Estoque customizado para produção de uma fábrica artesanal **Tribos Kombucha**

Bianca Viana **CP300547X**
Carolina Zanotto **CP3003817**
Deive Leal **CP3003825**

Tribos Kombucha

Kombucha é uma bebida fermentada, viva, refrescante, levemente ácida e rica em microrganismos.

O processo de produção é baseado na fermentação de uma colônia de bactérias e leveduras conhecida como “scoby”, alimenta-se a cultura com um chá adoçado (geralmente verde ou preto) que consome grande parte do açúcar.

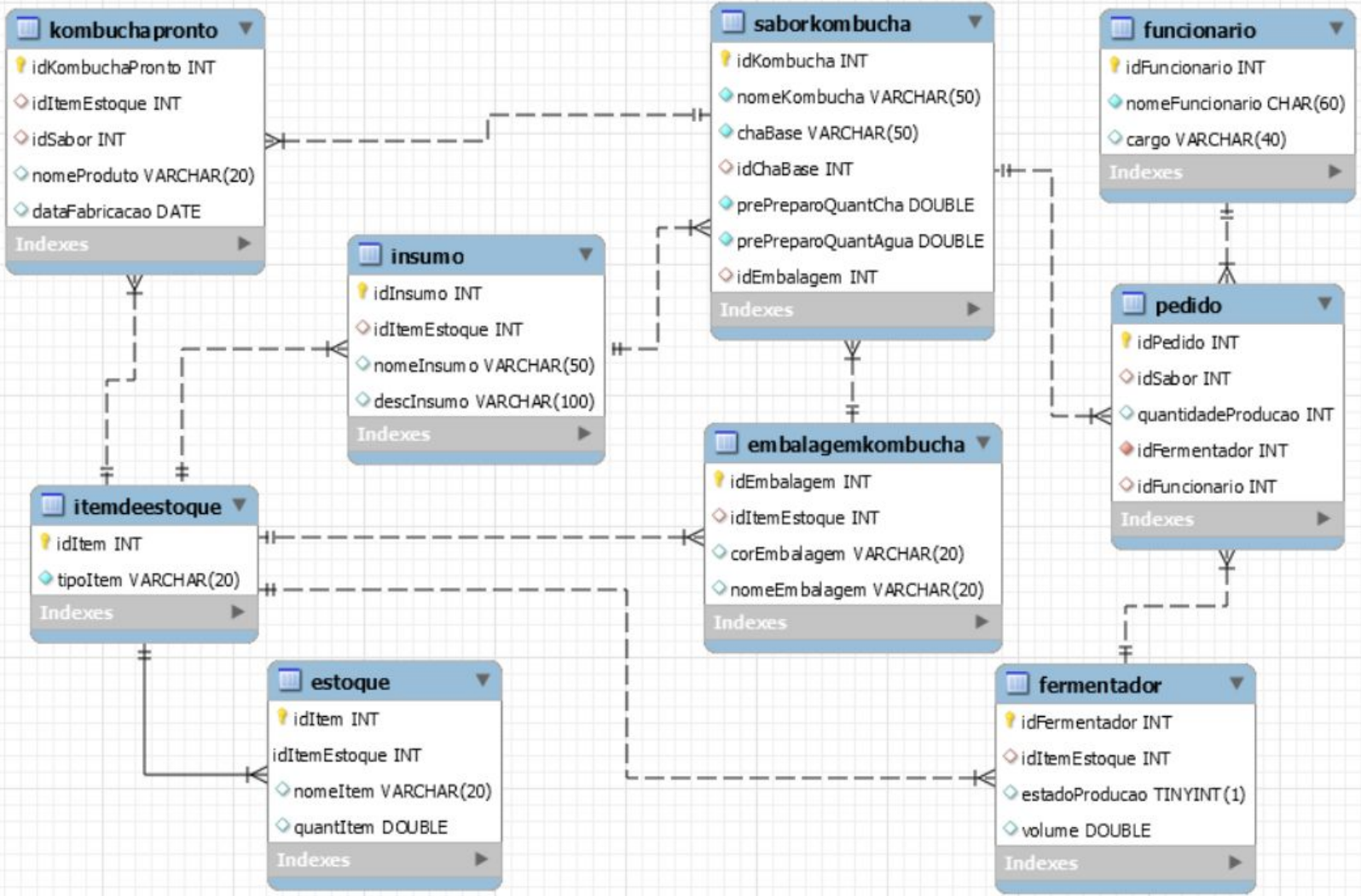
O segundo passo da produção envolve a adição de extratos naturais à base de frutas, vegetais e especiarias, que irão adicionar sabor e aroma, e promover a carbonatação da bebida, deixando-a frisante.

Tribos Kombucha é gaseificada naturalmente. E o álcool está presente em concentração abaixo de 0,5%, o que não caracteriza a bebida como alcoólica.

Sumário

// DER ou MER
// Scripts de Criação
// Procedures or functions
// Views
// Cursor
// Controle de Acesso
// Funções de Agregação
// Triggers

DER


kombuchapronto
idKombuchaPronto INT
idItemEstoque INT
idSabor INT
nomeProduto VARCHAR(20)
dataFabricacao DATE
Indexes
saborkombucha
idKombucha INT
nomeKombucha VARCHAR(50)
chaBase VARCHAR(50)
idChaBase INT
prePreparoQuantCha DOUBLE
prePreparoQuantAgua DOUBLE
idEmbalagem INT
Indexes
funcionario
idFuncionario INT
nomeFuncionario CHAR(60)
cargo VARCHAR(40)
Indexes
insumo
idInsumo INT
idItemEstoque INT
nomeInsumo VARCHAR(50)
descInsumo VARCHAR(100)
Indexes
pedido
idPedido INT
idSabor INT
quantidadeProducao INT
idFermentador INT
idFuncionario INT
Indexes
embalagemkombucha
idEmbalagem INT
idItemEstoque INT
corEmbalagem VARCHAR(20)
nomeEmbalagem VARCHAR(20)
Indexes
itemdeestoque
idItem INT
tipoItem VARCHAR(20)
Indexes
estoque
idItem INT
idItemEstoque INT
nomeItem VARCHAR(20)
quantItem DOUBLE
Indexes
fermentador
idFermentador INT
idItemEstoque INT
estadoProducao TINYINT(1)
volume DOUBLE
Indexes

Scripts de criação das tabelas de **Item Estoque** e **Estoque**

```
CREATE TABLE ItemDeEstoque (
    idItem INT PRIMARY KEY,
    tipoItem VARCHAR(20) NOT NULL
);
```

```
CREATE TABLE Estoque(
    idItem INT,
    idItemEstoque INT,
    nomeItem VARCHAR(20),
    quantItem DOUBLE,

FOREIGN KEY (idItemEstoque)
REFERENCES    ItemDeEstoque(idItem),
    PRIMARY KEY(idItem , idItemEstoque)
);
```

Scripts de criação da tabela
de **Sabor Base**

```
CREATE TABLE SaborKombucha(

        idKombucha  INT PRIMARY KEY,
        nomeKombucha VARCHAR(50) NOT NULL,
        chaBase VARCHAR(50) NOT NULL,
        idChaBase INT,
        prePreparoQuantCha DOUBLE NOT NULL,
        prePreparoQuantAgua DOUBLE NOT NULL,
        idEmbalagem INT,

        FOREIGN KEY (idEmbalagem) REFERENCES EmbalagemKombucha
(idEmbalagem),
        FOREIGN KEY (idChaBase) REFERENCES Insumo(idInsumo)
);
```

Scripts de criação da tabela de **Pedido**

```
CREATE TABLE Pedido(

    idPedido INT PRIMARY KEY,
    idSabor INT,
    quantidadeProducao INT,
    idFermentador INT NOT NULL,
    idFuncionario INT,

    FOREIGN KEY (idSabor) REFERENCES SaborKombucha(idKombucha),
    FOREIGN KEY (idFermentador) REFERENCES Fermentador(idFermentador),
    FOREIGN KEY (idFuncionario) REFERENCES Funcionario(idFuncionario)
);
```

Scripts de criação de procedure

```
DROP PROCEDURE IF EXISTS montaPedidoPreparo;
DELIMITER $$
   CREATE PROCEDURE montaPedidoPreparo (
    IN idKombucha INT,
    INOUT quantidadeDeProducao DOUBLE)
      BEGIN
      DECLARE idPedido_aux INT;
      DECLARE prePreparoQuantCha_aux DOUBLE;
      DECLARE quantEmEstoqueCha DOUBLE;
      DECLARE prePreparoQuantAgua_aux DOUBLE;
      DECLARE dataEntradaPedido DATETIME;

       SELECT NOW() INTO dataEntradaPedido;

      SET prePreparoQuantCha_aux = montagemCha(idKombucha, quantidadeDeProducao);
      -- continua
```

Scripts de criação de procedure

```
        SET prePreparoQuantAgua_aux = montagemAgua(idKombucha, quantidadeDeProducao);
        select dataEntradaPedido, prePreparoQuantAgua_aux;

        SELECT idPedido INTO idPedido_aux FROM tribos_kombucha.pedido
                ORDER BY idPedido DESC LIMIT 1;

        SET idPedido_aux = idPedido_aux + 1;

        INSERT INTO pedido( idPedido, idSabor, quantidadeProducao, idFermentador,
idFuncionario, quantidadeCha,quantidadeAgua, dataEntradaPedido)
                VALUES( idPedido_aux, idSabor,  quantidadeDeProducao, 1, 1,
prePreparoQuantCha_aux, prePreparoQuantAgua_aux, dataEntradaPedido);
    END;
$$
DELIMITER ;
```

Scripts de criação de function

```
DROP FUNCTION IF EXISTS contapedidos;
DELIMITER $$
CREATE FUNCTION contapedidos (numPedido int)
RETURNS INT DETERMINISTIC
BEGIN
    DECLARE contador INT;
        SELECT COUNT(*)
        INTO contador
        FROM tribos_kombucha.Pedido tkp
        WHERE tkp.idPedido = numPedido;
        RETURN contador;
END$$
```

Scripts de criação de views

```
CREATE VIEW ingredientes_saboreskombucha AS
    SELECT * FROM saborkombucha;
```

```
CREATE VIEW qtd_base_sabor AS
        SELECT   chaBase,
                 prePreparoQuantCha,
                 prePreparoQuantAgua
        FROM saborkombucha;
```

Scripts de criação de cursor

```
DELIMITER $$
CREATE PROCEDURE
CURUltimosPedidos()
BEGIN
    DECLARE idPedido INT;
    DECLARE idSabor INT;
    DECLARE quantidadeProducao INT;
    DECLARE idFuncionario INT;

    DECLARE curs_UltimosPedidos
        CURSOR FOR SELECT idPedido,
idSabor, quantidadeProducao,
idFuncionario
    FROM tribos_kombucha.Pedido;
    OPEN curs_UltimosPedidos;

    FETCH NEXT FROM
curs_UltimosPedidos
    INTO idPedido, idSabor,
quantidadeProducao, idFuncionario;

    SELECT idPedido, idSabor,
quantidadeProducao, idFuncionario;

    CLOSE curs_UltimosPedidos;

END $$

CALL CURUltimosPedidos();
```

Scripts de criação de controles de acesso

```
CREATE USER IF NOT EXISTS
'desenvolvedor'@'localhost'
    IDENTIFIED BY 'dev12345'
    PASSWORD EXPIRE INTERVAL 90 DAY;

CREATE USER IF NOT EXISTS
'admdb'@'localhost'
    IDENTIFIED BY 'mandb123'
        PASSWORD EXPIRE INTERVAL 365 DAY;

CREATE USER IF NOT EXISTS
'userdb'@'localhost'
    IDENTIFIED BY 'user1234'
        PASSWORD EXPIRE INTERVAL 365 DAY;
```

```
GRANT ALTER, DELETE, INSERT,
SELECT, UPDATE
    ON tribos_kombucha.*
        TO 'desenvolvedor'@'localhost'
WITH GRANT OPTION;

GRANT ALL
    ON tribos_kombucha.*
        TO 'admdb'@'localhost' WITH
GRANT OPTION;

GRANT DELETE, INSERT, SELECT,
UPDATE
    ON tribos_kombucha.*
        TO 'userdb'@'localhost';

            FLUSH PRIVILEGES;
```

Scripts de criação de funções de agregação

```
DROP PROCEDURE IF EXISTS NumeroPedidoPorData;
DELIMITER $$
CREATE PROCEDURE NumeroPedidoPorData()
BEGIN
    SELECT count(idPedido) FROM Pedido
        WHERE date(dataEntradaPedido) LIKE '%2020-09-14%';
END
$$
DELIMITER ;
```

Scripts de criação de funções de triggers

```
DELIMITER $$
CREATE TRIGGER darBaixaEstoque
AFTER INSERT ON pedido
FOR EACH ROW
        BEGIN
                UPDATE estoque
       SET quantItem = quantItem - NEW.quantidadeProducao
        WHERE idItem = NEW.idPedido;
        END $$
```

```
DROP TRIGGER IF EXISTS trg_contapedidos_insert;
DELIMITER $$
CREATE TRIGGER trg_contapedidos_insert
AFTER INSERT ON tribos_kombucha.Pedido
FOR EACH ROW
   BEGIN
   CALL contapedidos(Pedido.idPedido);
   END $$
```