**PHP DO ZERO**

Algoritimo: É uma série de condições que são feitas para finalizar uma tarefa.

O que é o WAMP?   
É um servidor local que é possivel utilizar o PHP com o Banco de dados.

W – Windows  
A – Apache  
M – MySQL  
P – PHP.

**Variáveis**

Como nomear uma variável?

Pode:  
$altura  
$variavel\_correta  
$minhaVariavel  
$MINHAVARIAVEL  
$Minha\_Variavel  
$variavel10

Não pode:  
$minha-variavel  
$1variavel  
$var iavel  
$ variavel  
$\*&variavel

Site para testar PHP na internet.  
<http://phptester.net>

**Declarando variáveis de forma correta:**

$nome = “Paulo Henrique”;  
$inteiro = 10000;  
$preco = 100.50;  
$lista = [10.20, 50, 40.23]; ou $lista = array(10.20, 50, 40.23);  
$eh\_verdade = true;  
$variavel10 = 10;

**Exibindo variáveis**

Ex: echo $variavel;

O Echo não exibe listas e para tipo booleano false, mas para exibir a lista e o valor booleano, é preciso usar o var\_dump.

Ex: var\_dump($inteiro);

**Constantes**

Não precisam de $.  
Variáveis que não mudam na execução do código.  
Seguem as mesmas regras das variáveis.

Ex: define(“nome”, valor);

**Operadores**

**O que são operadores?**  
São formas de representar operações matemáticas na computação.

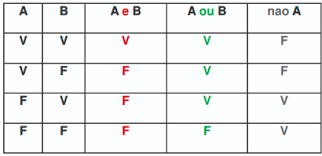
**Os tipos principais de operadores:**

Operador de atribuição  
=

Operadores aritiméticos  
+  
-  
\*  
/

Operadores de compração  
==  
===  
!=  
<  
>  
<=  
>=

Operadores lógicos  
&& - E  
|| - OU  
! – NÃO



**Estruturas de condições/condicionais**

**Para que servem?**

Exemplos:

O aluno só será aprovado se a nota for >= 6.  
O usuário só poderá fazer login se sua senha estiver correta.

Implementando:

If(condição) – Lembram dos operadores lógicos? (Não é 0 ou false -> true).

**Variações da estrutura if**

**else**

Se a nota for < 6, o aluno está reprovado.  
Se a senha estiver incorreta, exibir a mensagem: “Senha Incorreta”

**else if**

Verificar se o número é positivo, negativo ou zero.

**Loops**

**FOR**

for(variável; condição; incremento ou decremento){

}

**DO WHILE**

do {

} while(condição)

**WHILE**

while(condição) {

}

**FOREACH**

foreach(array as variaval) {

}

**Array**

Formas e possibilidades de se criar um array

$lista = array(1, 2, 3, 4); - Array normal

$lista2 = [1, 2, 3, 4]; - Outra forma

$lista3 = ["a", "palavras", 3, 3.23]; - Array com vários tipos

Também é possivel criar um array com indicies predefinidos:

$lista3 = array(

    10 => "Item 1",

    20 => "Item 2",

    30 => 3.4

);

É possivel criar um array dentro de outro array:

$lista = array(0, 0, 0, 0, 0);

$lista[5] = array(1, 1, 1, 1555, 1);

Já para acessa é preciso utilizar:

echo $lista[5][3];

Para adicionar um valor ao array já criado, é preciso usar:

// $lista = array(1, 2, 3, 4, 5);

// $lista[] = 6;

Ou pode utilizar a função push:

// array\_push($lista, 10); - Primeiro parâmetro é o array, que será adicionado no último lugar, já o segundo é o valor.

Para remover um valor e uma posição, usasse a função pop:

// array\_pop ($lista)

**Funções**

Para criar uma função no PHP:

function nomeDaFuncao($variaveis) {

}

Para utilizar a função, é preciso usar:

nomeDaFuncao($valordavariavel)

**Formulários**

HTTP/HTTPS

**Métodos**:

**GET** -  
Dados são passados pela URL, deve ser usado quando o intuito é RECUPERAR DADOS.

Ex:

Seusite.com.br/index.php?nome=Paulo&idade=19

A screen shot of a computer screen

Description automatically generated

O action do form vai servi para indicar aonde o form vai levar, caso tenha um botão com type submit.

O method utilizado é o GET, o mesmo que será usado no PHP.

O name vai servir para usar a váriavel no PHP.

Ex:

$\_GET[‘nome’];

$\_GET[‘idade’];

$\_GET[‘email’];

A screen shot of a computer code

Description automatically generated

**POST** -

Dados são passados dentro da requisição HTTP.

No método POST, os dados não será exibidos pela URL.

O method utilizado é o POST, o mesmo que será usado no PHP.

O name vai servir para usar a váriavel no PHP.

Ex:

$\_POST[‘nome’];

$\_POST[‘idade’];

$\_POST[‘email’];

Intval() – Faz com que a variável selecionada esteja sendo tratada como inteiro.

Ex:

$idade = intval($\_POST['idade']);

Para o PHP, se por exemplo você quiser colocar um class no código ou algo do gênero, não irá funcionar, pois dará erro nas aspas, mas existe duas maneiras de resolver.

Ou você coloca

Ex: “<p class=\” exemplo\”>”

Ou você pode começar com aspas simples

Ex: ‘<p class=”exemplo”>’

Die(); - Serve para acabar com a execução do script.

Isset é usado para saber se aquela variável está sendo usada no momento.

**Banco de dados**

É basicamente uma tabela no execel, mas com uma padronização.

MySQL- É um software que utiliza a liguagem SQL para manipulação dos dados de um Banco de dados.

**O que é um SGBD?**  
É um sistema de gerenciamento de uma banco de dados.

SQL – É uma linguagem de consuta para banco de dados.

PHPMyAdmin – É um software com gerenciamento visual para manipular um banco de dados.

**Tipos de Dados**

INT – Inteiro, são aqueles números que não possiem virgulas. Ex: (0, 2, 3, 4, 5, -1, -4).

VARCHAR – É basicamente uma string, pois ele irá aceitar textos. Ex: “(81) 99999-4444”, “email@gmail.com”, “abcd” e etc.

DATE – Data, é ano, mês e dia. Ex: “2023-12-21”.

DATETIME – É a data e a hora juntas. Ex: “2021-01-23 11:39:45”

DECIMAL – É usado para números decimais, ou seja os números quebrados. Para definir o tamanho e valor dele, é preciso usar o tamanho total do número mais o tamanho de casas decimais. Ex: “1000,00 = 6,2”. Um padrão é 10,2, pois vai existir uma margem de erro.

LONGTEXT – É um texto logo, ou seja, uma string que não se sabe o tamanho.

BOOLEAN – É um valor de duas escolhas. Ex: ”1 = True / 0 = False”.

**Comandos SQL**

**SELECT** – Serve para selecionar um ou diversos campos de uma tabela.

Ex: SELECT \* FROM veiculos; - Seleciona todos os veículos

**WHERE –** Da tradução literal é chamado de aonde, ou seja, é uma clausura para especificar uma busca.

Ex:SELECT \* FROM veiculos WHERE id = “9”; - Irá pegar o veículo com o ID 9.

**LIMIT –** Limita uma determinada pesquisa de acordo com o número desejado.

Ex:SELECT \* FROM veiculos WHERE fabricante = ‘Peugout’ LIMIT 10; - Esse commando irá pegar os veículos com o fabricante Peugout, mas irá ser exibido 10 carros, independente de quantos estejam no Banco de dados.

**Operadores:**

**AND –** Esse operador irá buscar algo a mais que seja preciso.

Ex: SELECT \* FROM veiculos WHERE fabricante = ‘Citroen’ AND id > 10; - Aqui irá pegar todos os veículos da Citroen, mas que tenha o ID maior que 10.

Podemos também pegar alguns especificos.

Ex: SELECT \* FROM veiculos WHERE fabricante = ‘Citroen’ AND id > 10 AND id < 15; - Esse irá aparecer apenas os maiores que 10 e menores que 15.

Ainda podemos retirar um ID especifico.

Ex: SELECT \* FROM veiculos WHERE fabricante = ‘Citroen’ AND veiculo = ‘C3’ AND id != 10;- Irá pegar todos C3 e não selecionar o Id 10.

**OR –** Esse operador é básimente um “OU”, ou seja, irá pegar um ou outro.

Ex: SELECT \* FROM veiculos WHERE fabricante = ‘Citroen’ OR fabricante = ‘Peugout’; - Aqui será pego ou o fabricante Citroen ou o Peugout, independente da ordem.

**ORDER BY –** Server para ordenar os resultados com base em uma coluna.

Existe dois tipos de ordenações, sendo elas ASC e DESC.

O ASC é de forma crescente, independente de ser número ou texto, Já o DESC é de forma decrescente.

Ex:

SELECT \* FROM veiculos ORDER BY fabricante ASC; - Irá pegar todos os veiculos pelo nome do fabricante em ordem crescente.

SELECT \* FROM veiculos WHERE fabricante = ‘Peugout’ OR fabricante = ‘Fiat’ ORDER BY fabricante ASC; - Dessa forma será pego os fabricantes Peugout ou Fiat, ordenados de forma crscente.

É possivel também usar dois ou mais Order By:

SELECT \* FROM veiculos ORDER BY fabricante ASC, modelo DESC; - Assim aparecera os veículos com as fabricantes em ordem crescente e ao mesmo tempo os modelos decrescente.

**Operações**

**COUNT() –** É uma forma de mostrar quantas linhas estão sendo exprimidas de tal consuta.

Ex: SELECT COUNT(\*) FROM veiculos WHERE fabricante = ‘Peugout’ AND veiculo = ‘106’; - Aqui irá mostrar a quantidade de linhas exprimidas dos veículos com fabricante Peugout, do modelo 106.

**SUM() –** É a soma de determinados valores de uma tabela.

Ex: SELECT SUM(preco) FROM livros; - Irá mostrar a soma do preço de todos os livros.

**AVG() –** É uma forma de tirar a média de todos os valores de uma tabela.

Ex: SELECT AVG(preco) FROM livros; - Irá pegar todos os preços dos livros e fazer a média deles.

Existe também uma forma de renomear a a linhas ao exprimir, sem trocar o nome original. É preciso utilizar o AS.  
Ex: SELECT autor as AutorDoLivro, isbn as CodigoNacional, preco \* estoque as valorTotalEmEstoque FROM livros;

**UPDATE –** O update fazer uma atualização nos dados, ou seja, trocar alguns dados especificos.

Ex: UPDATE livros SET autor = 'Joao Oliveira', preco = 30 WHERE id = 2; - Esse código irá trocar o nome do autor e o preço do livro, se o id igual a 2.

UPDATE livros SET assunto = 'Sem assunto' WHERE assunto = 'Biografia' – Já nesse irá trocar o assunto, mas só se o assunto anterior for Biografia.

**DELETE –** O delete apaga determinada linha do banco de dados, de acordo com as condições ou não desejadas.

Ex: DELETE FROM livros WHERE id = 2; - Vai deletar toda a linha que tiver o Id = 2.

DELETE FROM livros WHERE estoque > 10; - Vai deletar toda a linha que tiver o estoque maior que 10.

**LIKE –** Faz a pesquisa de todas as coisas que são parecidas.

Ex: SELECT \* FROM veiculos WHERE LIKE ‘Peu%’; - Dessa maneira, a pesquisa irá pegar todos os veículos que começam com Peu e que tenha qualquer outra coisa depois, pois o sinal de porcentagem colocado depois, significa que pode vim qualquer outra coisa depois, mas também é possível colocar antes, sendo assim irá selecionar tal palavra, mesmo que tenha algo na frente.

Ex: SELECT \* FROM veiculos WHERE modelo LIKE ‘%Passion%’;

**IN() –** Faz uma pesquisa mais enchuta de acordo com o que está dentro do parêntese.

Ao invés de fazer vários OR para selecionar vários valores, É possível utilizar o IN, ele faz com que seja pesquisado de acordo com os valores dentro dele.

Ex:

SELECT \* FROM veiculos WHERE id IN(10, 11, 13, 14); - Nessa pesquisa irá pegar todos os veículos com id selecionado no IN.

**Juntando resultado de várias tabelas**

Podemos juntar os resultado de várias tabelas, só precisamos fazer um select de acordo com o que for preciso.

Ex:

SELECT compras.id, compras.data as dataDaCompra, livros.preco as precoDoLivro, livros.assunto as assuntoDoLivro, clientes.nome as nomeDoCliente

FROM compras, livros, clientes

WHERE compras.id\_cliente = clientes.id

AND compras.id\_livro = livros.id;

Nesse select está pegando as tabelas Compras, Livros e Clientes, E estamos fazendo a pesquisa de acordo com a tabela e seu deteminado campo.

Ex: compras.id – Está pegando o ID de compras.

Já na parte do WHERE, estamos dizendo que determinado tabela.campo é igual a sua respectiva tabela.campo.

Ex: compras.id\_cliente = cliente.id – Então a tabela compras que tem o campo id\_cliente, vem da tabela clientes do campo id.

**Conexão de um BD em PHP**

Para fazer a conexão do banco em Php, é preciso criar um arquivo de conexão e o configurar:

A computer screen shot of a program

Description automatically generated

E para conectar em uma página, é só usar o include:

A screen shot of a computer program

Description automatically generated

**CRUD**

**C – Create** (Criar) – Criar um novo registro.

**R – Read** (Ler) – Ler (Exibir) as informações de um registro.

**U – Update** (Atualizar) – Atualizar os dados do registro.

**D – Delete** (Apagar) – Apagar um registro.