

# DESENVOLVIMENTO PARA WEB II

## Conceitos PHP e Prog. para Web



Professor: Alexandre Strapção Guedes Vianna – IFPE Igarassu  
[alexandrevianna.net](http://alexandrevianna.net)

# Roteiro da Aula

1. Apresentar os principais aspectos da linguagem PHP para o contexto da disciplina
  - Orientação a Objetos
  - Composer
  - Namespaces
  - Associative Arrays
  - Variáveis de Sessão
2. Apresentar os conceitos essenciais de programação para Web
  - Protocolo HTTP
  - Padrão MVC
  - Segurança: Entrada de Dados e Senhas
  - Codificação e Padrões: Datas e Horários
  - Banco de Dados



# Framework ?

Relembrando...



- Padronização
- **Velocidade de Desenvolvimento**
- Qualidade
- Performance
- Segurança
- Manutenção
- Comunidade

# Framework ?

## Relembrando...

Linguagem Java: JSF, Hibernate, Spring,...



Linguagem Ruby: Rails



Linguagem Python: Django



Linguagem JavaScript: Node.js



Linguagem C#: DotNet



Linguagem PHP: CakePHP



# PHP



## Linguagem PHP (Hypertext Preprocesso)

O PHP é uma linguagem de script open source de uso geral, muito popular, e especialmente adequada para o desenvolvimento web server-side.

A linguagem tem evoluído nos últimos anos graças a forte interação da comunidade e de grandes players como o Facebook, e tem melhorado em funcionalidades e performance. Aplicações:

- Scripts no lado do servidor (server-side)
- Scripts de linha de comando
- Escrever aplicações desktop.

# PHP



## **Sobre PHP** (*Versão Estável 7.0*)

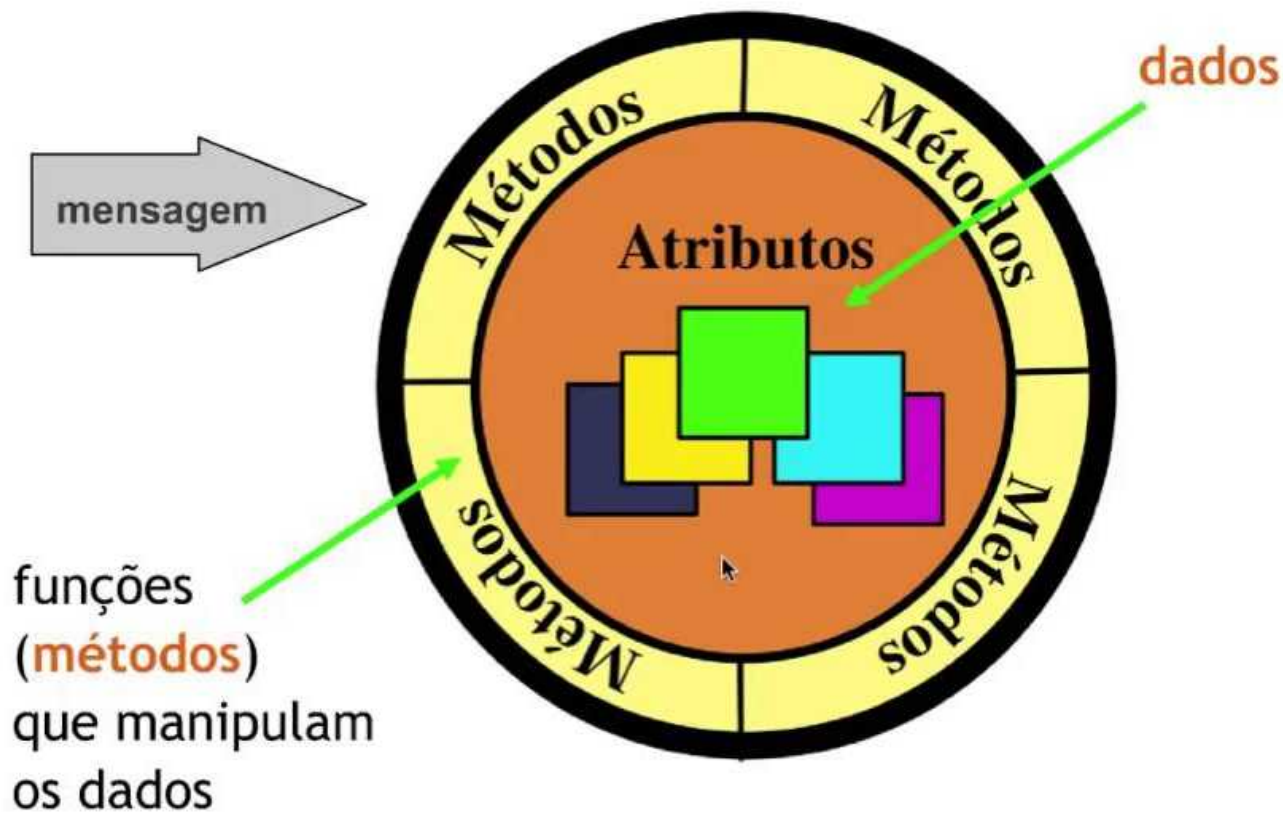
A constante evolução da linguagem, as radicais transformações e os novos recursos de PHP, a tornou a Internet um baú de documentação desatualizada e inapropriada.

Procurem sempre referências confiáveis e atuais:

<http://br.phptherightway.com/> PHP: Do Jeito Certo é uma referência rápida e fácil de ler, introduzindo desenvolvedores às melhores práticas, renomados padrões de código e links para tutoriais .

[http://php.net/manual/pt\\_BR/langref.php](http://php.net/manual/pt_BR/langref.php) Referência da API PHP

# Encapsulamento



# Orientação a Objetos

**Vantagens Diretas** – maior facilidade para reutilização de código e por consequência do projeto – possibilidade do desenvolvedor trabalhar em um nível mais elevado de abstração – utilização de um único padrão conceitual durante todo o processo de criação de software – maior adequação à arquitetura cliente/servidor – maior facilidade de comunicação com os usuários e com outros profissionais de informática.

**Vantagens Reais** – ciclo de vida mais longo para os sistemas – desenvolvimento acelerado de sistemas – possibilidade de se construir sistema muito mais complexos, pela incorporação de funções prontas – menor custo para desenvolvimento e manutenção de sistemas



# Programando OO em PHP

As classes são orientadas ao assunto, ou seja, cada classe é responsável por um assunto diferente e possui responsabilidade sobre o mesmo. Ela deve proteger o acesso ao seu conteúdo por meio de mecanismos como o de encapsulamento.

```
<?php  
    class Usuario{  
  
        }  
    ?>
```

# Exemplo completo

```
<?php
class Usuario{
    private $nome;
    private $cpf;
    private $endereco;

    function Usuario($nome,$cpf,$endereco){
        $this->nome = $nome;
        $this->cpf = $cpf;
        $this->endereco = $endereco;
    }
}
?>
```

# Programando OO em PHP

**Instanciando:** criando objetos de uma classe

```
$fulano = new Usuario("Fulando da silva",  
123456, "Avenida da Felicidade, 13, Terra do  
Nunca");
```

# Programando OO em PHP

**Herança:** A seguir veremos um exemplo prático de herança. Herança é a possibilidade de classes e objetos compartilharem funções e atributos. Por exemplo, podemos criar uma classe chamada professor, que é um tipo de usuário, porém possui algumas particularidades, como atributos área e siape.

Para acessar os métodos sobrescritos ou membros estáticos da classe pai, utilizamos a palavra reservada ***parent***.

```
require_once 'usuario.php';  
class Professor extends Usuario{  
    private $siape;  
    private $area;  
  
    function Professor($nome, $cpf, $endereco, $area) {  
        parent::Usuario($nome, $cpf, $endereco);  
        $this->area = $area;  
    }  
    public function getArea(){  
        return $this->area;  
    }  
    public function setArea($area){  
        $this->$area = $area;  
    }  
}
```

# PHP: Composer

## Composer:



O Composer é um gerenciador de dependências brilhante para o PHP. Liste as dependências do seu projeto em um arquivo `composer.json` e, com poucos comandos simples, o Composer irá fazer o download das dependências do seu projeto automaticamente e configurará o autoloading para você.

Dependency Manager for PHP

Já existem várias bibliotecas PHP que são compatíveis com o Composer, prontas para usar no seu projeto. Esses “pacotes” estão listados no Packagist, o repositório oficial das bibliotecas PHP compatíveis com o Composer.

# PHP: Composer

## Composer, como utilizar:



Primeiro você precisa instalar o composer, siga as instruções disponíveis em:

[http://br.phptherightway.com/#composer\\_e\\_packagist](http://br.phptherightway.com/#composer_e_packagist)

O Composer mantém o controle de dependências do seu projeto em um arquivo chamado composer.json.

Dependency Manager for PHP

Você pode controlá-lo na mão se preferir ou usar o próprio Composer. O comando composer require adiciona uma dependência do projeto e se você não tem um arquivo composer.json, ele será criado.

# PHP: Namespaces

## Namespaces PHP

- Os namespaces ajudam a organizar o código fonte de projetos em uma estrutura de diretórios, que devem ser declarados e importados com o comando **use**.
- Os namespaces de PHP são semelhantes aos pacotes de Java
- Ajudam a evitar colisão de nomes, por exemplo, pode haver mais de uma classe Calculadora em um projeto, porém são calculadoras diferentes.



# PHP: Namespaces

The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer displays a project structure with folders like 'examples' and 'src', and files like 'autoload.php' and 'Calculator.php'. The code editor shows PHP code using namespaces. Red arrows point from the file explorer to the code editor, highlighting the relationship between the file structure and the code.

**File Explorer Structure:**

- distical
  - examples
  - src
    - Distical
      - Calculator.php
    - .gitignore
    - README.md
    - composer.json
  - monolog
  - psr
  - autoload.php
  - .DS\_Store
  - composer.json
  - composer.lock
  - index.php
  - serialization.php
  - store

**Code Editor Content:**

```
2
3 /**
4  * Name space demo by Bobby Allen, 22nd March 2012
5  */
6
7
8
9
10
11 include 'vendor/autoload.php';
12
13
14 use nlogphp\logger\logr;
15 use Distical\Calculator;
16
17 $mylogger = new logr();
18
19 $an_eample_name = 'Mikey Mouse!';
20 $mylogger->setMyName($an_eample_name);
21
22 echo "Name has been set to: " . $mylogger->getMyName();
23
```

**Annotations:**

- The `autoload.php` file is automatically generated by Composer, all you need to do is add this line to your application!
- Using the namespaces translates to the folder structure under the 'src' folder in the package directory.

# PHP: Namespaces

## Declarando um namespaces PHP

```
<?php  
namespace MeuProjeto\Component\LeitorPdf;
```

- Sempre aparece na linha seguinte após abrir a tag do php <?php
- Por questão de organização, declare apenas 1 namespace por arquivo

# PHP: Namespaces

## Usando namespaces PHP

```
use MeuProjeto\Component\LeitorPdf;
```

- Basta usar a palavra use e o caminho do namespace
- Você poderá usar neste arquivo todas as classes que fazem parte do namespace

# PHP: Associative Arrays

Array comum:

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and
" . $cars[2] . ".";
?>
```

# PHP: Associative Arrays

## Associative Arrays:

- São arrays que utiliza-se nomes como como indices associados aos elementos

Exemplo 1:

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

Exemplo 2:

```
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```

# PHP: Associative Arrays

## Associative Arrays:

- Os índices com nomes podem ser utilizados assim:

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
echo "Peter is " . $age['Peter'] . " years old.";
```

# PHP: Associative Arrays

## Associative Arrays:

- Interagindo Associative Arrays:

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
foreach($age as $x => $x_value) {  
    echo "Key=" . $x . ", Value=" . $x_value;  
    echo "<br>";  
}
```

# PHP: Variáveis Session

- Sessão é um recurso do PHP que permite que você salve valores (variáveis) para serem usados ao longo da visita do usuário. Valores salvos na sessão podem ser usados em qualquer parte do script, mesmo em outras páginas do site. São variáveis que permanecem setadas até o visitante fechar o browser ou a sessão ser destruída.

```
<?php
```

```
session_start(); // Inicia a sessão
```

```
$_SESSION['usuario'] = 'Fulano'; // cria uma variável de sessão
```

```
unset($_SESSION['usuario']); // Deleta uma variável da sessão
```



# PHP: Variáveis Session

- Destruir sessão:

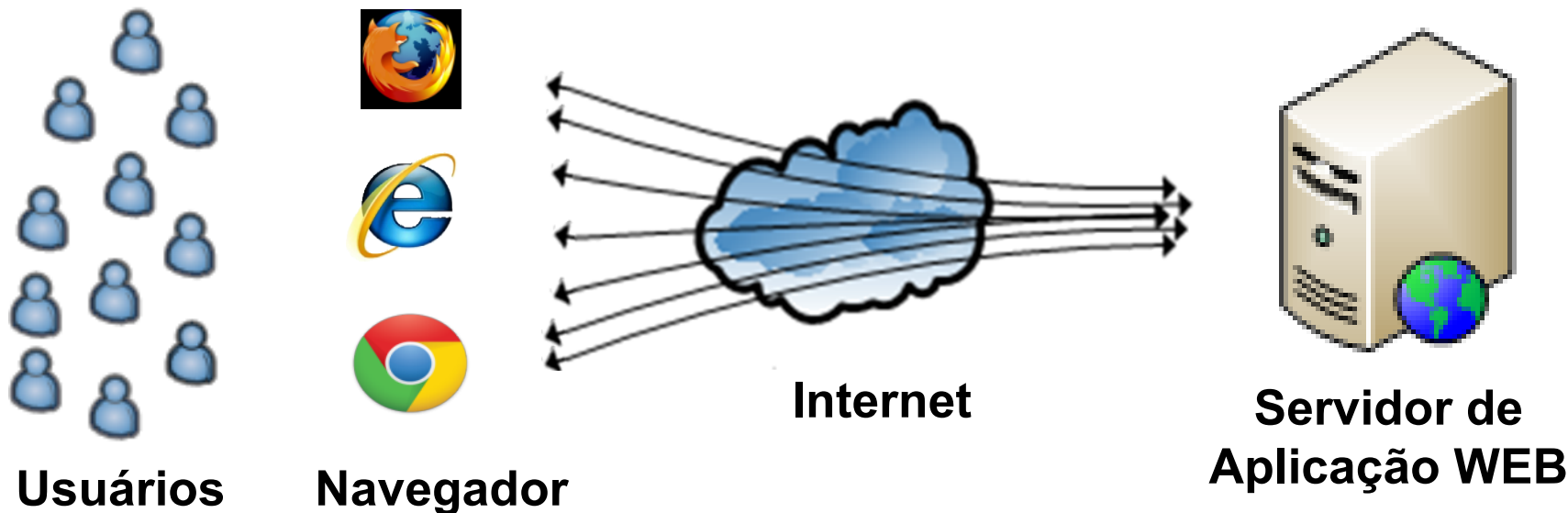
Você também pode apagar todos os dados armazenados em uma sessão

```
<?php  
session_destroy(); // Destrói toda sessão
```

# Programação para Web

## Aplicação Web:

Sistema projetado para a utilização através do navegador Web (Firefox, Internet Explorer, Chrome entre outros)



# Protocolo HTTP

## Protocolo HTTP, definição:

*“Hypertext Transfer Protocol (HTTP) é o método utilizado para transferência de dados de páginas na web entre aplicações clientes (browsers) e servidores.”*

Ex.: hipertexto (HTML),  
imagens, videos e outros

Este protocolo é usado desde 1990

# Protocolo HTTP

**Como funciona:**

Existe uma  
página WEB

Como a página que  
está na internet vai  
chegar até o meu  
computador ??



**Página na WEB**



Existe usuário que  
quer acessar a  
página!

**Usuário**

# Protocolo HTTP

## Como funciona:

A página está armazenada no Servidor conectado a internet



Servidor

A página é constituída de HTML + Imagens



Usuário



Página na WEB



HTML + arquivos associados

# Protocolo HTTP

Como funciona:



Browser



Servidor



Usuário

Usuário utiliza um  
browser para  
acessar a página



Página na WEB



HTML + arquivos associados

# Protocolo HTTP

Como funciona:



Browser

Browser solicita a página



Servidor

Servidor envia a página e todos os arquivos associados



Usuário



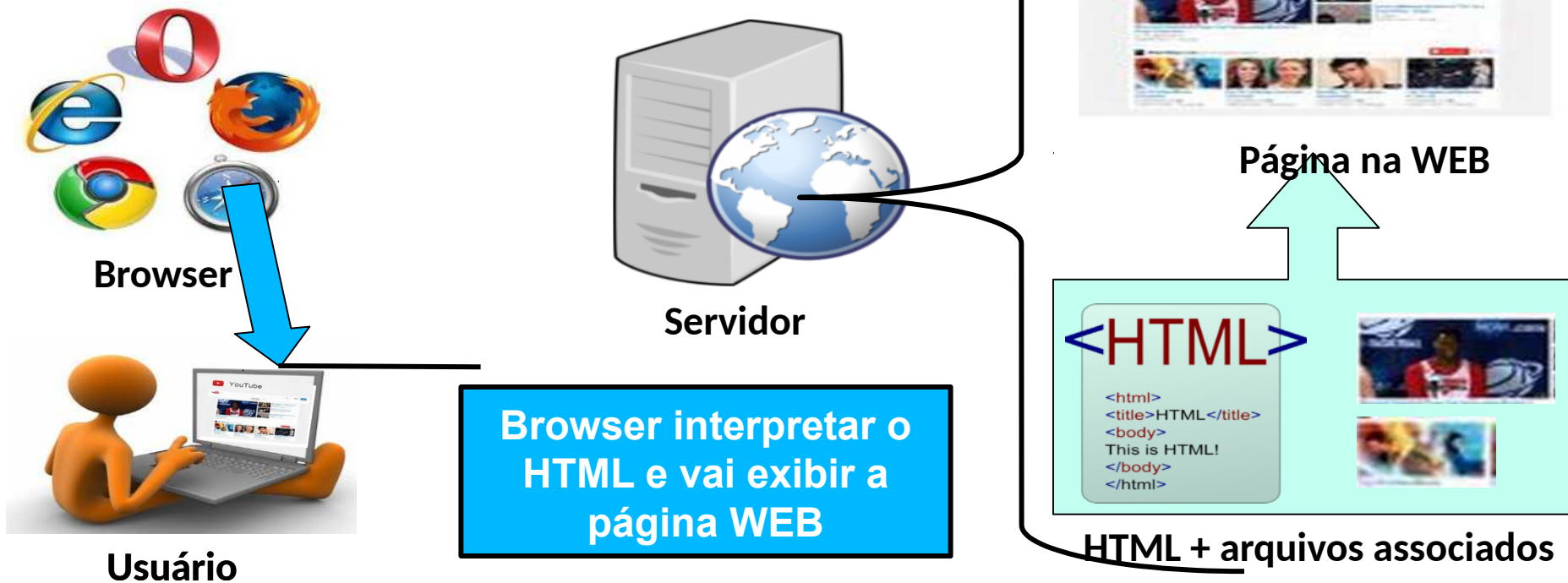
Página na WEB



HTML + arquivos associados

# Protocolo HTTP

Como funciona:





# Protocolo HTTP

## Recapitulando...

1. Usuário utiliza o browser e entra com o endereço da página.
2. Browser solicita ao servidor os dados da página
3. Servidor envia dados da página. (HTML + imagens, vídeos e outros)
4. Browser interpreta HTML e gera uma visualização da página para o usuário

# Programação para Web:MVC

E quando estamos tratando de software?



Também temos técnicas, boas práticas e processos estabelecidos e ao longo da evolução da Tecnologia de Informação

Um exemplo é padrão de projeto MVC – Model View Controller (Model Visão Controle)

# MVC – Model View Controller

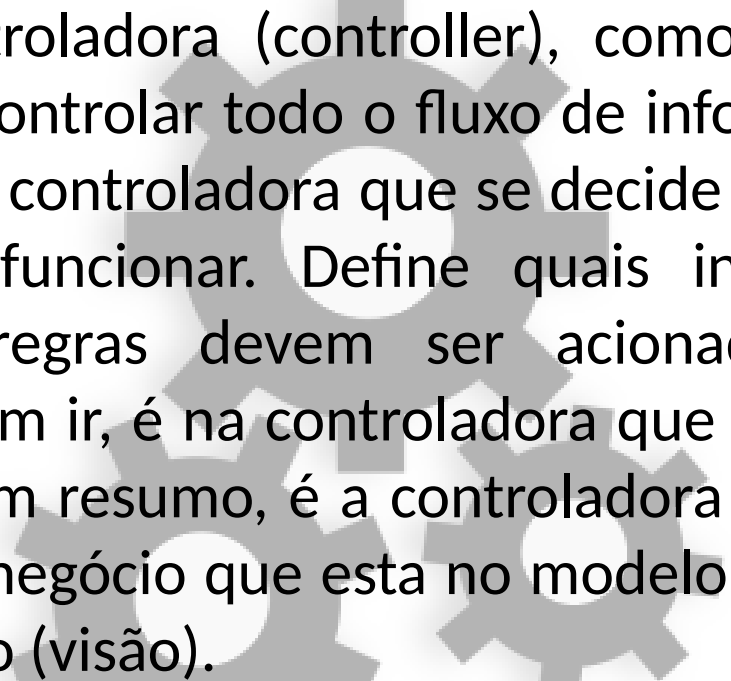
**Visão:** A visão (view) é responsável por tudo que o usuário final interage, toda a interface, informação, não importando sua fonte de origem, é disponibilizada ao usuário graças a camada de visão.



**Visualização**

# MVC – Model View Controller

**Controle:** A Controladora (controller), como o nome já sugere, é responsável por controlar todo o fluxo de informação que passa pelo site/sistema. É na controladora que se decide “se”, “o que”, “quando” e “onde” deve funcionar. Define quais informações devem ser geradas, quais regras devem ser acionadas e para onde as informações devem ir, é na controladora que essas operações devem ser executadas. Em resumo, é a controladora que solicita a execução de uma regra de negócio que está no modelo e repassa a informação para a visualização (visão).



# MVC – Model View Controller

**Modelo:** O modelo (Model) é utilizado para manipular informações de forma mais detalhada, sendo recomendado que, sempre que possível, se utilize dos modelos para realizar consultas, cálculos e todas as regras de negócio do nosso site ou sistema. É o modelo que tem acesso a toda e qualquer informação sendo essa vinda de um banco de dados, arquivo XML.



# Programação Web: Segurança

## Sanitizar entradas:

- Todas as entradas de dados do seu sistema devem ser tratadas com objetivo de evitar um ataque com a inserção de um script malicioso. Exemplo: SQLInjection
- Validar dados evitando causar erros na aplicação com a inserção de dados mal formatados ou invalidos
- Validação de informações cadastrais: e-mail, CPF...

# Programação Web: Segurança

## Senhas:

A segurança das senhas é extremamente importante!

- Jamais tome conhecimento das senhas dos usuário
- Jamais restrinja as senhas dos usuários, exija apenas um tamanho mínimo
- Colocar senhas comuns ou fáceis na lista negra. Ex.: 123456, abcdef
- Jamais envie senhas para os usuários
- Gere hash das senhas de usuários com o bcrypt (*hash e criptografica são diferentes*)
- Mantenha as credenciais do seu banco de dados secretas

# Programação Web: Padrões

## Datas, horas e fusos horários:

- Formatar e converte utilizando os recursos disponibilizados pelas classes: DateTime, DateIntervale DateTimeZone

Configure timezone no arquivo php.ini

```
Date.timezone = 'America/Recife' ;
```

Ou em tempo de execução

```
<?php  
date_default_timezone_set('America/Recife') ;
```



# Programação Web: Codificação

*Uma codificação de caracteres é um método de empacotar os dados Unicode em uma codificação de caracteres que possa ser armazenado em memória ou enviado pela rede entre um computador e outro.*

- Existem diversos tipos de codificação de caracteres, a UTF-8 é a codificação mais popular e é suportada por todos os browsers.
- **Utilize UTF-8!**

# Programação Web: Banco de Dados

## **Banco de dados:**

Responsável pela persistência dos dados da aplicação

## **Extensão PDO: PHP Data Objects**

- É uma coleção de classes que se comunica com o banco de dados por meio de uma única interface.
- As implementações dos banco de dados são abstraídas.

# Roteiro da Aula

1. Apresentar os principais aspectos da linguagem PHP para o contexto da disciplina
  - Orientação a Objetos
  - Composer
  - Namespaces
  - Associative Arrays
  - Variáveis de Sessão
2. Apresentar os conceitos essenciais de programação para Web
  - Protocolo HTTP
  - Padrão MVC
  - Segurança: Entrada de Dados e Senhas
  - Codificação e Padrões: Datas e Horários
  - Banco de Dados



# **DESENVOLVIMENTO PARA WEB II**

## **Conceitos PHP e Prog. para Web**



IPI – Desenvolvimento para WEB II – IFPE Igarassu  
Professor: Alexandre Strapação Guedes Vianna