

1. INTRODUÇÃO

No desenvolvimento de aplicações web com PHP, o uso de bancos de dados relacionais, como o MySQL, é uma prática comum. Para realizar a comunicação entre o PHP e o banco de dados, existem duas principais extensões: MySQLi (MySQL Improved) e PDO (PHP Data Objects). Ambas oferecem suporte às operações básicas de CRUD (Create, Read, Update, Delete), porém, possuem diferenças significativas que impactam diretamente a segurança, portabilidade, desempenho e manutenção dos sistemas.

Este trabalho tem como objetivo analisar e comparar as características, vantagens e desvantagens de ambas as abordagens, bem como apresentar os principais métodos utilizados em cada uma, orientando desenvolvedores na escolha da solução mais adequada ao seu projeto.

2. MYSQLI

2.1 Visão Geral

A extensão MySQLi foi introduzida no PHP 5 como uma melhoria da antiga e obsoleta extensão MySQL. A sigla "MySQLi" significa MySQL Improved, e como o nome sugere, ela oferece funcionalidades mais robustas e seguras, sendo recomendada para conexões com servidores MySQL 4.1.3 ou superior. A MySQLi suporta tanto a programação procedural quanto a orientada a objetos, o que facilita sua adoção por desenvolvedores de diferentes níveis de experiência.

2.2 Vantagens

- **Desempenho otimizado para MySQL:** A MySQLi é projetada especificamente para trabalhar com o banco de dados MySQL, o que permite uma interação mais eficiente em termos de desempenho.
- **Prepared Statements:** Permite o uso de declarações preparadas, que ajudam a prevenir ataques de SQL Injection.
- **Suporte a múltiplas queries:** Possui suporte nativo à execução de múltiplas instruções SQL em uma única chamada.
- **Programação Procedural e OO:** A flexibilidade de poder programar tanto de forma procedural quanto orientada a objetos facilita a adaptação do código.

2.3 Desvantagens

- **Compatibilidade limitada:** Só funciona com o banco de dados MySQL. Caso haja a necessidade de trocar de banco no futuro, o código precisará ser completamente reescrito.
- **Abordagem menos moderna:** Em projetos grandes e que exigem maior abstração, a abordagem procedural pode se tornar difícil de manter.



2.4 Principais Métodos

- `mysqli_connect()` / `new mysqli()` – Criação da conexão com o banco.
- `query()` – Executa instruções SQL diretas.
- `prepare()` – Prepara uma instrução SQL para execução posterior.
- `bind_param()` – Liga variáveis aos parâmetros da instrução preparada.
- `execute()` – Executa a instrução preparada.
- `fetch_assoc()` – Obtém os resultados em formato de array associativo.

3. PDO (PHP DATA OBJECTS)

3.1 Visão Geral

PDO é uma interface de abstração para acesso a banco de dados, que permite trabalhar com diferentes sistemas de gerenciamento de banco de dados (SGBDs) por meio de uma API única. Com suporte para mais de 12 SGBDs, como MySQL, PostgreSQL, SQLite, Oracle e SQL Server, a PDO se tornou uma escolha ideal para projetos que requerem portabilidade e escalabilidade.

3.2 Vantagens

- **Portabilidade:** A principal vantagem do PDO é sua capacidade de suportar múltiplos bancos de dados sem alteração significativa no código.
- **Segurança:** Com suporte a prepared statements, o PDO oferece excelente proteção contra injeção de SQL.
- **Manutenibilidade:** Como é orientado a objetos e segue boas práticas de encapsulamento, torna o código mais limpo e modular.
- **Flexibilidade:** Permite configurar o modo de retorno dos resultados (array associativo, numérico ou ambos).

3.3 Desvantagens

- **Curva de aprendizado:** Por ser totalmente orientado a objetos, pode ser mais difícil para iniciantes em PHP.
- **Desempenho levemente inferior:** Em alguns benchmarks específicos, o PDO apresenta desempenho levemente inferior ao MySQLi com MySQL.

3.4 Principais Métodos

- `new PDO()` – Cria uma nova conexão com o banco de dados.
- `prepare()` – Prepara uma instrução SQL para execução.
- `bindParam()` – Liga variáveis aos parâmetros da instrução preparada.
- `execute()` – Executa a instrução SQL.
- `fetch()` / `fetchAll()` – Recupera os resultados da consulta.



4. COMPARAÇÃO ENTRE MYSQLI E PDO

A tabela a seguir resume as principais diferenças entre as duas abordagens:

Característica	MySQLi	PDO
Suporte a múltiplos bancos	Não	Sim
Interface procedural	Sim	Não
Interface OO	Sim	Sim
Prepared statements	Sim	Sim
Facilidade de migração	Baixa	Alta
Performance com MySQL	Alta	Boa
Segurança contra SQLi	Boa	Excelente
Suporte a múltiplas queries	Sim	Não (nativamente)
Suporte a transações	Sim	Sim
API moderna	Parcialmente	Sim

5. BOAS PRÁTICAS E CONSIDERAÇÕES DE SEGURANÇA

Ao trabalhar com qualquer uma das extensões, é essencial seguir boas práticas de segurança para evitar vulnerabilidades comuns, como SQL Injection. A utilização de prepared statements é indispensável para proteger os dados do usuário e garantir a integridade do sistema. Além disso, recomenda-se:

- Nunca concatenar diretamente dados de entrada em instruções SQL.
- Validar e sanitizar dados antes de inseri-los no banco.
- Utilizar variáveis tipadas e funções específicas para manipulação de dados.
- Evitar o uso de senhas hardcoded no código-fonte.

6. CENÁRIOS DE USO RECOMENDADOS

- **Uso do MySQLi:** Ideal para aplicações que utilizarão exclusivamente MySQL, com foco em desempenho e simplicidade.
- **Uso do PDO:** Indicado para aplicações que podem evoluir para utilizar outros SGBDs, ou que necessitam de um código mais limpo, seguro e modular.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02
Setor Leste - Gama - DF
CEP: 72445-020



7. CONCLUSÃO

A escolha entre MySQLi e PDO depende dos requisitos técnicos e da escalabilidade esperada para o projeto. MySQLi se destaca pela simplicidade e desempenho ao trabalhar exclusivamente com MySQL, sendo ideal para projetos pequenos ou quando a performance é prioridade. Em contrapartida, o PDO oferece uma solução mais robusta e moderna, garantindo segurança e portabilidade, o que o torna uma escolha preferencial para aplicações escaláveis e mais complexas.

A recomendação geral para novos projetos é a adoção do PDO, especialmente em ambientes profissionais que exigem manutenção contínua, segurança de dados e integração com múltiplos sistemas.

REFERÊNCIAS

PHP Manual. Disponível em: https://www.php.net/manual/pt_BR/ DigitalOcean. "PDO vs. MySQLi: Choosing the Right PHP Database Extension." Acesso em: 17 abr. 2025.
W3Schools. "PHP MySQL Database." Disponível em: https://www.w3schools.com/php/php_mysql_intro.asp DevDocs. "PHP: PDO." Disponível em: <https://devdocs.io/php/pdo/> Stack Overflow. "Differences between PDO and MySQLi." Acesso em: 17 abr. 2025.

