

# **Desenvolvimento para Servidores 2**

**Prof. Orlando Saraiva Júnior**  
**[Orlando.nascimento@fatec.sp.gov.br](mailto:Orlando.nascimento@fatec.sp.gov.br)**

# **Do estruturado à orientação a objetos**

O objetivo desta aula é demonstrar a evolução de um programa PHP desde sua concepção, usando o método estruturado até a utilização do método orientado a objetos.

# Nível 1: Procedural, um script por ação

---

Neste nível de aprendizado, construiremos um programa estruturado em que cada script (.php) representa uma ação do usuário ( incluir, editar, excluir, listar).

Neste nível, os programas criados representarão eventualmente uma combinação de PHP, HTML e SQL às vezes no mesmo arquivo, o que não é uma boa prática, porque mistura diferentes aspectos de programação.

## Nível 2: Agrupando ações comuns em scripts

---

Um dos problemas do cenário apresentado no nível 1 é a grande quantidade de scripts para realizar as operações. Considerando que, naquele modelo, cada ação do usuário era respondida por um script.

Neste novo cenário apresentado no nível 2, reuniremos scripts com ações em comum. Assim, conseguiremos reduzir a quantidade total de scripts, que foi de seis scripts no nível 1 para apenas dois scripts no nível 2.

# Nível 3: Separando o HTML com micro-templates

---

No nível anterior, reunimos todas as ações que o usuário precisava executar em dois scripts, um que representava tudo o que estava relacionado ao formulário ( inserção e edição) e tudo o que estava relacionado à listagem ( apresentação, exclusão ).

Entretanto, vimos que, apesar de evoluirmos, alguns problemas ainda estão presentes. Um deles é a mistura de HTML e PHP no mesmo arquivo.

## Nível 4: Separando o acesso a dados com funções

---

No nível anterior, conseguimos separar o PHP do HTML, o que tornou os scripts mais fáceis de serem lidos e compreendidos. A manutenção também melhorou, pois o isolamento entre diferentes questões ( lógica, apresentação) faz com que melhorias nesse script sejam mais rápidas de ser aplicadas.

No entanto, nossos scripts ainda tem aspectos de acesso à base de dados espalhados no código, ou seja, teremos INSERTs, UPDATEs, DELETEs e SELECTs espalhados por aí.

# Nível 5: Separando o acesso a dados com classes

---

No nível anterior, separamos as funções de manipulação de dados do script principal e as colocamos em um arquivo de funções. Com essa separação, já pudemos perceber as vantagens de separar diferentes aspectos do programa, mas ainda temos algo a melhorar, pois as funções de acesso ao banco de dados não utilizam classes e métodos.



## Nível 6: Melhorando as conexões e a segurança

---

No nível anterior, transformamos funções de manipulação de dados em classes com métodos estáticos. O código ficou mais organizado, e com melhor legibilidade. Porém ainda podemos fazer melhor. Neste novo nível, vamos melhorar dois aspectos relacionados à segurança: Quanto à criação da conexão e segurança.

O método `getConnection()` criará uma conexão e a armazenará na variável estática.

# Nível 7: Transformando páginas em classes de controle

---

No nível anterior, melhoramos a criação de conexões e a segurança de nossa classe de manipulação de dados. Porém, nossos principais programas ( `pessoa_form.php` e `pessoa_list.php`) ainda são totalmente estruturados e dependem de que o usuário acesse cada um deles individualmente, ou seja, não há um ponto central a partir do qual as requisições possam ser feitas.

Ter vários pontos por meio dos quais certas funcionalidades podem ser acessadas pode ser perigoso.

# Dúvidas

**Prof. Orlando Saraiva Júnior**  
**[orlando.nascimento@fatec.sp.gov.br](mailto:orlando.nascimento@fatec.sp.gov.br)**

O código do capítulo 3 do livro “PHP – Programando com Orientação a Objetos” estão prontos para ser executados com o banco de dados PostgreSQL.

Migre o nível 6 para que o programa acesse o banco de dados MySQL.