

Relatório de Sistema - Livraria

Relatório de Sistema - Livraria

DIAGRAMAS

CASO DE USO

DIAGRAMA DE CLASS

PROTÓTIPO

DER

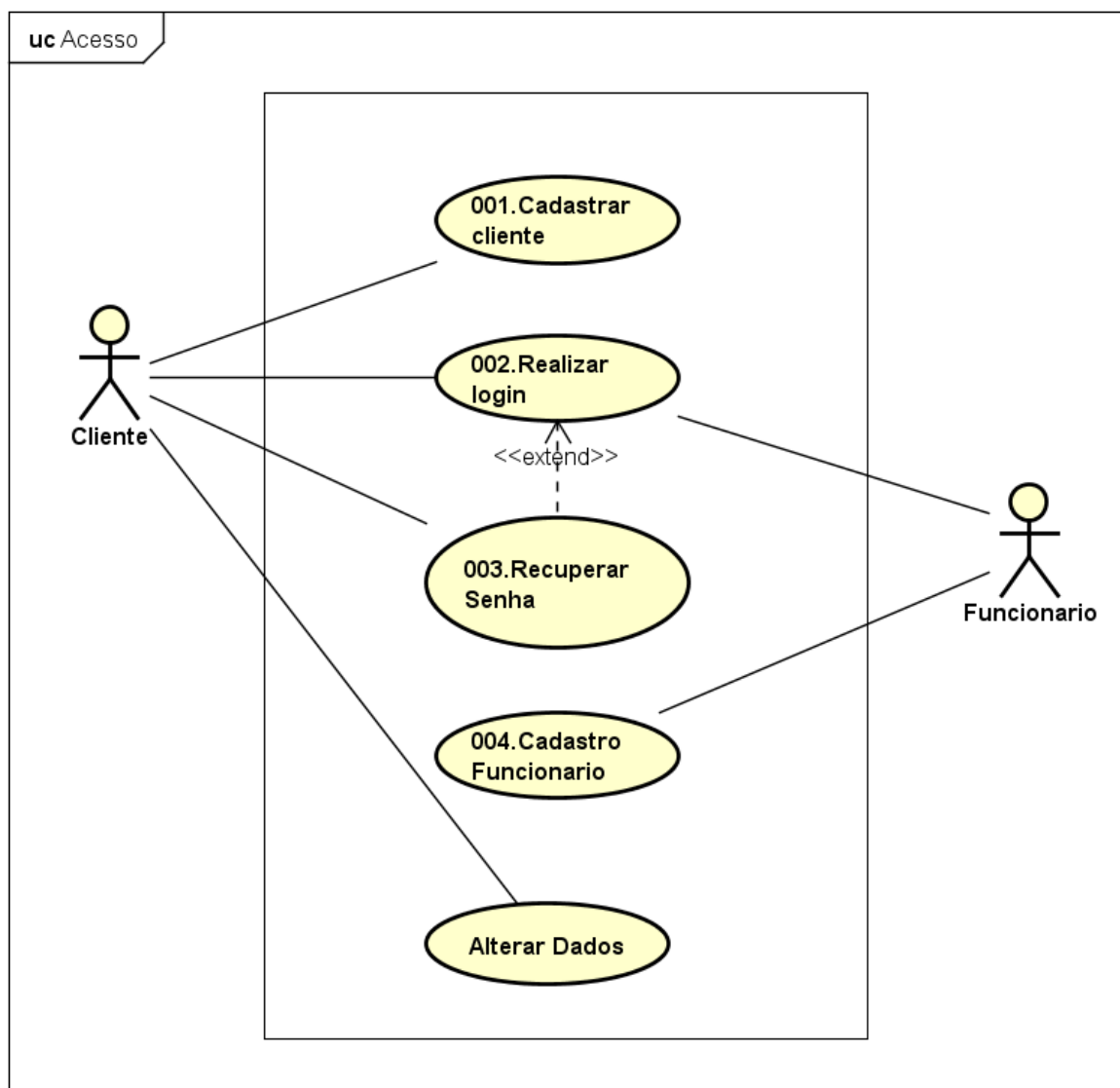
MODELAGEM

SCRIPT SQL

LINKS ÚTEIS

DIAGRAMAS

CASO DE USO



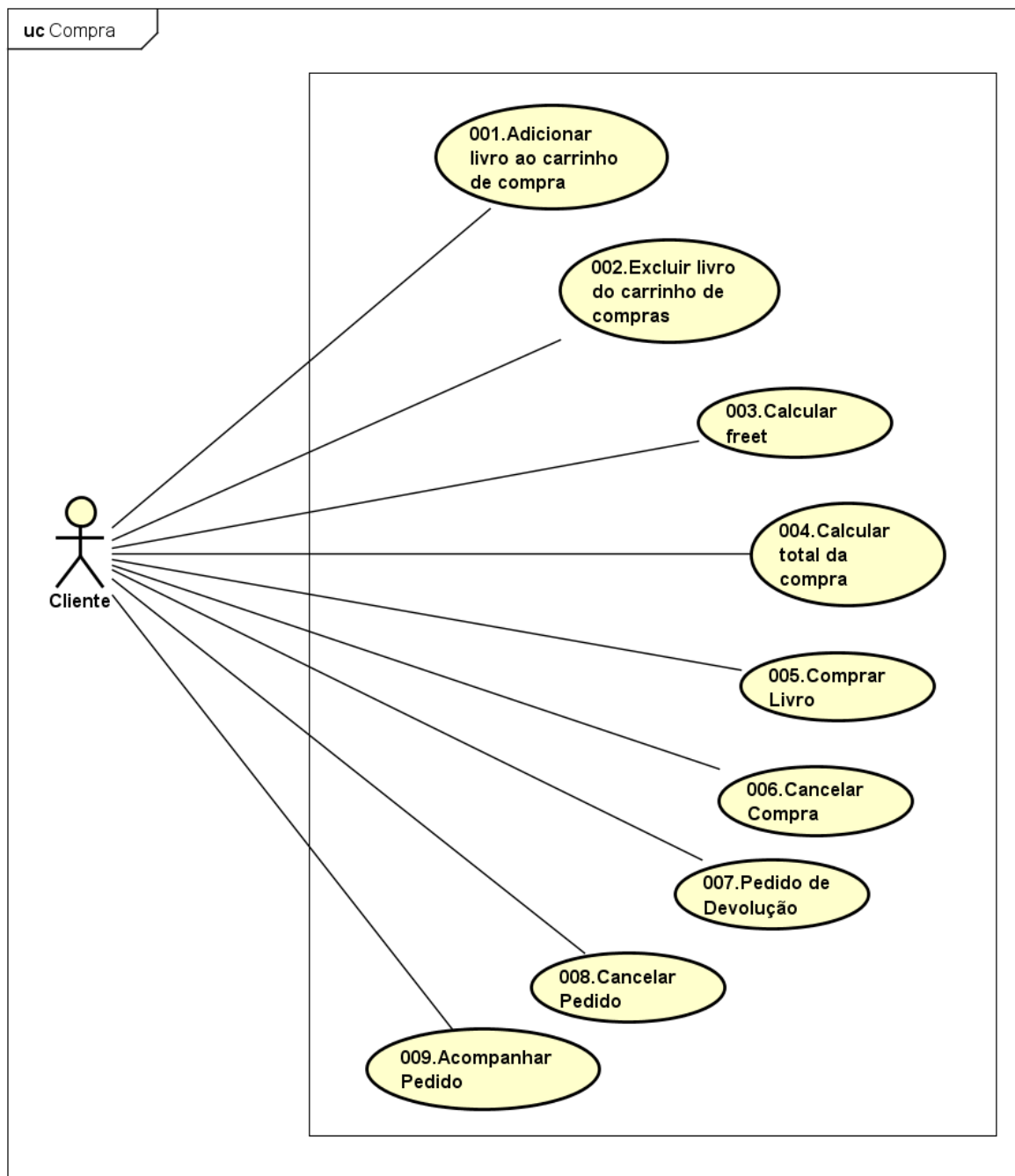
Usuario que acessarem o site deve conseguir realizar seu cadastro para terem acesso a [compra, favoritos, carinho], neste cadastro é obrigatório a coleta no nome completo, cpf, email, usuario e senha.

Já devidamente cadastrado no sistema deve ser possível realizar login com seu email ou usuário e senha.

Caso necessário o usuário deve poder realizar o cadastro de sua senha, para isso deve informar o cpf e numero de telefone ou e-mail cadastrado, após informar os dados será enviado para ele um código que deverá digitar no código para poder alterar sua senha, tempo válido do código é de 2 horas.

O cadastro de funcionário é permitido apenas aos funcionários com cargo de administrador do sistema, dados a eles a função de cadastrar todos os funcionários que adicionaram novos anúncios ao sistema.

Cliente deve poder realizar a alteração de seus dados cadastrais.



Não é necessário ter cadastro de para acessar essa funcionalidade, nela deve ser possível visualizar todos os livros cadastrados no sistema. Caso o usuário esteja logado o sistema poderá se basear em suas últimas compras, favoritos e produtos adicionados como favoritos para trazer um catálogo mais utilizado ao usuário.

Não é necessário ter cadastro de para acessar essa funcionalidade, nela deve ser possível adicionar filtros a pesquisa[por gênero, ano de lançamento, título, autor, editora, edição ...].

Nesta funcionalidade usuário deve estar devidamente cadastrado como listado no requisito 001. Cadastrar Cliente do módulo de acesso, nela o usuário do sistema poderá adicionar os livros que tem interesse em sua lista de favoritos facilitando o acesso a ele.

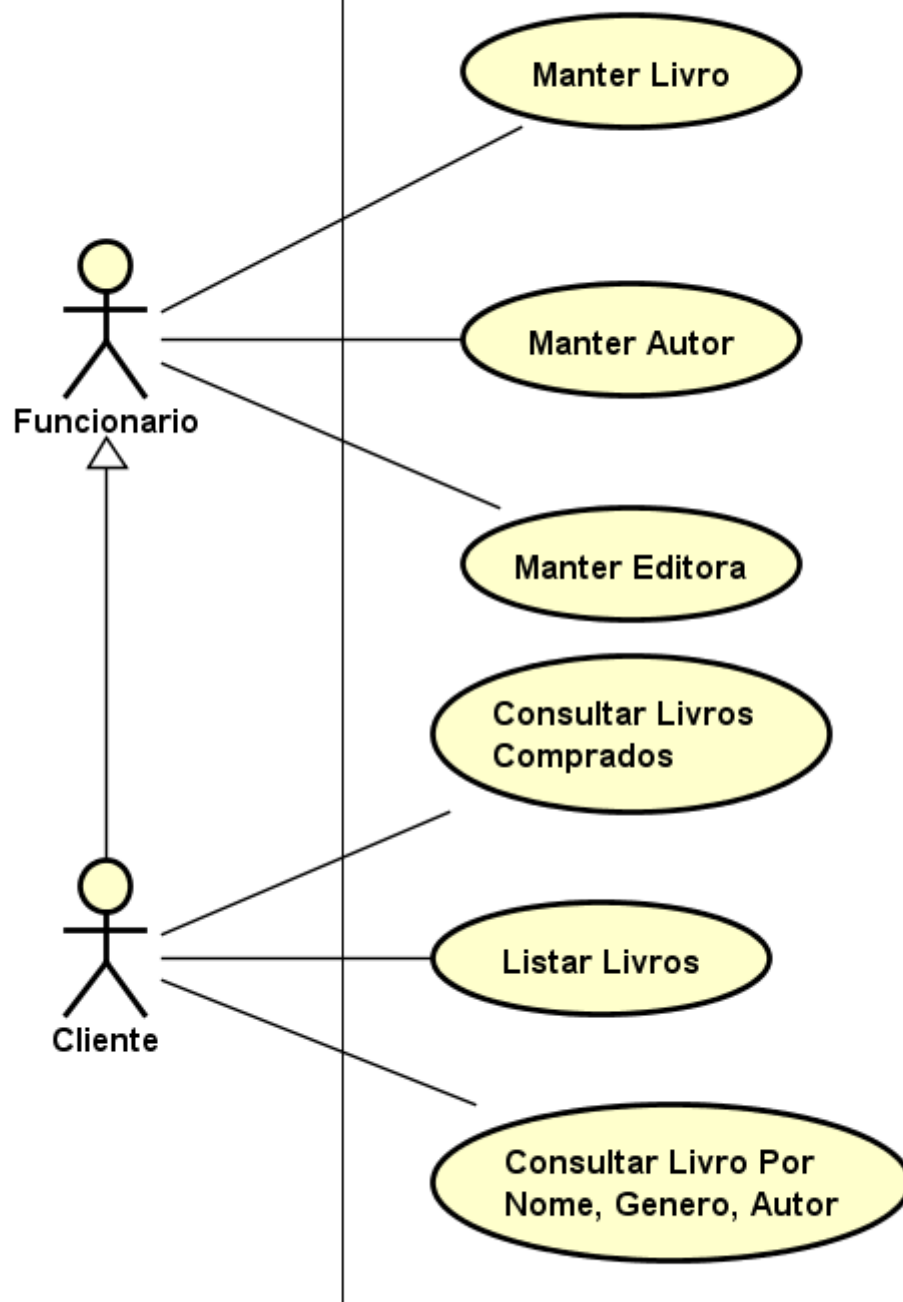
Não é necessário ter cadastro de para acessar essa funcionalidade, após consultar o livro no catálogo, deve ser possível visualizar todas as informações de um livro ao selecioná-lo, tendo também a opção de adicionar ao carrinho, ler a sinopse do livro, calcular frete.

Funcionários devem conseguir adicionar livros ao sistema, a partir da página de adicionar livro deve ser possível adicionar autor, editora e todas as outras informações de um livro

Deve ser permitido ao cliente realizar a consulta e monitoramento de status de todos os seus pedidos.

Cliente deve conseguir realizar o pedido de devolução, podendo realizar essa devolução dentro de um prazo de no máximo 10 dias após o recebimento do pedido.

Cliente deve poder cancelar uma compra, essa compra não deve ser cancelada caso o item ainda não tenha sido enviado caso já tenha o cliente deve aguardar sua chegada e fazer o pedido de devolução .



Não é necessário ter cadastro de para acessar essa funcionalidade, nela deve ser possível visualizar todos os livros cadastrados no sistema. Caso o usuário esteja logado o sistema poderá se basear em suas últimas compras, favoritos e produtos adicionados como favoritos para trazer um catálogo mais utilizado ao usuário.

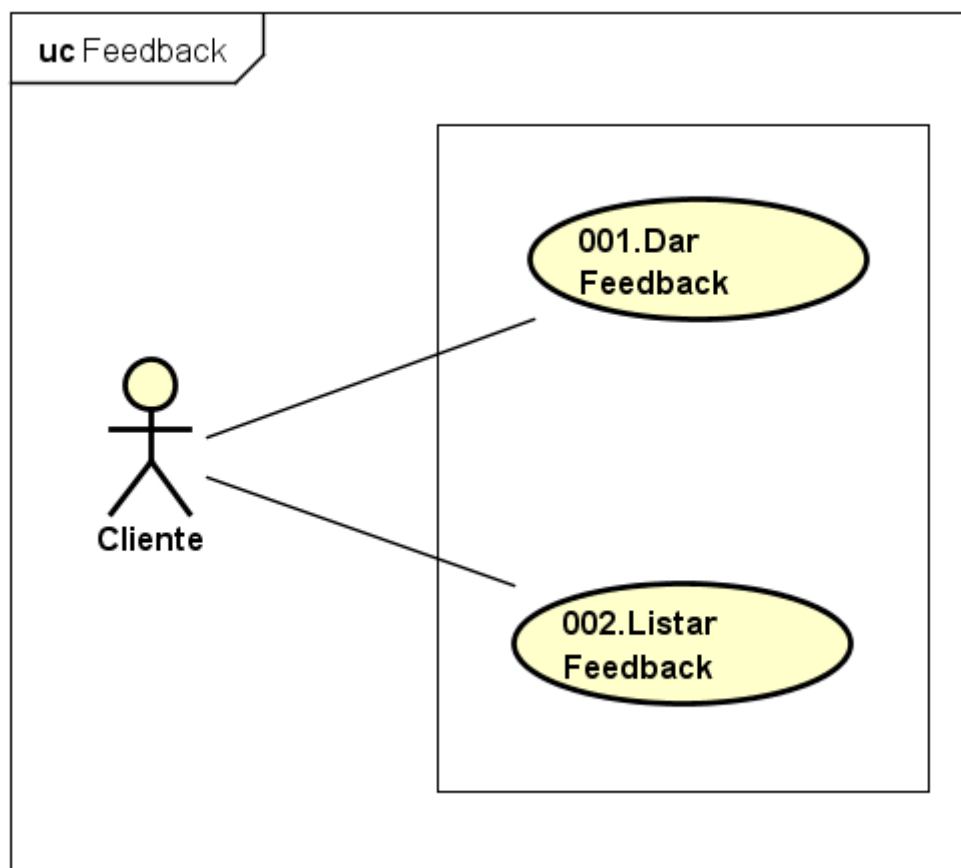
Não é necessário ter cadastro de para acessar essa funcionalidade, nela deve ser possível adicionar filtros a pesquisa [por gênero, ano de lançamento, título, autor, editora, edição ...].

Nesta funcionalidade usuário deve estar devidamente cadastrado como listado no requisito 001. Cadastrar Cliente do módulo de acesso, nela o usuário do sistema poderá adicionar os livros que tem interesse em sua lista de favoritos facilitando o acesso a ele.

Funcionário deve conseguir inserir e realizar a manutenção dos Autores cadastrados no sistema (podendo inserir, excluir e alterar qualquer um caso necessário).

Funcionário deve conseguir inserir e realizar a manutenção dos Editores cadastrados no sistema (podendo inserir, excluir e alterar qualquer um caso necessário).

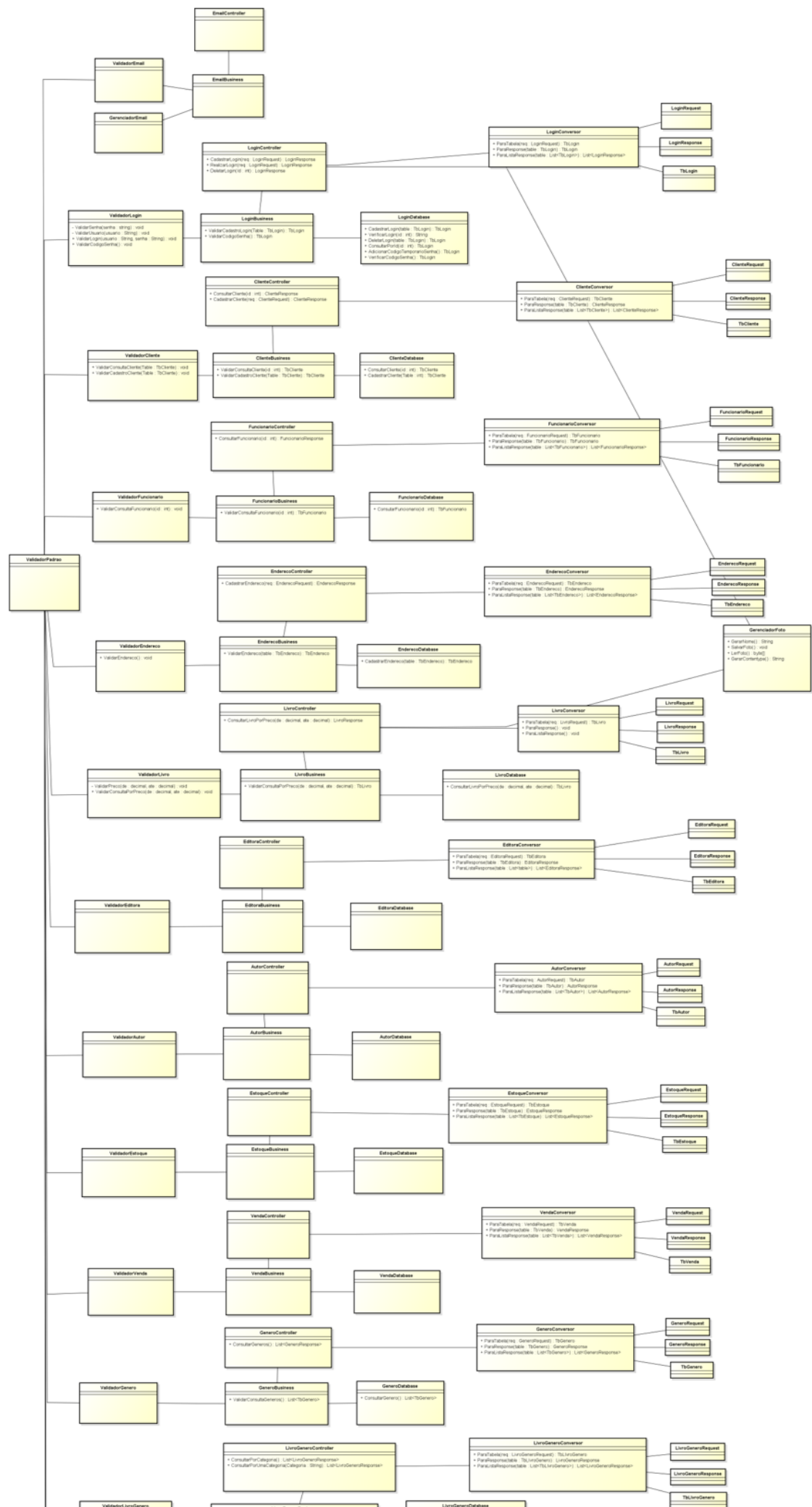
Funcionários devem conseguir adicionar livros ao sistema, a partir da página de adicionar livro deve ser possível adicionar autor, editora e todas as outras informações de um livro

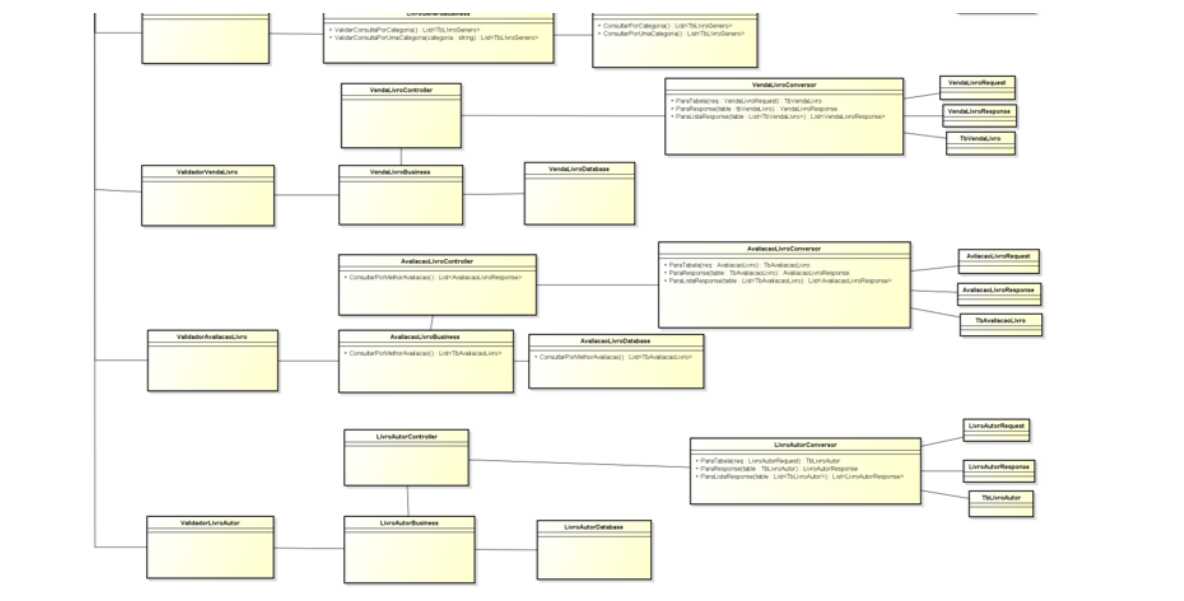


Deve ser possível que cada livro que um cliente compra ele possa avaliar e comentar sua experiência com essa compra.

Clientes devem poder visualizar todos os seus feedbacks, outros clientes devem também poder consultar os comentários de outros clientes na aba de consulta de livros para poderem ter conhecimento de outros usuários sobre o livro que desejam adquirir.

DIAGRAMA DE CLASS





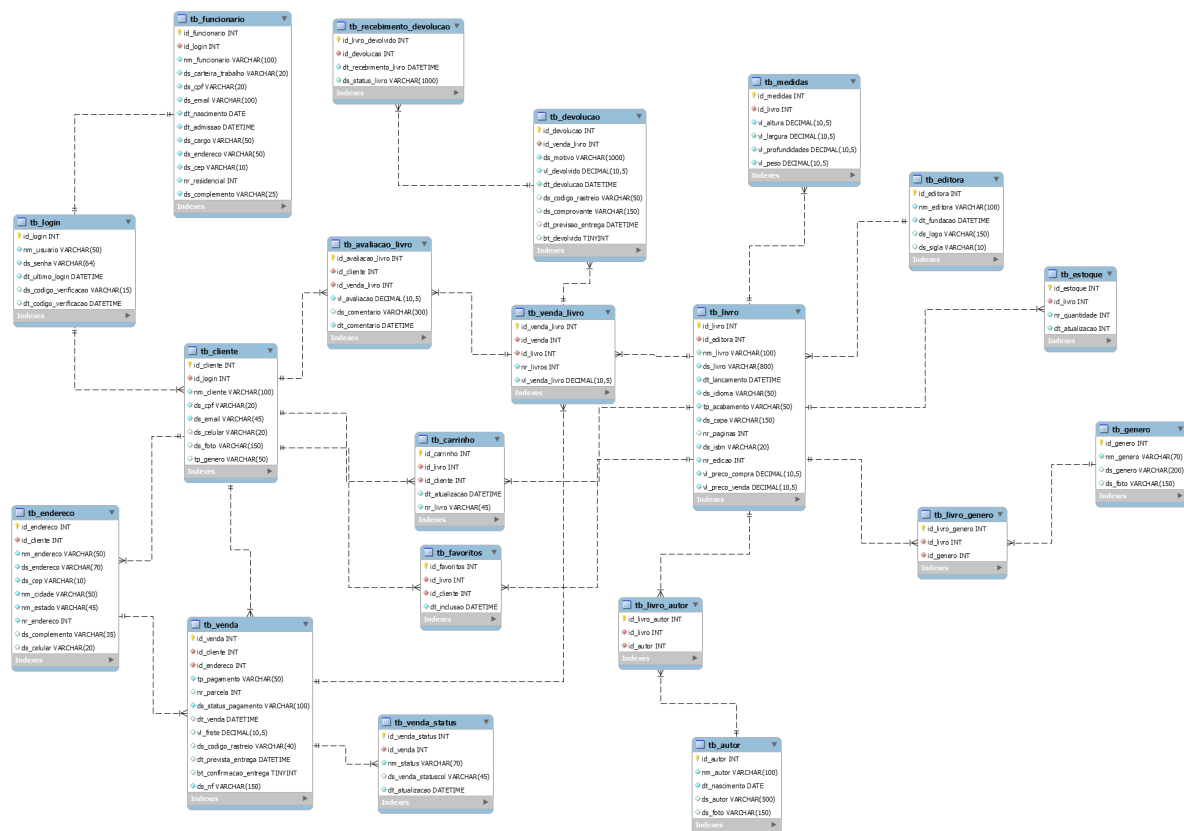
[Diagrama de class no GitHub](#)

PROTÓTIPO

[PROTÓTIPO FIGMA](#)

DER

MODELAGEM



[Link da Modelagem de Dados no GitHub](#)

SCRIPT SQL

-- Mapear banco de dados:


```

-- dotnet ef dbcontext scaffold "server=localhost;user
id=root;password=45923617xx;database=db_next_gen_books"
Pomelo.EntityFrameworkCore.MySql -o Models --data-annotations --force

-- -----

-- MySQL Script generated by MySQL Workbench

-- Thu Oct 15 22:06:42 2020

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

-- -----

-- Schema db_next_gen_books

-- -----

-- Schema db_next_gen_books

-- -----

DROP DATABASE IF EXISTS `db_next_gen_books`;

CREATE DATABASE IF NOT EXISTS `db_next_gen_books`;

USE `db_next_gen_books` ;

-- -----

-- Table `db_next_gen_books`.`tb_login`

-- -----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_login` (

  `id_login` INT NOT NULL AUTO_INCREMENT,

  `nm_usuario` VARCHAR(50) NOT NULL,

  `ds_senha` VARCHAR(64) NOT NULL,

  `dt_ultimo_login` DATETIME NOT NULL,

  `ds_codigo_verificacao` VARCHAR(15) NULL,

```

```

`dt_codigo_verificacao` DATETIME NULL,

PRIMARY KEY (`id_login`))

ENGINE = InnoDB;

-----

-- Table `db_next_gen_books`.`tb_funcionario`
-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_funcionario` (

  `id_funcionario` INT NOT NULL AUTO_INCREMENT,

  `id_login` INT NOT NULL,

  `nm_funcionario` VARCHAR(100) NOT NULL,

  `ds_carteira_trabalho` VARCHAR(20) NOT NULL,

  `ds_cpf` VARCHAR(20) NOT NULL,

  `ds_email` VARCHAR(100) NOT NULL,

  `dt_nascimento` DATE NOT NULL,

  `dt_admissao` DATETIME NOT NULL,

  `ds_cargo` VARCHAR(50) NOT NULL,

  `ds_endereco` VARCHAR(50) NOT NULL,

  `ds_cep` VARCHAR(10) NOT NULL,

  `nr_residencial` INT NOT NULL,

  `ds_complemento` VARCHAR(25) NOT NULL,

  PRIMARY KEY (`id_funcionario`),

  INDEX `fk_tb_funcionario_tb_login1_idx` (`id_login` ASC) VISIBLE,

  FOREIGN KEY (`id_login`)

    REFERENCES `db_next_gen_books`.`tb_login` (`id_login`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

```
-- Table `db_next_gen_books`.`tb_cliente`
```

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_cliente` (  
  `id_cliente` INT NOT NULL AUTO_INCREMENT,  
  `id_login` INT NOT NULL,  
  `nm_cliente` VARCHAR(100) NOT NULL,  
  `ds_cpf` VARCHAR(20) NOT NULL,  
  `ds_email` VARCHAR(45) NOT NULL,  
  `ds_celular` VARCHAR(20) NULL,  
  `ds_foto` VARCHAR(150) NULL,  
  `tp_genero` VARCHAR(50) NULL,  
  PRIMARY KEY (`id_cliente`),  
  INDEX `id_login_idx` (`id_login` ASC) VISIBLE,  
  UNIQUE INDEX `ds_foto_UNIQUE` (`ds_foto` ASC) VISIBLE,  
  FOREIGN KEY (`id_login`)  
    REFERENCES `db_next_gen_books`.`tb_login` (`id_login`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `db_next_gen_books`.`tb_endereco`
```

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_endereco` (  
  `id_endereco` INT NOT NULL AUTO_INCREMENT,  
  `id_cliente` INT NOT NULL,  
  `nm_endereco` VARCHAR(50) NOT NULL,  
  `ds_endereco` VARCHAR(70) NOT NULL,
```

```

`ds_cep` VARCHAR(10) NOT NULL,

`nm_cidade` VARCHAR(50) NOT NULL,

`nm_estado` VARCHAR(45) NOT NULL,

`nr_endereco` INT NOT NULL,

`ds_complemento` VARCHAR(35) NOT NULL,

`ds_celular` VARCHAR(20) NULL,

PRIMARY KEY (`id_endereco`),

INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,

FOREIGN KEY (`id_cliente`)

REFERENCES `db_next_gen_books`.`tb_cliente` (`id_cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_editora`

```

```

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_editora` (

  `id_editora` INT NOT NULL AUTO_INCREMENT,

  `nm_editora` VARCHAR(100) NOT NULL,

  `dt_fundacao` VARCHAR(45) NOT NULL,

  `ds_logo` VARCHAR(150) NULL,

  `ds_sigla` VARCHAR(10) NULL,

  PRIMARY KEY (`id_editora`))

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_livro`

```

```

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_livro` (

  `id_livro` INT NOT NULL AUTO_INCREMENT,

  `id_editora` INT NOT NULL,

  `nm_livro` VARCHAR(100) NOT NULL,

  `ds_livro` VARCHAR(800) NOT NULL,

  `dt_lancamento` DATETIME NOT NULL,

  `ds_idioma` VARCHAR(50) NOT NULL,

  `tp_acabamento` VARCHAR(50) NOT NULL,

  `ds_capa` VARCHAR(150) NOT NULL,

  `nr_paginas` INT NULL,

  `ds_isbn_10` VARCHAR(20) NOT NULL,

  `ds_isbn_13` VARCHAR(20) NOT NULL,

  `nr_edicao` INT NOT NULL,

  `vl_preco_compra` DECIMAL(10,5) NOT NULL,

  `vl_preco_venda` DECIMAL(10,5) NOT NULL,

  PRIMARY KEY (`id_livro`),

  INDEX `id_editora_idx` (`id_editora` ASC) VISIBLE,

  UNIQUE INDEX `ds_capa_UNIQUE` (`ds_capa` ASC) VISIBLE,

  FOREIGN KEY (`id_editora`)

    REFERENCES `db_next_gen_books`.`tb_editora` (`id_editora`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_autor`

```

```

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_autor` (

  `id_autor` INT NOT NULL AUTO_INCREMENT,

```

```

`nm_autor` VARCHAR(100) NOT NULL,

`dt_nascimento` DATE NOT NULL,

`ds_autor` VARCHAR(500) NULL,

`ds_foto` VARCHAR(150) NULL,

PRIMARY KEY (`id_autor`))

ENGINE = InnoDB;

-----

-- Table `db_next_gen_books`.`tb_livro_autor`

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_livro_autor` (

  `id_livro_autor` INT NOT NULL AUTO_INCREMENT,

  `id_livro` INT NOT NULL,

  `id_autor` INT NOT NULL,

  PRIMARY KEY (`id_livro_autor`),

  INDEX `id_livro_idx` (`id_livro` ASC) VISIBLE,

  INDEX `id_autor_idx` (`id_autor` ASC) VISIBLE,

  FOREIGN KEY (`id_livro`)

    REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  FOREIGN KEY (`id_autor`)

    REFERENCES `db_next_gen_books`.`tb_autor` (`id_autor`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `db_next_gen_books`.`tb_genero`

```

```

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_genero` (

  `id_genero` INT NOT NULL AUTO_INCREMENT,

  `nm_genero` VARCHAR(70) NOT NULL,

  `ds_genero` VARCHAR(200) NULL,

  `ds_foto` VARCHAR(150) NULL,

  PRIMARY KEY (`id_genero`))

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_livro_genero`

```

```

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_livro_genero` (

  `id_livro_genero` INT NOT NULL AUTO_INCREMENT,

  `id_livro` INT NOT NULL,

  `id_genero` INT NOT NULL,

  PRIMARY KEY (`id_livro_genero`),

  INDEX `id_livro_idx` (`id_livro` ASC) VISIBLE,

  INDEX `id_genero_idx` (`id_genero` ASC) VISIBLE,

  FOREIGN KEY (`id_livro`)

    REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  FOREIGN KEY (`id_genero`)

    REFERENCES `db_next_gen_books`.`tb_genero` (`id_genero`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_venda`

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_venda` (

  `id_venda` INT NOT NULL AUTO_INCREMENT,

  `id_cliente` INT NOT NULL,

  `id_endereco` INT NOT NULL,

  `tp_pagamento` VARCHAR(50) NOT NULL,

  `nr_parcela` INT NULL,

  `ds_status_pagamento` VARCHAR(100) NOT NULL,

  `dt_venda` DATETIME NULL,

  `vl_frete` DECIMAL(10,5) NULL,

  `ds_codigo_rastreio` VARCHAR(40) NULL,

  `dt_prevista_entrega` DATETIME NULL,

  `bt_confirmacao_entrega` TINYINT NULL,

  `ds_nf` VARCHAR(150) NOT NULL,

  PRIMARY KEY (`id_venda`),

  INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,

  INDEX `id_endereco_idx` (`id_endereco` ASC) VISIBLE,

  FOREIGN KEY (`id_cliente`)

    REFERENCES `db_next_gen_books`.`tb_cliente` (`id_cliente`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  FOREIGN KEY (`id_endereco`)

    REFERENCES `db_next_gen_books`.`tb_endereco` (`id_endereco`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

```



```
-- -----  
-- Table `db_next_gen_books`.`tb_venda_livro`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_venda_livro` (  
  `id_venda_livro` INT NOT NULL AUTO_INCREMENT,  
  `id_venda` INT NOT NULL,  
  `id_livro` INT NOT NULL,  
  `nr_livros` INT NOT NULL,  
  `vl_venda_livro` DECIMAL(10,5) NOT NULL,  
  PRIMARY KEY (`id_venda_livro`),  
  INDEX `id_venda_idx` (`id_venda` ASC) VISIBLE,  
  INDEX `id_livro_idx` (`id_livro` ASC) VISIBLE,  
  FOREIGN KEY (`id_venda`)  
    REFERENCES `db_next_gen_books`.`tb_venda` (`id_venda`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  FOREIGN KEY (`id_livro`)  
    REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `db_next_gen_books`.`tb_avaliao_livro`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_avaliao_livro` (  
  `id_avaliao_livro` INT NOT NULL AUTO_INCREMENT,  
  `id_cliente` INT NOT NULL,  
  `id_venda_livro` INT NOT NULL,
```

```

`vl_avaliacao` DECIMAL(10,5) NOT NULL,

`ds_comentario` VARCHAR(300) NULL,

`dt_comentario` DATETIME NOT NULL,

INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,

INDEX `id_venda_livro_idx` (`id_venda_livro` ASC) VISIBLE,

PRIMARY KEY (`id_avaliacao_livro`),

FOREIGN KEY (`id_cliente`)

REFERENCES `db_next_gen_books`.`tb_cliente` (`id_cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

FOREIGN KEY (`id_venda_livro`)

REFERENCES `db_next_gen_books`.`tb_venda_livro` (`id_venda_livro`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_estoque`
-----

```

```

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_estoque` (

`id_estoque` INT NOT NULL AUTO_INCREMENT,

`id_livro` INT NOT NULL,

`nr_quantidade` INT NOT NULL,

`dt_atualizacao` INT NOT NULL,

PRIMARY KEY (`id_estoque`),

INDEX `id_livro_idx` (`id_livro` ASC) VISIBLE,

FOREIGN KEY (`id_livro`)

REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)

ON DELETE NO ACTION

```

```

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `db_next_gen_books`.`tb_favoritos`
-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_favoritos` (
  `id_favoritos` INT NOT NULL AUTO_INCREMENT,
  `id_livro` INT NOT NULL,
  `id_cliente` INT NOT NULL,
  `dt_inclusao` DATETIME NOT NULL,
  PRIMARY KEY (`id_favoritos`),
  UNIQUE INDEX `id_livro_UNIQUE` (`id_livro` ASC) VISIBLE,
  INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,
  FOREIGN KEY (`id_cliente`)
    REFERENCES `db_next_gen_books`.`tb_cliente` (`id_cliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  FOREIGN KEY (`id_livro`)
    REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `db_next_gen_books`.`tb_venda_status`
-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_venda_status` (
  `id_venda_status` INT NOT NULL AUTO_INCREMENT,

```

```

`id_venda` INT NOT NULL,

`nm_status` VARCHAR(70) NOT NULL,

`ds_venda_statuscol` VARCHAR(45) NULL,

`dt_atualizacao` DATETIME NOT NULL,

PRIMARY KEY (`id_venda_status`),

INDEX `id_venda_idx` (`id_venda` ASC) VISIBLE,

FOREIGN KEY (`id_venda`)

REFERENCES `db_next_gen_books`.`tb_venda` (`id_venda`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

```

-----

-- Table `db_next_gen_books`.`tb_medidas`

```

```

-----

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_medidas` (

`id_medidas` INT NOT NULL AUTO_INCREMENT,

`id_livro` INT NOT NULL,

`vl_altura` DECIMAL(10,5) NOT NULL,

`vl_largura` DECIMAL(10,5) NOT NULL,

`vl_profundidades` DECIMAL(10,5) NOT NULL,

`vl_peso` DECIMAL(10,5) NOT NULL,

PRIMARY KEY (`id_medidas`),

INDEX `id_livro_idx` (`id_livro` ASC) VISIBLE,

UNIQUE INDEX `id_livro_UNIQUE` (`id_livro` ASC) VISIBLE,

FOREIGN KEY (`id_livro`)

REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB;
```

```
-- Table `db_next_gen_books`.`tb_devolucao`
```

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_devolucao` (  
  `id_devolucao` INT NOT NULL AUTO_INCREMENT,  
  `id_venda_livro` INT NOT NULL,  
  `ds_motivo` VARCHAR(1000) NOT NULL,  
  `vl_devolvido` DECIMAL(10,5) NOT NULL,  
  `dt_devolucao` DATETIME NOT NULL,  
  `ds_codigo_rastreio` VARCHAR(50) NULL,  
  `ds_comprovante` VARCHAR(150) NULL,  
  `dt_previsao_entrega` DATETIME NULL,  
  `bt_devolvido` TINYINT NULL,  
  PRIMARY KEY (`id_devolucao`),  
  FOREIGN KEY (`id_venda_livro`)  
    REFERENCES `db_next_gen_books`.`tb_venda_livro` (`id_venda_livro`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `db_next_gen_books`.`tb_recebimento_devolucao`
```

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_recebimento_devolucao` (  
  `id_livro_devolvido` INT NOT NULL AUTO_INCREMENT,  
  `id_devolucao` INT NOT NULL,  
  `dt_recebimento_livro` DATETIME NOT NULL,
```

```

`ds_status_livro` VARCHAR(1000) NOT NULL,

PRIMARY KEY (`id_livro_devolvido`),

INDEX `id_devolucao_idx` (`id_devolucao` ASC) VISIBLE,

FOREIGN KEY (`id_devolucao`)

REFERENCES `db_next_gen_books`.`tb_devolucao` (`id_devolucao`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

-- Table `db_next_gen_books`.`tb_carrinho`

```

CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_carrinho` (

`id_carrinho` INT NOT NULL AUTO_INCREMENT,

`id_livro` INT NOT NULL,

`id_cliente` INT NOT NULL,

`dt_atualizacao` DATETIME NOT NULL,

`nr_livro` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id_carrinho`),

INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,

INDEX `id_livro_idx` (`id_livro` ASC) VISIBLE,

FOREIGN KEY (`id_cliente`)

REFERENCES `db_next_gen_books`.`tb_cliente` (`id_cliente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

FOREIGN KEY (`id_livro`)

REFERENCES `db_next_gen_books`.`tb_livro` (`id_livro`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB;
```

LINKS ÚTEIS

[Trello](#)

[GitHub](#)

[Google Drive](#)