

Desenvolvimento de Sistemas

Aluna: Louise Geppert Varela.

Turma: 01.

Hypertext preprocessor (PHP)

O PHP (acrônimo recursivo para "PHP: Hypertext Preprocessor") é uma das linguagens de programação mais populares, versáteis e amplamente utilizadas para o desenvolvimento web. É uma linguagem de script de código aberto, o que significa que é gratuita e mantida por uma vasta comunidade global de desenvolvedores.

Características e funcionamento

- Execução no lado do servidor (Back-end): Ao contrário de linguagens como o JavaScript, que roda no navegador do usuário, o código **PHP é processado diretamente no servidor web**. Isso permite gerar conteúdo dinâmico antes que a página seja enviada para o navegador do cliente em formato HTML simples, mantendo o código-fonte oculto.
- Facilidade de uso: A sintaxe do PHP é simples e intuitiva, com a possibilidade de ser **embutida diretamente em documentos HTML**. Isso torna a curva de aprendizado mais suave, especialmente para iniciantes.
- Orientação a objetos: O PHP evoluiu ao longo dos anos para suportar o paradigma de Programação Orientada a Objetos (POO), permitindo **a criação de sistemas mais robustos e bem estruturados**.
- **Multiplataforma**: O PHP pode ser executado em diversos sistemas operacionais, como Windows, Linux e macOS, e é compatível com a maioria dos servidores web, como Apache e IIS.
- Integração com bancos de dados: Oferece **excelente suporte e fácil integração com vários sistemas de gerenciamento de bancos de dados**, como MySQL, PostgreSQL e Oracle, permitindo a criação de aplicações dinâmicas e baseadas em dados.
- Comunidade e ecossistema: Com mais de 25 anos de existência, o PHP possui uma comunidade ativa e uma vasta documentação, além de um grande número de frameworks (como Laravel e Symfony) e bibliotecas que facilitam e aceleram o desenvolvimento.

PHP e Banco de dados

A combinação de PHP com um banco de dados, como o MySQL, é a base para a **criação de sites e sistemas dinâmicos**. O PHP atua como a "**ponte**" entre o servidor, o navegador do usuário e o banco de dados. Para usar PHP com um banco de dados, é preciso ter as credenciais de acesso (host, nome do banco, usuário e senha) para criar uma conexão e, em seguida, executar consultas SQL usando os métodos dessas extensões para manipular os dados.

Principais extensões para conexão

- **PDO (PHP Data Objects):**
 - **O que é:** Fornece uma camada de abstração de dados para interagir com diversos bancos de dados.
 - **Vantagens:** Permite a troca de banco de dados com o mesmo código PHP, sendo mais versátil e recomendado.
 - **Exemplo de uso:** `new PDO('mysql:host=localhost;dbname=test', $user, $pass);`
- **MySQLi (MySQL Improved):**
 - **O que é:** Uma extensão mais moderna e focada exclusivamente em bancos de dados MySQL.
 - **Vantagens:** Mais simples para usar apenas com MySQL, oferece API orientada a objetos e procedural.
 - **Exemplo de uso:** `$conn = new mysqli($host, $username, $password, $dbname);`

Como funciona

1. **Requisição:** O usuário interage com uma página web, por exemplo, preenchendo e enviando um formulário.
2. **Processamento pelo PHP:** O servidor web recebe a requisição e a envia para o PHP processar o código.
3. **Interação com o banco de dados:** O script PHP se conecta ao banco de dados e executa comandos para inserir, ler, atualizar ou excluir dados (conhecido como operação CRUD – Create, Read, Update, Delete). Para isso, o PHP utiliza extensões como o PDO (PHP Data Objects) ou MySQLi.
4. **Resposta:** O banco de dados envia a resposta ao PHP. Com base nisso, o PHP gera o código HTML que será exibido ao usuário.

Para que serve

A interação entre PHP e um banco de dados é essencial para a criação de aplicações dinâmicas e interativas, que não dependem de páginas estáticas. Com essa combinação, é possível criar:

- **Sistemas de gerenciamento de conteúdo (CMS):** Como o WordPress, para criar e gerenciar blogs e sites.
- **Plataformas de e-commerce:** Para armazenar informações de produtos, clientes, pedidos e transações.
- **Redes sociais e fóruns:** Para salvar dados de usuários, postagens, comentários e interações.
- **Sistemas de login:** Para armazenar e validar dados de login de usuários, como nome e senha.
- **Formulários de contato:** Para coletar e armazenar informações enviadas pelos visitantes do site.