

## MARIA DB ESSENTIALS

O que é Maria DB?

É um banco de dados relacional: todas as informações são relacionadas em torno de um problema que deve ser resolvido; relacionamento 1...1, relacionamento 1...N e relacionamento de N... M.

Transacional: uma sequência de processos, que quando executadas formam apenas uma ação; ACID

1. Atomicidade: todo processo deve ter um fim,
2. Consistência: todos os processos devem ser executados obedecendo todas as regras e restrições impostas,
3. Isolamento: nenhuma transação pode afetar outra em andamento,
4. Durabilidade: [ou persistência]: toda informação escrita no repositório só pode ser desfeita/refeita por outra transação.

Normalizado: nas 5 formas normais tende a ser extremamente dinâmico, porém com perda de desempenho.

### CARACTERÍSTICA DO MARIA DB

Implementação C e C++,

Multiplataforma,

Open Source,

Aceita várias linguagens de programação,

Comparação mariaDB e MySQL,

Repositório

## ESCOLHA SEU STAGE

### TIPOS DE AMBIENTE

Produção - Serviço do MariaDB no cPanel,

Homologação (sandbox) - Localhost,

Serviço dedicado.

### INSTALAÇÃO NO UBUNTU

Instalação feita através do terminal seguindo o passo a passo do site: [Produtos e Ferramentas MariaDB baixa | MariaDB](#)

Após selecionar o SO, clicar no link MariaDB Package repository

Abra o terminal e utilize as linhas de comando da página do MariaDB.

Após instalação, execute o comando: `mysql -u root -p`

CREATE DATABASE teste;

SHOW DATABASES;

USE teste;

## ESCOLHA SEU APLICATIVO



## PARAMETRIZANDO ESTAÇÃO

TESTANDO CONEXÃO - telnet nome do domínio 3306

Teste conexão: telnet localhost 3306 -> Deve ser habilitado no Windows adicionar ou remover componentes do Windows.

## COMPREENDENDO O PROCESSO

EXECUTANDO IMPORT

EXECUTANDO EXPORT

Marcar Personalizado – estrutura

## TABELAS CAMPOS E ATRIBUTOS

### TIPOS DE DADOS INTEIROS

### TIPO DE DADOS [NÚMEROS REAIS]

### TIPO DE DADOS [TEXTOS]

### TIPOS DE DADOS [DATAS E HORAS]

TIPOS DE DADOS [OTHERS] -> tipos geométricos de dados, autoincremento e null.

## OVERVIEW DDL

Linguagem de definição de dados (Data Definition language)

**TRABALHANDO COM CREATE [DATABASE] -> CRIANDO BANCO DE DADOS!**

```
CREATE DATABASE mod_essentials DEFAULT CHARACTER SET 'utf8' DEFAULT COLLATE = 'utf8_general_ci';
```

```
CREATE DATABASE IF NOT EXISTS mod_essentials DEFAULT CHARACTER SET 'utf8' DEFAULT COLLATE = 'utf8_general_ci';
```

----->>> O "IF NOT EXISTS" verifica se o banco já existe

USE mod\_essentials; --->>> Podemos utilizar o BD

**TRABALHANDO COM CREATE [TABLE]**

```
CREATE TABLE teste(
```

```
    teste_id INT(11) NOT NULL AUTO_INCREMENT UNIQUE PRIMARY KEY COMMENT 'Campo para armazenar o ID',
```

```
    teste_nome VARCHAR(255) NOT NULL,
```

```
    teste_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP()
```

```
)
```

**MODIFICANDO COM ALTER**

```
ALTER DATABASE mod_essentials DEFAULT COLLATE = ' ';
```

```
ALTER TABLE teste MODIFY COLUMN teste_name VARCHAR(100) NOT NULL;
```

```
ALTER TABLE teste ADD COLUMN teste_descricao TEXT AFTER teste_nome;
```

----->> Acrescentou uma coluna após a coluna teste\_nome.

```
ALTER TABLE teste ADD COLUMN teste_title TEXT FIRST teste_nome;
```

----->> Inseriu uma coluna antes da coluna teste\_nome.

```
ALTER TABLE teste DROP COLUMN IF EXISTS teste_titulo;
```

----->> Excluiu a coluna teste\_titulo da tabela.

**DELETANDO COM TRUNCATE**

```
TRUNCATE TABLE teste;
```

----->> Apaga todos os dados da tabela

```
DELETE FROM teste;
```

----->> Apaga todos os dados da tabela

## RENAME DE TABELA

```
RENAME TABLE teste TO teste_geral;
```

```
RENAME TABLE teste_geral TO usuarios, teste2 TO produtos; -->>> Renomeando duas tabelas
```

## REMOVENDO ESTRUTURA COM DROP

```
DROP TABLE usuario; -->> Apaga a tabela do banco de dados
```