

EVANDRO MANARA MILETTO  
SILVIA DE CASTRO BERTAGNOLLI



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO SUL

# DESENVOLVIMENTO DE SOFTWARE II

INTRODUÇÃO AO DESENVOLVIMENTO WEB  
COM HTML, CSS, JAVASCRIPT E PHP

EIXO INFORMAÇÃO E COMUNICAÇÃO

>> série tekne





D451      Desenvolvimento de software II [recurso eletrônico] : introdução ao desenvolvimento web com HTML, CSS, JavaScript e PHP / Organizadores, Evandro Manara Miletto, Sílvia de Castro Bertagnolli. – Dados eletrônicos. – Porto Alegre : Bookman, 2014.

Editado também como livro impresso em 2014.  
ISBN 978-85-8260-196-9

1. Informática – Desenvolvimento de software. 2. HTML.  
3. CSS. 4. JavaScript. 5. PHP. I. Miletto, Evandro Manara.  
II. Bertagnolli, Sílvia de Castro.

CDU 004.41



#### » IMPORTANTE

Observe o uso, dentro da classe, da variável especial `$this`. Esta variável pode ser utilizada para acessar internamente atributos (`$this->nome`) ou métodos da classe.

apresentados outros dois métodos (GET e set) para manipulação do atributo `nome` da classe (GETs e sets para os demais atributos foram omitidos neste exemplo).

Após a especificação da classe, há a instanciação de um objeto chamado de `$produtoatual` por meio da utilização da palavra-chave `new` em conjunto com o nome da classe e de todos os dados requeridos pelo seu construtor. Por fim, apresentamos a chamada ao método de atribuição do nome do produto (`$produtoatual->setNome("Notebook - edição especial")`) instanciado no objeto atual, passando como argumento uma nova string. O acesso a atributos ou métodos sempre se dá com o uso dos sinais `->` depois da variável que identifica o objeto e antes do elemento/método que se deseja acessar.

Evite a definição dos atributos de uma classe como `public`, a menos que seja realmente necessário. Como boa prática, priorize o acesso aos atributos por meio de métodos disponibilizados pela própria classe, definindo esses métodos como `public`, e os atributos como `private`.



## » Agora é a sua vez!

1. A partir da classe `Produto`, apresentada no exemplo anterior, crie um script PHP (`produto.php`) para conter especificamente as definições dessa classe e o armazene dentro de um diretório de classes (`./class/`). O diretório deve conter todos os scripts de definições de classes de seu sistema.
2. No arquivo recém-criado, insira as definições da classe apresentadas em nosso exemplo e amplie sua especificação incluindo os métodos GETs e sets para acesso aos demais atributos. Usaremos esse arquivo criado no decorrer dos exemplos de nosso livro.

## » Herança

A implementação de herança em PHP se assemelha muito ao padrão adotado por outras linguagens, como em Java, devendo-se utilizar a palavra-chave `extends` na declaração de uma classe para explicitar que ela herdará atributos e métodos de outra classe informada. O exemplo a seguir, ao ser apresentado em conjunto com o exemplo anterior, formaliza uma classe chamada de `ProdutoVip`, que,

além de ter todos os atributos e métodos da classe `Produto`, também apresenta um novo atributo chamado de `desconto`, bem como os métodos para alterá-lo.



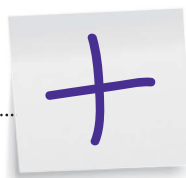
## >> EXEMPLO

```
<?php
class ProdutoVip extends Produto {
    private $desconto; // novo atributo, presente apenas em ProdutoVip
    // os demais atributos de Produto são herdados por ProdutoVip
    function __construct ($varnome, $varpreco, $vardescricao, $varquantidade,
        $varcodigo, $vardesconto) { // método construtor da classe
        parent::__construct ($varnome, $varpreco, $vardescricao, $varquantidade,
            $varcodigo); // chamada ao construtor da superclasse Produto
        $this->desconto = $vardesconto; // inicialização do novo atributo da subclasse
        ProdutoVip, não inicializada pelo construtor da superclasse Produto
    }
    public function getDesconto() { // método para obter valor do atributo
        return $this->desconto;
    }
    public function setDesconto($vardesconto) { // método para definir valor do atributo
        $this->nome = $vardesconto;
    }
    // os demais métodos de Produto são herdados por ProdutoVip
}

$produtovipatual = new ProdutoVip("Notebook", 2999, "Intel Core i5 6GB 1TB LED 14
    Touchscreen Windows 8", 200, 28, 17.5); // instanciando um objeto da subclasse,
    chamando seu próprio construtor
echo 'O valor de desconto do produto vip '.$produtovipatual->getNome().' é
    '.$produtovipatual->getDesconto().'%. ' . '%.';
?>
```

Observe que, além do novo atributo `desconto`, a subclasse `ProdutoVip` apresenta o seu próprio construtor, o qual recebe os mesmos parâmetros da superclasse `Produto` e também o valor para o novo atributo `desconto`. Sempre que instanciarmos um objeto de uma classe herdeira, uma chamada será feita ao seu construtor, e não ao construtor da superclasse.

No entanto, note que, dentro do construtor de `ProdutoVip`, é utilizada a palavra-chave `parent` seguida de `::`, o que determina que o atributo ou método que vier a seguir está se referindo à superclasse da qual se está herdando. No caso do construtor de `ProdutoVip`, com o trecho `parent::__construct` estamos determinando que será chamado o construtor da superclasse `Produto` e este inicializa as variáveis presentes na superclasse. Já o novo atributo apresentado pela subclasse `ProdutoVip` passa a ser inicializado logo em seguida.



Por fim, a última linha de instruções do código solicita a impressão de uma frase em que serão apresentados atributos do objeto. Observe que, mesmo se tratando de um objeto da classe `ProdutoVip`, é possível acessar o método `getNome` que se encontra definido na classe `Produto` (esse método foi herdado por `ProdutoVip`). O método `getDesconto`, definido na classe `ProdutoVip`, também é solicitado nessa linha, cujo resultado a ser impresso no navegador seria: "O valor de desconto do produto vip Notebook é 17,5%".

## >> PARA SABER MAIS

Informações mais aprofundadas sobre conceitos de orientação a objetos fogem do escopo desta obra. Para a obtenção de um maior detalhamento, sugere-se a consulta do livro *Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*, 3. ed. (Veja referência ao final do capítulo.)

# >> Reuso de código: Include e Require

No processo de desenvolvimento de um sistema, muitas vezes pode ser interessante realizar o reaproveitamento de trechos de códigos já desenvolvidos. O reuso de código proporciona benefícios ao reduzir o tempo necessário para a realização de procedimentos já formalizados e testados, bem como ao favorecer a manutenção desses procedimentos em caso de necessidade de futuras alterações.

Assim, o `Include` e o `Require` são recursos muito semelhantes no PHP. Ambos são utilizados para a inclusão de um determinado arquivo no script em que está sendo chamado. Para usar o `include`, devemos especificar o arquivo que será incluído no script em que a função é chamada. O arquivo será incluído tomando como base o caminho relativo apresentado (conceito abordado no Capítulo 4) ou, nos casos de não haver especificação, o caminho definido por `include_path` nas configurações do PHP no servidor.

Veja a seguir um exemplo para a inclusão de um arquivo que contém as definições da classe `Produto`. Ele permite visualizar como podemos fazer uso no script atual dos recursos definidos pelo arquivo incluído, sem a necessidade de reescrita do trecho de código com as definições da classe.

Página propositalmente deixada em branco



## >> capítulo 8

# Integração de PHP e MySQL

*Neste capítulo, apresentaremos como se dá a utilização do PHP em conjunto com o Sistema Gerenciador de Banco de Dados (SGBD), mais especificamente com o MySQL. Serão abordados os procedimentos necessários para permitir o gerenciamento e a consulta das informações armazenadas no exemplo de sistema de compras online trabalhado em todos os capítulos.*

### Objetivos de aprendizagem

- >> Realizar os procedimentos necessários para a estruturação das bases de dados utilizadas em sistemas Web.
- >> Gerenciar as informações armazenadas nas bases de dados.
- >> Realizar e tratar os resultados de consultas realizadas sobre bases de dados MySQL.



#### >> IMPORTANTE

O MySQL também possibilita a aquisição de licenças específicas para uso comercial do SGBD.

## >> Introdução

Conforme visto no capítulo anterior, o PHP apresenta compatibilidade com um conjunto bastante significativo de Sistemas Gerenciadores de Bancos de Dados (SGBD). Dentre os sistemas relacionais suportados, destaca-se o **MySQL**, distribuído sob licença GPL (*GNU General Public License*), e de ampla utilização e compatibilidade com distintas linguagem de programação e plataformas de operação.



#### >> CURIOSIDADE

O MySQL foi disponibilizado, em sua primeira versão, em 1996, apesar de já ter ocorrido um lançamento interno do sistema em maio de 1995. É considerado o segundo SGBD mais utilizado no mundo, de acordo com dados de outubro de 2013, da pesquisa da DB-Engines.

www.

#### >> NO SITE

Para obter detalhes sobre os procedimentos específicos para instalação do MySQL, nas mais distintas plataformas, acesse o ambiente virtual de aprendizagem Tekne: [www.bookman.com.br/tekne](http://www.bookman.com.br/tekne).

Para atuar em conjunto com o SGBD MySQL, a linguagem PHP apresenta um amplo conjunto de funções. Apresentaremos, no decorrer deste capítulo, as fundamentais para habilitá-lo a implementar um sistema Web que possibilite o gerenciamento e a consulta de informações armazenadas em um banco de dados.

A fim de que seja possível realizar testes e implementar os exemplos que serão apresentados, é fundamental que você tenha o MySQL instalado e configurado em um servidor ou em seu computador pessoal.

Inicialmente, abordaremos os aspectos fundamentais para que se estabeleça a conexão com um SGBD MySQL. Na sequência, apresentaremos os procedimentos para o gerenciamento e seleção de bases de dados. A interação com uma base de dados é apresentada no tópico seguinte, em que será explorada a realização de consultas SQL. Por fim, chegará a vez das funções e dos procedimentos para o tratamento das consultas realizadas e das possibilidades de recursos para utilização do MySQL com orientação a objetos.

www.

#### >> NO SITE

Para mais informações sobre a pesquisa da DB-Engines, acesse o ambiente virtual de aprendizagem Tekne.





## >> EXEMPLO

```
<?php
$con = mysql_connect('localhost', 'administrador', 'minhasenha');
if (!$con)
    exit('Não foi possível conectar: ' . mysql_error());
// insira aqui o restante do código para utilização da conexão...
mysql_close($con); // encerramento da conexão
?>
```



### >> IMPORTANTE

O PHP disponibiliza também a função `mysql_pconnect()` para o estabelecimento de conexão com o MySQL, diferenciando-se da `mysql_connect()` por estabelecer uma conexão persistente com o banco, não sendo encerrada ao final da execução do script.

Observe, no exemplo, que é feito um teste após a tentativa de estabelecimento de conexão, para verificar se o procedimento foi realizado com sucesso. Caso tenha ocorrido algum problema, `$con` receberá o valor `false` e será utilizada a função `mysql_error()` para apresentar informações complementares sobre a falha no estabelecimento, encerrando a execução do script com a utilização da construção de linguagem `exit()`.

Após a inclusão do código adicional que fará uso da conexão recém-estabelecida, deve-se inserir uma chamada à função `mysql_close()`, a qual recebe como parâmetro o identificador da conexão para proceder com seu encerramento. Caso não seja passado um identificador como parâmetro, a última conexão estabelecida será encerrada. Se não for feito uso dessa função, a conexão será encerrada tão logo a execução do script terminar.



## >> CURIOSIDADE

O procedimento de fazer uso da última conexão estabelecida em caso de supressão do identificador é geralmente adotado nas demais funções relacionadas ao MySQL que não apresentam esse parâmetro como obrigatório.

# >> Seleção de bases de dados

Uma vez estabelecida uma conexão com o MySQL e partindo-se de uma base de dados já existente, deve-se passar à seleção dessa base de dados, para que seja possível o gerenciamento das informações nela armazenadas. É a função `mysql_select_db()` que possibilita essa seleção. Para realizar a seleção de uma base de dados, é necessário especificar o nome da base que se deseja selecionar e, opcionalmente, o identificador de conexão adotado. Em caso de sucesso na seleção, aparece o retorno do valor `true`, e, para casos de insucesso, o retorno `false`.

A sintaxe da função `mysql_select_db()` é:

```
bool mysql_select_db (string $database_name [,  
resource $ link_identifier ])
```

A criação de uma base de dados pode ser feita com a função específica do PHP `mysql_create_db()`. No entanto, o uso dessa função tem se tornado obsoleto, sendo preferencial o uso da consulta, em SQL, `CREATE DATABASE`, em conjunto com a função `mysql_query()`. Veremos os aspectos relacionados ao uso desta função de execução de comandos SQL no decorrer deste capítulo.

Veja o exemplo que apresenta um trecho de código do procedimento de seleção para interação de uma base de dados já existente denominada “sitedecompras”.

## >> EXEMPLO

```
<?php  
include 'conectamysql.php'; // arquivo que estabelece a conexão  
$db_selecionado = mysql_select_db('sitedecompras', $con);  
if (!$db_selecionado)  
    exit('Não foi possível selecionar a base de dados: '. mysql_error());  
// insira aqui seu código para uso da base de dados...  
mysql_close($con); // encerramento da conexão  
?>
```

No código apresentado no exemplo anterior, foi estabelecida uma conexão com o banco de dados do servidor e, em seguida, realizada a seleção da base de dados “sitedecompras”. Caso tenha ocorrido algum problema na seleção da base, é apresentada uma mensagem ao usuário especificando o problema.

Observe que o procedimento de estabelecimento de conexão apresentado no exemplo da seção “Conexão com o banco de dados” foi disponibilizado dentro do arquivo `conectamysql.php`, o qual foi incluído no script na primeira linha. Esse procedimento é recomendado para evitar que informações referentes ao usuário e à senha sejam disponibilizadas em todos os scripts que interajam com o banco de dados, gerando problemas de segurança e dificuldade na manutenção do sistema em caso de alterações nas configurações do servidor MySQL.

## » Realização de consultas SQL

A partir do estabelecimento de conexão com o banco de dados e da seleção da base de dados adotada, é possível passar à interação com esta última. Esse processo se dará principalmente por meio da realização de consultas SQL, possibilitada pelo uso da função `mysql_query()`.

A sintaxe para `mysql_query()` é:

```
resource mysql_query (string $query [, resource  
$link_identifier ])
```

Ao fazer uso dessa função, deve-se passar como primeiro parâmetro a consulta SQL que se deseja realizar (`$query`) e, opcionalmente, o identificador de uma conexão já estabelecida com o banco de dados. O valor retornado pela execução com sucesso da função dependerá do tipo de comando SQL utilizado: `SELECT`, `SHOW`, `DESCRIBE` ou `EXPLAIN` retornam um `resource` que poderá ser tratado no decorrer do script; já as demais consultas SQL (como `UPDATE` ou `DELETE`) retornam `true`. Para qualquer SQL realizado, o valor `false` será retornado sempre que ocorrerem falhas.

Observe a seguir o exemplo sobre o uso dessa função no processo de criação da base de dados “`sitedecompras`”, caso ela ainda não houvesse sido criada.

