

# MySQL e PHP

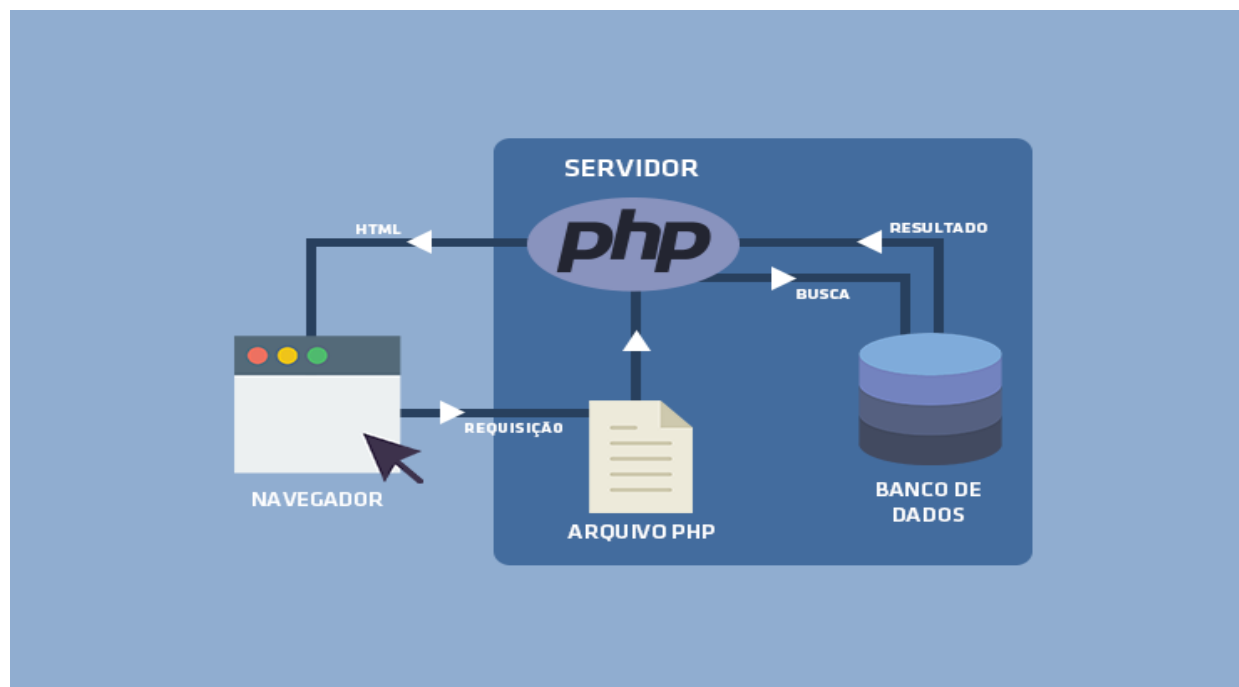


ECOMP

# MYSQL

## O QUE É BANCO DE DADOS E SUA RELAÇÃO COM PHP E NAVEGADOR

BD é um conjunto de informações que se relacionam entre si de acordo com suas características. Sua função é guardar esses dados e, que processados, formam uma página para um usuário acessar.



Disponível em: <https://www.tutoriart.com.br/como-criar-e-configurar-o-banco-de-dados-para-seu-blog-wordpress/>

- O usuário acessa uma página, na qual é um arquivo no formato PHP;
- O que a página requisitar o PHP buscará somente esses dados no banco, que é organizado em tabelas;
- Buscado esses dados, o PHP completa o pedido, inserindo em uma página no formato formato HTML;
- Ao final, a página é entregue ao navegador do visitante.

## CHAVE PRIMÁRIA

A chave primária (**primary key**) é um atributo na qual identifica apenas um registro dentro do BD. Com isso, o valor dentro de uma chave primária não poderá se repetir e não poderá conter um valor nulo.



### Exemplo Criado

Para chave primária o atributo escolhido foi o GRR, que é único de cada pessoa, porém outros identificadores podem ser usados, como o CPF.

## CHAVE ESTRANGEIRA

A chave estrangeira (**foreign key**) é um atributo que estabelece uma relação entre tabelas, através da referência de uma **primary key**.

## CRIANDO DATABASE

```
CREATE DATABASE nome_database;
```

### Código pro Database

```
CREATE DATABASE ecomp;
```

## CRIANDO TABELA

```
CREATE TABLE nome_tabela (  
    coluna1 datatype,  
    coluna2 datatype,  
    coluna3 datatype,  
    ....  
);
```



### Código da Tabela

```
CREATE TABLE projetos(  
    grr    int auto_increment,  
    nome_membro varchar(45),  
    cargo    varchar(45),  
    PRIMARY KEY (grr)  
);
```

## INSERINDO DADOS

```
INSERT INTO nome_tabela(coluna1,coluna2,coluna3)  
VALUES ('valor1','valor2','valor3');
```

### Código pra Inserir

```
INSERT INTO projetos(nome_membro,cargo)  
VALUES ('João Picolo','Diretor');
```

## SELECIONANDO DADOS

```
SELECT *, coluna1,coluna2, ...  
FROM nome_tabela;  
WHERE condição;
```

### Código pra Selecionar

```
SELECT *  
FROM projetos  
WHERE nome_membro LIKE 'J%';
```

Aqui retornará os membros  
que começam com J.

## EXCLUINDO DADOS

```
DELETE FROM nome_tabela  
WHERE condição;
```

### Código pra Excluir

```
DELETE  
FROM projetos  
WHERE cargo = 'Diretor';
```

Aqui deletará todos os diretores.

## ATUALIZANDO DADOS

```
UPDATE nome_tabela  
SET coluna1 = valor1, coluna2 = valor2  
WHERE condição;
```

### Código pra Atualizar

```
UPDATE projetos  
SET cargo = 'mentor'  
WHERE nome_membro = 'João Picolo';
```

O cargo de João Picolo, irá mudar para mentor

## LISTANDO DADOS

```
SELECT *, coluna1, coluna2..  
FROM nome_tabela;
```

### Código pra Listar

```
SELECT nome_membro  
FROM projetos;
```

Aqui mostrará  
todos os membros.



**PHP**

## **SOBRE PHP E UTILIDADE**

A linguagem de programação PHP é utilizada por diversos desenvolvedores no mundo. Ela fornece uma interação dinâmica com o usuário através de formulários, link e fornecimento de parâmetros de URL.

Sua principal utilidade é em torno de aplicativos web, visto que ela é capaz de interpretar scripts e fornecer/gerar páginas HTML.

## **ESTRUTURA BÁSICA**

### **EXEMPLO DE USO**

```
<?PHP  
    CONTEÚDO;  
?>
```

### **VARIÁVEIS**

```
<?php  
    $VARIÁVEL = "valor";  
?>
```

## INCLUDE E REQUIRE

Include e require servem basicamente para importar um arquivo php para dentro da sua página. É comum que uma aplicação tenha vários arquivos .php, portanto importar é necessário.

```
INCLUDE('CODIGO.PHP');
```

**Ele importará o arquivo codigo.php e, caso algum erro aconteça, o PHP irá retornar um WARNING, contudo continuará a execução.**

```
REQUIRE('CODIGO.PHP');
```

**Ele importará o arquivo codigo.php e, caso algum erro aconteça, o PHP irá retornar um ERRO DE COMPILAÇÃO, encerrando a execução do script.**

## MÉTODO POST

O funcionamento desse método é similar ao GET, contudo a capacidade de passar dados/informações não é limitada e é possível passar outros tipos de dados, como imagens.

```
<?php  
$variável = $_POST['campo']  
?>
```

O valor 'campo' colocado no exemplo acima é buscado no formulário pelo NAME do INPUT ou alguma TAG compatível.

## MÉTODO GET

Este método é utilizado com o objetivo de passar pequenos dados/informações e também passar um dado para outra página através da URL.

Quando o objetivo é transmitir informações sigilosas, como senhas, o método GET **não** deve ser usado.

```
<?php  
$variável = $_GET['campo']  
?>
```

Ao criar uma tag FORM no HTML é necessário usar o METHOD="tipo\_de\_método" nela. No exemplo do GET, ficaria METHOD="GET".

## ORIENTAÇÃO AO OBJETO

A programação orientada a objetos é um método de programação que usa Objetos e Classes como elementos principais para o processamento e a representação de dados usados nos programas. Algumas definições importantes:

**Classe:** é uma estrutura que contém um conjunto de objetos contendo características em comum. Ela usa métodos (functions) para definir o comportamento de seus atributos.

**Objeto:** é uma instância, um exemplar de uma classe . Ele se define na execução do código da classe.

**Atributos:** é a organização dos dados que irão representar a classe, ou seja, são as especificações de um objeto.

É interessante saber a definição formal, contudo é mais fácil o entendimento com um exemplo:

### Exemplo de Classe

```
Class Ecomp{  
  
}
```

## Exemplo de Atribuições

```
Class Ecomp{  
    public $quemSomos;  
    public $valores;  
    public $totalDiretorias;  
}
```

## Exemplo de Objeto

```
$ecomp = new Ecomp();  
  
$ecomp→quemSomos = "A maior empresa júnior";  
$ecomp→valores = "Paixão, profissionalismo, crescimento e inovação";  
$ecomp→totalDiretorias= 7;
```

## CONEXÃO BANCO DE DADOS

```
MYSQLI ("host", "usuário", "senha", "banco_dados");
```

OU

```
new PDO("mysql:host=HOST;dbname=BASE", "USUARIO",  
"SENHA");
```



FIM!

