



NEXT GEN BOOK

Relatório de Requisitos

São Paulo (SP)

2020

Sumário

Integrantes	4
MODULOS	5
1. ACESSO	5
2. ANÚNCIOS	5
3. COMPRAR	5
4. FEEDBACK	5
LINKS	5
TRELLO:	5
GITHUB:	5
RELATÓRIO:	5
CASO DE USO	6
UC - ACESSO	7
001. Cadastrar Cliente	7
002. Logar	7
003. Alterar Senha	7
004. Cadastro de Funcionário	7
005. Remover Funcionário	7
UC – ANÚNCIOS	8
001. Listar Livros	8
002. Pesquisar Livros por Filtro	8
003. Favorita Livro	8
004. Selecionar Produto	8
005. Adicionar Livro	8
UC – COMPRA	9
001. Adicionar Livro ao Carrinho de Compra	9
002. Excluir Livro ao Carrinho de Compra	9
003. Calcular frete	9
004. Calcular total da Compra	9
005. Comprar Livro	9
006. Cancelar Compra	9
UC - FEEDBACK	10
001. Dar Feedback	10
002. Listar Feedback	10
003. Alterar Feedback	10
PROTÓTIPOS	11

PROT.001 - HOME	12
PROT.002 – LOGIN/ CADASTRO	13
PROT – 003.VALIDAR USUÁRIO.....	14
PROT - 004.ALTERAR SENHA	15
PROT – 005.CADASTRO FUNCIONARIO	16
PROT 006.FAVORITOS	18
PROT – 007.INFORMACÕES DE PRODUTO	19
PROT – 008.PERFIL	20
PROT – 009.CARRINHO.....	21
DER	22
MODELAGEM DO BANCO DE DADOS.....	23
SCRIPT SQL	24

Integrantes

MODULOS

1. ACESSO
2. ANÚNCIOS
3. COMPRAR
4. FEEDBACK

LINKS

TRELLO: <https://trello.com/b/fWcf3EPE/tcc>

GITHUB: <https://github.com/RebeccaSantos/TCC/>

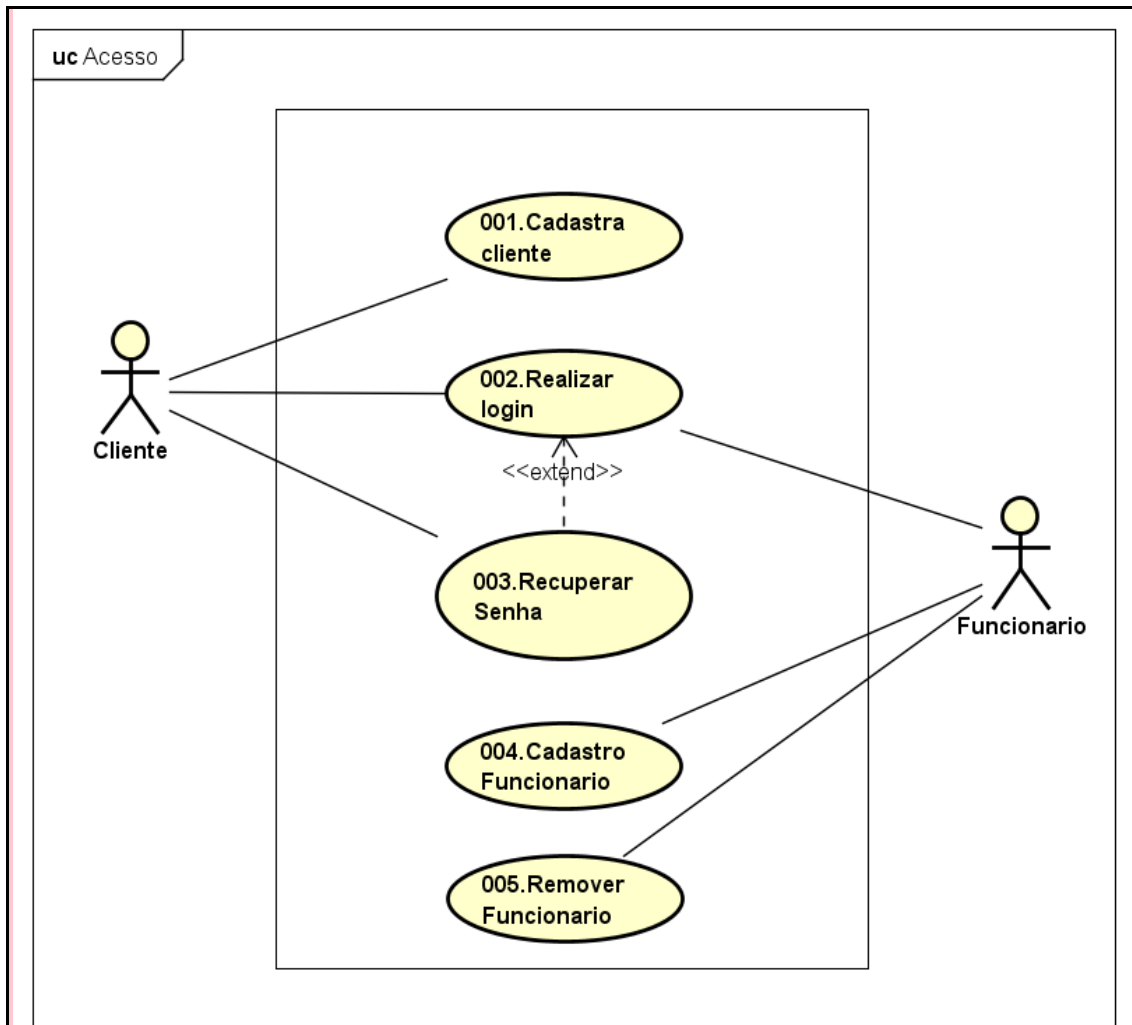
RELATÓRIO:

<https://github.com/RebeccaSantos/TCC/blob/master/analise/Relatorio%20de%20Sistema.pdf>

FIGMA: <https://www.figma.com/file/LEPJhYa0xzfAVbS0lkzy7m/Untitled?node-id=0%3A1>

CASO DE USO

UC - ACESSO



001. Cadastrar Cliente - 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.

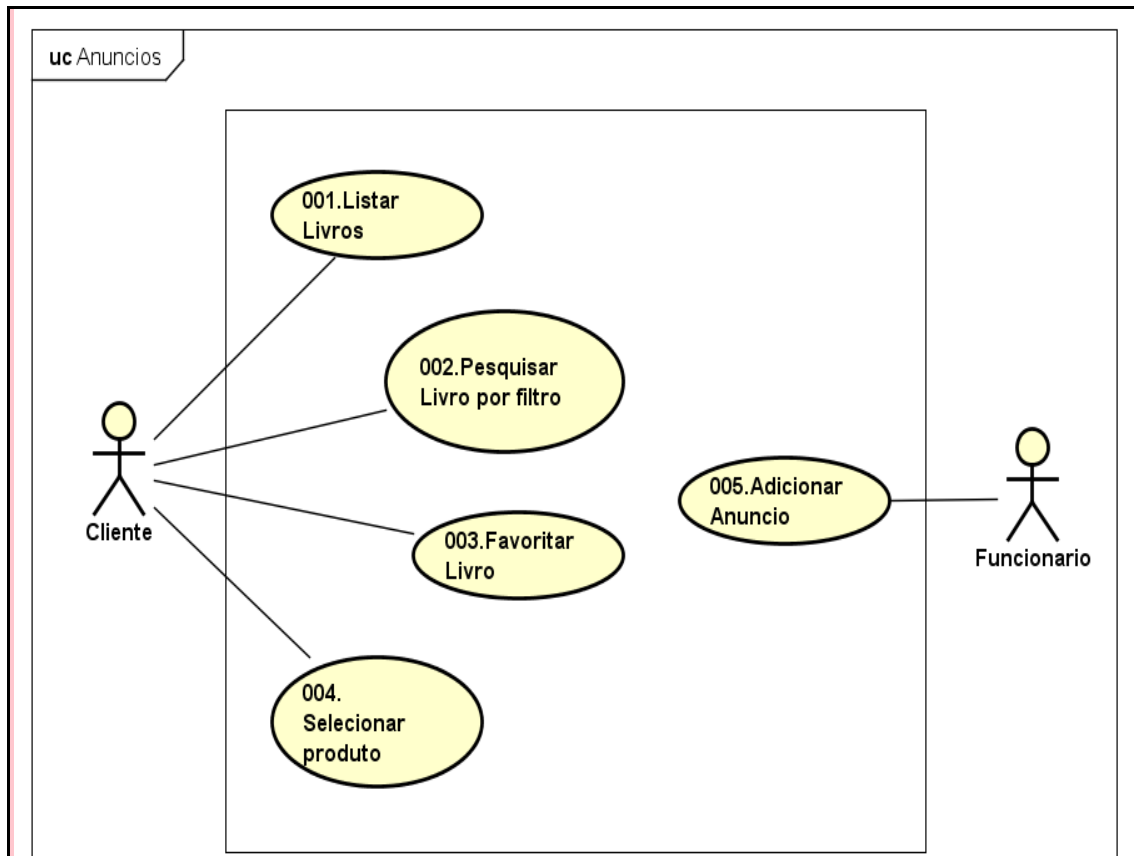
002. Logar -Já devidamente cadastrado no sistema deve ser possível realizar login com seu email ou usuário e senha.

003. Alterar 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 codigo que deverá digitar no codigo par pode alterar sua senha, tempo valido do código é de 2 horas.

004. Cadastro de Funcionário – 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.

005. Remover Funcionário - Os funcionários administradores ponderam remover os funcionários cadastrados no sistema.

UC – ANÚNCIOS



001. Listar Livros - 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.

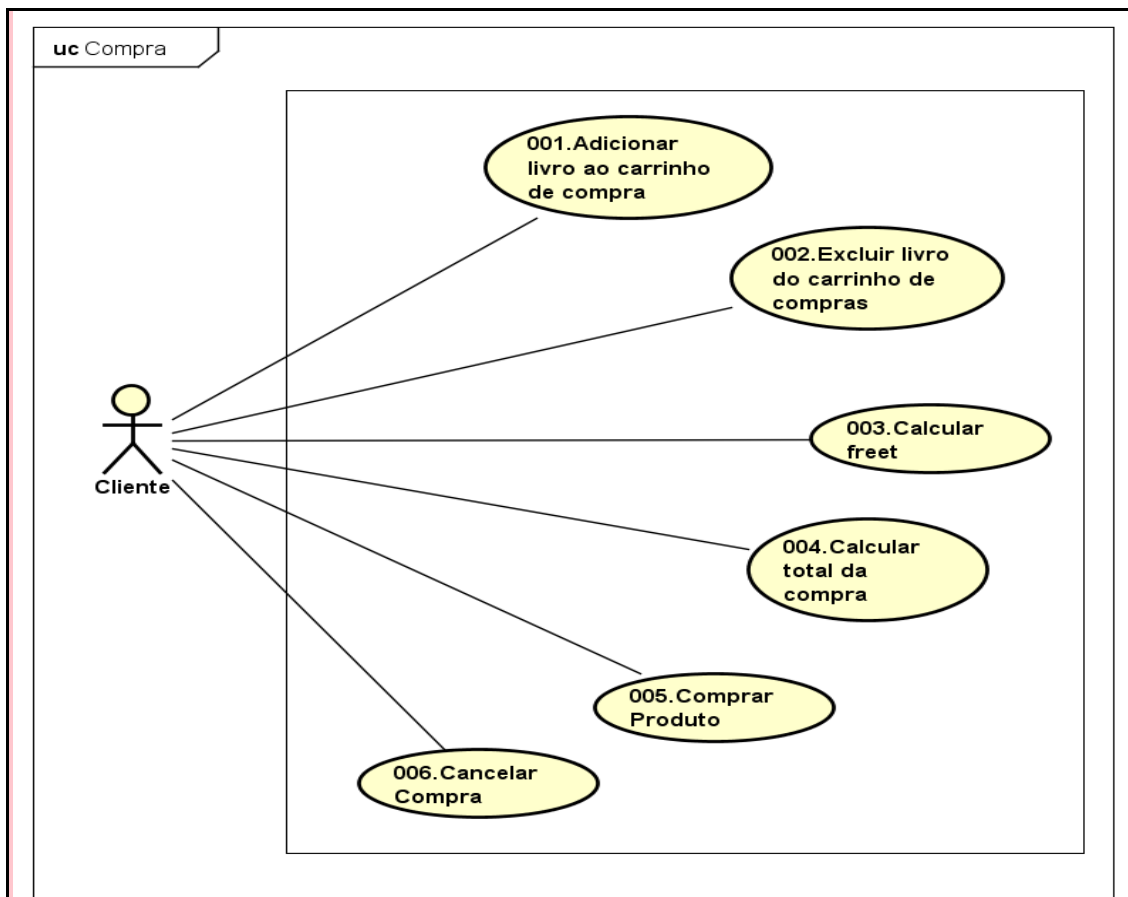
002. Pesquisar Livros por Filtro - Não é necessário ter cadastro de para acessar essa funcionalidade, nela deve ser possível adicionar filtros a pesquisa [por gênero, ano de lançamento, título, autor, editora, edição ...].

003. Favoritar Livro - 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 uma sua lista de favoritos facilitando o acesso a ele.

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

005. Adicionar Livro - 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.

UC – COMPRA



001. Adicionar Livro ao Carrinho de Compra - 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, carrinhos e produtos adicionados como favoritos para trazer um catálogo mais utilizado ao usuário.

002. Excluir Livro ao Carrinho de Compra - 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 ...].

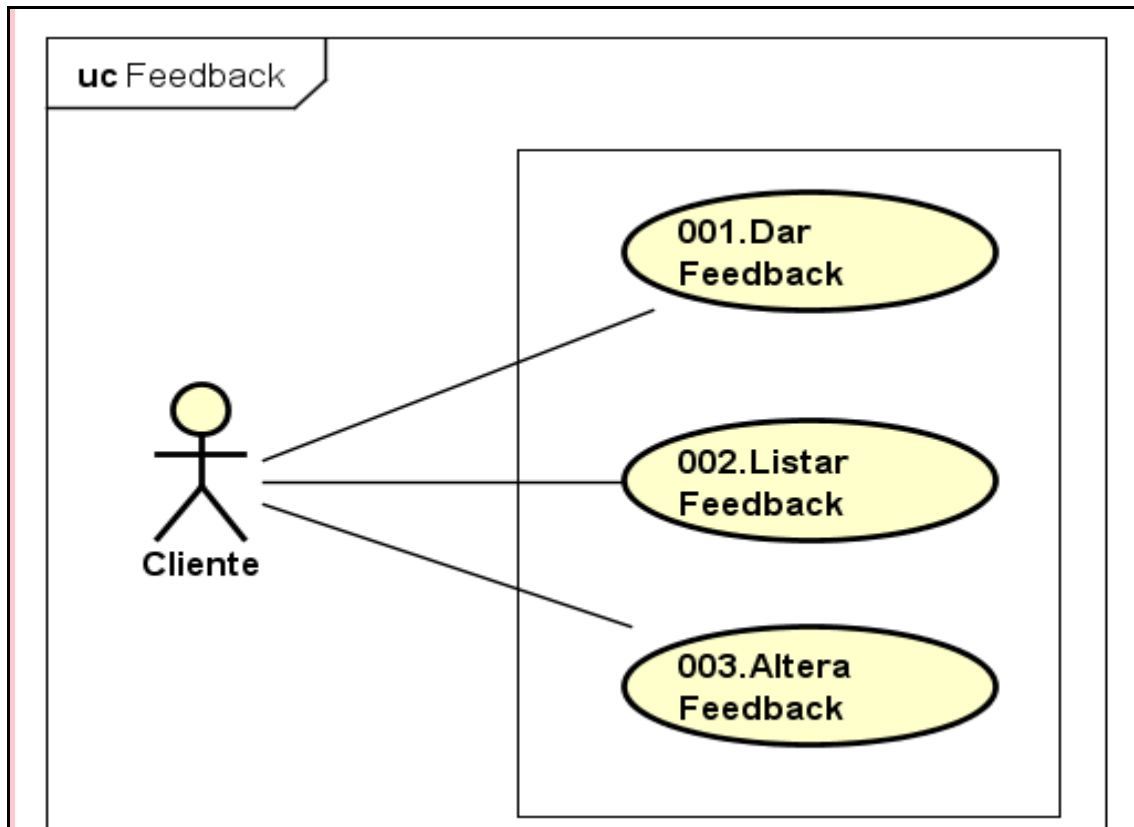
003. Calcular frete - Nesta funcionalidade usuário deve estar devidamente cadastrado como listado no requisito 001. Cadastrar Cliente do módulo de acesso, nela o usuário do sistema poderá adicionar os livros que tem interesse em uma sua lista de favoritos facilitando o acesso a ele.

004. Calcular total da Compra - Não é necessário ter cadastro de para acessar essa funcionalidade, após consultar os livros no catálogo, deve ser possível visualizar todas as informações de um livro ao selecionar, tendo também a opção de adicionar ao carrinho, ler a sinopse do livro, calcular frete.

005. Comprar Livro - 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.

006. Cancelar Compra - Cliente deve poder cancelar uma compra, essa compra não deve ser comprada caso o item ainda não tenha sido enviado caso já tenha o cliente deve aguardar sua chegada e fazer o pedido de reembolso.

UC - FEEDBACK



001. Dar Feedback - Deve ser possível que a cada livro que um cliente compre ele possa avaliar e comentar sua experiencia com essa compra.

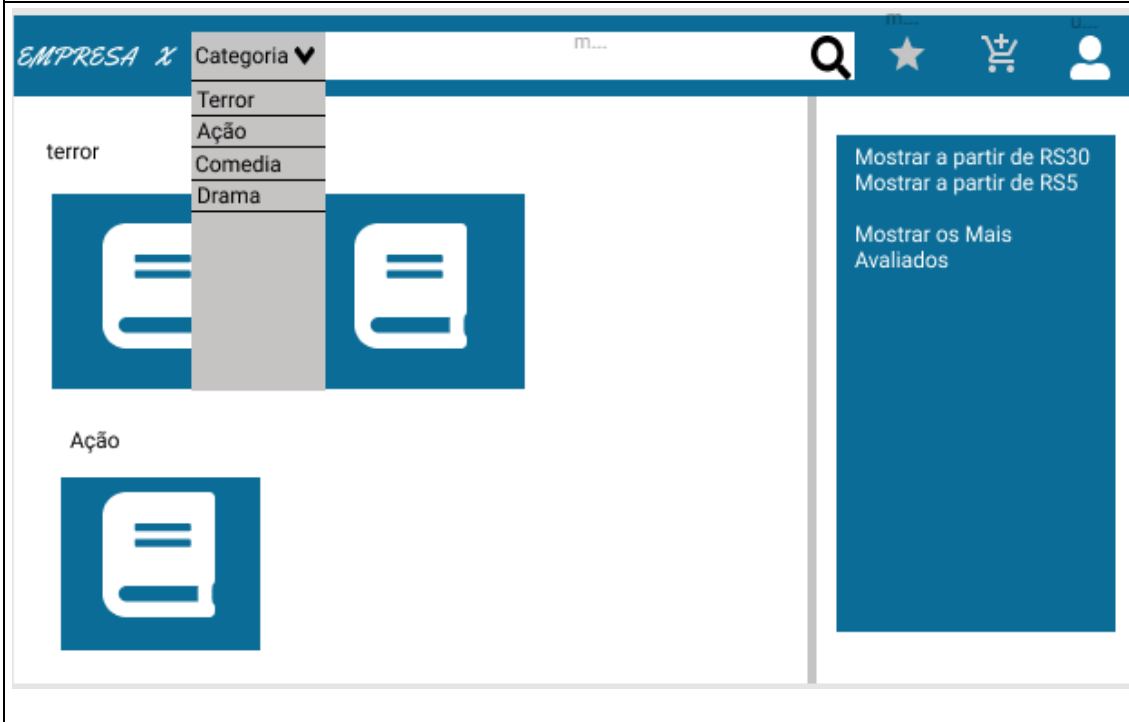
002. Listar Feedback - Clientes devem poder visualizar todos os seus feedbacks outros clientes devem também pode consultar os comentários de outros clientes na aba de consulta os livros para poderem ter conhecimento de outros usuários sobre o livre que desejam adquirir.

003. Alterar Feedback - Deve ser possível alterar a avaliação e os comentários de um feedback.

Níveis	
	Emergentes
	Urgente
	Não Urgente

PROTÓTIPOS

PROT.001 - HOME



EVENTO: 001.CATEGORIA.CLICK

ENTRADA: SELEÇÃO DE CATEGORIA

SAIDA: ATUALIZAÇÃO DAS PAGINA COM ITENS DA CATEGORIA SELECIONADA

EVENTO: 002.PESQUISAR.CLICK

ENTRADA: TEXTO – PESQUISA[NOME, AUTOR, EDITORA, AVALIACAO]

SAIDA: ATUALIZAÇÃO DAS PAGINA COM ITENS DA PESQUISA

EVENTO: 002.FAVARITOS.CLICK

SAIDA: REDIRE3IONA[PROT 006.FAVORITOS]

EVENTO: 004.CARRINHO.CLICK

SAIDA: REDIRECIONA[PROT 007.FAVORITOS]

EVENTO: 005.PERFIL.CLICK

SAIDA: SER ESTIVER LOGADO => REDIRECIONA[PROT – 009.PERFIL], SE NÃO REDIRECIONADO PARA [PROT.002 – LOGIN/CADASTRO]

EVENTO: 006.LIVRO.CLICK

SAIDA: REDIRECIONAMENTO PARA [PROT – 007.INFORMACÕES DE PRODUTO]

EMPRESA X

Cadastro

Nome

Sobrenome

Sexo

 ▼

Proxima Etapa

Login

Email:

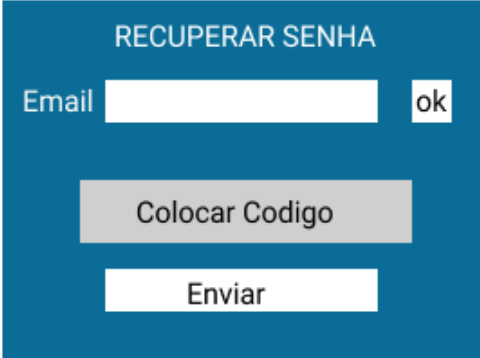
Senha:

LOGAR

[Esqueci a senha](#)

PROT – 003.VALIDAR USUÁRIO

EMPRESA X

A screenshot of a web form titled "RECUPERAR SENHA" (Recover Password). The form is set against a blue background. It contains an "Email" label followed by a white text input field and a small "ok" button. Below this is a grey button labeled "Colocar Codigo" (Place Code). At the bottom is a white button labeled "Enviar" (Send).

EVENTO: 001.OK.CLICK

ENTRADA: EMAIL

SAIDA: ENVIO DE EMAIL COM CÓDIGO DE VERIFICAÇÃO

EVENTO: 002.ENVIAR.CLICK

ENTRADA: CÓDIGO DE VERIFICAÇÃO

SAIDA: REDIRECIONAMENTO PARA [PROT - 004.ALTERAR SENHA]

PROT - 004.ALTERAR SENHA

EMPRESA X

Senha Nova

Repetir Senha

Trocar Senha

EVENTO: 001.TROCAR SENHA.CLICK

ENTRADA: NOVA SENHA, CONFIRMAÇÃO NOVA SENHA

SAÍDA: REDIRECIONAMENTO PARA [PROT – 008.PERFIL]

EMPRESA X

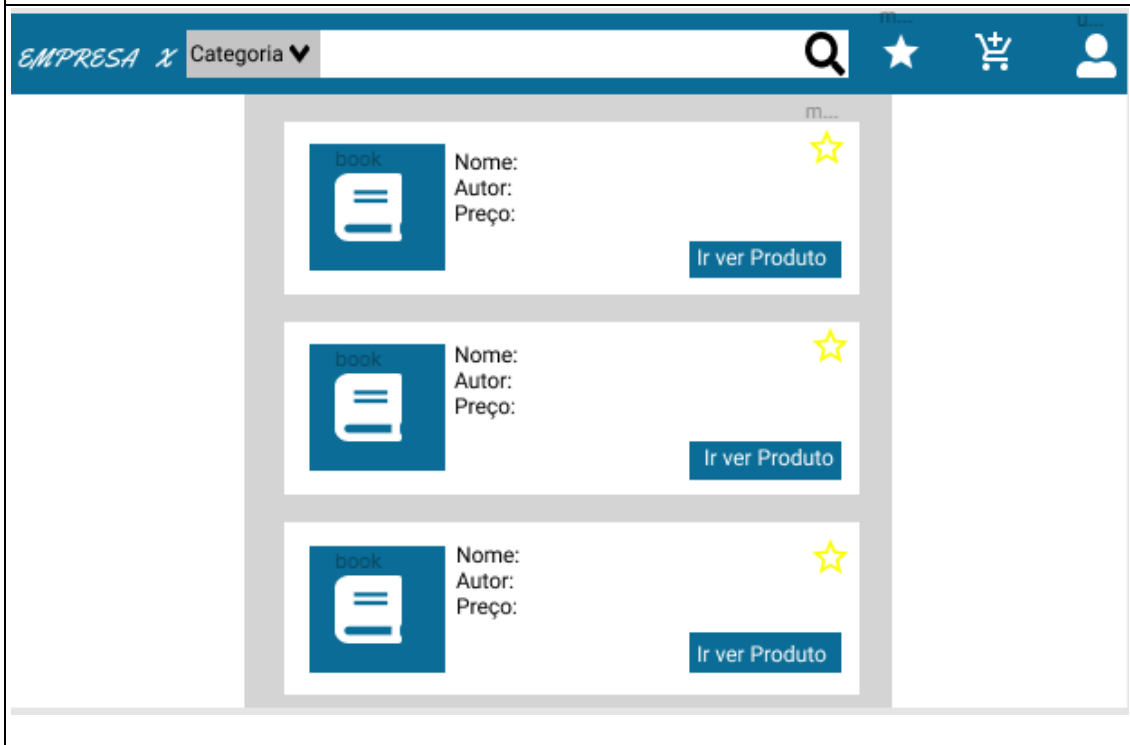
CPF	<input type="text"/>	Nascimento	<input type="text"/>
Estado	<input type="text"/>	Telefone	<input type="text"/>
Cidade	<input type="text"/>	Email	<input type="text"/>
Endereço	<input type="text"/>	Repetir Email	<input type="text"/>
Complemento	<input type="text"/>	Senha	<input type="text"/>
		Repetir Senha	<input type="text"/>
<input type="button" value="Pular"/>		<input type="button" value="Confirmar"/>	

EVENTO: 001.COMFIRMAR.CLICK

ENTRADA: NOME, CPF, DATA DE NASCIMENTO, EMAIL, SENHA, REPITIR SENHA, TELEFONE/CELULAR, ESTADO, CIDADE, ENDEREÇO, CEP, NUMERO RESIDENCIAL, COMPLEMENTO.

SAIDA: REDIRECIONAMENTO PARA PAGINA DE INSERÇÃO DE PRODUTO

PROT 006.FAVORITOS



EVENTO: 001.VER PRODUTO.CLICK

SAÍDA: REDIRECIONAMENTO PARA [PROT – 007.INFORMACÕES DE PRODUTO]

EVENTO: 002.FAVORITO(ESTRELA).CLICK

SAÍDA: REMOVE ITEM DOS FAVORITOS E RECARREGA A PAGINA

PROT – 007.INFORMACÕES DE PRODUTO

EMPRESA X

Categoria ▼


Q

★

🛒

👤

book



Resenha

Valor:

Nome

Generos

Data de Publicação

Quantidade Disponível

Autor(a)

Numero de Paginas

Editora

Adicionar no Carrinho

EVENTO: 001.ADICIONAR NO CARRINHO.CLICK

SAÍDA: REDIRECIONAMENTO PARA [PROT – 009.CARRINHO]

EMPRESA X

Categoria ▼

Q

★

🛒

👤

user-alt





DESCRIÇÃO

NOME
IDADE
SEXO

ULTIMAS COMPRAS

PROT – 009.CARRINHO

EMPRESA X Categoria ▼ 🔍 ★ 🛒 👤

	Nome: Autor: Preço: Quantidade:	Excluir Item
	Nome: Autor: Preço: Quantidade:	Excluir Item

Calcular o valor do Frete, digite seu CEP [Calcular](#)

Quantidade de Produtos:
Frete:
Valor Dos Produtos:
Valor Total:

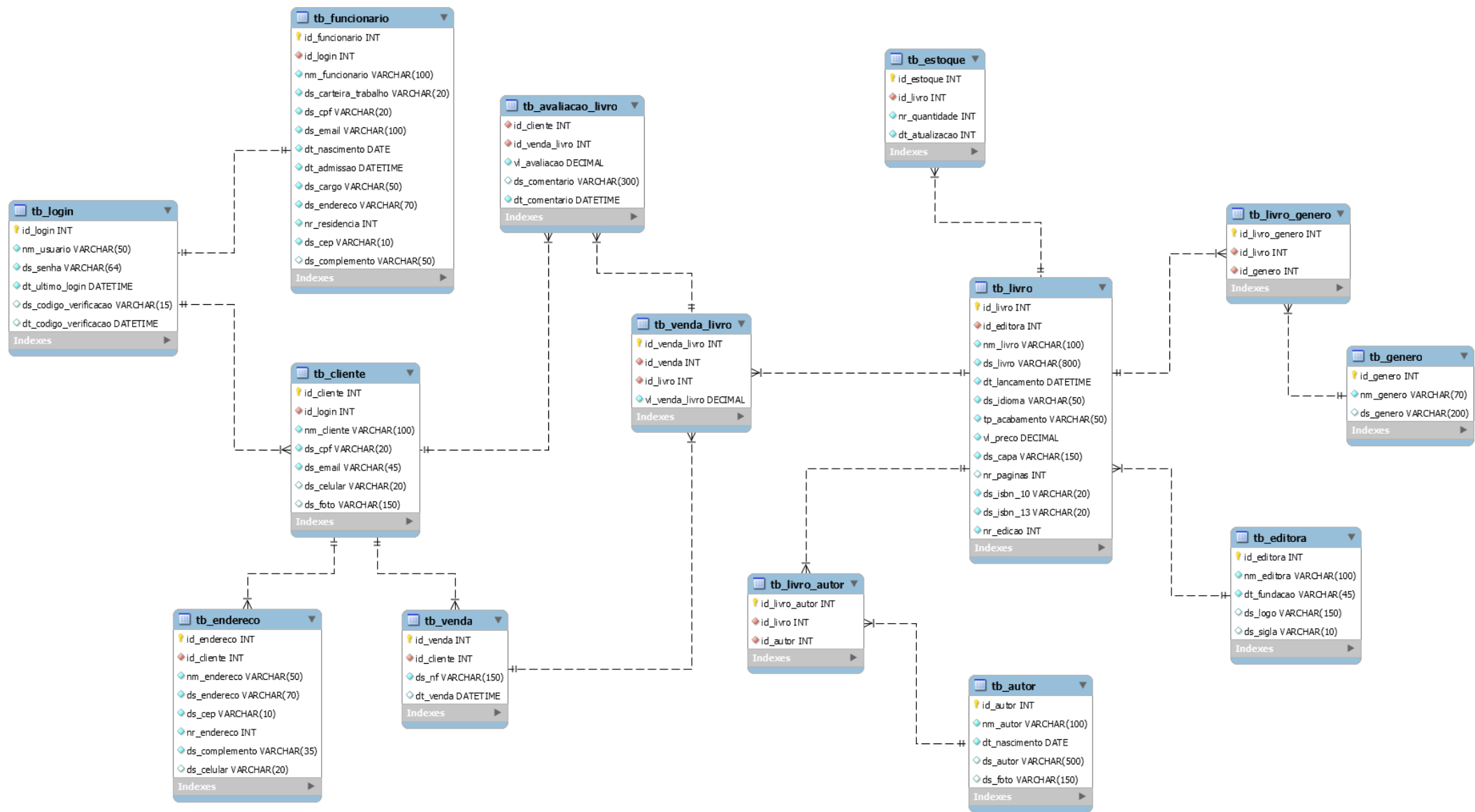
[Finalizar Compra](#)

EVENTO: 001.FINALIZAR COMPRA.CLICK

SAÍDA: REDIRECIONAMENTO PARA [\[PROT.001 - HOME\]](#)

DER

MODELAGEM DO BANCO DE DADOS



SCRIPT SQL

[LINK DO SCRIPT NO GIT HUB](#)

-- 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
```

-- 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(70) NOT NULL,  
  `nr_residencia` INT NOT NULL,  
  `ds_cep` VARCHAR(10) NOT NULL,  
  `ds_complemento` VARCHAR(50) 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,
        PRIMARY KEY (`id_cliente`),
        INDEX `id_login_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_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,
    `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,
  `id_autor` INT NOT NULL,
  `ds_genero` VARCHAR(60) NOT NULL,
  `dt_lancamento` DATETIME NOT NULL,
  `ds_idioma` VARCHAR(50) NOT NULL,
  `tp_acabamento` VARCHAR(50) NOT NULL,
  `vl_preco` DECIMAL NOT NULL,
  `nr_paginas` INT NULL,
  `ds_isbn_10` VARCHAR(20) NULL,
  `ds_isbn_13` VARCHAR(20) NULL,
  `nr_edicao` VARCHAR(45) NULL,
  `tb_livrocol` VARCHAR(45) NULL,
  PRIMARY KEY (`id_livro`),
  INDEX `id_editora_idx` (`id_editora` 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,  
  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,
  `ds_nf` VARCHAR(150) NOT NULL,
  `dt_venda` DATETIME NULL,
  PRIMARY KEY (`id_venda`),
  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_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,
  `vl_venda_livro` DECIMAL 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_avaliacao_livro`

```
CREATE TABLE IF NOT EXISTS `db_next_gen_books`.`tb_avaliacao_livro` (  
  `id_cliente` INT NOT NULL,  
  `id_venda_livro` INT NOT NULL,  
  `vl_avaliacao` DECIMAL 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,  
  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;
```