gecotel/

├── models/

│ ├── User.js

│ ├── PostBox.js

│ ├── VirtualAddress.js

│ └── Package.js

├── routes/

│ ├── userRoutes.js

│ ├── postBoxRoutes.js

│ ├── virtualAddressRoutes.js

│ └── packageRoutes.js

├── app.js

└── config/

└── db.js

**Paso 3: Hacer Solicitudes POST en Postman**

**Crear un PostBox**

1. Abre Postman.
2. Selecciona el método POST.
3. Ingresa la URL: http://localhost:5000/api/postboxes.
4. En la pestaña Body, selecciona raw y el tipo JSON.
5. Ingresa el JSON con los datos del PostBox, por ejemplo:

json

Copy code

{

"ubicacion": "Ubicacion 1",

"usuarioId": "60c72b1f4f1a2c6d88f8e5a9"

}

1. Haz clic en Send.

**Crear una VirtualAddress (Crear una dirección virtual)**

1. Abre Postman.
2. Selecciona el método POST.
3. Ingresa la URL: http://localhost:5000/api/virtualaddresses.
4. En la pestaña Body, selecciona raw y el tipo JSON.
5. Ingresa el JSON con los datos de la VirtualAddress, por ejemplo:

json

Copy code

{

"direccion": "Calle Falsa 123",

"apartadoPostalId": "60c72b2f4f1a2c6d88f8e5b0"

}

1. Haz clic en Send.

**Crear un Package (Crear un paquete)x`**

1. Abre Postman.
2. Selecciona el método POST.
3. Ingresa la URL: http://localhost:5000/api/packages.
4. En la pestaña Body, selecciona raw y el tipo JSON.
5. Ingresa el JSON con los datos del Package, por ejemplo:

json

Copy code

{

"estado": "En tránsito",

"destino": "Calle Falsa 123",

"apartadoPostalId": "60c72b2f4f1a2c6d88f8e5b0"

}

1. Haz clic en Send.

**Resumen**

Siguiendo estos pasos, podrás agregar datos a tu base de datos MongoDB utilizando las rutas definidas en tu aplicación Node.js con Express. Utiliza Postman para enviar solicitudes POST a las rutas correspondientes con los datos necesarios en el cuerpo de la solicitud. Asegúrate de que tu servidor esté en funcionamiento y que las rutas estén correctamente configuradas.

**Usando la Línea de Comandos (mongo shell)**

1. **Accede a tu terminal o línea de comandos**: Abre una terminal (en Linux o macOS) o el símbolo del sistema (en Windows).
2. **Conéctate a tu servidor MongoDB**: Usa el comando mongo para conectarte a tu base de datos. Escribe:

bash

Copy code

mongo --host localhost --port 27017

1. **Selecciona tu base de datos**: Una vez conectado, selecciona la base de datos gecotel escribiendo:

javascript

Copy code

use gecotel

1. **Verifica las colecciones en tu base de datos**: Para ver todas las colecciones en la base de datos gecotel, usa:

javascript

Copy code

show collections

1. **Consulta los datos de una colección específica**: Por ejemplo, si tienes una colección llamada usuarios, puedes ver todos sus documentos usando:

javascript

Copy code

db.usuarios.find().pretty()

**Usando MongoDB Compass**

1. **Descarga e instala MongoDB Compass**: Puedes descargar MongoDB Compass desde [el sitio oficial](https://www.mongodb.com/try/download/compass).
2. **Abre MongoDB Compass**: Una vez instalado, abre la aplicación.
3. **Conéctate a tu servidor MongoDB**:
   * En la pantalla de inicio de MongoDB Compass, ingresa mongodb://localhost:27017 en el campo de conexión y haz clic en el botón "Connect".
4. **Selecciona tu base de datos**:
   * Después de conectarte, verás una lista de todas las bases de datos en tu servidor MongoDB.
   * Selecciona gecotel de la lista.
5. **Explora las colecciones y documentos**:
   * Una vez seleccionada la base de datos, verás todas las colecciones dentro de gecotel.
   * Haz clic en una colección para ver y explorar los documentos contenidos en ella.

**Ejemplo con MongoDB Compass**

Siguiendo estos pasos, podrás ver y gestionar los datos de tu base de datos MongoDB local gecotel de manera efectiva. Si necesitas ejecutar consultas más complejas o administrar tus datos de forma avanzada, tanto el mongo shell como MongoDB Compass te proporcionan las herramientas necesarias para hacerlo.

**Esquema Mejorado de la Base de Datos para Apartado Postal Virtual y Direcciones Virtuales para Gecotel Postal**

**1. Tabla de Usuarios**

Almacena información general sobre los usuarios (individuos y entidades).

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| usuario\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada usuario |
| nombre | VARCHAR(60) | Nombre del usuario |
| apellidos | VARCHAR(60) | Apellidos del usuario |
| fecha\_nacimiento | DATE | Fecha de nacimiento del usuario |
| correo\_electronico | VARCHAR(100) | Correo electrónico del usuario |
| telefono | VARCHAR(50) | Teléfono del usuario |
| direccion\_id | INT | Clave foránea a la tabla Direcciones |
| apartado\_postal\_id | INT | Clave foránea a la tabla ApartadosPostales |
| contraseña\_hash | VARCHAR(255) | Contraseña cifrada para seguridad |
| rol\_id | INT | Clave foránea a la tabla Roles |

**2. Tabla de Direcciones**

Almacena información de las direcciones.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| direccion\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada dirección |
| calle | VARCHAR(100) | Nombre de la calle |
| barrio | VARCHAR(100) | Barrio |
| numero | VARCHAR(10) | Número de la casa/edificio |
| distrito | VARCHAR(100) | Distrito |
| provincia | VARCHAR(100) | Provincia |
| entidad\_id | INT | Clave foránea a la tabla Entidades |

**3. Tabla de Apartados Postales**

Almacena información sobre los apartados postales asignados a los usuarios.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| apartado\_postal\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada apartado postal |
| cantidad\_documentos | INT | Cantidad de documentos dirigidos al usuario |

**4. Tabla de Roles**

Define los roles de los usuarios dentro del sistema.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| rol\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada rol |
| nombre\_rol | VARCHAR(100) | Nombre del rol (por ejemplo, Admin, Empleado, Usuario) |

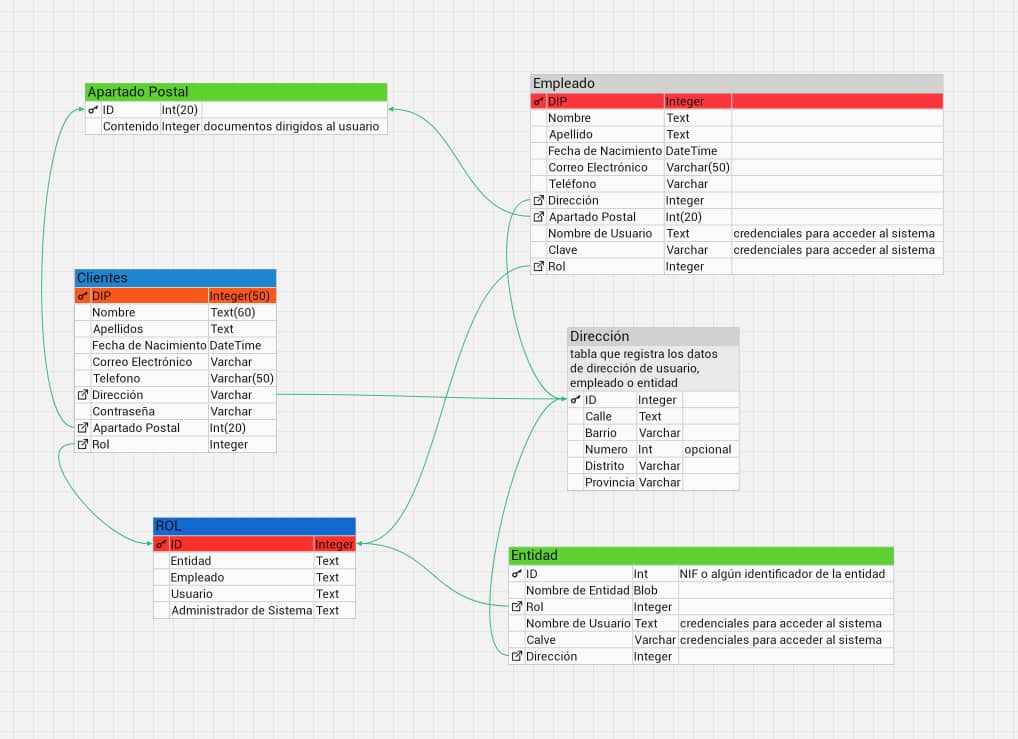
**5. Tabla de Entidades**

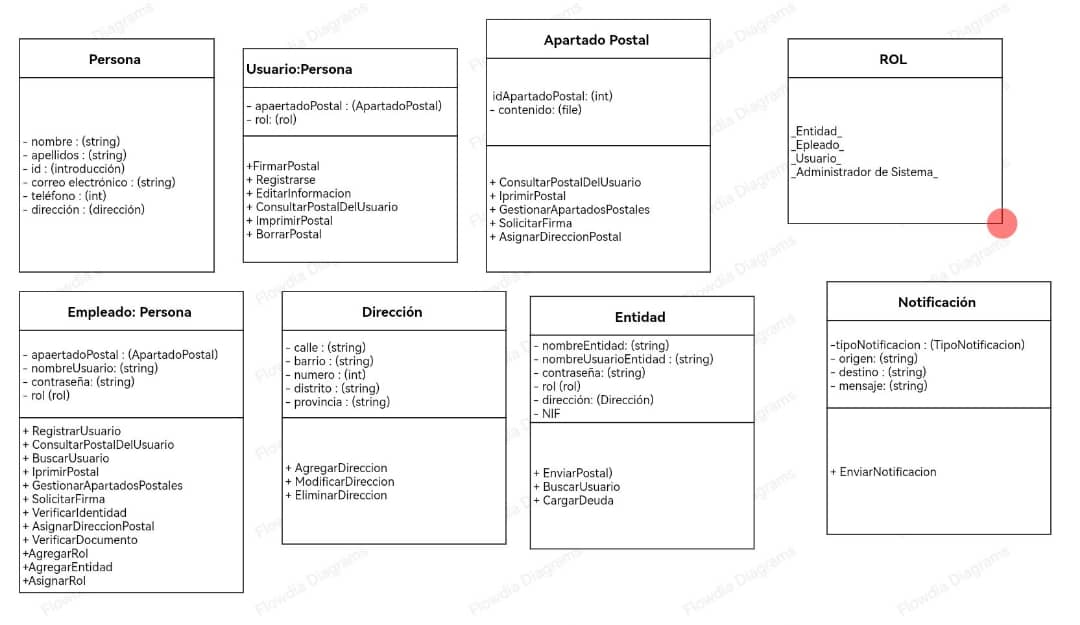
Almacena información sobre varias entidades (bancos, ministerios, hospitales, etc.).

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| entidad\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada entidad |
| nombre\_entidad | VARCHAR(100) | Nombre de la entidad |
| rol\_id | INT | Clave foránea a la tabla Roles |
| direccion\_id | INT | Clave foránea a la tabla Direcciones |
| nif | VARCHAR(50) | NIF u otro identificador |

**Relaciones**

1. **Usuarios y Direcciones**: Relación uno a uno. Cada usuario tiene una dirección.
2. **Usuarios y ApartadosPostales**: Relación uno a uno. Cada usuario tiene asignado un apartado postal.
3. **Usuarios y Roles**: Relación muchos a uno. Cada usuario tiene un rol.
4. **Entidades y Direcciones**: Relación uno a uno. Cada entidad tiene una dirección.
5. **Entidades y Roles**: Relación muchos a uno. Cada entidad tiene un rol.





**Esquema Mejorado de la** **Base de Datos para Apartado Postal Virtual y Direcciones Virtuales para Gecotel Postal**

**1. Tabla de Clientes**

Almacena información general sobre los clientes (individuos y entidades).

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| cliente\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada cliente |
| DIP | INT(50) | Documento de Identificación Personal |
| nombre | VARCHAR(60) | Nombre del cliente |
| apellidos | VARCHAR(60) | Apellidos del cliente |
| fecha\_nacimiento | DATETIME | Fecha de nacimiento del cliente |
| correo\_electronico | VARCHAR(100) | Correo electrónico del cliente |
| telefono | VARCHAR(50) | Teléfono del cliente |
| direccion | VARCHAR(255) | Dirección del cliente |
| contraseña | VARCHAR(255) | Contraseña cifrada para seguridad |
| apartado\_postal\_id | INT(20) | Clave foránea a la tabla ApartadosPostales |
| rol\_id | INT | Clave foránea a la tabla Roles |

**2. Tabla de Direcciones**

Almacena información de las direcciones.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| direccion\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada dirección |
| calle | VARCHAR(100) | Nombre de la calle |
| barrio | VARCHAR(100) | Barrio |
| numero | VARCHAR(10) | Número de la casa/edificio (opcional) |
| distrito | VARCHAR(100) | Distrito |
| provincia | VARCHAR(100) | Provincia |
| entidad\_id | INT | Clave foránea a la tabla Entidades |

**3. Tabla de Apartados Postales**

Almacena información sobre los apartados postales asignados a los clientes.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| apartado\_postal\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada apartado postal |
| contenido | INT | Cantidad de documentos dirigidos al cliente |

**4. Tabla de Roles**

Define los roles de los clientes dentro del sistema.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| rol\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada rol |
| nombre\_rol | VARCHAR(100) | Nombre del rol (por ejemplo, Admin, Empleado, Usuario) |

**5. Tabla de Entidades**

Almacena información sobre varias entidades (bancos, ministerios, hospitales, etc.).

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| entidad\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada entidad |
| nif | VARCHAR(50) | NIF u otro identificador |
| nombre\_entidad | VARCHAR(100) | Nombre de la entidad |
| direccion\_id | INT | Clave foránea a la tabla Direcciones |

**Relaciones**

1. **Clientes y Direcciones**: Relación uno a uno. Cada cliente tiene una dirección.
2. **Clientes y ApartadosPostales**: Relación uno a uno. Cada cliente tiene asignado un apartado postal.
3. **Clientes y Roles**: Relación muchos a uno. Cada cliente tiene un rol.
4. **Entidades y Direcciones**: Relación uno a uno. Cada entidad tiene una dirección.

**Esquema SQL**

sql

Copy code

CREATE TABLE Roles (

rol\_id INT AUTO\_INCREMENT PRIMARY KEY,

nombre\_rol VARCHAR(100) NOT NULL

);

CREATE TABLE Direcciones (

direccion\_id INT AUTO\_INCREMENT PRIMARY KEY,

calle VARCHAR(100),

barrio VARCHAR(100),

numero VARCHAR(10),

distrito VARCHAR(100),

provincia VARCHAR(100),

entidad\_id INT,

FOREIGN KEY (entidad\_id) REFERENCES Entidades(entidad\_id)

);

CREATE TABLE Entidades (

entidad\_id INT AUTO\_INCREMENT PRIMARY KEY,

nif VARCHAR(50),

nombre\_entidad VARCHAR(100) NOT NULL,

direccion\_id INT,

FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id)

);

CREATE TABLE ApartadosPostales (

apartado\_postal\_id INT AUTO\_INCREMENT PRIMARY KEY,

contenido INT DEFAULT 0

);

CREATE TABLE Clientes (

cliente\_id INT AUTO\_INCREMENT PRIMARY KEY,

DIP INT(50) NOT NULL,

nombre VARCHAR(60) NOT NULL,

apellidos VARCHAR(60) NOT NULL,

fecha\_nacimiento DATETIME,

correo\_electronico VARCHAR(100) UNIQUE NOT NULL,

telefono VARCHAR(50),

direccion VARCHAR(255),

contrasena VARCHAR(255) NOT NULL,

apartado\_postal\_id INT,

rol\_id INT,

FOREIGN KEY (apartado\_postal\_id) REFERENCES ApartadosPostales(apartado\_postal\_id),

FOREIGN KEY (rol\_id) REFERENCES Roles(rol\_id)

);

**Notas**

1. **Almacenamiento de Contraseñas**: Asegúrese de que las contraseñas se almacenen de manera segura utilizando algoritmos de hash.
2. **Validación de Datos**: Implemente la validación de datos para mantener la integridad de los datos.
3. **Índices**: Considere agregar índices a las columnas que se usan frecuentemente en las consultas para mejorar el rendimiento.
4. **Roles**: Defina roles y permisos adicionales según sea necesario para mejorar la seguridad y funcionalidad.

4o

//Tabla modificada

**Esquema Mejorado de la Base de Datos para Apartado Postal Virtual y Direcciones Virtuales para Gecotel Postal**

**1. Tabla de Clientes**

Almacena información general sobre los clientes (individuos y entidades).

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| cliente\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada cliente |
| DIP | INT(50) | Documento de Identificación Personal |
| nombre | VARCHAR(60) | Nombre del cliente |
| apellidos | VARCHAR(60) | Apellidos del cliente |
| fecha\_nacimiento | DATETIME | Fecha de nacimiento del cliente |
| correo\_electronico | VARCHAR(100) | Correo electrónico del cliente |
| telefono | VARCHAR(50) | Teléfono del cliente |
| direccion | VARCHAR(255) | Dirección del cliente |
| contrasena | VARCHAR(255) | Contraseña cifrada para seguridad |
| apartado\_postal\_id | INT(20) | Clave foránea a la tabla ApartadosPostales |
| rol\_id | INT | Clave foránea a la tabla Roles |

**2. Tabla de Empleados**

Almacena información general sobre los empleados.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| empleado\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada empleado |
| DIP | INT(50) | Documento de Identificación Personal |
| nombre | VARCHAR(60) | Nombre del empleado |
| apellidos | VARCHAR(60) | Apellidos del empleado |
| fecha\_nacimiento | DATETIME | Fecha de nacimiento del empleado |
| correo\_electronico | VARCHAR(50) | Correo electrónico del empleado |
| telefono | VARCHAR(50) | Teléfono del empleado |
| direccion\_id | INT | Clave foránea a la tabla Direcciones |
| apartado\_postal\_id | INT(20) | Clave foránea a la tabla ApartadosPostales |
| nombre\_usuario | VARCHAR(60) | Nombre de usuario del empleado |
| clave | VARCHAR(255) | Contraseña cifrada para seguridad |
| rol\_id | INT | Clave foránea a la tabla Roles |

**3. Tabla de Direcciones**

Almacena información de las direcciones.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| direccion\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada dirección |
| calle | VARCHAR(100) | Nombre de la calle |
| barrio | VARCHAR(100) | Barrio |
| numero | VARCHAR(10) | Número de la casa/edificio (opcional) |
| distrito | VARCHAR(100) | Distrito |
| provincia | VARCHAR(100) | Provincia |
| entidad\_id | INT | Clave foránea a la tabla Entidades |

**4. Tabla de Apartados Postales**

Almacena información sobre los apartados postales asignados a los clientes y empleados.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| apartado\_postal\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada apartado postal |
| contenido | INT | Cantidad de documentos dirigidos al usuario |

**5. Tabla de Roles**

Define los roles de los clientes y empleados dentro del sistema.

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| rol\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada rol |
| nombre\_rol | VARCHAR(100) | Nombre del rol (por ejemplo, Admin, Empleado, Usuario) |

**6. Tabla de Entidades**

Almacena información sobre varias entidades (bancos, ministerios, hospitales, etc.).

| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| --- | --- | --- |
| entidad\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada entidad |
| nif | VARCHAR(50) | NIF u otro identificador |
| nombre\_entidad | VARCHAR(100) | Nombre de la entidad |
| direccion\_id | INT | Clave foránea a la tabla Direcciones |

**Relaciones**

1. **Clientes y Direcciones**: Relación uno a uno. Cada cliente tiene una dirección.
2. **Clientes y ApartadosPostales**: Relación uno a uno. Cada cliente tiene asignado un apartado postal.
3. **Clientes y Roles**: Relación muchos a uno. Cada cliente tiene un rol.
4. **Empleados y Direcciones**: Relación uno a uno. Cada empleado tiene una dirección.
5. **Empleados y ApartadosPostales**: Relación uno a uno. Cada empleado tiene asignado un apartado postal.
6. **Empleados y Roles**: Relación muchos a uno. Cada empleado tiene un rol.
7. **Entidades y Direcciones**: Relación uno a uno. Cada entidad tiene una dirección.

**Esquema SQL**

sql

Copy code

CREATE TABLE Roles (

rol\_id INT AUTO\_INCREMENT PRIMARY KEY,

nombre\_rol VARCHAR(100) NOT NULL

);

CREATE TABLE Direcciones (

direccion\_id INT AUTO\_INCREMENT PRIMARY KEY,

calle VARCHAR(100),

barrio VARCHAR(100),

numero VARCHAR(10),

distrito VARCHAR(100),

provincia VARCHAR(100),

entidad\_id INT,

FOREIGN KEY (entidad\_id) REFERENCES Entidades(entidad\_id)

);

CREATE TABLE Entidades (

entidad\_id INT AUTO\_INCREMENT PRIMARY KEY,

nif VARCHAR(50),

nombre\_entidad VARCHAR(100) NOT NULL,

direccion\_id INT,

FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id)

);

CREATE TABLE ApartadosPostales (

apartado\_postal\_id INT AUTO\_INCREMENT PRIMARY KEY,

contenido INT DEFAULT 0

);

CREATE TABLE Clientes (

cliente\_id INT AUTO\_INCREMENT PRIMARY KEY,

DIP INT(50) NOT NULL,

nombre VARCHAR(60) NOT NULL,

apellidos VARCHAR(60) NOT NULL,

fecha\_nacimiento DATETIME,

correo\_electronico VARCHAR(100) UNIQUE NOT NULL,

telefono VARCHAR(50),

direccion VARCHAR(255),

contrasena VARCHAR(255) NOT NULL,

apartado\_postal\_id INT,

rol\_id INT,

FOREIGN KEY (apartado\_postal\_id) REFERENCES ApartadosPostales(apartado\_postal\_id),

FOREIGN KEY (rol\_id) REFERENCES Roles(rol\_id)

);

CREATE TABLE Empleados (

empleado\_id INT AUTO\_INCREMENT PRIMARY KEY,

DIP INT(50) NOT NULL,

nombre VARCHAR(60) NOT NULL,

apellidos VARCHAR(60) NOT NULL,

fecha\_nacimiento DATETIME,

correo\_electronico VARCHAR(50) UNIQUE NOT NULL,

telefono VARCHAR(50),

direccion\_id INT,

apartado\_postal\_id INT,

nombre\_usuario VARCHAR(60),

clave VARCHAR(255) NOT NULL,

rol\_id INT,

FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id),

FOREIGN KEY (apartado\_postal\_id) REFERENCES ApartadosPostales(apartado\_postal\_id),

FOREIGN KEY (rol\_id) REFERENCES Roles(rol\_id)

);

**Notas**

1. **Almacenamiento de Contraseñas**: Asegúrese de que las contraseñas se almacenen de manera segura utilizando algoritmos de hash.
2. **Validación de Datos**: Implemente la validación de datos para mantener la integridad de los datos.
3. **Índices**: Considere agregar índices a las columnas que se usan frecuentemente en las consultas para mejorar el rendimiento.
4. **Roles**: Defina roles y permisos adicionales según sea necesario para mejorar la seguridad y funcionalidad.

4o

**#Codigo para crear las tablas en Postgred en formato Query**

CREATE TABLE Roles (

rol\_id SERIAL PRIMARY KEY,

nombre\_rol VARCHAR(100) NOT NULL

);

CREATE TABLE Direcciones (

direccion\_id SERIAL PRIMARY KEY,

calle VARCHAR(100),

barrio VARCHAR(100),

numero VARCHAR(10),

distrito VARCHAR(100),

provincia VARCHAR(100)

);

CREATE TABLE Entidades (

entidad\_id SERIAL PRIMARY KEY,

nif VARCHAR(50) NOT NULL,

nombre\_entidad VARCHAR(100) NOT NULL,

direccion\_id INT,

FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id)

);

CREATE TABLE ApartadosPostales (

apartado\_postal\_id SERIAL PRIMARY KEY,

contenido INT DEFAULT 0

);

CREATE TABLE Clientes (

cliente\_id SERIAL PRIMARY KEY,

DIP INT NOT NULL,

nombre VARCHAR(60) NOT NULL,

apellidos VARCHAR(60) NOT NULL,

fecha\_nacimiento TIMESTAMP,

correo\_electronico VARCHAR(100) UNIQUE NOT NULL,

telefono VARCHAR(50),

direccion VARCHAR(255),

contrasena VARCHAR(255) NOT NULL,

apartado\_postal\_id INT,

rol\_id INT,

FOREIGN KEY (apartado\_postal\_id) REFERENCES ApartadosPostales(apartado\_postal\_id),

FOREIGN KEY (rol\_id) REFERENCES Roles(rol\_id)

);

CREATE TABLE Empleados (

empleado\_id SERIAL PRIMARY KEY,

DIP INT NOT NULL,

nombre VARCHAR(60) NOT NULL,

apellidos VARCHAR(60) NOT NULL,

fecha\_nacimiento TIMESTAMP,

correo\_electronico VARCHAR(100) UNIQUE NOT NULL,

telefono VARCHAR(50),

direccion\_id INT,

apartado\_postal\_id INT,

nombre\_usuario VARCHAR(60),

clave VARCHAR(255) NOT NULL,

rol\_id INT,

FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id),

FOREIGN KEY (apartado\_postal\_id) REFERENCES ApartadosPostales(apartado\_postal\_id),

FOREIGN KEY (rol\_id) REFERENCES Roles(rol\_id)

);

Codigo para crear tipos, sexo etc, de forma enumerada

CREATE TYPE tipo\_sexo as enum

('M', 'F');

#Codigo para relacionar types con nuestras tablas

ALTER TABLE nombre de la tabla

ALTER COLUMN nombre de la columna que queremos alterar TYPE tipo\_sexo USING sex::tipo\_sexo;

Para alterar una tabla utilizamos el siguiente código

ALTER TABLE nombre de la tabla ADD Nombre de la nueva columna VARCHAT(8) \*Tipo de datos

ALTER TABLE nombre de la tabla ALTER COLUMN nombre de la Columna SET NOT NULL;

**Tabla de Notificaciones**

Almacena información sobre las notificaciones enviadas a los usuarios.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de Columna** |  | **Tipo de Datos** | **Descripción** |
| notificacion\_id |  | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada notificación |
| usuario\_id |  | INT | Clave foránea a la tabla Clientes |
| remitente |  | VARCHAR(100) | Nombre del remitente |
| mensaje |  | TEXT | Contenido del mensaje |
| fecha\_notificacion |  | DATETIME | Fecha y hora en que se envió la notificación |
| leido |  | BOOLEAN | Indica si la notificación ha sido leída |

**Relaciones**

* **Notificaciones y Clientes (Usuarios)**: Relación muchos a uno. Cada notificación está asociada a un cliente (usuario).

**Estructura Completa de la Base de Datos**

**Tabla de Clientes (Individuo)**

Almacena información general sobre los clientes (individuos y entidades).

|  |  |  |
| --- | --- | --- |
| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| cliente\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada cliente |
| DIP | INT(9) | Documento de Identidad Personal |
| nombre | VARCHAR(60) | Nombre del cliente |
| apellidos | VARCHAR(60) | Apellidos del cliente |
| fecha\_nacimiento | DATETIME | Fecha de nacimiento del cliente |
| correo\_electronico | VARCHAR(100) | Correo electrónico del cliente |
| telefono | VARCHAR(50) | Teléfono del cliente |
| direccion | VARCHAR(255) | Dirección del cliente |
| contraseña | VARCHAR(255) | Contraseña cifrada para seguridad |
| apartado\_postal\_id | INT(20) | Clave foránea a la tabla ApartadosPostales |
| rol\_id | INT | Clave foránea a la tabla Roles |

**Tabla de Empleados**

Almacena las credenciales de los empleados.

|  |  |  |
| --- | --- | --- |
| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| empleado\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada empleado |
| clave | VARCHAR(255) | Contraseña cifrada para seguridad |
| Otros tipos… | VARCHAR(255) | Más datos… |
| cliente\_id | INT | Clave foránea a la tabla Clientes o Individuos |

**Tabla de Direcciones**

Almacena información de las direcciones.

|  |  |  |
| --- | --- | --- |
| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| direccion\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada dirección |
| calle | VARCHAR(100) | Nombre de la calle |
| barrio | VARCHAR(100) | Barrio |
| numero | VARCHAR(10) | Número de la casa/edificio (opcional) |
| distrito | VARCHAR(100) | Distrito |
| provincia | VARCHAR(100) | Provincia |
| entidad\_id | INT | Clave foránea a la tabla Entidades |

**Tabla de Apartados Postales**

Almacena información sobre los apartados postales asignados a los clientes.

|  |  |  |
| --- | --- | --- |
| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| apartado\_postal\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada apartado postal |
| contenido | INT | Cantidad de documentos dirigidos al cliente |

**Tabla de Roles**

Define los roles de los clientes dentro del sistema.

|  |  |  |
| --- | --- | --- |
| **Nombre de Columna** | **Tipo de Datos** | **Descripción** |
| rol\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada rol |
| nombre\_rol | VARCHAR(100) | Nombre del rol (por ejemplo, Admin, Empleado, Usuario) |

**Tabla de Entidades**

**Columna** Almacena información sobre varias entidades (bancos, ministerios, hospitales, etc.).

|  |  |  |
| --- | --- | --- |
| **Columna Nombre de** | **Tipo de Datos** | **Descripción** |
| entidad\_id | INT AUTO\_INCREMENT PRIMARY KEY | Identificador único para cada entidad |
| nif | VARCHAR(50) | NIF u otro identificador |
| nombre\_entidad | VARCHAR(100) | Nombre de la entidad |
| direccion\_id | INT | Clave foránea a la tabla Direcciones |

**Relaciones**

* **Clientes y Direcciones**: Relación uno a uno. Cada cliente tiene una dirección.
* **Clientes y ApartadosPostales**: Relación uno a uno. Cada cliente tiene asignado un apartado postal.
* **Clientes y Roles**: Relación muchos a uno. Cada cliente tiene un rol.
* **Entidades y Direcciones**: Relación uno a uno. Cada entidad tiene una dirección.
* **Notificaciones y Clientes**: Relación muchos a uno. Cada notificación está asociada a un cliente.

Actualizado

-- Tabla de Clientes

CREATE TABLE Clientes (

cliente\_id SERIAL PRIMARY KEY,

DIP INT NOT NULL,

nombre VARCHAR(60) NOT NULL,

apellidos VARCHAR(60) NOT NULL,

fecha\_nacimiento TIMESTAMP NOT NULL,

correo\_electronico VARCHAR(100) NOT NULL,

telefono VARCHAR(50),

direccion VARCHAR(255),

contraseña VARCHAR(255) NOT NULL,

apartado\_postal\_id INT,

rol\_id INT,

FOREIGN KEY (apartado\_postal\_id) REFERENCES ApartadosPostales(apartado\_postal\_id),

FOREIGN KEY (rol\_id) REFERENCES Roles(rol\_id)

);

-- Tabla de Empleados

CREATE TABLE Empleados (

empleado\_id SERIAL PRIMARY KEY,

clave VARCHAR(255) NOT NULL,

otros\_datos VARCHAR(255),

cliente\_id INT,

FOREIGN KEY (cliente\_id) REFERENCES Clientes(cliente\_id)

);

-- Tabla de Direcciones

CREATE TABLE Direcciones (

direccion\_id SERIAL PRIMARY KEY,

calle VARCHAR(100) NOT NULL,

barrio VARCHAR(100) NOT NULL,

numero VARCHAR(10),

distrito VARCHAR(100) NOT NULL,

provincia VARCHAR(100) NOT NULL,

entidad\_id INT,

FOREIGN KEY (entidad\_id) REFERENCES Entidades(entidad\_id)

);

-- Tabla de Apartados Postales

CREATE TABLE ApartadosPostales (

apartado\_postal\_id SERIAL PRIMARY KEY,

contenido INT NOT NULL

);

-- Tabla de Roles

CREATE TABLE Roles (

rol\_id SERIAL PRIMARY KEY,

nombre\_rol VARCHAR(100) NOT NULL

);

-- Tabla de Entidades

CREATE TABLE Entidades (

entidad\_id SERIAL PRIMARY KEY,

nif VARCHAR(50) NOT NULL,

nombre\_entidad VARCHAR(100) NOT NULL,

direccion\_id INT,

FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id)

);

-- Tabla de Notificaciones

CREATE TABLE Notificaciones (

notificacion\_id SERIAL PRIMARY KEY,

usuario\_id INT NOT NULL,

remitente VARCHAR(100) NOT NULL,

mensaje TEXT NOT NULL,

fecha\_notificacion TIMESTAMP NOT NULL,

leido BOOLEAN DEFAULT FALSE,

FOREIGN KEY (usuario\_id) REFERENCES Clientes(cliente\_id)

);

-- Relación uno a uno entre Clientes y Direcciones

ALTER TABLE Clientes ADD COLUMN direccion\_id INT;

ALTER TABLE Clientes ADD FOREIGN KEY (direccion\_id) REFERENCES Direcciones(direccion\_id);

-- Relación uno a uno entre Clientes y ApartadosPostales

ALTER TABLE ApartadosPostales ADD COLUMN cliente\_id INT;

ALTER TABLE ApartadosPostales ADD FOREIGN KEY (cliente\_id) REFERENCES Clientes(cliente\_id);

Comandos utilizados en postgred

Para ver detalles de una tabla de forma general cliqueamos en propiedades

Código para eliminar una tabla

DROP TABLE IF EXISTS Empleados;

#Codigo para crear una tabla

CREATE TABLE Customer(

first\_name VARCHAR(30) NOT NULL,

last\_name VARCHAR(30) NOT NULL,

email VARCHAR(60) NOT NULL,

company VARCHAR(60) NOT NULL,

street VARCHAR(50) NOT NULL,

city VARCHAR(40) NOT NULL,

state CHAR(2) NOT NULL,

zip SMALLINT NOT NULL,

phone VARCHAR(20) NOT NULL,

birth\_date DATE NULL,

sex CHAR(1) NOT NULL,

date\_entered TIMESTAMP NOT NULL,

id SERIAL PRIMARY KEY

);

#Codigo para insertar datos en una tabla

INSERT INTO customer (first\_name, last\_name, email, company, street, city,

state, zip, phone,birth\_date, sex, date\_entered)

VALUES ('Mas', 'Onewe', 'masonewe@gmail.com', 'I-Tec', 'Cabo San Juan', 'EQG', '30033',

'222245678', '1997-08-23', 'M', current\_timestamp);

#Codigo para crear sexo o tipos

CREATE TYPE sex\_type as enum

('M', 'F');

#Funcion para alterar una tabla

ALTER TABLE "customer "

ALTER COLUMN sex TYPE sex\_type USING sex::sex\_type;

