

Módulo 5 – Persistência

Neste workbook do módulo 5, revisaremos os principais conceitos vistos nas videoaulas.

Table Data Gateway

Uma classe que gerencia a persistência de tabela na base de dados, sem retenção de estado, ou seja, você sempre precisa informar todos os atributos necessários para a operação.

classes/tdg/ProdutoGateway.php

```
<?php
class ProdutoGateway {
    private static $conn;

    public function find($id, $class = 'stdClass') {
        $sql = "SELECT * FROM produto where id = '$id' ";
        print "$sql <br>\n";
        $result = self::$conn->query($sql);
        return $result->fetchObject($class);
    }
}
```

Active Record

Uma classe que gerencia a persistência e a lógica de negócios de uma tabela. Como ela retém estado, você se baseará em atributos do objeto (`$this->preco`), pois existirá um objeto para cada registro que for carregado da base de dados para a memória.

```
<?php
class Produto {

    public static function find($id) {
        $sql = "SELECT * FROM produto where id = '$id' ";
        print "$sql <br>\n";
        $result = self::$conn->query($sql);
        return $result->fetchObject(__CLASS__);
    }

    public function getMargemLucro() {
        return (($this->preco_venda-$this->preco_custo) / $this->preco_custo) * 100;
    }
}
```

Data Mapper

Uma classe que persiste um objeto ou conjunto de objetos. Possui um método que recebe o objeto a ser persistido, investiga sobre ele, e grava na base de dados.

```
<?php
class VendaMapper {

    public static function save(Venda $venda) {
        $date = date("Y-m-d");

        $sql = "INSERT INTO venda (data_venda) values ('$date')";
        foreach ($venda->getItens() as $item) {
            ....
        }
    }
}
```

Layer supertype

Uma classe superpai de toda uma camada da aplicação. Neste exemplo, todas classes de modelo são filhas da classe Record.

```
<?php
class Produto extends Record {
    const TABLENAME = 'produto';
}

$p1 = new Produto;
$p1->descricao = 'Cerveja artesanal IPA';
$p1->estoque = 50;
$p1->store();
```

Query object

Um objeto ou estrutura de objetos que representa um critério de filtro que será utilizado como filtro de uma consulta.

```
<?php
$criteria = new Criteria;
$criteria->add('idade', '<', 16);
$criteria->add('idade', '>', 60, 'or');
```

Collections

Trabalhando com coleções de objetos. Neste exemplo, utilizamos o pattern Repository para carregar objetos de uma classe (Produto) que satisfazem um critério (Criteria).

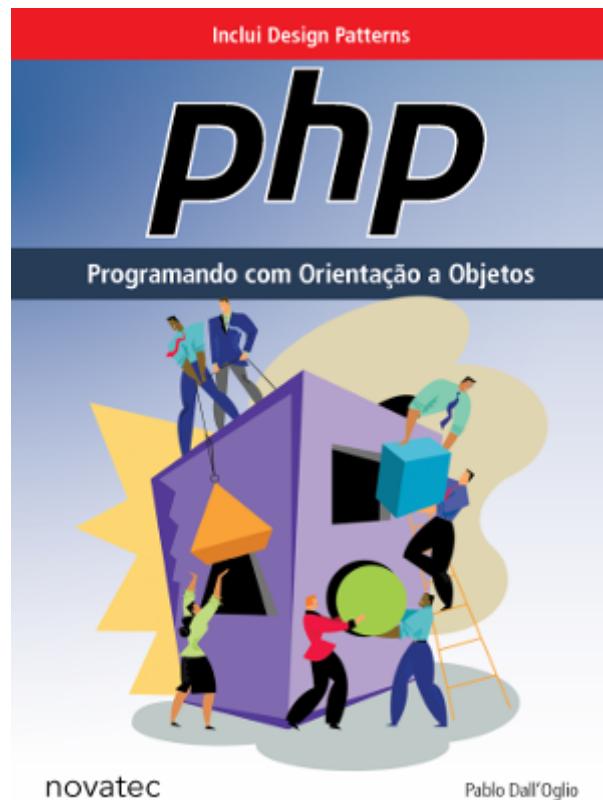
```
<?php
try {
    Transaction::open('estoque');

    $criteria = new Criteria;
    $criteria->add('estoque', '>', 10);
    $criteria->add('origem', '=', 'N');

    $repository = new Repository('Produto');
$produtos = $repository->load($criteria);

    if ($produtos) {
        echo "Produtos <br>\n";
        foreach ($produtos as $produto) {
            echo ' ID: ' . $produto->id;
        }
    }
}
```

Livro PHP Programando com Orientação a Objetos



www.adianti.com.br/phpoo