

## **Diagrama de Entidade-Relacionamento (DER) – Referente ao estudo de caso – Projeto de Extensão.**

O Diagrama de Entidade-Relacionamento tem por objetivo estabelecer uma representação e descrição das entidades, relacionamentos e atributos de um sistema de informações. O DER é uma ferramenta importante para a modelagem de dados, pois permite visualizar a estrutura e os relacionamentos entre as entidades de um sistema de informação, sendo possível criar um modelo conceitual e lógico que represente com precisão as necessidades do negócio, garantindo que o banco de dados seja eficiente e consistente. Além disso, o DER é uma ferramenta útil para a comunicação entre diferentes membros de uma equipe de desenvolvimento, permitindo que todos entendam e trabalhem com a mesma visão do sistema.

Um DER é composto por entidades, que são objetos ou conceitos do mundo real que precisam ser armazenados no banco de dados, e relacionamentos, que são conexões entre as entidades. Cada entidade é representada por um retângulo e os relacionamentos são representados por linhas que conectam as entidades.

No presente trabalho, o DER será utilizado para a construção da modelagem lógica do Banco de Dados do Projeto de Extensão. Para tal, as entidades são, nesse caso, as tabelas que compõem o Banco de Dados, sendo estes: Os produtos; Os usuários; Os supermercados; e os preços dos produtos. Os relacionamentos e suas cardinalidades são representadas pelas linhas que unem as entidades (Figura 1).

Desta forma, para a construção eficiente do BD, os nomes foram transformados em títulos das tabelas, sendo: *produtos*; *usuario*; *mercado*; e *preco\_prod*.

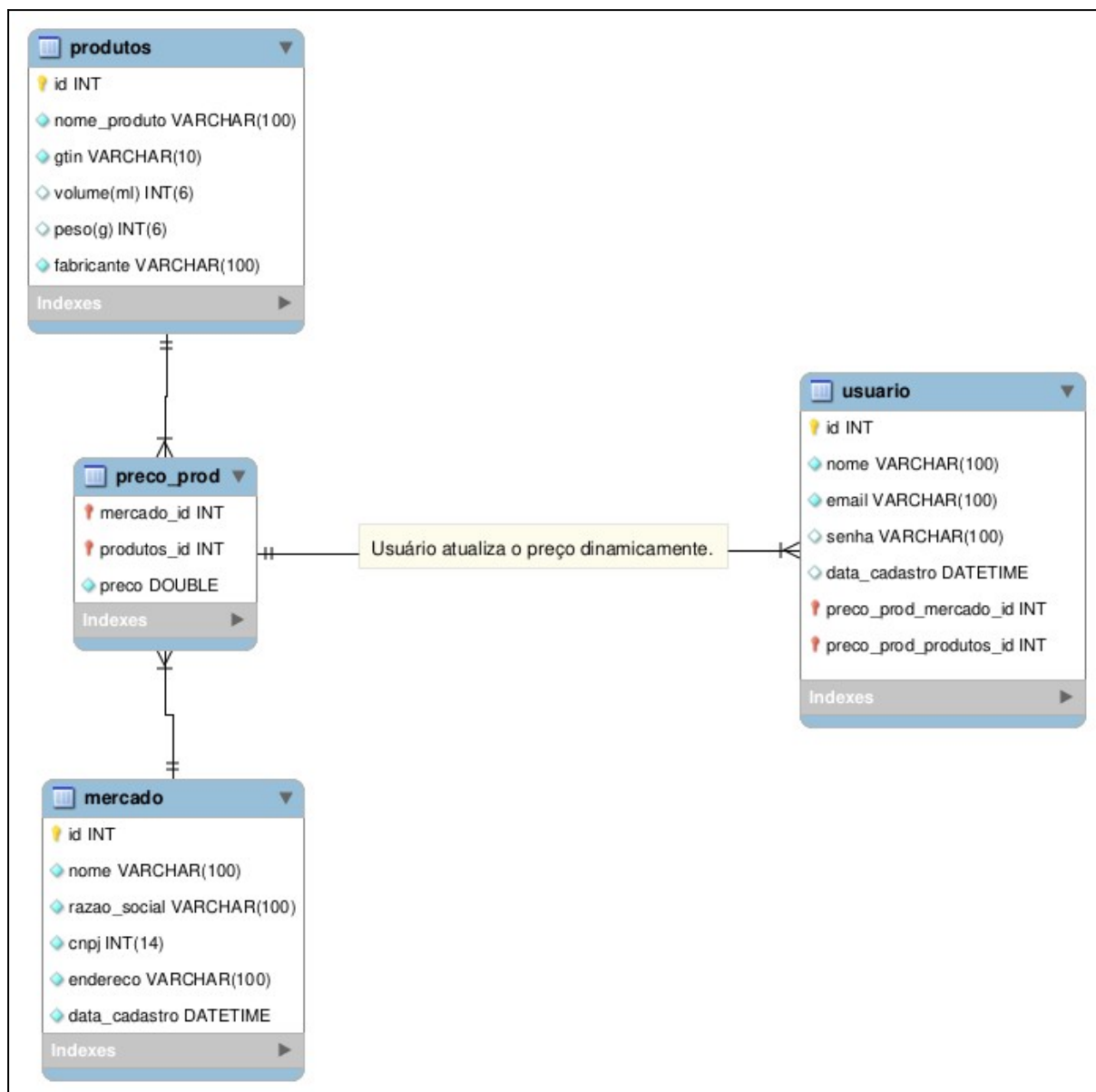


Figura 1: Diagrama de Entidade-Relacionamento – Projeto de Extensão.

Ademais, para validação da estrutura dos dados representados no DER e consolidação do Banco de Dados, é importante que se delimite um script baseado no Diagrama, de forma a construir o BD de fato. Para tal, utilizou-se a ferramenta *MySQL Workbench* através do módulo de construção de um DER (Figura 1) e geração do *script* por meio do *Forward Engineer*. O *script* gerado segue abaixo:

```
-- MySQL Workbench Forward Engineering
```

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;  
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;  
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

```
-----  
-- Schema mydb  
-----
```

```
-----  
-- Schema mydb  
-----
```

```
CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;  
USE `mydb` ;
```

```
-----  
-- Table `mydb`.`produtos`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`produtos` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome_produto` VARCHAR(100) NOT NULL,  
  `gtin` VARCHAR(10) NOT NULL,  
  `volume(ml)` INT(6) NULL,  
  `peso(g)` INT(6) NULL,  
  `fabricante` VARCHAR(100) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `gtin_UNIQUE` (`gtin` ASC) VISIBLE)  
ENGINE = InnoDB;
```

```
-----  
-- Table `mydb`.`mercado`  
-----
```

```
CREATE TABLE IF NOT EXISTS `mydb`.`mercado` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `razao_social` VARCHAR(100) NOT NULL,  
  `cnpj` INT(14) NOT NULL,  
  `endereco` VARCHAR(100) NOT NULL,  
  `data_cadastro` DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`),  
  UNIQUE INDEX `id_UNIQUE` (`id` ASC) VISIBLE,  
  UNIQUE INDEX `cnpj_UNIQUE` (`cnpj` ASC) VISIBLE)  
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`preco\_prod`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`preco_prod` (  
  `mercado_id` INT NOT NULL,  
  `produtos_id` INT NOT NULL,  
  `preco` DOUBLE NOT NULL,  
  PRIMARY KEY (`mercado_id`, `produtos_id`),  
  INDEX `fk_mercado_has_produtos_produtos2_idx` (`produtos_id` ASC) VISIBLE,  
  INDEX `fk_mercado_has_produtos_mercado1_idx` (`mercado_id` ASC) VISIBLE,  
  CONSTRAINT `fk_mercado_has_produtos_mercado1`  
    FOREIGN KEY (`mercado_id`)  
    REFERENCES `mydb`.`mercado` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_mercado_has_produtos_produtos2`  
    FOREIGN KEY (`produtos_id`)  
    REFERENCES `mydb`.`produtos` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-----  
-- Table `mydb`.`usuario`  
-----

```
CREATE TABLE IF NOT EXISTS `mydb`.`usuario` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `nome` VARCHAR(100) NOT NULL,  
  `email` VARCHAR(100) NOT NULL,  
  `senha` VARCHAR(100) NULL,  
  `data_cadastro` DATETIME NULL DEFAULT CURRENT_TIMESTAMP,  
  `preco_prod_mercado_id` INT NOT NULL,  
  `preco_prod_produtos_id` INT NOT NULL,  
  PRIMARY KEY (`id`, `preco_prod_mercado_id`, `preco_prod_produtos_id`),  
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,  
  INDEX `fk_usuario_preco_prod1_idx` (`preco_prod_mercado_id` ASC, `preco_prod_produtos_id`  
ASC) VISIBLE,  
  CONSTRAINT `fk_usuario_preco_prod1`  
    FOREIGN KEY (`preco_prod_mercado_id`, `preco_prod_produtos_id`)  
    REFERENCES `mydb`.`preco_prod` (`mercado_id`, `produtos_id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```