



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

# Disciplina: Desenvolvimento Web III

Prof. Risiberg Teixeira  
email: [risiberg.teixeira@ifrj.edu.br](mailto:risiberg.teixeira@ifrj.edu.br)

# POO – private e protect (1)

```
<?php
class classe1 {
    private $var1 = "Olá, var1!\n"; // ver na classe
    protected $var2 = "Olá, var2!\n"; // ver na outra
    classe
    protected $var3 = "Olá, var3!\n"; // ver na outra
    classe
    function bomdia(){
        echo ".....Classe 1.....<br/>";
        print "Classe1: " . $this->var1 . "<br/>";
        print "Classe2: " . $this->var2 . "<br/>";
        print "Classe3: " . $this->var3 . "<br/>";
    }
}
```

# POO – private e protect (2)

```
class classe2 extends classe1 {  
    function bomdia(){  
        echo ".....Classe 2 herda.....<br/>";  
        print "Classe2: " . $this->var2 . "<br/>";  
        print "classe2: " . $this->var3 . "<br/>";  
    }  
}  
$obj1 = new classe1();  
$obj1->bomdia();  
$obj2 = new classe2();  
$obj2->bomdia();  
?>
```

# POO – Métodos privados e protegidos

```
<?php
class classe1 {
    private function metodoPrivado() {
        echo "Classe1::metodoPrivado() chamado. <br/>";
    }
    protected function metodoProtegido(){
        echo "Classe1::metodoProtegido() chamado.<br/>";
        $this->metodoPrivado();
    }
}
class classe2 extends classe1 {
    public function metodoPublico(){
        echo "classe2::metodoPublico() chamado.<br/>";
        $this->metodoProtegido();
    }
}
$obj = new classe2();
$obj->metodoPublico();
?>
```

# POO – método abstrato

```
<?php
    abstract class ClasseAbstrata {
        abstract public function teste();
    }
    class ClasseImplementa extends ClasseAbstrata{
        public function teste() {
            echo "Metodo teste() chamado. <br/>";
        }
    }
    $obj = new ClasseImplementa();
    $obj->teste();
?>
```

# POO – Variáveis e métodos estáticos

```
<?php
    Class Classe {
        static $variavel_estatica = 10;
        function soma ($numero) {
            Classe::$variavel_estatica += $numero;
        }
    }
    $obj = new Classe();
    $obj->soma(2);
    $obj2 = new Classe();
    $obj2->soma(2);
    print Classe::$variavel_estatica;
?>
```

# Acesso a dados por método

```
<?php
class Carro {
    protected $chassi;
    function __construct($chassi) {
        $this->chassi = $chassi;
        echo "Objeto carro construído<br />";
    }
    public function getChassi() {
        return $this->chassi; }
    }
    $c1 = new Carro("ABC1235");
    echo "[VIA MÉTODO] o chassi é: " .
        $c1->getChassi();
```

?>