**WYDEN FACULDADE ÁREA 1 / CENTRO UNIVERSITÁRIO UNIRUY**

Andreza da Silva Lima - 202304509506

Caio Neves da Silva - 202308305956

Gustavo Cerqueira Bonfim Oliveira – 202303392877

Leonardo Alves das Mercês Calisto - 202309882141

Luís Felipe Tavares Alencar dos Reis – 202302417922

Thiago Santos Soares -202302923348

**PROJETO DE BANCO DE DADOS - FLORICULTURA**

SALVADOR – BA 2023

Andreza da Silva Lima - 202304509506

Caio Neves da Silva - 202308305956

Gustavo Cerqueira Bonfim Oliveira – 202303392877

Leonardo Alves das Mercês Calisto - 202309882141

Luís Felipe Tavares Alencar dos Reis – 202302417922

Thiago Santos Soares -202302923348

**PROJETO DE BANCO DE DADOS - FLORICULTURA**

Projeto apresentado como requisito parcial para obtenção de aprovação na disciplina de Banco de Dados na instituição Centro Universitário UniRuy.

Professora: Heleno Cardoso da Silva Filho

SALVADOR-BA 2023

Sumário

1. [INTRODUÇÃO 5](#_bookmark0)
2. [MODELO CONCEITUAL 6](#_bookmark1)
   1. [Modelo Conceitual Textual 6](#_bookmark2)
   2. [Diagrama Entidade-Relacionamento: 7](#_bookmark3)

[. 9](#_bookmark9)

1. [MODELO LÓGICO 1](#_bookmark13)0
   1. [Tabela e suas referências 11](#_bookmark15)
2. [MODELO FÍSICO 13](#_bookmark16)
3. [CONCLUSÃO 16](#_bookmark17)

Lista de Figuras

[Figura 1- Diagrama ER 7](#_bookmark4)

[Figura 2- Relacionamento entre dadofuncionario\_funcionario 7](#_bookmark5)

[Figura 3 - Relacionamento entre produto e pedido 8](#_bookmark6)

[Figura 4 - Relacionamento entre pedido e cliente 8](#_bookmark7)

[Figura 5 - Relacionamento entre produto e fornecedor . 9](#_bookmark8)

[Figura 6 - Relacionamento entre funcionario e pedido 9](#_bookmark10)

[Figura 7 – Diagrama do Modelo Lógico. 10](#_bookmark11)

# INTRODUÇÃO

Esta atividade envolve a criação de um modelo conceitual e um diagrama de Entidade-Relacionamento (E-R) para um banco de dados. A representação gráfica das entidades e seus relacionamentos é o modelo conceitual, enquanto o diagrama E-R mostra as entidades, seus atributos e os relacionamentos entre elas.

Para construir um modelo conceitual, é importante definir primeiro o tipo de sistema que será utilizado. Isso pode incluir a identificação dos usuários, a natureza dos dados a serem armazenados e processados suas necessidades e requisitos. Em seguida, é necessário identificar as entidades do sistema, e seus relacionamentos.

Nos relacionamentos, é importante definir tanto a cardinalidade quanto a totalidade de cada um deles. A cardinalidade se refere ao número de entidades que participam de um relacionamento, enquanto a totalidade indica se todas as entidades em um relacionamento precisam estar presentes ou não.

No caso específico do banco de dados escolhido pela equipe, existem 6 entidades principais: Fornecedor, funcionário, pedido, cliente, dado\_funcionario, produto. Cada uma dessas entidades possui atributos únicos e relacionamentos específicos com as outras entidades.

Ao todo, foram identificados 5 relacionamentos entre as entidades. É importante lembrar que a identificação desses relacionamentos foi feita com base nos requisitos do sistema e nos dados que serão armazenados e processados.

# MODELO CONCEITUAL

## Modelo Conceitual Textual:

Fornecedor: IDFornecedor, Nome, Endereço, Número, Bairro, Cep, Contato, CNPJ.

Funcionário: IDFuncionário, Nome, Sobrenome, CPF, Data De Nascimento.

Pedido: IDPedido, Data, Status, Valor.

Cliente: IDCliente, Nome, Sobrenome, Cpf, Data De Nascimento, E-Mail, Contato.

Dado\_Funcionario: IDDadoFuncionario, Logradouro, Bairro, Estado, CEP, Contato, E-Mail.

Produto: IDProduto, Descrição, Peso, Quantidade, Preço.

Item: IDProduto, IDPedido, Quantidade.

Fornecimento: IDProduto, IDFornecedor.

## Diagrama Entidade-Relacionamento:

**Diagrama

Descrição gerada automaticamente**

*Figura 1- Diagrama ER*

*Fonte: De autoria própria.*

Diagrama

Descrição gerada automaticamente

*Figura 2- Relacionamento entre dado\_funcionario e funcionário*

*Fonte: De autoria própria.*

**Dado\_funcionário - Funcionário** = O dado\_funcionário e a tabela funcionário tem uma relação igual, pois ambos fazem solicitações simultaneas um ao outro.

Diagrama

Descrição gerada automaticamente

*Figura 3 - Relacionamento entre Produto e Pedido*

**Produto - Pedido** = A tabela produto aciona o pedido através do item, que contém a id do produto e pedido, dizendo a sua quantidade.

Diagrama

Descrição gerada automaticamente

*Figura 4 - Relacionamento entre Pedido e Cliente*

*Fonte: De autoria própria.*

**Pedido – Cliente** = O Cliente efetua um pedido, mas o pedido não se relaciona com o cliente, então temos a relação de 1:N, nesse caso.

Diagrama

Descrição gerada automaticamente

*.*

*Figura 5 - Relacionamento entre Produto e Fornecedor*

*Fonte: De autoria própria.*

**Produto – Fornecedor** = O produto se relaciona com o fornecedor pela tabela fornecimento, que contém a ID do produto e do forncedor. Com uma relação N:N.

Diagrama

Descrição gerada automaticamente

*Figura 6 - Relacionamento entre Funcionário e Pedido*

**Funcionário – Pedido** = O funcionário emite um pedido para o cliente, mas o pedido não faz esse mecanismo com o funcionário, por isso temos uma relação 1:N.

# MODELO LÓGICO

Diagrama

Descrição gerada automaticamente

*Figura 7 - Diagrama do modelo lógico*

*Fonte: De autoria própria.*

**Modelo lógico** se dá através das tabelas apresentadas a cima na imagem, onde representam a relação entre elas, atributos de cada uma e a importância de trabalharem juntas, formando assim a nossa base de dados. A tabela produtos tem ligação com todas as tabelas fazendo com que seja a movimentação de todos os processos, pois é por ela que a tabela funcionário tem a relação de vendas, a Tabela cliente tem os produtos, a tabela Importadora solicita a reposição e a tabela fornecedores abastece o estoque.

## Tabela e suas referências

**Fornecedor** tem como seus atributos:

IDfornece: Chave primária

Nome: identificar o nome da empresa

Endereço: Armazena endereço da empresa. Contato: Armazena contatos da empresa.

E-mail: Armazena e-mail oficial da empresa.

**Funcionário** tem como seus atributos:

IDFuncionário: Utilizado como identificador único, permitindo pesquisar informações especificas de cada funcionário.

Nome: Onde são registrados nomes de cada funcionário CPF: onde armazena número de cada um funcionário.

Data de Nascimento: Armazena data de nascimento de cada trabalhador, onde a empresa manter um registro preciso dos aniversários e da faixa etária dos funcionários.

**Pedido** tem como atributos:

IDPedido: Códigos dos Pedidos.

Status: Informando atualizações do estado do pedido.

Valor: Valor do pedido.

E-mail: E-mails de cada fornecedor.

Chave estrangeira Cliente: Onde estabelece uma interação entre Pedido e Cliente.

Chave estrangeira Produdo: Onde estabelece uma interação entre Pedido e Produto.

Chave estrangeira Funcionário: Onde estabelece uma interação entre Pedido e Funcionário que fez a venda.

**Cliente** tem como seus atributos: IDCliente: Armazena código do cliente. Nomes: Armazena nomes dos clientes. E-mail: Armazena E-mail dos clientes. Data de nascimento: Armazena data de nascimento dos clientes.

CPF: Armazena CPF dos clientes.

**Produto** tem como seus atributos:

Descrição: Onde existe a descrição do produto.

Peso: Onde informa o peso do produto.

Quantidade: A quantidade do produto.

**Dado\_funcionário** tem como seus atributos:

Logradouro: Onde existe a descrição do produto.

Bairro: Informa o bairro do funcionário.

Estado: Informa o Estado do funcionário.

CEP: Informa o CEP do funcionário.

Contato: Informa o Contato do funcionário.

E-mail: Informa o e-mail do funcionário.

**Itens** tem como seus atributos:

Quantidade: Informa a quantidade dos intens.

# MODELO FÍSICO

**CLIENTE:**

CREATE TABLE IF NOT EXISTS `floricutura`.`cliente` (

`clieid` INT NOT NULL AUTO\_INCREMENT,

`clie\_nome` VARCHAR(20) NOT NULL,

`clie\_sobrenome` VARCHAR(20) NOT NULL,

`clie\_cpf` VARCHAR(11) NOT NULL,

`clie\_dt\_nascimento` DATE NULL,

`clie\_email` VARCHAR(45) NOT NULL,

`clie\_contato` VARCHAR(15) NOT NULL,

PRIMARY KEY (`clieid`),

UNIQUE INDEX `clie\_cpf` (`clie\_cpf` ASC) VISIBLE,

UNIQUE INDEX `clie\_email\_UNIQUE` (`clie\_email` ASC) VISIBLE,

UNIQUE INDEX `clie\_contato\_UNIQUE` (`clie\_contato` ASC) VISIBLE);

**FUNCIONÁRIO:**

CREATE TABLE IF NOT EXISTS `floricutura`.`funcionario` (

`funcioid` INT NOT NULL AUTO\_INCREMENT,

`funcio\_nome` VARCHAR(20) NOT NULL,

`funcio\_sobrenome` VARCHAR(20) NOT NULL,

`funcio\_cpf` VARCHAR(11) NOT NULL,

`funcio\_dt\_nascimento` DATE NULL DEFAULT NULL,

PRIMARY KEY (`funcioid`),

UNIQUE INDEX `funcio\_cpf` (`funcio\_cpf` ASC) VISIBLE);

**DADO\_FUNCIONARIO:**

CREATE TABLE IF NOT EXISTS `floricutura`.`dado\_funcionario` (

`dadofuncio\_logradouro` VARCHAR(20) NOT NULL,

`dadofuncio\_bairro` VARCHAR(20) NOT NULL,

`dadofuncio\_estado` VARCHAR(20) NOT NULL,

`dadofuncio\_cep` VARCHAR(12) NOT NULL,

`dadofuncio\_contato` VARCHAR(15) NOT NULL,

`dadofuncio\_email` VARCHAR(45) NOT NULL,

`funcio\_id` INT NOT NULL,

PRIMARY KEY (`dadofuncioid`),

UNIQUE INDEX `dadofuncio\_email` (`dadofuncio\_email` ASC) VISIBLE,

INDEX `funcio\_id` (`funcio\_id` ASC) VISIBLE,

UNIQUE INDEX `dadofuncio\_contato\_UNIQUE` (`dadofuncio\_contato` ASC) VISIBLE,

CONSTRAINT `dado\_funcionario\_ibfk\_1`

FOREIGN KEY (`funcio\_id`)

REFERENCES `floricutura`.`funcionario` (`funcioid`));

**FORNECEDOR:**

CREATE TABLE IF NOT EXISTS `floricutura`.`fornecedor` (

`forneceid` INT NOT NULL AUTO\_INCREMENT,

`fornece\_nome` VARCHAR(20) NOT NULL,

`fornece\_logradouro` VARCHAR(20) NULL,

`fornece\_numero` SMALLINT NOT NULL,

`fornece\_cep` VARCHAR(12) NOT NULL,

`fornece\_cidade` VARCHAR(25) NULL,

`fornece\_bairro` VARCHAR(20) NULL,

`fornece\_contato` VARCHAR(12) NOT NULL,

`fornece\_cnpj` VARCHAR(15) NOT NULL,

PRIMARY KEY (`forneceid`),

UNIQUE INDEX `fornece\_contato` (`fornece\_contato` ASC) VISIBLE,

UNIQUE INDEX `fornece\_cnpj` (`fornece\_cnpj` ASC) VISIBLE);

**PEDIDO:**

CREATE TABLE IF NOT EXISTS `floricutura`.`pedido` (

`pedidoid` INT NOT NULL AUTO\_INCREMENT,

`pedido\_dt` DATE NOT NULL,

`pedido\_status` VARCHAR(20) NOT NULL,

`pedido\_valor` FLOAT NOT NULL,

`clie\_id` INT NOT NULL,

`funcio\_id` INT NOT NULL,

PRIMARY KEY (`pedidoid`),

INDEX `clie\_id` (`clie\_id` ASC) VISIBLE,

INDEX `funcio\_id` (`funcio\_id` ASC) VISIBLE,

CONSTRAINT `pedido\_ibfk\_1`

FOREIGN KEY (`clie\_id`)

REFERENCES `floricutura`.`cliente` (`clieid`),

CONSTRAINT `pedido\_ibfk\_2`

FOREIGN KEY (`funcio\_id`)

REFERENCES `floricutura`.`funcionario` (`funcioid`));

**PRODUTO:**

CREATE TABLE IF NOT EXISTS `floricutura`.`produto` (

`produtoid` INT NOT NULL AUTO\_INCREMENT,

`produto\_descricao` VARCHAR(20) NOT NULL,

`produto\_peso` VARCHAR(20) NULL,

`produto\_quantidade` VARCHAR(11) NOT NULL,

`produto\_preco` FLOAT NOT NULL,

PRIMARY KEY (`produtoid`));

**Fornecimento:**

CREATE TABLE IF NOT EXISTS `floricutura`.`fornecimento` (

`idproduto` INT NOT NULL,

`idfornecedor` INT NOT NULL,

PRIMARY KEY (`idproduto`, `idfornecedor`),

INDEX `fk\_idfornecedor` (`idfornecedor` ASC) VISIBLE,

CONSTRAINT `fk\_idfornecedor`

FOREIGN KEY (`idfornecedor`)

REFERENCES `floricutura`.`fornecedor` (`forneceid`),

CONSTRAINT `fk\_idproduto`

FOREIGN KEY (`idproduto`)

REFERENCES `floricutura`.`produto` (`produtoid`));

**ITENS:**

CREATE TABLE IF NOT EXISTS `floricutura`.`itens` (

`idprod` INT NOT NULL,

`idped` INT NOT NULL,

`quantidade` INT NOT NULL,

PRIMARY KEY (`idprod`, `idped`),

INDEX `fk\_ped` (`idped` ASC) VISIBLE,

CONSTRAINT `fk\_ped`

FOREIGN KEY (`idped`)

REFERENCES `floricutura`.`pedido` (`pedidoid`),

CONSTRAINT `fk\_prod`

FOREIGN KEY (`idprod`)

REFERENCES `floricutura`.`produto` (`produtoid`));

# CONCLUSÃO

Portanto, o trabalho foi concluído atingindo os objetivos esperados pela equipe. Durante o processo foi possível entender melhor como funciona às tabelas e seus relacionamentos no banco de dados, além de suas necessidades fundamentais para as necessidades de uma floricultura. Dessa forma buscamos continuar adquirindo ainda mais conhecimento para o desenvolvimento profissional.