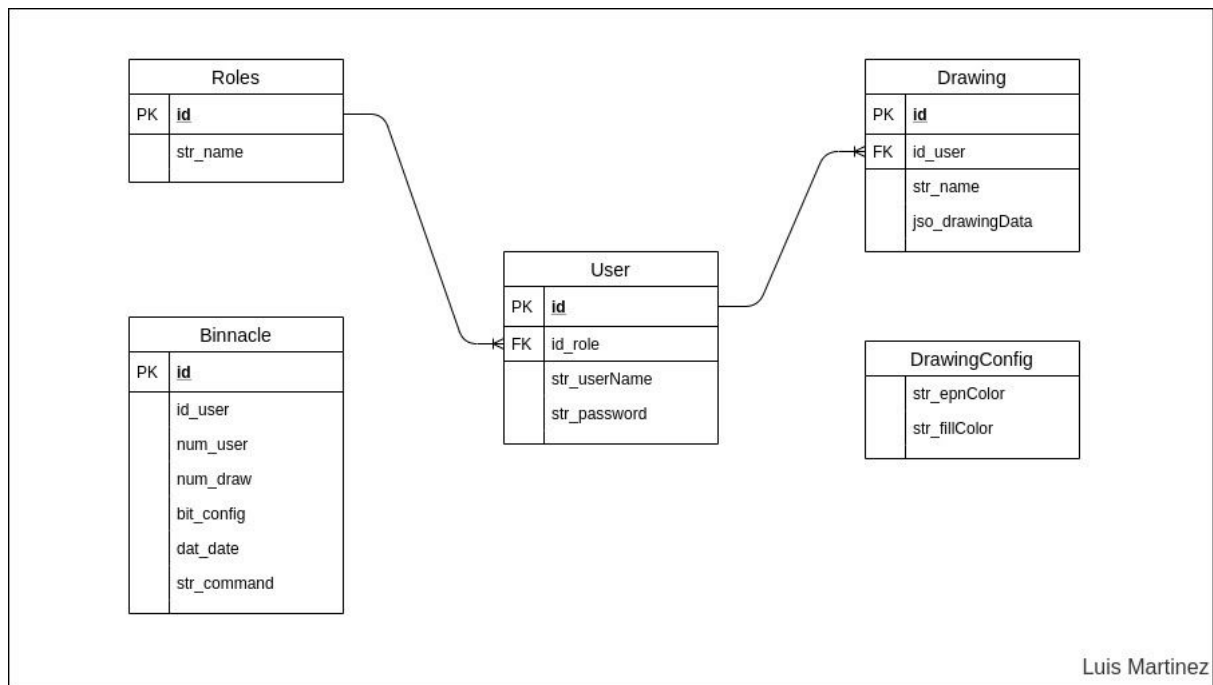
Luego de la creación de las tablas anteriores, se procede a crear una cuarta tabla, que posee la configuración de colores del programa de forma global, es decir, la configuración de lápiz y de relleno, que es utilizada por todos los usuarios, la cual solo puede ser modificada por el usuario administrador del programa. Se decide llamar a la tabla **Drawing Config**, la cual posee los siguientes campos:

1. **str\_penColor:** Un campo de tipo CHAR de longitud 7, el cual contiene el color de lápiz que utiliza el programa en formato hexadecimal, el valor de éste campo por defecto es el color Negro (#000000).
2. **str\_fillColor:** Un campo de tipo CHAR de longitud 7, el cual contiene el color de relleno que utiliza el programa en formato hexadecimal, el valor por defecto de dicho campo es el color Negro (#000000).
3. Se decide no agregar un campo “id”, ni una llave primaria, pues en esta tabla sólo debe existir un único registro.

Al tener creadas éstas 4 tablas se procede a comenzar a crear la bitácora de la Base de Datos principal llamada **Blnnacle**, la cual posee los siguientes campos:

1. **id:** Un campo autoincremental que sirve como llave primaria de esta tabla.
2. **id\_user:** Un campo tipo INT donde se almacena el id del usuario que realiza determinada acción (autenticación, visualización, inserción, modificación, eliminación).
3. **num\_user:** Un campo de tipo INT donde se almacena el id del usuario que es modificado, eliminado o agregado, si no se realiza ninguna de estas acciones, el valor del campo es NULL.
4. **num\_draw:** Un campo de tipo INT donde se almacena el id del dibujo que es agregado o modificado, si no se realiza ninguna de estas acciones el valor del campo es NULL.
5. **bit\_config:** Un campo de tipo BIT donde se almacena un “1” si se realiza un cambio en la configuración de colores del programa, si no se realiza ninguna modificación el valor de este campo es cero.
6. **dat\_date:** Un campo de tipo DATETIME donde se almacena la hora y fecha en la cual se realiza determinada acción (autenticación, visualización, inserción, modificación, eliminación).
7. **str\_command:** Un campo de tipo TEXT donde se almacena la acción que realiza un usuario (autenticación, visualización, inserción, modificación, eliminación).





### **Base de Datos de Respaldo**

Al tener finalizada la Base de Datos principal usada por el programa, se comienza a trabajar en la Base de Datos de Respaldo. Se decide diseñar la Base de Datos de Respaldo con las siguientes tablas:

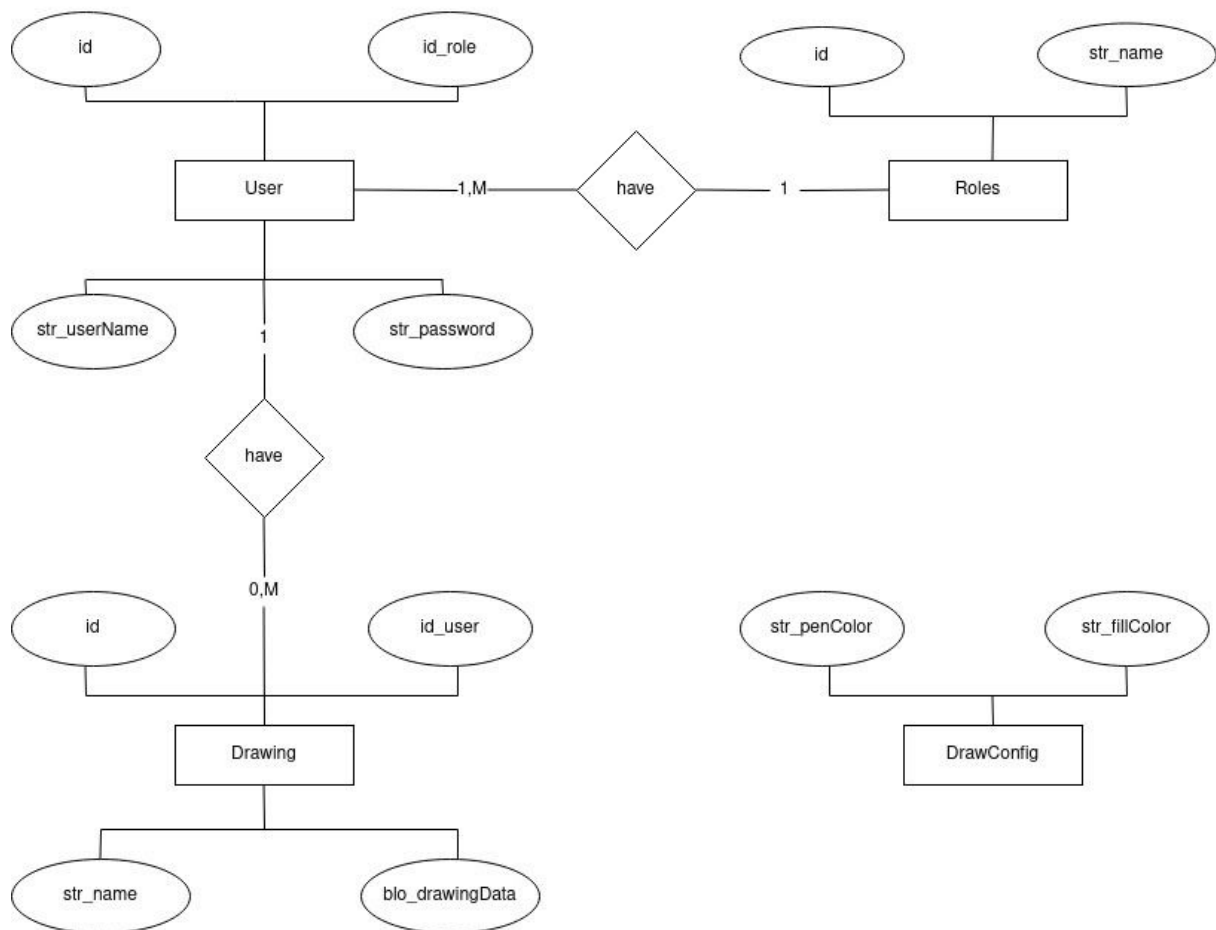
- Roles
- User
- Drawing
- DrawingConfig

Utilizando los mismos campos que poseen dichas tablas en la Base de Datos Principal, excepto por el campo **jso\_drawingData** de la tabla **Drawing**, donde este campo se reemplaza por:

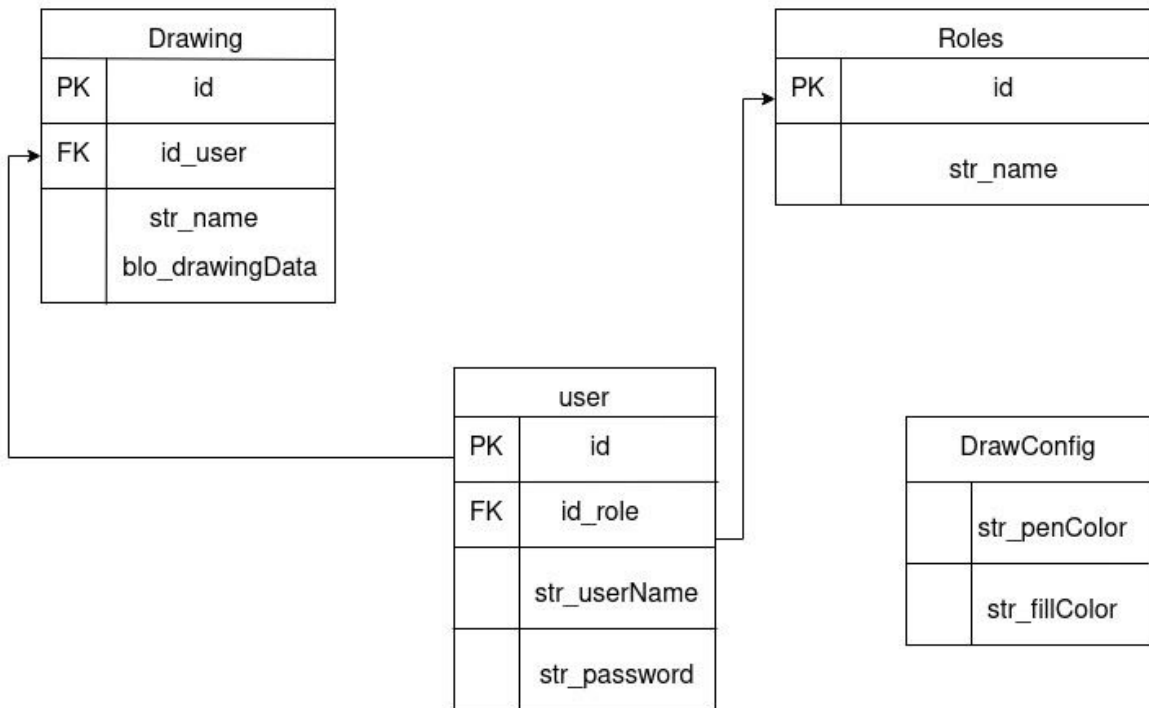
1. **blo\_drawingData**: Campo de tipo BLOB donde se almacena la información necesaria para que el dibujo guardado pueda ser cargado de nuevo, siendo los datos comprimidos, lo que permite que se pueda guardar la información en un campo de tipo BLOB.

Además, ya que esta es una Base de Respaldo, se decide que solo se puede hacer inserciones y modificaciones en la misma, de esta manera en caso que se elimine un registro por error, se puede revisar en esta Base de Datos la información que contiene el registro eliminado. También se decide no crear una bitácora en esta Base de Datos, ya que al ser una Base de Respaldo, su función solamente es la de servir como backup.

### Diagrama Entidad - Relación Base "B"



Eduardo Lopez



Eduardo Lopez