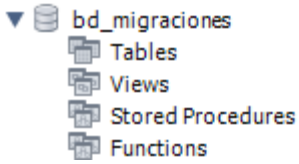
Migraciones

Pasos para crear migraciones, **un ejemplo en el cual se crean dos tablas (usuario y publicación)**

1. Primero se debe crear un base de datos vacía en mysql, ya que más adelante se necesita el nombre de la base de datos, para este ejemplo se crea la base de datos llamada: bd_migraciones



2. Se ejecutan los siguientes comandos en la terminal siguiendo el mismo orden
   Primero se ejecuta:

   npm init –y

   npm install prisma --save-dev

   npm install prisma@5.20 @prisma/client@5.20

   npx prisma init

3. Se modifica del archivo ".env" el "DATABASE_URL" que trae por defecto, por lo tanto se debe tener la siguiente información:

   - Administrador (por defecto es *root*, si no ha sido personalizado)
   - Contraseña (si tiene o no)
   - Puerto (puede ser 3306 o 3307 u otra si ha sido personalizado)
   - Nombre de la base de datos

   En mi caso utilizo "MySQL Workbench", y para crear alguna base de datos en la aplicación solicita un usuario que es "root" y una contraseña la cual es "soft123", su puerto de conexión configurado es "3306" y el nombre de la base de datos es "bd_migraciones". Recuerde que esta información puede cambiar en su computadora

   Según los datos así queda esa línea de código del DATABASE_URL en el archivo ".env"

   DATABASE_URL="mysql://root:soft123@localhost:3306/bd_migraciones"

   En caso de que no pida contraseña no se escribe en el espacio que le corresponde, ejemplo:

   DATABASE_URL="mysql://root:@localhost:3306/bd_migraciones"

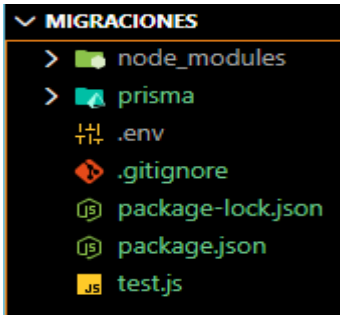
4. Se modifica el archivo "schema.prisma", aquí es donde se define la base de datos para Prisma: aquí escribes los modelos (tablas), sus campos, relaciones y el proveedor que usarás en este caso (MySQL)

```
generator client {
  provider = "prisma-client-js"
}
datasource db {
  provider = "mysql"
  url      = env("DATABASE_URL")
}
model Usuario {
  id     Int    @id @default(autoincrement())
  nombre String
  email  String @unique
  publicaciones  Publicacion[]
}
model Publicacion {
  id          Int @id @default(autoincrement())
  titulo      String
  contenido   String
  usuario     Usuario    @relation(fields: [usuarioId], references: [id])
  usuarioId   Int
}
```

5. Se ejecuta el comando

npx prisma generate

6. Se realiza una prueba de conexión a la base de datos, creando un archivo llamado **test.js** en el proyecto



El cual contiene el siguiente código

```
1   const { PrismaClient } = require('@prisma/client');
2   const prisma = new PrismaClient();
3
4   async function main() {
5       try {
6           await prisma.$connect();
7           console.log("Prisma se conectó correctamente");
8       } catch (error) {
9           console.error("Error al conectar Prisma:", error);
10      } finally{
11          await prisma.$disconnect();
12      }
13  }
14  main();
```

7. Se ejecuta el comando

node test.js

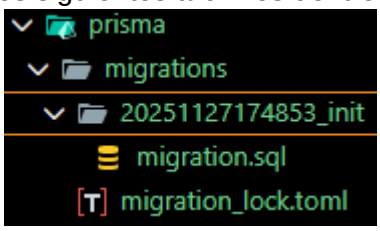
Y debe mostrar lo siguiente en la terminal

```
Prisma se conectó correctamente
```

8. Una vez comprobado que existe conexión se procede a realizar las migraciones creando las tablas en la base de datos, por lo tanto se ejecuta el siguiente comando

npx prisma migrate dev --name init

Esto genera la carpeta migrations y los siguientes archivos dentro de ella



Y automáticamente se crean en la base de datos las tablas, finalizando así la migración