

Orientação a Objetos

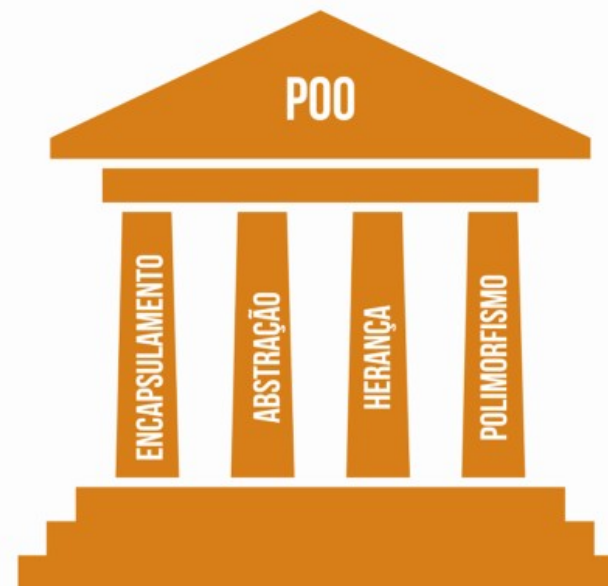
Prof. Daniel Di Domenico

Persistência de dados



O que já sabemos?

- Linguagem Java
- Orientação a objetos
 - Abstração
 - **Classes e Objetos**
 - Atributos e Métodos
 - Listas
 - **Encapsulamento**
 - **Associação**
 - **Herança**
 - **Polimorfismo**



Objetivos da aula

- **Persistir objetos em base de dados**
 - Salvar os objetos de forma persistente
 - Manter os objetos acessíveis após finalizar a execução do programa
 - Os objetos poderão ser buscados quando necessário

Requisitos de software

- PHP
 - Conexão ao banco de dados MySQL habilitada
- Banco de dados MySQL
- Cliente do banco de dados (phpMyAdmin ou MySQLWorkbench)

PHP com banco de dados

- Conectar uma aplicação PHP ao banco de dados
 - Utilizar a classe de conexão PDO
 - Gravar e buscar os dados de uma base



PHP com banco de dados

- Utilizaremos a classe **PDO** para conectar ao MySQL (PDO tem suporte a outros bancos)
 - https://www.php.net/manual/pt_BR/class.pdo.php
- Exemplo para criar uma conexão:

```
$opcoes = array(<...OPÇÕES>);  
  
$conn =  
    new PDO("mysql:host=XXXX;dbname=XXX",  
            "user", "pass", $opcoes);
```

PHP com banco de dados

- Exemplo de **opções** da conexão PDO:

```
$opcoes =  
    array(  
        //Define o charset da conexão  
        PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",  
        //Define o tipo do erro como exceção  
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,  
        //Define o tipo do retorno das consultas  
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC  
    );
```

Persistência: conexão

- Acesso ao banco de dados com PDO:
 - **1-** Criar uma classe que retorna a conexão com o banco (objeto PDO)
 - **2-** Incluir o arquivo da conexão onde for necessário acesso ao banco de dados

```
//Arquivo Connection.php
class Connection {

    private static $conn = null;

    public static
    function getConnection() {

        if(self::$conn == null) {
            //Cria a conexão...
        }
        return self::$conn;
    }
}
```

require_once: adiciona um arquivo .php na página

```
//Arquivo que utilizará a conexão

//Adiciona o arquivo Connection.php
require_once("Connection.php");

//Chama o método que cria a conexão
$conn = Connection::getConnection();
```


Persistência: buscar

- Exemplo de código para **buscar** registros da base de dados

```
//Obtém a conexão já implementada
$conn = Connection::getConnection();

$sql = "SELECT * FROM exemplo";

//Prepara e executa o comando SQL
$stmt = $conn->prepare($sql);
$stmt->execute();

//Armazena os resultados ($result é uma matriz)
$result = $stmt->fetchAll();
```

Persistência: buscar

- Retornando os dados da consulta

```
//... continuação do slide anterior  
  
//Acesso aos registros da matriz $result  
foreach ($result as $reg) {  
    echo $reg['coluna1'];  
    echo ' - ';  
    echo $reg['coluna2'];  
    echo '<br>';  
}
```

\$result é um *array* indexado
que possui um *array*
associativo dentro dele

\$reg é um *array* associativo.
As **chaves** são os **nomes das**
colunas na base de dados

Persistência: retorno dos dados

Tabela **peessoas**

id	nome	idade
1	Juarez	17
2	Sônia	32



Os dados da tabela na base de dados são convertidos para uma matriz no PHP

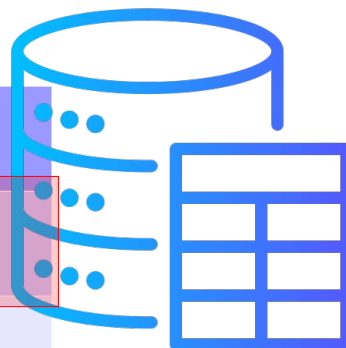


```
Array(  
  [0] => Array([ 'id' ] => 1,  
                [ 'nome' ] => 'Juarez',  
                [ 'idade' ] => 17),  
  
  [1] => Array([ 'id' ] => 2,  
                [ 'nome' ] => 'Sônia',  
                [ 'idade' ] => 32)  
)
```

Persistência: retorno dos dados

Tabela **pessoas**

id	nome	idade
1	Juarez	17
2	Sônia	32



Os dados da tabela na base de dados são convertidos para uma matriz no PHP



Array(

```
[0] => Array(['id'] => 1,  
              ['nome'] => 'Juarez',  
              ['idade'] => 17),
```

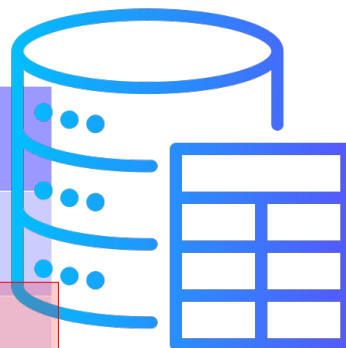
```
[1] => Array(['id'] => 2,  
              ['nome'] => 'Sônia',  
              ['idade'] => 32)
```

)

Persistência: retorno dos dados

Tabela **peessoas**

id	nome	idade
1	Juarez	17
2	Sônia	32



Os dados da tabela na base de dados são convertidos para uma matriz no PHP



```
Array(  
  [0] => Array([ 'id' ] => 1,  
                [ 'nome' ] => 'Juarez',  
                [ 'idade' ] => 17),  
  
  [1] => Array([ 'id' ] => 2,  
                [ 'nome' ] => 'Sônia',  
                [ 'idade' ] => 32)  
)
```

Persistência: inserir

- Exemplo de código para **inserir** um registro na base de dados

```
//Obtém a conexão já implementada
$conn = Connection::getConnection();

$sql = "INSERT/UPDATE/DELETE... <?, ?, ?...>";

$stmt = $conn->prepare($sql);
$stmt->execute(...PARAMETROS);
```

Array de parâmetros:
deve ser passado um valor
para cada ? do SQL

Parâmetros da
instrução SQL

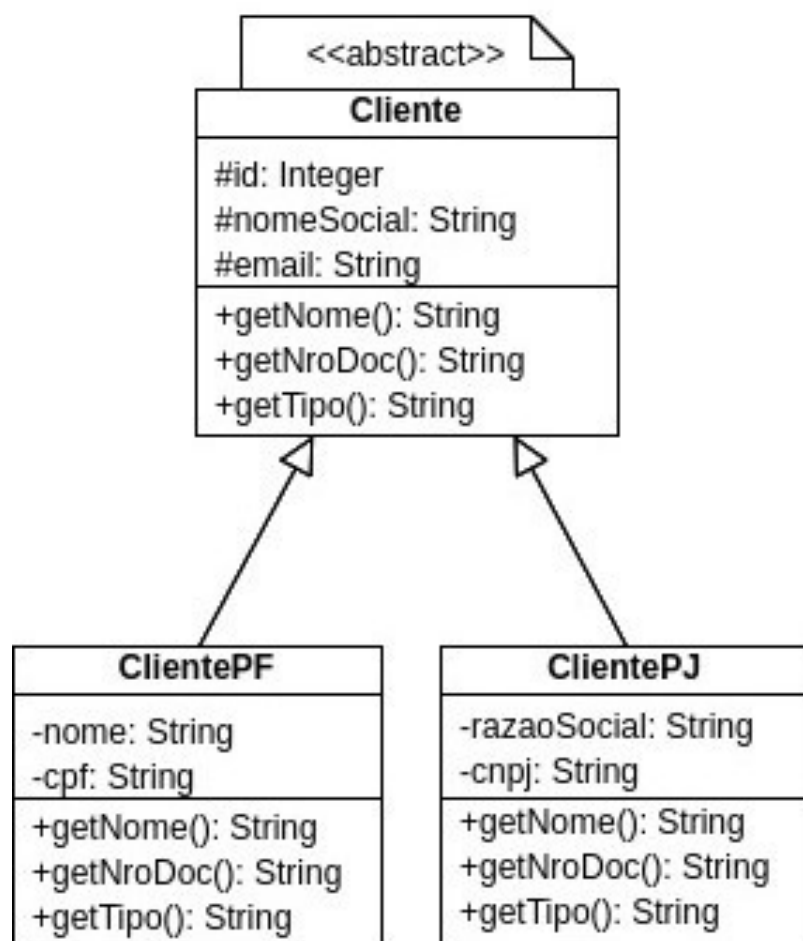
Diagramas propostos



INSTITUTO
FEDERAL

Paraná

Campus
Foz do Iguaçu



teste clientes	
id	int
tipo	varchar(1)
nome_social	varchar(70)
email	varchar(70)
nome	varchar(70)
cpf	varchar(30)
razao_social	varchar(70)
cnpj	varchar(30)