

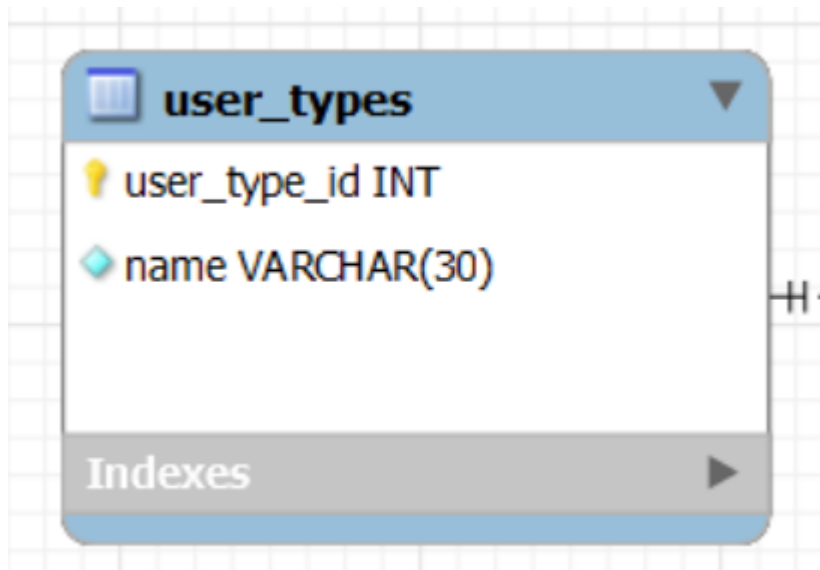
Documentación Biblioteca DB

user_types

Almacena los nombres de los roles que pueden tener las personas que interactúen con la aplicación.

Atributos:

- user_type_id: un entero auto incremental que representa el índice del registro dentro de la tabla.
- name: una cadena de caracteres que denota el rol que tiene este usuario en la aplicación. Sus posibles valores pueden ser "**Estudiante**" o "**Administrador**".



Relaciones:

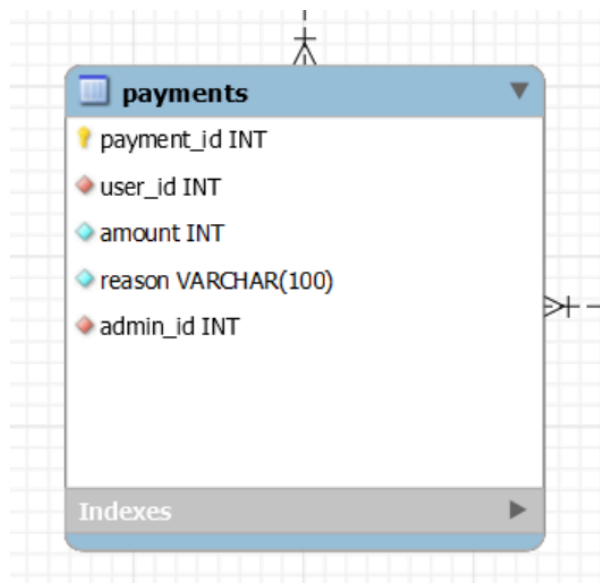
- 1:1 con la tabla **users**.

payments

En el caso que un estudiante pida un libro y éste lo devuelva con daños muy graves o lo pierda, los administradores pueden cobrar un monto adecuado a la situación. La tabla **payments** almacena toda la información relacionada a cada caso donde halla sido necesario aplicar un cobro.

Atributos:

- **payment_id**: un entero auto incremental que representa el índice del registro dentro de la tabla.
- **user_id**: llave foránea que hace referencia a la llave primaria de la tabla Usuarios (para poder identificar a la persona responsable).
- **amount**: Monto total del cobro (valor de los daños o del libro).
- **reason**: Especifica el motivo del cobro al estudiante. Esta información debe ser proveída por l administrador a cargo. Por tanto, este campo no tiene valores posibles como el campo **user_types**.
- **admin_id**: llave foránea a la tabla **users**, representa el id del administrador que aprobó ese cobro.



Relaciones:

- m : 1 con la tabla **users** (campo ***user_id**).
- m : 1 con la tabla **users** (campo **admin_id**).

authors

En esta tabla se almacena la información necesaria de cada autor de los libros almacenado en el sistema.

Atributos:

- `author_id`: un entero auto incremental que representa el índice del registro dentro de la tabla.
- `first_name`: contiene ambos nombres del autor (o solo uno según sea el caso).
- `last_name`: contiene ambos apellidos del autor (o solo uno según sea el caso).



Relaciones:

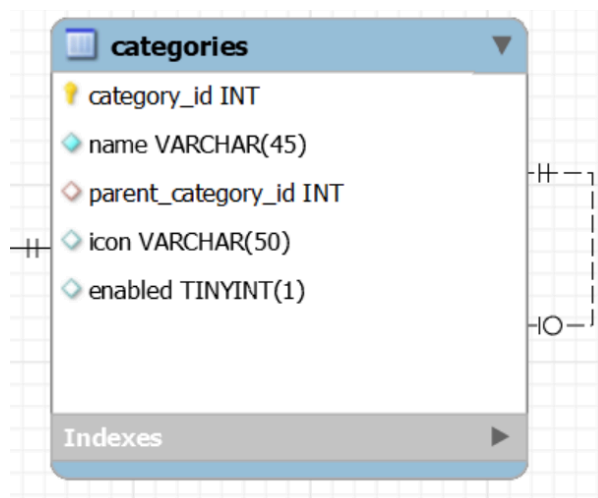
- 1:m con [authors_per_book](#).

categories

En la aplicación, cada libro pertenece a una o varias categorías. Donde las categorías va desde las mas generales (como ser [matemática](#) o [física](#)) hasta las mas especificas como ([Ecuaciones Diferenciales](#) o [Análisis de atractores](#)). Donde las categorías mas generales son "padre" de categorías mas especificas y así mismo las categorías hijas pueden ser padre de otras mas especificas.

Atributos:

- `category_id`: un entero auto incremental que representa el índice del registro dentro de la tabla.
- `name`: cadena de caracteres que almacena el nombre de la categoría.
- `parent_category_id`: referencia al id de la categoría padre (puede ser nulo para las categorías mas especificas como [física](#) o [matemática](#)).
- `icon`: almacena la ruta donde se ubica el icono de la categoría. Un valor de ejemplo puede ser `./catFisica.jpg` para un icono que se encuentra en el directorio public del frontEnd.
- `enabled`: un valor booleano que se usa para hacer un "Soft Delete" de todas las categorías de que ya no se encuentren en uso por algun motivo (*ver inciso 1 en la sección de Observaciones*)



Relaciones:

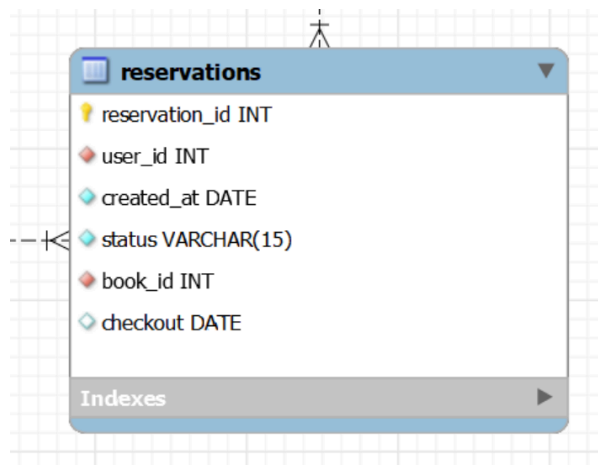
- 1:1 con [categories](#).
- 1:m con [categories_per_book](#).

reservations

En la aplicación la usuario puede generar una reserva, esto le permite pasar recogiendo un libro en la fecha estipulada. Existen situación durante la reserva que pueden afectar la reputación del usuario, para mayor información [ver el inciso 2 de la sección de Observaciones](#).

Atributos:

- reservation_id: entero auto incremental que representa el índice del registro dentro de la tabla.
- user_id: **llave foránea** a la tabla Usuarios que referencia al **id** del usuario que hizo la reserva (estudiante al que se le prestará el libro).
- ~~FechaReserva: fecha en la que el estudiante *debe* pasar recogiendo el libro en las instalaciones de la biblioteca.~~
- status: indica el cual es el estado actual de la reserva (si ya se atendió, se encuentra pendiente, fue aceptada, etc.). Los posibles valores son:
 - **Rechazada**: un administrador no aprobó la solicitud de reserva de un libro.
 - **Aceptada**: un administrador aprobó la solicitud de reserva de un libro.
 - **Pendiente**: la solicitud de reserva aún no ha sido aprobada ni rechazada por un administrador.
- book_id: **llave foránea** a la tabla **books**, indica el id del libro que se pide prestado.
- checkout: es la *fecha real* en la que el estudiante se avocó a la biblioteca para recibir su libro.
- created_at: fecha de creación de la reserva.



Relaciones:

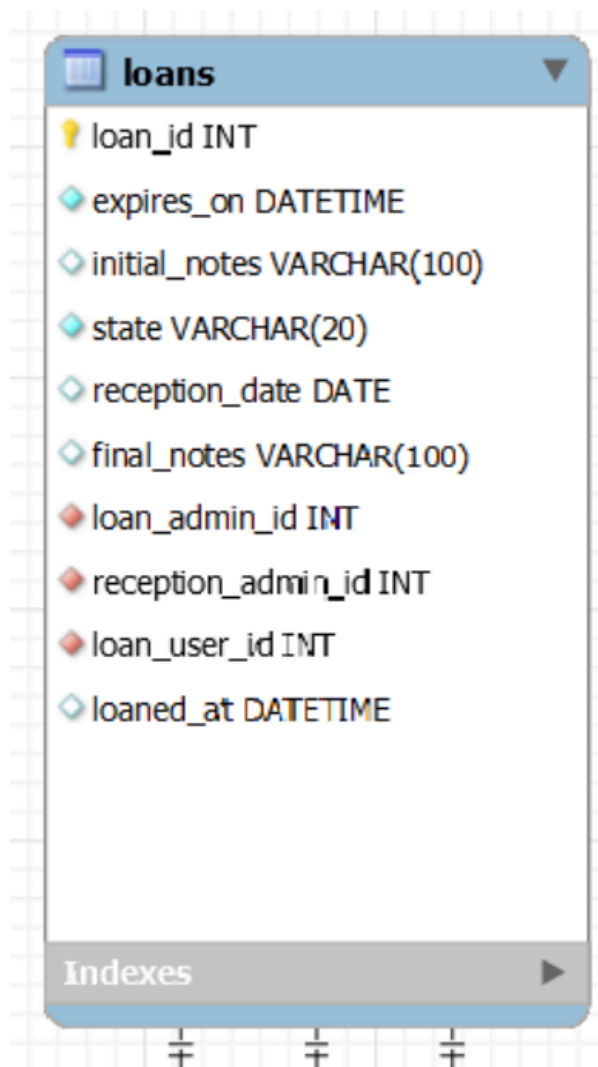
- m : 1 con **books**.
- m : 1 con **users**.
- 1 : 1 con **loans**.

loans

Registra todos los prestamos que se han hecho a través del tiempo.

Atributos:

- **loan_id**: entero auto incremental que representa el índice del registro dentro de la tabla.
- **expires_on**: es la fecha *en que se espera* que el estudiante *devuelva* el libro que se le prestó.
- **initial_notes**: en este campo el administrador puede dar una descripción de los daños mas relevantes que tiene el libro antes de ser prestado. De esa forma se puede verificar si el libro regresa con más daños.
- **state**: representa el estado actual del préstamo. Sus valores pueden ser:
 - **En Curso**: si la reserva del usuario fué aprobada y ya se le entregó el libro al usuario pero aun no llega el momento de regresarlo.
 - **Finalizada**: si la reserva del usuario fué aprobada y el usuario devolvió el libro en tiempo y forma.
 - **Sin Entregar**: si la reserva fué aprobada y el usuario no devolvió el libro dentro del tiempo definido o de plano no ha regresado aún.
- **reception_date**: es la *fecha real* en la que el usuario llega a biblioteca a *devolver* el libro.
- **final_notes**: en este campo el administrador puede escribir sobre daños adicionales (diferentes a los daños que ya tenía) que tenga el libro al ser recibido.
- **loan_admin_id**: es una **llave foránea** a la tabla **Usuarios** que hace referencia al id del Administrador que le presta el libro al estudiante.
- ~~**idReserva**: **llave foránea** a la tabla **Reserva** que hace referencia al id de la Reserva a la que le corresponde este registro de historial de préstamo.~~
- **reception_admin_id**: **llave foránea** a la tabla **Usuarios** que hace referencia al id del administrador que recibe el libro de vuelta, una vez que se ha completado el tiempo de préstamo.
- **loaned_at**: fecha y hora de la creacion de este registro (momento en que el usuario recibe el libro reservado).



loans	
loan_id	INT
expires_on	DATETIME
initial_notes	VARCHAR(100)
state	VARCHAR(20)
reception_date	DATE
final_notes	VARCHAR(100)
loan_admin_id	INT
reception_admin_id	INT
loan_user_id	INT
loaned_at	DATETIME

Indexes

Relaciones:

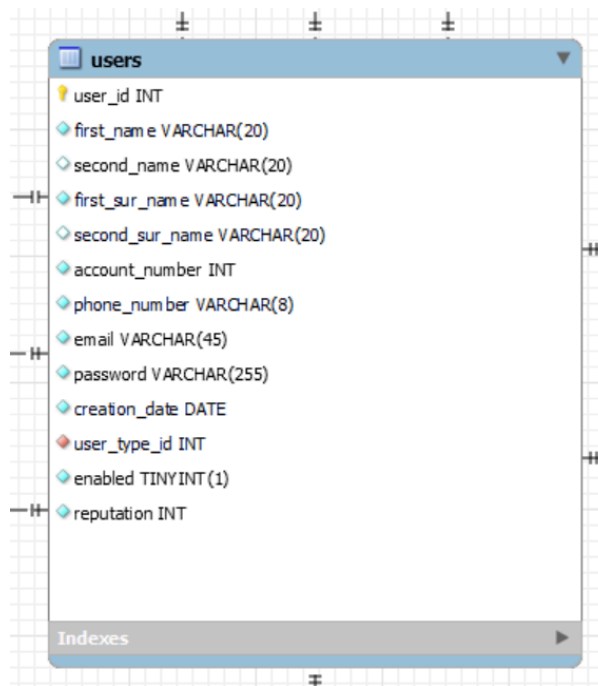
- I : I con [users](#) (*IDAdministradorPrestamo*).
- I : I con [users](#) (*idAdministradorRecepcion*).
- I : I con [reservations](#).

users

Almacena la información de todos los usuarios de la aplicación (tanto de administradores como de estudiantes).

Atributos:

- `user_id`: entero auto incremental que representa el índice del registro dentro de la tabla.
- `first_name`: almacena el primer nombre del usuario.
- `second_name`: almacena el segundo nombre del usuario.
- `first_sur_name`: almacena el primer apellido del usuario.
- `second_sur_name`: almacena el segundo apellido del usuario.
- `account_number`: almacena el número de cuenta (ej. 20211009089) del usuario.
- `phone_number`: almacena el número de teléfono (*sin guiones ni código de área*) del usuario.
- `email`: cadena de caracteres que almacena el correo institucional del usuario.
- `passord`: cadena de caracteres que almacena la *contraseña encriptada* del usuario.
- `creation_date`: la fecha de creacion de esa cuneta de usuario dentro de la aplicación.
- `user_type_id`: llave foránea a la tabla `user_types` que hace referencia al rol (*Estudiante o Administrador*) que tiene el usuario dentro de la aplicación.
- `enabled`: campo booleano que sirve para hacer un *Soft Delete* de las cuentas de Usuario. Cuando un usuario tiene la reputación mínima su cuenta se deshabilita (se banea del sistema) y pierde derecho a reservar libros.
- `reputation`: campo entero que almacena la reputación del usuario (0-10) donde todo nuevo usuario inicia con una reputación de 10 (la mejor reputación). *Ver inciso 2 en la Sección de Observaciones*



Relaciones:

- 1 : 1 con [user_types](#).
- 1 : m con [payments](#).
- 1 : m con [payments](#).
- 1 : m con [reservations](#).
- 1 : 1 con [loans](#).
- 1 : 1 con [loans](#).

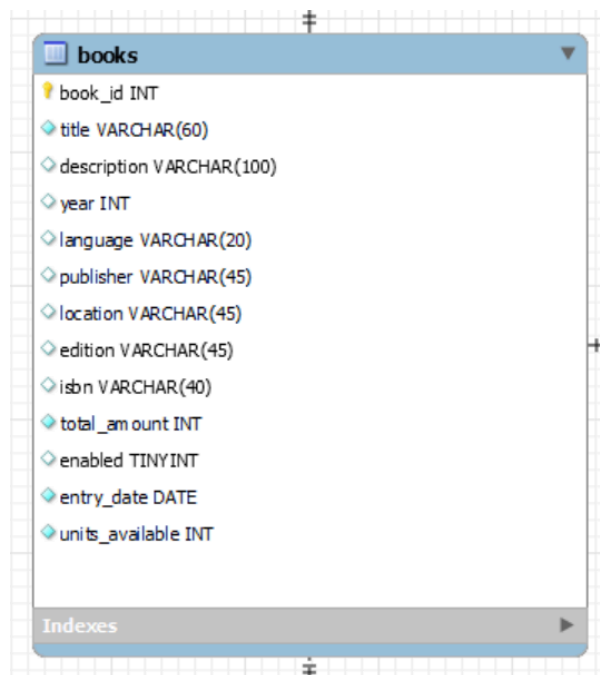
Nota: Se sugiere ver la definición de las tablas [payments](#) e [loans](#) en caso de duda sobre por que estas relaciones aparecen dos veces.

books

Almacena toda la información que define a un libro.

Atributos:

- `book_id`: entero auto incremental que representa el índice del registro dentro de la tabla.
- `title`: almacena el título del libro.
- `edition`: un entero que representa el número de edición del libro (ej. si es 6 se entiende que es la sexta edición del libro).
- `year`: año de publicación del libro.
- `language`: idioma del libro (puede ser "español" o "inglés").
- `publisher`: nombre de la editorial a cargo de la publicación del libro.
- `location`: país donde se ubica la editorial.
- `description`: una breve descripción sobre lo que trata el libro.
- `isbn`: *código Internacional normalizado para libros* es un identificador único para libros.
- `total_amount`: es el número total de ejemplares en existencia de ese libro.
- `enabled`: campo booleano que sirve para hacer un *Soft Delete* de los Libros que por algún motivo ya no estén disponibles para ser prestados.
- `entry_date`: representa la fecha en que se ingresa un nuevo libro al sistema.
- `units_available`: representa la cantidad de ejemplares que están disponibles para ser reservados. Si se presta un ejemplar entonces este número disminuye, cuando el ejemplar es regresado, este número aumenta.



Relaciones:

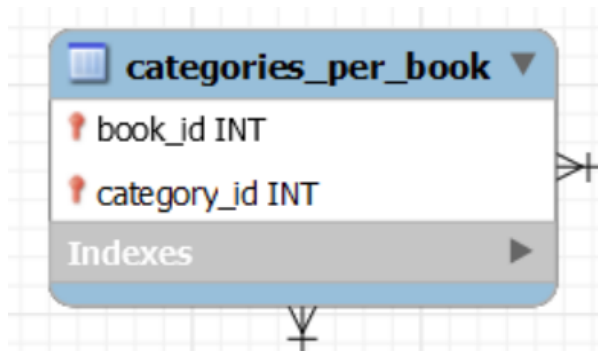
- 1 : m con [reservations](#).
- 1 : 1 con [categories_per_book](#).
- 1 : 1 con [authors_per_book](#).

categories_per_book

Tabla intermedia que relaciona a cada libro con sus respectivas categorías.

Atributos:

- book_id: llave foránea a la tabla [books](#).
- category_id: llave foránea a la tabla [categories](#).



Relaciones:

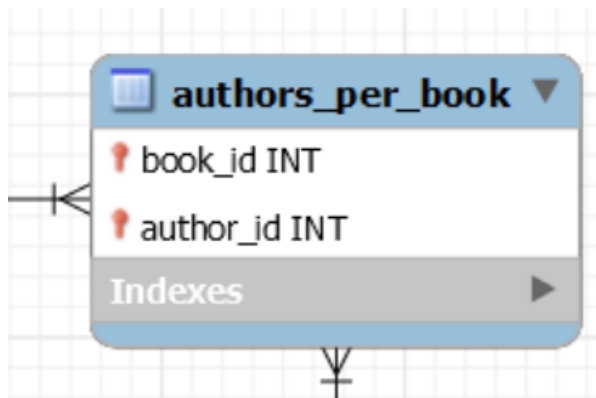
- m : 1 con [books](#).
- m : 1 con [categories](#).

authors_per_book

Tabla intermedia que relaciona los Libros con los Autores

Atributos:

- book_id: llave foránea a la tabla [books](#).
- author_id: llave foránea a la tabla [authors](#).



Relaciones:

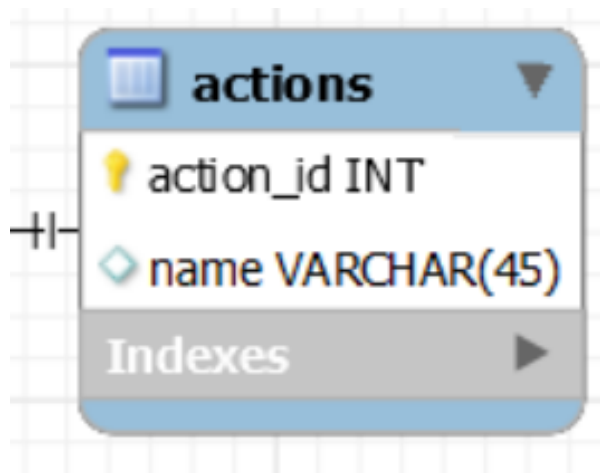
- m : 1 con [books](#).
- m : 1 con [authors](#).

actions

Contiene los nombre de las posibles acciones dentro del sistema.

Atributos:

- action_id: entero auto incremental que representa el índice del registro dentro de la tabla.
- name: almacena el nombre de la acción para mejor orden los posibles valores se han agrupado según la entidad principal a la que afectan (los posibles valores están en **negrita**).
 - acciones sobre libros
 - **editar Libro**
 - **crear Libro**
 - **eliminar Libro**
 - acciones sobre usuarios (estudiantes y administradores)
 - **crear Usuario**
 - **eliminar Usuario**
 - **cambiar reputación**
 - **editar usuario**
 - acciones sobre prestamos
 - **prestar libro**
 - **recibir libro**
 - acciones sobre reservas
 - **aceptar reserva de libro**
 - **rechazar reserva de libro**
 - acciones sobre categorías
 - **crear categoría**
 - **editar categoría**
 - **eliminar categoría**
 - acciones sobre autores
 - **crear autor**
 - **editar autor**
 - **eliminar autor**



Relaciones:

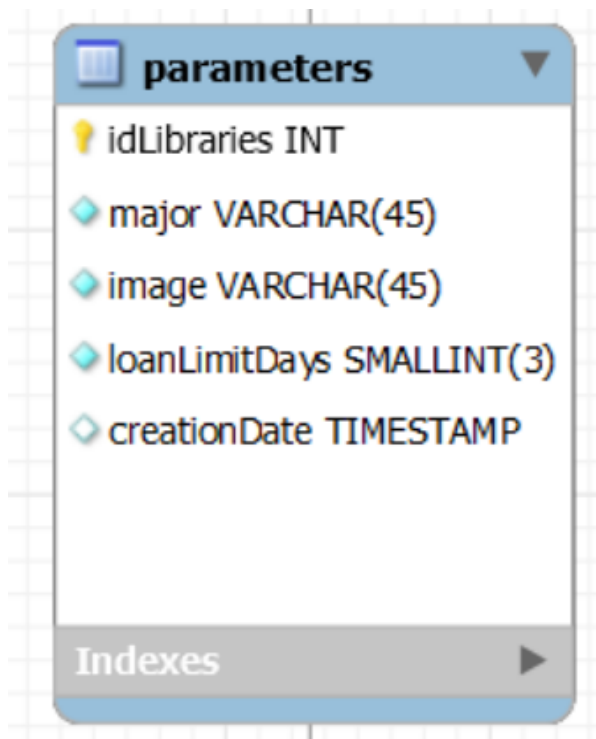
- 1 : m con [actions_history](#)

parameters

Almacena las características de distinguen a una biblioteca.

Atributos:

- idLibraries: entero auto incremental que representa el índice del registro dentro de la tabla.
- major: carrera a la que pertenece la biblioteca
- image: ruta a la imagen del logo de la carrera.
- loanLimitDays: entero que representa el limite de días que se puede prestar un libro.
- creationDate: almacena la fecha y hora de creación del registro de la biblioteca.



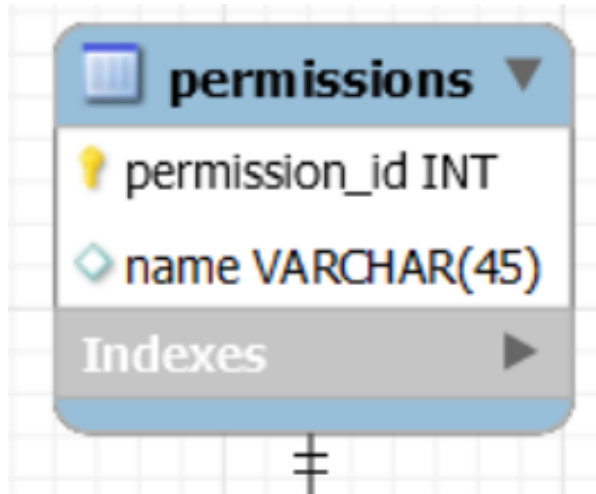
Relaciones: [Ninguna](#)

permissions

Almacena todos los permisos que puede tener un usuario.

Atributos:

- `permission_id`: entero auto incremental que representa el índice del registro dentro de la tabla.
- `name`: nombre del permiso. Sus posibles valores son:
 - **gestión de libros** (creacion, edición, borrado de autores y categorías)
 - **gestión de usuarios** (creacion, edición, borrado de reputación)



Relaciones:

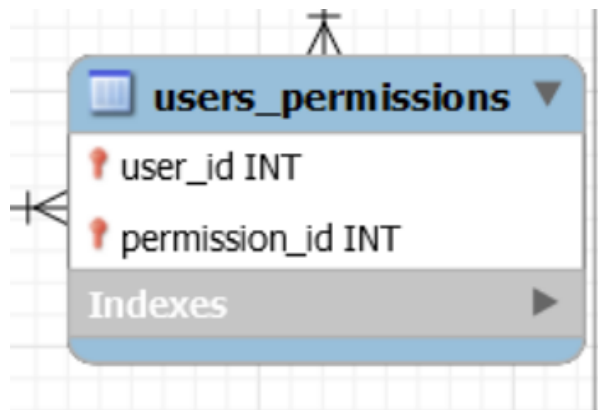
- 1 : m con [users_permissions](#)

users_permissions

Tabla intermedia entre la tabla [users](#) y la tabla [permissions](#).

Atributos:

- user_id: llave foranea a la tabla [users](#).
- permission_id: llave foranea a la tabla [permissions](#).



Relaciones:

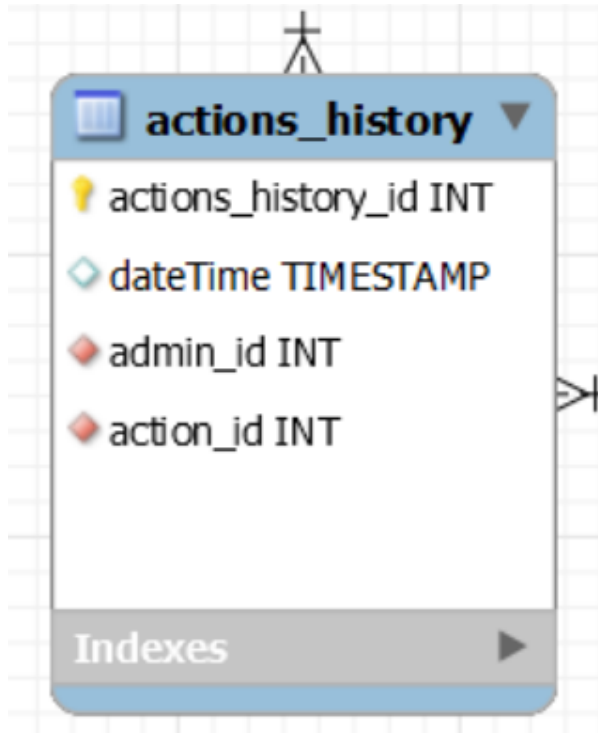
- m : 1 con [permissions](#)
- m : 1 con [users](#)

actions_history

Tabla intermedia entre la tabla [actions](#) y [users](#).

Atributos:

- actions_history_id: entero auto incremental que representa el índice del registro dentro de la tabla.
- dateTime: timestamp (fecha y hora) en la que se ejecutó la acción.
- admin_id: llave foránea a la tabla [users](#) que indica el usuario que ejecutó la acción.
- action_id: llave foránea a la tabla [actions](#) que indica la acción ejecutada.



Relaciones:

- m : 1 con [users](#)
- m : 1 con [actions](#)

Observaciones

A continuación se presentan explicaciones a mas detalladas sobre casos especiales en la base de datos.

1. Explicación de *soft delete*

El Soft Delete permite "*borrar registros*" sin borrarlos necesariamente. Todos los registros con el campo Habilitado en 0 serán los registro borrados. Por lo que en todas las operaciones solo se consideraran aquellos registros con el campo Habilitado en 1. Se hace de esta manera con la intención de que quede registro de la información dentro de la base de datos, a demás esto es útil en caso de auditorias a la base de datos.

2. Explicación de *sistema de reputación*

La reputación es un valor entero entre el rango de 0-10 donde cero es la peor reputación y 10 la mejor. Inicialmente todo usuario tendrá una reputación de 10. Sin embargo, existen situaciones que reducen su reputación:

- **Entrega Tarde:** Al usuario que ha reservado un libro y lo recibió en tiempo y forma se le establece una fecha de entrega del libro. Si el usuario entrega el libro despues de la fecha indicada su reputación se verá disminuida.
- **Daño al libro:** Si el usuario regresa el libro con daños considerables su reputación también se vera disminuida.
- **Reserva no completada:** Si el usuario reserva un libro pero no se avoca a las instalaciones de la biblioteca para recibir el libro, su reputación disminuye.
- **No entrega el libro:** Si el usuario no entrega el libro, disminuye su reputación considerablemente