



# 4LINUX

OPEN SOFTWARE SPECIALISTS



**ESPECIALISTA EM "JUNTAR AS PEÇAS" DO MUNDO OPEN SOURCE**

[WWW.4LINUX.COM.BR](http://WWW.4LINUX.COM.BR)

# Manipulação de Data e Hora

---



# Objetivos da Aula

---

- Funções nativas do PHP;
- Classe DateTime;
- Configuração de data e hora.

# Manipulação de Data e Hora

---

## Funções nativas

- A linguagem possui várias funções nativas, como já conhecemos nos cursos anteriores;
- Algumas delas são voltadas à manipulação de data e hora;
- Nos slides a seguir vamos conhecer as principais funções relacionadas à manipulação de data e hora.

# Manipulação de Data e Hora

---

## Funções nativas

### Função `date( <formato> [, <timestamp> ] )`

- Esta função permite que você crie uma string de data ou formate uma data com base em um timestamp UNIX.

### Parâmetros da função

- **<formato>**: string com o formato que a data/hora deverá ser retornada;
- **<timestamp>**: parâmetro não obrigatório que, se informado, servirá de base para a data gerada.

# Manipulação de Data e Hora

---

## Funções nativas

### Função time()

- A função retorna a hora atual em segundos desde a Era UNIX (1º de Janeiro de 1970 00:00:00 GMT).

# Manipulação de Data e Hora

---

## Funções nativas

### Função `microtime(<get_as_float>)`

- A função retorna um TimeStamp Unix da hora atual em microssegundos;
- O valor informado é baseado na Era UNIX (1º de Janeiro de 1970 00:00:00 GMT).

### Parâmetros da função

- **`<get_as_float>`**: Se definido como TRUE então o retorno terá um float dos segundos.



# Manipulação de Data e Hora

---

## Funções nativas

### Função `strptime( [<formato>] )`

- A função converte uma string passada como parâmetro em um timestamp Unix. Quando omitido o parâmetro, a conversão é realizada baseado na data e hora atual.

### Parâmetros da função

- **<formato>**: string com o formato de data em inglês. Ex:
  - “now”: data e hora atual;
  - “next sunday”: próximo domingo, baseado na data atual.

# Manipulação de Data e Hora

---

## Funções nativas

**Função mktime( <hora>, <min>, <seg>, <mes>, <dia>, <ano> )**

- A função retorna um timestamp Unix baseado nos parâmetros.

### Parâmetros da função

- **<hora>**: inteiro, equivalente a hora;
- **<min>**: inteiro, equivalente ao minuto;
- **<seg>**: inteiro, equivalente a segundo;
- **<mes>**: inteiro, equivalente a mês;
- **<dia>**: inteiro, equivalente ao dia;
- **<ano>**: inteiro, equivalente a ano;

# Manipulação de Data e Hora

---

## DateTime

### Introdução

- A classe nativa DateTime permite que nós trabalhemos com data e hora, como se fossem objetos que podem ser facilmente manipulados, através de diversos métodos;
- Ao criar um objeto da classe DateTime será mais fácil de manipular datas e horários na sua aplicação.

# Manipulação de Data e Hora

---

## DateTime

### Criando um objeto da classe DateTime( <data\_hora> )

- A criação do objeto DateTime é simples basta que você instancie o objeto DateTime. Quando omitido o parâmetro do construtor, o objeto é gerado a partir da data e hora atual do sistema.

### Parâmetros da função

- **<data\_hora>**: string contendo uma data e hora válida, por exemplo: '2013-09-20 13:00:00'.

# Manipulação de Data e Hora

---

## DateTime

### Modificando o valor do objeto

- Podemos modificar os valores de data e hora armazenados no objeto; para isso temos um método chamado `'modify( <string> )'`;

### Parâmetros do método

- **<string>**: string com o formato de data em inglês. Ex:
  - “now”: data e hora atual;
  - “next sunday”: próximo domingo, baseado na data atual.

# Manipulação de Data e Hora

---

## DateTime

### Ajustando a data

- Podemos alterar a data do objeto através do método 'setDate( <ano>, <mes>, <dia>)'.

### Parâmetros do método

- **<ano>**: inteiro, equivalente a ano;
- **<mes>**: inteiro, equivalente a mês;
- **<dia>**: inteiro, equivalente ao dia.

# Manipulação de Data e Hora

---

## DateTime

### Ajustando a hora

- Podemos alterar a hora do objeto através do método 'setTime( <hora>, <minuto>, <segundo>)'.

### Parâmetros do método

- **<hora>**: inteiro, equivalente a hora;
- **<minuto>**: inteiro, equivalente a minuto;
- **<segundo>**: inteiro, equivalente a segundos.

# Manipulação de Data e Hora

---

## DateTime

### Ajustando o fuso horário

- Toda vez que quisermos ajustar o fuso horário de um objeto de data, precisamos criar um objeto da classe `DateTimeZone`;
- Um objeto desta classe representa um fuso horário válido e pode ser utilizado junto ao método `setTimezone()` da classe `DateTime`.



# Manipulação de Data e Hora

---

## DateTime

### Intervalos

- Objetos da classe `DateInterval` representam um intervalo entre datas. Este intervalo pode armazenar um tempo (em anos, meses, dias ou horas);
- Podemos criar intervalos utilizando um padrão que será passado ao método construtor da classe `DateInterval`. O padrão começa com a letra `P`, de período.

# Manipulação de Data e Hora

---

## DateTime

### Comparando datas

- Uma das grandes vantagens de trabalhar com objetos da classe DateTime é que podemos compará-los utilizando operadores comuns (==, <, >, <=, >= e !=), sem a necessidade de fazer conversões da data para Timestamp.

# Manipulação de Data e Hora

---

## DateTime

### Diferença entre datas

- Dentro do objeto da classe DateTime existe um método chamado **diff()**;
- O parâmetro utilizado no método precisa ser, obrigatoriamente, um objeto da classe DateTime, quando utilizado retorna um objeto da classe DateInterval com a diferença existente entre os objetos da DateTime.

# Manipulação de Data e Hora

---

## DateTime

### Somando intervalo a um data

- Dentro do objeto da classe DateTime existe um método chamado add(), que permite acrescentar um período de tempo ao objeto DateTime existente;
- O parâmetro utilizado no método precisa ser obrigatoriamente um objeto da classe Dateinterval.

# Manipulação de Data e Hora

---

## DateTime

### Subtraindo intervalo a um data

- De forma semelhante à adição, temos um método chamado `sub()`, que permite subtrair um período de tempo ao objeto `DateTime` existente;
- O parâmetro utilizado no método precisa ser obrigatoriamente um objeto da classe `DateInterval`.

# Manipulação de Data e Hora

---

## Configuração

- Podemos configurar nossas aplicações para gerenciar data e hora da maneira que desejarmos;
- Essas configurações geralmente envolvem o ajuste de fuso horário ou o uso de datas no idioma de nossa aplicação.

# Manipulação de Data e Hora

---

## Configuração

### Fuso horário

- Por padrão, o PHP utiliza sempre como fuso horário padrão o do sistema operacional;
- Isso pode ser um problema quando você tem múltiplas aplicações rodando em um mesmo servidor que podem possuir um fuso horário diferente;
- Para especificar a timezone correta para cada aplicação, utilizamos a função `date_default_timezone_set()`.

# Manipulação de Data e Hora

---

## Configuração

### Fuso horário

- Há momentos em que, ao imprimir datas na aplicação, o nome dos meses ou dias de semana precisam vir em português ou em outro idioma;
- Para fazer isso, o comportamento de nossa aplicação deve mudar um pouco. Não poderemos mais utilizar a função `date()`, e sim a função `strftime()`.



# Manipulação de Arquivo

---



# Objetivos da Aula

---

- Streams;
- Acesso, leitura e escrita de arquivos;
- Sistemas de arquivos;
- Locks;
- Wrappers;
- Upload;

# Manipulação de Arquivos

---

## Acessando um arquivo

- Antes de extrair, adicionar ou remover o conteúdo de um arquivo, precisamos acessá-lo;
- Este tipo de acesso é conhecido como **stream**;
- Quando falamos de stream estamos falando de algo que pode ser lido/escrito;

# Manipulação de Arquivos

---

## Acessando um arquivo

### Função fopen ( <arquivo> , <modo de acesso> )

- Utilizamos esta função para criar um stream com um arquivo, sua execução retorna um recurso correspondente a um stream.

### Parâmetros da função

- **<arquivos>**: string com o caminho do arquivo que será manipulado;
- **<modo de acesso>**: string, determina o que o PHP poderá fazer com o arquivo a ser acessado.

# Manipulação de Arquivos

---

## Leitura

**Função fread( <resource> , <quantidade byte a ser lido> )**

- Utilizamos esta função para ler parte do arquivo.

## Parâmetros da função

- **<resource>**: recurso gerado pelo fopen();
- **<quantidade byte a ser lido>**: determina a quantidade de bytes a ser lido do arquivo.

# Manipulação de Arquivos

---

## Leitura

Função `file_get_contents (<filename>,<include_path>,<context>,<offset>,<maxlen>)`

- Utilizamos esta função para ler o conteúdo do arquivo.

## Parâmetros da função

- **<filename>**: nome do ou caminho para o arquivo;
- **<include\_path>**: se true irá buscar os arquivos no caminho especificado na diretiva `include_path`
- **<context>**: contexto(recurso criado com `stream_context_create`)
- **<offset>**: Início da leitura; **<maxlen>**: Tamanho máximo que será lido;

# Manipulação de Arquivos

---

## Escrita

### Função `fwrite( <resource> , <string> )`

- Função utilizada para escrever uma string (segundo parâmetro), no arquivo do primeiro parâmetro.

### Parâmetros da função

- **<resource>**: recurso gerado pelo `fopen()`;
- **<string>**: string que será escrito no arquivo.

# Manipulação de Arquivos

---

## Escrita

Função `file_put_contents (<filename>,<data>, <flags>, <context>)`

- Utilizamos para escrever algo no documento;

## Parâmetros da função

- **<filename>**: nome do ou caminho para o arquivo;
- **<data>**: conteúdo que será escrito no arquivo;
- **<flags>**: Parâmetro que determina como será escrito no documento
- **<context>**: contexto válido criado com a `stream_context_create`



# Manipulação de Arquivos

---

## SPL

- No curso de desenvolvimento orientado a objetos estudamos a SPL, veremos agora algumas classes para trabalhar com arquivos;

- **Classe SplFileObject (<filename>) ;**

### Métodos da Classe:

- **fgets:** retorna uma linha do arquivo
  - **fgetcsv:** retorna uma linha em formato csv, conforme parâmetros;
  - **fwrite:** escreve algo no documento;

# Manipulação de Arquivos

---

## SPL

- Em alguns momentos podemos ter a necessidade de criarmos arquivos que não terão utilidade por muito tempo, temos uma SPL para trabalhar com arquivos temporários.

➤ **Classe SplTempFileObject();**

### Métodos da Classe:

- **fwrite:** escreve algo no documento;

# Manipulação de Arquivos

---

## Sistemas de arquivos

- Além de permitir que nós efetuem operações de leitura e escrita em arquivos locais e remotos através da abertura de um stream, o PHP permite que manipulemos o sistema de arquivos local através de algumas funções;
- Para quem tem experiência com UNIX, algumas funções podem ser bastante familiares.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função chmod( <arquivo> , <permissão> )

- Função para alterar o nível de permissão do arquivo.

### Parâmetros da função

- **<arquivo>**: string com o caminho do arquivo;
- **<permissão>**: inteiro, com o nível de permissão.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função `chown( <arquivo> , <usuário> )`

- Função para alterar o dono do arquivo.

### Parâmetros da função

- **<arquivo>**: string com o caminho do arquivo;
- **<usuário>**: string, com o nome do usuário.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função chgrp( <arquivo> , <grupo> )

- Função para alterar o grupo do arquivo.

### Parâmetros da função

- **<arquivo>**: string com o caminho do arquivo;
- **<grupo>**: mixed, com o nome ou número do grupo.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função stat( <arquivo> )

- Função retorna um array com informações de um determinado arquivo.

### Parâmetros da função

- **<arquivo>**: string com o caminho do arquivo.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função touch( <arquivo> )

- Função cria um arquivo vazio no caminho indicado.

### Parâmetros da função

- **<arquivo>**: string com o caminho do arquivo.



# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função unlink( <arquivo> )

- Função remove o arquivo existente no caminho indicado.

### Parâmetros da função

- **<arquivo>**: string com o caminho do arquivo.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função copy( <origem>, <destino> )

- Função para copiar arquivo da origem indicada, para o destino indicado.

### Parâmetros da função

- **<origem>**: string com o caminho do arquivo;
- **<destino>**: string com o caminho do arquivo.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função rename( <nome\_atual>, <novo\_nome> )

- Função para renomear arquivo

### Parâmetros da função

- **<nome\_atual>**: string com o nome do arquivo atual;
- **<novo\_nome>**: string, novo nome do arquivo.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função `link( <caminho_arquivo>, <caminho_link> )`

- Função para criar uma referência para um arquivo/diretório.

### Parâmetros da função

- **<caminho\_arquivo>**: string com o caminho para o arquivo;
- **<caminho\_link>**: string com o caminho onde o link deverá ser criado.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função `symlink( <caminho_arquivo>, <caminho_link> )`

- Função para criar um link simbólico, para um arquivo/diretório

### Parâmetros da função

- **<caminho\_arquivo>**: string com o caminho para o arquivo;
- **<caminho\_link>**: string com o caminho onde o link deverá ser criado.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função readlink( <caminho\_link> )

- Função para ler informações de um link, a função retorna conteúdo do link.

### Parâmetros da função

- <caminho\_link>: string com o caminho do link.

# Manipulação de Arquivos

---

## Sistemas de arquivos

### Função glob( <string\_consulta> )

- Função para consultar todos os caminhos que combinem com o parâmetro informado.

### Parâmetros da função

- **<string\_consulta>**: string com o conteúdo a ser consultado.

# Manipulação de Arquivos

---

## Locks

- Durante a leitura e escrita de arquivos em nosso sistema, podem ocorrer conflitos. Isto é: uma mesma aplicação escrevendo no mesmo arquivo, ao mesmo tempo;
- Isso pode ser fatal! Então, é preciso adquirir o lock de um arquivo antes de efetuar essa operação;
- O **flock** garante que só você irá trabalhar neste arquivo naquele momento, até que o lock seja retirado.



# Manipulação de Arquivos

---

## Wrappers

- Os wrappers podem ser definidos como códigos responsáveis por fazer a conexão com N destinos diferentes;
- Uma pagina web respondendo em HTTP é diferente do sistema de arquivos local, sendo assim, há um wrapper para cada um deles;
- É possível adicionar um wrapper customizado usando a função `stream_wrapper_register()`;

# Manipulação de Arquivos

---

## Lista dos principais Wrappers

- **file://** é o wrapper padrão utilizado quando nenhum é passado explicitamente e serve para conexão com arquivos locais;
- **http://** é o wrapper utilizado para comunicação usando protocolo HTTP;
- **ftp://** para trabalhar com arquivos FTP;
- **php://** para trabalhar com buffers nativos do PHP;

# Manipulação de Arquivos

---

## Contextos

- Durante o acesso a streams podemos precisar manipular o cabeçalho dessa abertura de stream;
- Um exemplo muito comum é a manipulação de um cabeçalho HTTP para criar uma requisição;
- Este cabeçalho é chamado de contexto e este tipo de manipulação é feito através da criação de um contexto utilizando a função **stream\_context\_create()**.

# Manipulação de Arquivos

---

## Uploads

- Podemos trabalhar com formulários que oferecem uma opção de upload de arquivos, sejam eles arquivos de texto ou binários;
- O PHP possui suporte a upload de arquivos a partir de qualquer navegador;
- O formulário deve possuir *method* definido como **POST**, e o *enctype* como **multipart/form-data**;
- O arquivo enviado é acessado através da superglobal `$_FILES`.

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Manipulação de Arquivos;
- Resolver o **Laboratório** proposto nesta aula;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**

# Manipulação de XML - SimpleXML

---



# Objetivos da Aula

---

- Introdução ao XML;
- SimpleXML;
- XMLParser;
- Xpath;

# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- O XML propõem uma forma genérica de codificação de documentos, permitindo que aplicações independentes da plataforma e/ou linguagem utilizada possam armazenar e compartilhar informações;
- Suas especificações e regras de criação e padronização de documentos XML são mantidas pelo W3C.



# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- Documento estruturado de forma hierárquica;
- Semelhante ao HTML, porém não possui um conjunto de tags predefinidas;
- Objetivo de viabilizar a criação de estruturas de armazenamento de informações.

# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- Todo documento XML deve sempre possuir cabeçalho;
- A unidade básica de informação em XML é definida como elemento;
- Os elementos são delimitados por uma tag inicial e uma tag final;

# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- Em elementos podem ainda existir atributos;
- As chaves são separadas dos valores por um sinal de igual e os valores ficam contidos entre aspas simples ou duplas;
- Os atributos devem ser utilizados como metadados, informações gerais sobre o que irá conter naquele elemento.

# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- Em elementos podem ainda existir atributos;
- As chaves são separadas dos valores por um sinal de igual e os valores ficam contidos entre aspas simples ou duplas;
- Os atributos devem ser utilizados como metadados, informações gerais sobre o que irá conter naquele elemento.

# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- Utilize nomes descritivos que expressem fielmente a finalidade do elemento;
- Para nomes extensos ou compostos por duas ou mais palavras faça uso de 'underline';
- Procure dar nomes objetivos e preferencialmente curtos;
- Evite utilizar hífen nos nomes dos elementos e atributos;
- Evite utilizar ponto final nos nomes dos elementos e atributos;

# Manipulação de XML - SimpleXML

---

## O que é um documento XML?

- Evite utilizar dois pontos nos nomes dos elementos e atributos;
- Dois pontos são reservados para documentos XML com namespaces.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Carregando um documento XML

- Para carregar um documento XML, utilizamos o construtor da classe SimpleXMLElement.

### Parâmetros do construtor

- **<data>**: string com um XML, ou o nome do arquivo XML;
- **<options>**: parâmetro opcional para configuração da libxml;
- **<data\_is\_url>**: parâmetro define o comportamento do primeiro parâmetro; se true, ele considera o primeiro parâmetro como sendo o caminho de XML.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Alterando elemento

- Uma vez que nosso documento XML foi transformado em um objeto podemos alterar as informações de um elemento, através de uma simples atribuição de valor a uma determinada propriedade do objeto.



# Manipulação de XML - SimpleXML

---

## SimpleXML

### Adicionando elemento

- Podemos adicionar novos elementos através do método `addChild()`, que criará um novo elemento no documento armazenado na memória.

### Parâmetros do método

- **<nome\_elemento>**: string com nome do elemento;
- **<valor\_elemento>**: string com valor do elemento.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Removendo elemento

- Podemos remover elementos do nosso documento através da função nativa `unset()`.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Recuperar todos atributos de um elemento

- Podemos também manipular os atributos de cada elemento. Para isso, basta selecionar o elemento cujos atributos você deseja alterar e fazer uso do método `attributes()`, que retorna um objeto com os atributos daquele elemento.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Alterando atributos

➤ Podemos alterar os atributos de um elemento; para isso basta recuperar os atributos do elemento com o método `attributes()`, conforme vimos anteriormente, e através do objeto retornado, modificar o valor da propriedade (que agora estará representando o atributo) através de simples atribuição.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Adicionando atributos

- Além de alterar os atributos que já estão nos elementos de nosso documento, podemos adicionar novos atributos através do método `addAttribute()`, que irá criar um novo atributo em um elemento no documento armazenado na memória.

### Parâmetros do método

- **<nome\_atributo>**: string com nome do atributo;
- **<valor\_atributo>**: string com valor do atributo.

# Manipulação de XML - SimpleXML

---

## SimpleXML

### Removendo atributos

- Podemos remover atributos de um XML; para isso basta utilizarmos a função nativa `unset()`.

# Manipulação de XML

## XMLParser / XPath

---

### XMLParser

#### O que é e porque utilizar?

- Quando utilizamos a classe SimpleXML e a extensão DOM, o documento XML é carregado por completo na memória;
- Porém, nem sempre fazer isso é viável;
- Então, a extensão XMLParser nos propõe uma solução...

# Manipulação de XML

## XMLParser / XPath

---

### XMLParser

Mas antes, qual a ideia de um parser?

- **Parser** representa a ação de realizar uma análise sobre algo, ou então, de criar regras que analisem uma determinada informação.



# Manipulação de XML

## XMLParser / XPath

---

### XMLParser

#### Como utilizar o XMLParser?

- Com a extensão XMLParse podemos definir regras de leitura e extração de informações de um documento XML;
- Tais regras de análise, para funcionar, precisam ser aliadas à técnica conhecida como SAX.

# Manipulação de XML

## XMLParser / XPath

---

### XMLParser

#### SAX?

- Abreviação de Simple API for XML;
- API para realizar parser de um XML, baseando-se em eventos;
- SAX é o nome dado a técnica de ler um documento XML em blocos;

# Manipulação de XML

## XMLParser / XPath

---

### XMLParser

#### Evento em um XML? Quais?

- É considerado como evento:
  - Abertura de um elemento, ou seja, uma nova tag;
  - Fechamento de um elemento;
  - Nós do tipo texto, ou seja, o valor de um elemento.

# Manipulação de XML

## XMLParser / XPath

---

### XMLParser

#### Na prática...

- Criamos funções para tratar os 3 eventos existentes;
- Criamos um 'analisador', utilizando as funções:
  - `xml_parser_create();`
  - `xml_set_element_handler();`
  - `xml_set_character_data_handler();`
- Aplicamos o SAX para leitura do documento.

# Manipulação de XML

## XMLParser / XPath

---

### XPath

#### O que é?

- Permite realizar consultas em um XML, através de expressões;
- Estas expressões, funcionam de forma semelhante aos seletores do CSS em um arquivo HTML, permitindo que você identifique os elementos, organizando-os em grupos;
- Vamos conhecer melhor como essas expressões funcionam.

# Manipulação de XML

## XMLParser / XPath

---

### XPath

#### Como funciona?

- Permite navegação, consulta e acesso aos elementos, como em sistema de arquivo;

- Exemplos:

/apostilas

/apostila

/apostilas/apostila[@codigo=500]

# Manipulação de XML

## XMLParser / XPath

---

### XPath

#### Xpath com SimpleXML

- Podemos realizar consultas Xpath sobre objetos da classe SimpleXML;
- Para isso, basta utilizarmos o método 'xpath()', que está presente nos objetos da classe SimpleXMLElement;

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Manipulação de XML - SimpleXML;
- Resolver o **Laboratório** proposto nesta aula;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**



# Manipulação de XML - DOM

---



# Objetivos da Aula

---

- O que é DOM?
- A classe DOMDocument;
- Xpath;

# Manipulação de XML - DOM

---

## O que é DOM?

- O DOM, do inglês Document Object Model, é um padrão da W3C para representar documentos baseados em XML;
- Representação realizada a partir de objetos;
- Propõe um modelo padrão para combinar, acessar e manipular esses objetos, baseado nas informações contidas em documentos.

# Manipulação de XML - DOM

---

## DOMDocument

### Carregando um documento XML a partir de uma variável

- Para carregar um documento XML a partir de uma variável precisamos de um objeto da classe DOMDocument;
- Através do objeto utilizamos o método 'loadXML()' passando como parâmetro a variável que contém o XML.

# Manipulação de XML - DOM

---

## DOMDocument

### Carregando um documento XML a partir de arquivo

- Para carregar um documento XML a partir de uma variável precisamos de um objeto da classe DOMDocument;
- Através do objeto utilizamos o método 'load()', passando como parâmetro o caminho para o arquivo que contém o XML.

# Manipulação de XML - DOM

---

## DOMDocument

### Alterando elemento

- Uma vez que nosso documento XML foi carregado em um objeto DOMDocument, podemos alterar as informações de um elemento. Para isso:
  - Precisamos localizar o elemento através do método 'getElementsByTagName()';
  - Ele retornará um objeto da classe DOMNodeList, com todos os elementos que possuem a tag consultada.

# Manipulação de XML - DOM

---

## DOMDocument

### Alterando elemento

- Então, precisamos utilizar o método 'item()', para especificar qual dos elementos encontrados vamos alterar;
- Por fim, através de atribuição simples, modificamos a propriedade 'nodeValue', indicando o novo valor do elemento.

# Manipulação de XML - DOM

---

## DOMDocument

### Adicionando elemento

- Podemos criar novos elementos através do método 'createElement()', que irá gerar um novo elemento;
- Com um novo elemento criado, com o método 'appendChild()', colocamos o elemento criado onde desejamos.



# Manipulação de XML - DOM

---

## DOMDocument

### Removendo elemento

- Para remover um elemento precisamos localizá-lo através do método 'getElementsByTagName()';
- Ele retornará um objeto da classe DOMNodeList, com todos os elementos que possuem a tag consultada;
- Feito isso, com o método 'item()' acessamos o objeto que queremos eliminar;

# Manipulação de XML - DOM

---

## DOMDocument

### Removendo elemento

- E, através da propriedade 'parentNode', acessamos o objeto pai do elemento que queremos remover;
- Após isso, basta utilizarmos o método 'removeChild()' para, enfim, remover o elemento desejado.

# Manipulação de XML - DOM

---

## DOMDocument

### Recuperar atributo de um elemento

- Podemos também manipular os atributos de cada elemento;
- Para recuperar o valor de um atributo específico basta localizar o elemento que queremos, em seguida extrair o valor do atributo desejado através do método 'getAttribute()'.

# Manipulação de XML - DOM

---

## DOMDocument

### Alterando e adicionando atributos

- Podemos adicionar e alterar os atributos de um elemento, para isso, basta utilizar o método 'setAttribute()', passando a ele dois parâmetros, o primeiro o nome do atributo, e o segundo o valor.

# Manipulação de XML - DOM

---

## DOMDocument

### Removendo atributos

- Podemos remover atributos de um XML;
- Para isso basta utilizarmos o método 'removeAttribute()', passando como parâmetro a ele o nome do atributo que desejamos remover.

# Manipulação de XML

## XMLParser / XPath

---

### XPath

#### Xpath com DOMDocument

- Para realizar uma consulta XPath em um objeto DOM precisamos:
  - Criar uma instância da classe DOMDocument e deixar o objeto com o XML carregado;
  - Criar uma instância da classe DOMXPath. Para criar este objeto é necessário informar ao construtor o objeto da DOMDocument;

# Manipulação de XML

## XMLParser / XPath

---

### XPath

#### Xpath com DOMDocument

- Com o objeto DOMXPath criado basta utilizarmos o método 'query()' informando, para seu parâmetro, a expressão que deseja consultar.

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Manipulação de XML - DOM;
- Resolver o **Laboratório** proposto nesta aula;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**



# Objetivos da Aula

---

- Introdução a WebServices;
  - Verbos HTTP / Status Code;
  - Content Negotiation;
  - WebServices RPC;
  - WebServices REST;
- Json;

# WebServices

---

## Introdução

### O que é um WebService?

- São meios que viabilizam comunicação/compartilhamento de informação entre aplicações;
- Funcionam independentemente da plataforma, linguagem e arquitetura;
- Seu uso tornou-se mais comum com o crescimento de aplicações orientadas a serviço (SOA), cuja principal vantagem é “facilitar” a escalabilidade da aplicação.

# WebServices

---

## Verbos HTTP

- **GET**: Recupera um ou vários recursos;
- **POST**: Cria um recurso;
- **PATCH**: atualiza parte de um recurso;
- **PUT**: atualiza um recurso como um todo;
- **DELETE**: deleta um recurso;

# WebServices

---

## Status Code

- **200**: Sucesso;
- **300**: Redirecionamento;
- **400**: Requisição inválida;
- **500**: Erro no servidor;

# WebServices

---

## Content-Type

- Este header vai na requisição HTTP pro serviço, contendo, por exemplo: application/xml para XML ou application/json para JSON.
- Chegando no serviço, este olha o header Content-Type e portanto sabe decodificar aquele dado vindo da requisição;
- Para acessar essa informação pelo PHP basta acessar a chave CONTENT\_TYPE de \$\_SERVER.

# WebServices

---

## Accept

- Este header é enviado na requisição HTTP pro serviço, contendo quais formatos nós queremos como resposta, podendo ser mais de um, passando prioridades;
  - Accept:application/json(somente JSON);
  - Accept:application/xml(somente XML);
  - Accept:application/json;q=0.5,application/xml(prefiro XML);
  - Accept:application/\*(xml ou json - ou outro que comece com application /);

# WebServices

---

## Content Negotiation

- Um recurso bastante interessante do HTTP é o uso dos headers Content-Type e Accept;
- Trabalhando em conjunto, eles especificam o formato de envio do dado e o formato de resposta;
- O cliente pode mandar diferentes formatos e escolher um formato de resposta à sua escolha;

# WebServices

---

## WebService RPC

- Remote Procedure Call é uma arquitetura onde dois sistemas se comunicam solicitando um ao outro a execução de procedimentos;
- O cliente manda para o servidor o método que ele quer executar e os parâmetros (quando houver);
- O servidor recebe e executa o método, e responde com o que foi solicitado;
- Dentro deste modelo, temos várias implementações possíveis, como JSON-RPC, XML-RPC e SOAP.



# WebServices

---

## Json

- O Json é atualmente o formato de dados mais utilizado para troca de dados entre sistemas;
- Ele é mais leve e portanto mais rápido que o formato Xml;
- O trabalho com Json é muito mais simples que o xml;
- Para manipular Json trabalhamos apenas com array, após toda manipulação codificamos para o Json: **json\_encode / json\_decode;**

# WebServices

---

## JSON-RPC

- Neste tipo de implementação, o cliente deve gerar um JSON no seguinte formato;

```
{  
    " method " : " echo " ,  
    " params " : [ " Hello JSON - RPC " ] ,  
    " id " : 1  
}
```

- Method: Nome do método que será chamado;
- Params: Array de argumentos;
- Id: identificação da requisição;

# WebServices

---

## JSON-RPC

- O servidor remoto irá executar o método solicitado e deve retornar a resposta como mostra no formato json:

```
{  
    "result" : "Resultado da Requisição" ,  
    "error" : null ,  
    "id" : 1  
}
```

- result: Resultado do retorno do método executado;
- error: erro caso ocorra;
- Id: identificação da requisição;

# WebServices

---

## WebService REST

### O que é e como funciona?

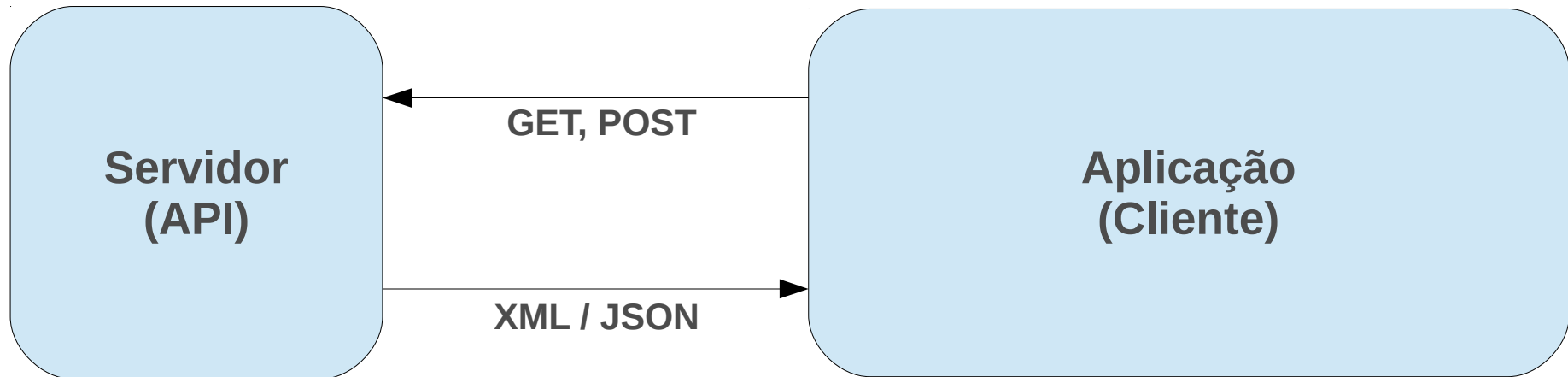
- Baseado no design do protocolo HTTP (códigos de status, tipo de conteúdo, cabeçalho, etc);
- A requisição devolverá as informações em formato XML, JSON ou outros;
- A ideia do seu funcionamento é semelhante a uma requisição comum, porém, ao invés de enviar como resposta um HTML, enviamos dados.

# WebServices

---

## WebService REST

O que é e como funciona?



# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** WebServices;
- Resolver o **Laboratório** proposto nesta aula.

**Mãos à obra!**

# WebServices

---



# Objetivos da Aula

---

- WSDL;
- WebServices SOAP;



# WebServices

---

## WebService SOAP

### O que é e como funciona?

- Simple Object Access Protocol;
- Padrão desenvolvido pela W3C para padronizar a troca de informações entre ambientes;
- Informações são transmitidas no formato XML;
- Para realizar essa padronização, um serviço SOAP é especificado por um WSDL.

# WebServices

---

## WSDL

- WSDL (Web Services Description Language) é um documento XML utilizado para definir os serviços de um servidor SOAP;
  - o endpoint do serviço;
  - o nome e descrição do serviço;
  - os argumentos que o serviço aceitará;
  - a resposta do serviço.

# WebServices

---

## WSDL

- Web Service Description Language;
- Utilizado para descrever:
  - as funcionalidades oferecidas no WebService;
  - Os métodos do serviço disponível;
  - Os parâmetros que cada um receberá;
  - Como as informações serão transmitidas.

# WebServices

---

## Principais tags Wsdl

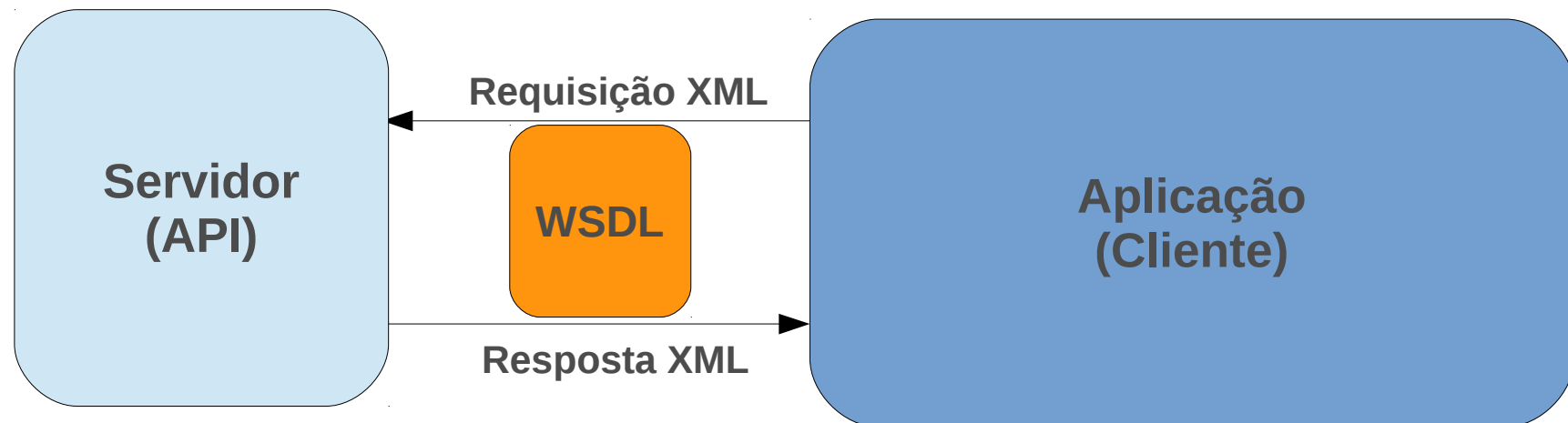
- Messages: Especificam as mensagens de entrada e saída;
- Ports: Especificam as funcionalidades dos serviços;
- Bindings: Especificam como serão armazenados e transportados os dados;
- Service: Descreve todo o serviço, quais são os ports, bindings e endereço do servidor;

# WebServices

---

## WebService SOAP

O que é e como funciona?



# WebServices

---

## WebService SOAP

### Criando um servidor

- Para disponibilizar um webservice para uso precisamos criar um objeto da classe 'SoapServer()' e, para seu construtor, devemos informar o caminho para o arquivo WSDL;
- Temos ainda, que indicar ao objeto da classe SoapServer, qual classe ou função, será 'servida', para isso, utilizamos respectivamente os métodos, 'setObject' ou 'addFunction'.

# WebServices

---

## WebService SOAP

### Consumindo um serviço

- Existe uma classe para consumir serviços chamada 'SoapClient()'. Ao criar um objeto da classe precisamos informar o endereço para o arquivo WSDL;
- Então, a partir do objeto criado, você simplesmente consome um método disponibilizado pelo WSDL;

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** WebServices;
- Resolver o **Laboratório** proposto nesta aula;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**



# Configuração de Ambiente

---



# Objetivos da Aula

---

- Configuração do Apache;
- Virtual Host;
- Rewrite;
- Configuração do PHP.

# Configuração de Ambiente

---

## Ambiente de desenvolvimento

### Porque é importante conhecer a configuração?

- Todo desenvolvedor não deve limitar-se em apenas conhecer a linguagem de programação que trabalha;
- Conhecer as ferramentas e recursos que envolvem o meio que a linguagem está inserida é de extrema importância;
- Assim como a infraestrutura onde a aplicação está sendo executada é muito importante, pois ajuda principalmente a detectar, prevenir e corrigir possíveis erros.

# Configuração de Ambiente

---

## Apache

### Configurações do Apache



# Configuração de Ambiente

---

## Apache

### Arquivos de configuração

- O arquivo padrão de configuração, chama-se 'apache2.conf';
- Nele estão presentes diversos parâmetros de configuração do servidor HTTP;
- Além de incluir outros arquivos com configurações específicas;
- No sistema UNIX, todas as configurações do Apache estão localizadas no diretório /etc/apache2.

# Configuração de Ambiente

---

## Apache

### Trabalhando com múltiplos sites

- O Apache permite servir mais de uma aplicação simultaneamente;
- Para permitir isso, podemos definir um arquivo de configuração para cada site, onde setamos o diretório onde está localizado a aplicação, além do endereço correspondente a ela;
- Esse tipo de configuração, é o que chamamos de 'virtual host'.

# Configuração de Ambiente

---

## Virtual Host

```
<VirtualHost *:80>
    ServerName site.com.br
    DocumentRoot path
    <Directory /var/www/html/cursos/REST_MVC/public>
        DirectoryIndex index.php
        AllowOverride All
        Order allow,deny
        Allow from all
        <IfModule mod_authz_core.c>
            Require all granted
        </IfModule>
    </Directory>
    ErrorLog /var/log/mvc_error.log
    CustomLog /var/log/mvc_access_log common
</VirtualHost>
```

# Configuração de Ambiente

---

## Virtual Host

- **VirtualHost** : Porta que será utilizada para aplicação
- **ServerName** : Nome do Site
- **DocumentRoot** : Caminho da aplicação
- **Directory**: Itens referentes ao comportamento do diretório da aplicação
  - **AllowOverride**: permite sobrescrever as regras padrão;
  - **Order**: Ordem de permissão;
  - **Allow**: Permissão;



# Configuração de Ambiente

---

## Rewrite

- O módulo de rewrite do apache é um módulo para podermos trabalhar com reescrita das URLs. Algo como a URL está de um jeito e reescrevemos-na de outro. Porquê?
  - `http://meu-sistema.com.br/index.php?pagina=clientes&acao=listar`
  - `http://meu-sistema.com.br/clientes/listar`
- Para habilitar o módulo rewrite usamos:
  - `a2enmod rewrite`
- Podemos fazer uso de um arquivo chamado `.htaccess` para esta tarefa;

# Configuração de Ambiente

---

## PHP

### Configurações do PHP



# Configuração de Ambiente

---

## PHP

### Arquivo de configuração do PHP

- O PHP possui um arquivo de configuração, chamado 'php.ini';
- Nele está definida toda a configuração do interpretador PHP;
- E, com a manipulação desse arquivo, podemos modificar como a execução dos scripts irá se comportar;
- Por questão de segurança, o arquivo de configuração do PHP possui duas versões, um que atende especialmente o Apache, localizado em /etc/php5/apache2/php.ini;

# Configuração de Ambiente

---

## PHP

### Arquivo de configuração do PHP

- E outro, que atende a execução do PHP via shell (terminal), localizado em `/etc/php5/cli/php.ini`.

# Configuração de Ambiente

---

## PHP

### Algumas configurações:

- **expose\_php:** Expõe a versão do PHP;
- **disable\_functions:** Lista de funções nativas que serão desabilitadas;
- **disable\_classes:** Lista de classes nativas que serão desabilitadas;
- **include\_path:** Lista de diretórios que será usada caso um arquivo não seja encontrado;

# Configuração de Ambiente

---

- **memory\_limit:** Limite de memória que um script PHP pode utilizar;
- **max\_execution\_time:** Tempo máximo de execução em segundos de um script php;
- **max\_input\_time:** Tempo máximo de leitura de dados via GET/POST;
- **post\_max\_size:** Tamanho máximo de uma requisição POST;
- **upload\_max\_filesize:** Tamanho máximo de um arquivo enviado(upload)
- **max\_file\_uploads:** Tamanho máximo de arquivos enviados via POST;

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Configuração de Ambiente;
- Resolver o **Laboratório** proposto nesta aula;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**

# Ecosistema ZEND

---





# Objetivos da Aula

---

- Introdução aos produtos Zend;
- ZendServer;
- ZendStudio;
- ZendServer + ZendStudio: combinação perfeita.

# Ecossistema ZEND

---

## A Zend

### Quem são?

- Empresa fundada pelos desenvolvedores responsáveis pela reescrita do PHP, os israelenses Zeev e Andy;
- Empresa com o principal foco de profissionalizar o PHP em todo seu ciclo de vida, do desenvolvimento ao ambiente de produção;
- Possui como principais produtos ZendStudio, ZendServer e ZendFramework.

# Ecossistema ZEND

---

## ZendServer

### Recursos disponíveis

- Servidor de aplicação PHP integrado ao Apache;
- Permite gerenciar aplicações críticas de negócios em PHP;
- Binários do PHP certificados pela Zend;
- Foco em recursos que aumentam a produtividade do ciclo de desenvolvimento;
- Recursos de otimização de desempenho;

# Ecossistema ZEND

---

## ZendServer

### Recursos disponíveis

- Fila de execução para realização de tarefas assíncronas;
- Padronização para deploy de aplicações;
- Integração com Zend Studio;
- Monitoramento;
- Escalável.

# Ecossistema ZEND

---

## ZendServer

### Resolução de problemas

- Identificar e reproduzir problemas é uma das etapas que mais tomam tempo durante o ciclo de desenvolvimento:
  - Quem encontra o erro quase sempre é usuário final;
  - O erro é reportado de forma genérica;
  - Gasta-se tempo para entender, reproduzir e detectar a real causa.

# Ecossistema ZEND

---

## ZendServer

### Resolução de problemas

- Com o Zend Server o tempo desta etapa é reduzido consideravelmente, utilizando-se de:
  - Code tracing;
  - Monitoramento.

# Ecossistema ZEND

---

## ZendServer

### Deploy de aplicações

- Deploy padronizado através de pacotes “zpk”;
- Realizado pela interface web do Zend Server;
- Pode ser realizado através de API;
- Permite criar e definir “virtual hosts” durante o deploy;
- É possível realizar rollback para versão anterior.

# Ecosystem ZEND

## ZendServer

### Isolamento de bibliotecas

- É possível criar um repositório de bibliotecas, não sendo mais necessário deixá-las disponíveis dentro da aplicação.

The screenshot displays the ZendServer Enterprise web interface. The top navigation bar includes 'Overview', 'Applications' (selected), 'Configurations', and 'Administration'. The user is logged in as 'administrator' at '11:24'. Below the navigation bar, there are tabs for 'Apps', 'Libraries' (selected), 'Monitoring Rules', 'Caching Rules', and 'Job Rules'. The main content area is titled 'Applications > Libraries > Libraries List'. It features a 'Deploy Library' button on the left and a 'Check for Updates' button on the right. A table lists installed libraries with their names, version counts, latest versions, and messages. The 'Actions' column contains icons for refreshing, deleting, and expanding. The libraries listed are: FPDF Library (1 version, latest: 1.7), GanttGraph for PHP (1 version, latest: 1.0.0), Symfony Framework 1 (1 version, latest: 1.4.20), Zend Framework 1 (1 version, latest: 1.12.3), Zend Framework 2 (2 versions, latest: 2.2.2), and Zend Server Gateway (1 version, latest: 0.9.0). Messages indicate that new versions (2.2.0 and 2.2.4) are available for FPDF and Zend Framework 2, respectively, with links to 'Download' and 'Release notes'.

Name	Messages	Actions
✓ FPDF Library (1 version, latest: 1.7)	A new library version (2.2.0) is available. <a href="#">Download</a> ( <a href="#">Release notes</a> )	
✓ GanttGraph for PHP (1 version, latest: 1.0.0)		
✓ Symfony Framework 1 (1 version, latest: 1.4.20)		
✓ Zend Framework 1 (1 version, latest: 1.12.3)		
✓ Zend Framework 2 (2 versions, latest: 2.2.2)	A new library version (2.2.4) is available. <a href="#">Download</a> ( <a href="#">Release notes</a> )	
✓ Zend Server Gateway (1 version, latest: 0.9.0)		



# Ecossistema ZEND

---

## ZendServer

### Fila de execução – JobQueue

- Execução de tarefas assíncronas;
- Disparada pela aplicação;
- Permite agendamento estilo CRON;
- Chamadas HTTP;
- Log e controle de execução.

# Ecossistema ZEND

## ZendServer

### Fila de execução - JobQueue

The screenshot displays the ZendServer Enterprise Job Queue management interface. The top navigation bar includes tabs for Overview, Applications, Configurations, and Administration. The 'Overview' tab is active, showing a 'Job Queue' sub-tab. The interface is divided into a header section with navigation links (Dashboard, Events, Code Tracing, Job Queue, Server Info, Logs) and a main content area. The main content area features a 'Statistics and Activity' section with various metrics, a filter section, and a table of jobs.

**Statistics and Activity**

Jobs waiting for execution	0	Job Queue startup time	Tuesday, 12:57:04	Jobs that failed execution	0
Jobs waiting for predecessor	0	Average job wait time	0	Jobs that reported an error	0
Jobs set to run once	542740	Average job execution	0	Completed jobs	48083
Jobs scheduled for execution	13	Jobs currently executing	0	Total jobs served	597457

**Filter and Search**

Show Filters ▾ All Jobs ▾ Time Range All ▾ From 09/Oct/2013 12:04 To 10/Oct/2013 12:04 Search...

Filter By : Currently no filter selected

Shown: 1-20 / 48096 Delete Selected Delete Filtered Requeue

ID	Name	Application	URL	Status	Priority	Start Time	Creation Time
544530	Flush Fila	CR4 - PROD	http://cr4.4linux.com.br:80/zend/swiftFlush	Scheduled	normal	Today, 12:05:45	Today, 12:04:00
544529	Flush Fila	CR4 - PROD	http://cr4.4linux.com.br:80/zend/swiftFlush	Scheduled	normal	Today, 12:05:15	Today, 12:04:00
544528	Preparar Fila	CR4 - PROD	http://cr4.4linux.com.br:80/zend/swiftFlush	Scheduled	normal	Today, 12:05:00	Today, 12:04:00
544527	Flush Fila	CR4 - PROD	http://cr4.4linux.com.br:80/zend/swiftFlush	Scheduled	normal	Today, 12:05:00	Today, 12:04:00
544526	Limpar Reserva F...	CR4 - PROD	http://cr4.4linux.com.br:80/zend/limparReservaFicha	Scheduled	normal	Today, 12:05:00	Today, 12:04:00

Find: update Previous Next Highlight all Match case

# Ecossistema ZEND

---

## ZendStudio

### Recursos disponíveis

- Auto-complete 100% funcional;
- Templates para os mais diversos tipos de arquivo;
- Preparado para trabalhar com TDD (Desenvolvimento orientado a testes), permitindo fácil integração a IDE com o PHPUnit;
- Permite debug integrado com o ZendServer;

# Ecossistema ZEND

---

## ZendStudio

### Recursos disponíveis

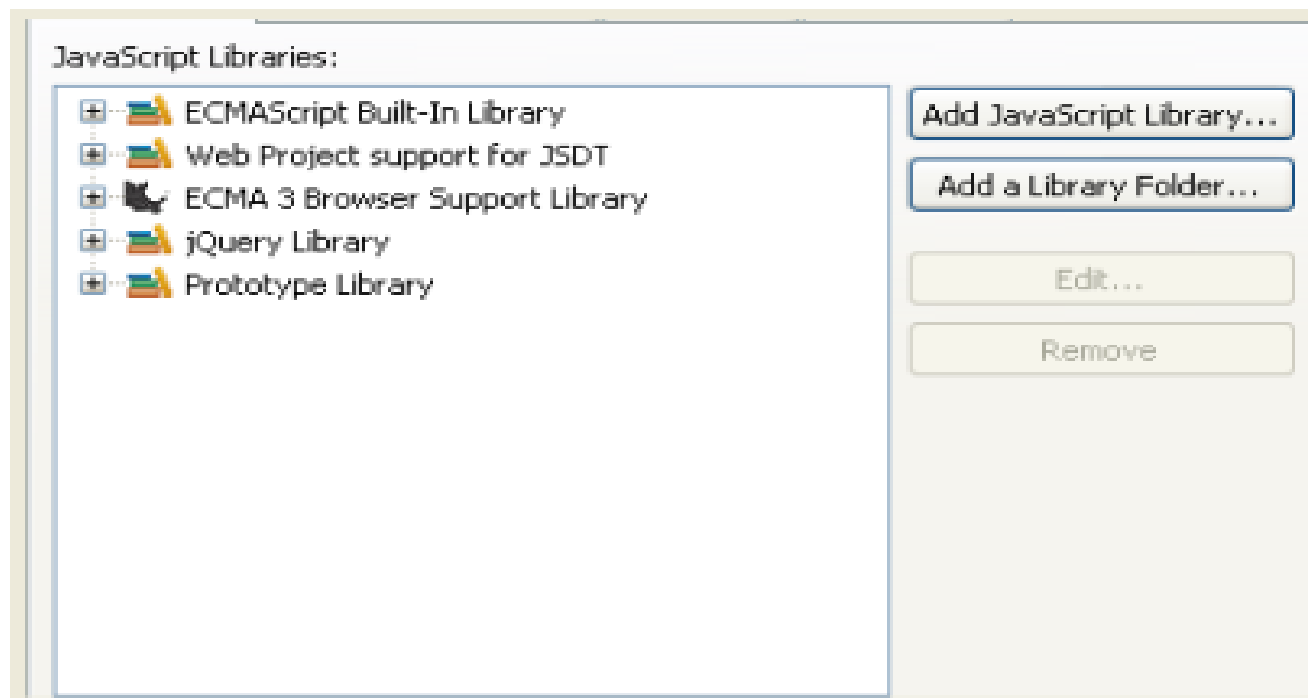
- Agilidade no desenvolvimento como ZendFramework 2;
- Permite criar aplicações Web para mobiles.

# Ecosystem ZEND

## ZendStudio

### Consistente não apenas para desenvolvimento PHP

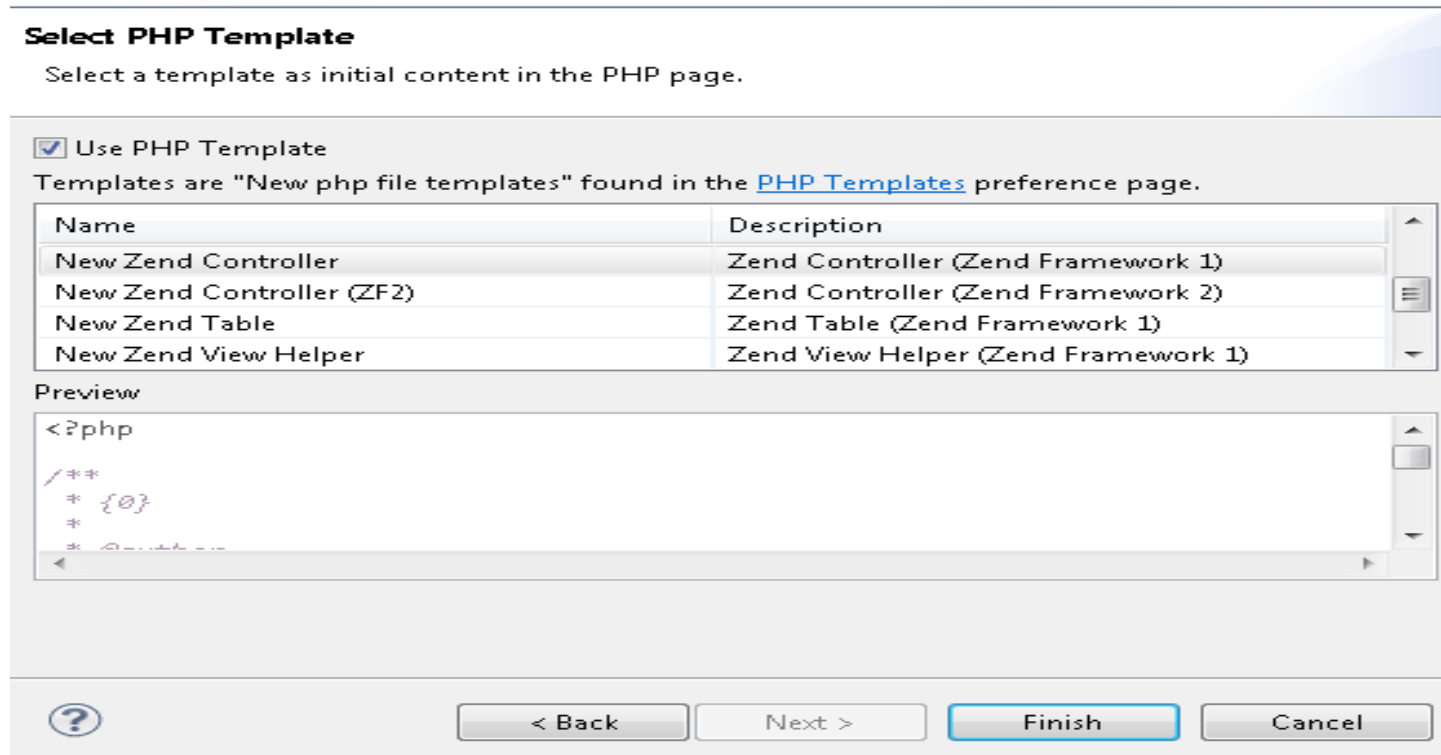
- Disponibiliza as principais bibliotecas JavaScript;



# Ecosystema ZEND

## ZendStudio

### Helpers para Zend Framework e PHP





# Ecosistema ZEND

## ZendStudio

## Debug

The screenshot displays the Zend Studio IDE interface during a debug session. The top menu bar includes File, Edit, Refactor, Source, Navigate, Search, Project, Run, Window, Help, and Zend Store. The toolbar contains various icons for file operations, debugging, and navigation.

The **Debug** tab is active, showing the **PHP Debug [Remote PHP Launch]** configuration. The **PHP Application** section lists the application being debugged, with the current execution point at `/usr/local/zend/var/libraries/Symfony_Framework_1/1.4.20/lib/util/sfContext.php` at line 7. The **Variables** tab shows the current state of variables:

Name	Value
<code>\$configuration</code>	Object of: frontendConfiguration (string:8) frontend
<code>\$name</code>	(string:9) sfContext
<code>\$class</code>	(string:24) Undefined variable: this

The **sfContext.class.php** file is open in the editor, showing the `createInstance` method. The code is as follows:

```
43 static public function createInstance(sfApplicationConfiguration $configuration, $name = null, $class = __CLASS__)  
44 {  
45     if (null === $name)  
46     {  
47         $name = $configuration->getApplication();  
48     }  
49     self::$current = $name;  
50     self::$instances[$name] = new $class();  
51 }
```

The **Outline** tab on the right shows the class structure, including `$dispatcher`, `$configuration`, `$mailerConfiguration`, `$factories`, `$instances`, `$current`, and the `createInstance` method.

The **Console** tab at the bottom shows the output of the debug session, indicating the application is running at `http://cr4-local.local/cursos`.



# Próximos Passos

---

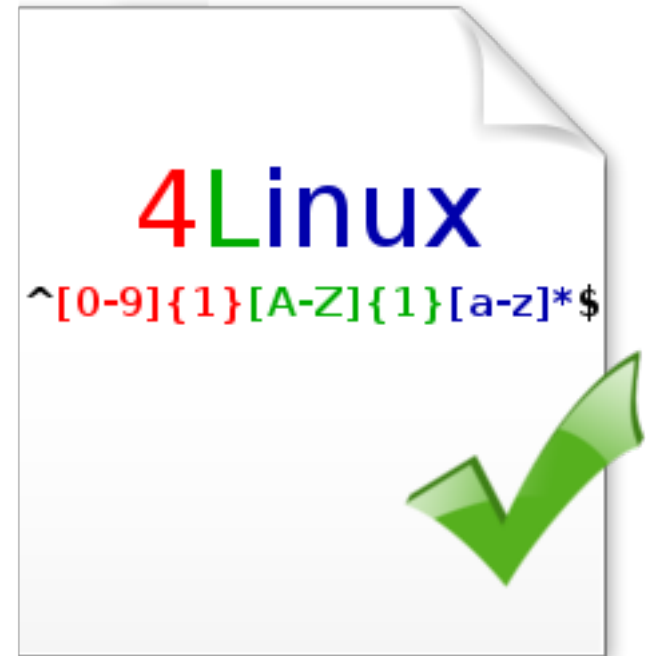
Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Ecossistema ZEND;
- Resolver o **Laboratório** proposto nesta aula.

**Mãos à obra!**

# Expressão Regular

---



# Objetivos da Aula

---

- Introdução a Expressão Regular;
- Funções do PHP para expressão regular.

# Expressão Regular

---

## Introdução a Expressão Regular

### O que é?

- Expressões regulares, também conhecidas como **regex** ou **regexp**, permitem que encontremos caracteres, palavras ou pedaços de string em uma outra string através de regras e comandos;
- Podemos dizer que expressões regulares é quase uma linguagem de programação.

# Expressão Regular

---

## Alguns usos comuns:

- Validação de dados que seguem um certo padrão (url, email etc.);
- Substituição de dados que seguem um certo padrão (acrescentar pontuação num cpf ou telefone por exemplo);
- Extrair informações que seguem um certo padrão (extrair todos os emails de uma página web para spam);

# Expressão Regular

## Introdução a Expressão Regular

Expressão	Descrição
foo	A String “foo”
^foo	“foo” no início de um string
foo\$	“foo” no final de uma string
^foo\$	“foo” sozinho em uma string
[abc]	“a”, “b” ou “c”
[a-z]	Qualquer letra minúscula
[^A-Z]	Qualquer letra que não seja maiúscula
( png   jpg )	Que seja “png” ou “jpg”
[a-z] +	Uma ou mais letras minúsculas
[0-9.-]	Qualquer número, ponto ou sinal de menos
^[a-zA-Z0-9_]{1,}\$	Mínimo de uma letra, número ou underline

# Expressão Regular

## Introdução a Expressão Regular

Expressão	Descrição
<code>([wx])([yz])</code>	"wy", "wz", "xy" ou "xz"
<code>[^A-Za-z0-9]</code>	Qualquer símbolo que não seja número ou underline
<code>[A-Z]{3} [0-9]{4}</code>	Encontra três letras ou quatro números

- Quantificadores:
  - (\*) O asterisco multiplica nosso padrão de 0 até N vezes.
  - (?) O interrogação multiplica nosso padrão de 0 a 1 vezes. Ou seja, o interrogação pode ser utilizado como um "ou nada".
  - (+) O sinal de mais multiplica nosso padrão 1 a N vezes. Ou seja, é a mesma coisa que o asterisco, o padrão existe pelo menos uma vez.

# Expressão Regular

---

## Introdução a Expressão Regular

- Captura e subexpressão:
  - Podemos utilizar os parênteses para gerar grupos de captura, assim como sub expressões. Vejamos alguns exemplos de sub expressões:
    - **(abc)\*** : casa com abc , abcabcab , abcabcabcab , string vazia , etc;
    - **([ A - Z ]+)?** : casa com string vazia , mas também casa com qualquer string composta por pelo um caractere maiúsculo;
    - **(/[ a - z ]+(/[ 0 - 9 ]+ )?)?** : casa com / clientes , / clientes / 2 , / fornecedores / 45 , string vazia;



# Expressão Regular

---

## Funções do PHP para Expressão Regular

### **preg\_match( <expressao>, <string entrada> )**

- Permite que você utilize uma expressão regular para analisar uma string.

### **Retorno**

- Será retornado se foi possível ou não achar alguma informação baseado na expressão regular. Independente do número de resultados, a função irá trazer o primeiro resultado que respeitou a expressão regular utilizada.

# Expressão Regular

---

## Funções do PHP para Expressão Regular

**preg\_match\_all( <expressao>, <string entrada> )**

- Permite que você utilize uma expressão regular para analisar uma string.

### Retorno

- Será retornado todos os resultados que a expressão regular conseguiu encontrar.

# Expressão Regular

---

## Funções do PHP para Expressão Regular

**preg\_replace( <expressao>, <novo valor>, <string entrada> )**

- Permite que você localize e substitua informações baseado em uma expressão regular.

### Retorno

- Retorna a nova string com os valores substituídos.

# Expressão Regular

---

## Funções do PHP para Expressão Regular

**preg\_split( <expressao>, <string entrada> )**

- Permite dividir uma string através de uma expressão regular.

### Retorno

- Retorna um array contendo os elementos que a expressão regular conseguiu dividir.

# Expressão Regular

---

## Funções do PHP para Expressão Regular

**preg\_grep( <expressao>, <array entrada> )**

- Permite analisar todos os elementos de um array, utilizando uma expressão regular

### Retorno

- Retorna um novo array contendo apenas os elementos que foram válidos a expressão regular.

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Expressões Regulares;
- Resolver o **Laboratório** proposto nesta aula;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**

# Segurança - Parte 1

---



# Objetivos da Aula

---

- Cross-site scripting;
- Cross-site request forgery;
- SQL injection;
- Remote injection.
- E-mail Injection;
- Filtros de Entrada e Saída;
- Validadores;
- Criptografia;



# Segurança de Aplicações Web

---

## Cross-site scripting

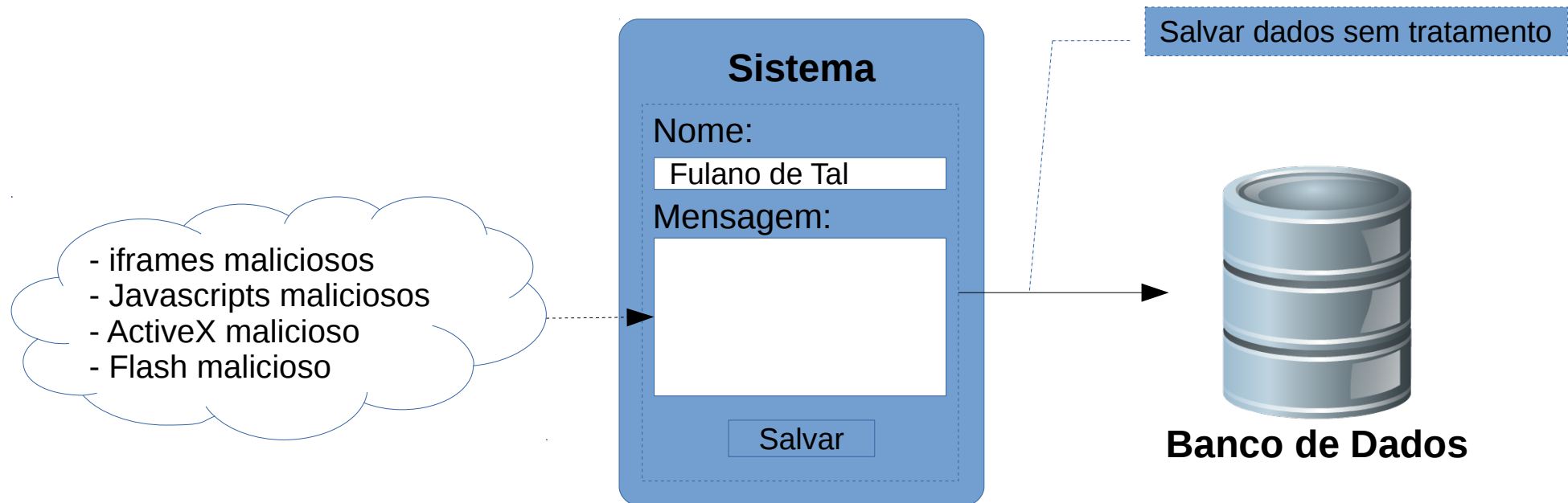
### O que é?

Cross-site scripting, também conhecido como **XSS**, é um tipo de vulnerabilidade de segurança em aplicações web que permite que um usuário injete um script malicioso no lado do cliente, visto por outros usuários.

# Segurança de Aplicações Web

## Cross-site scripting

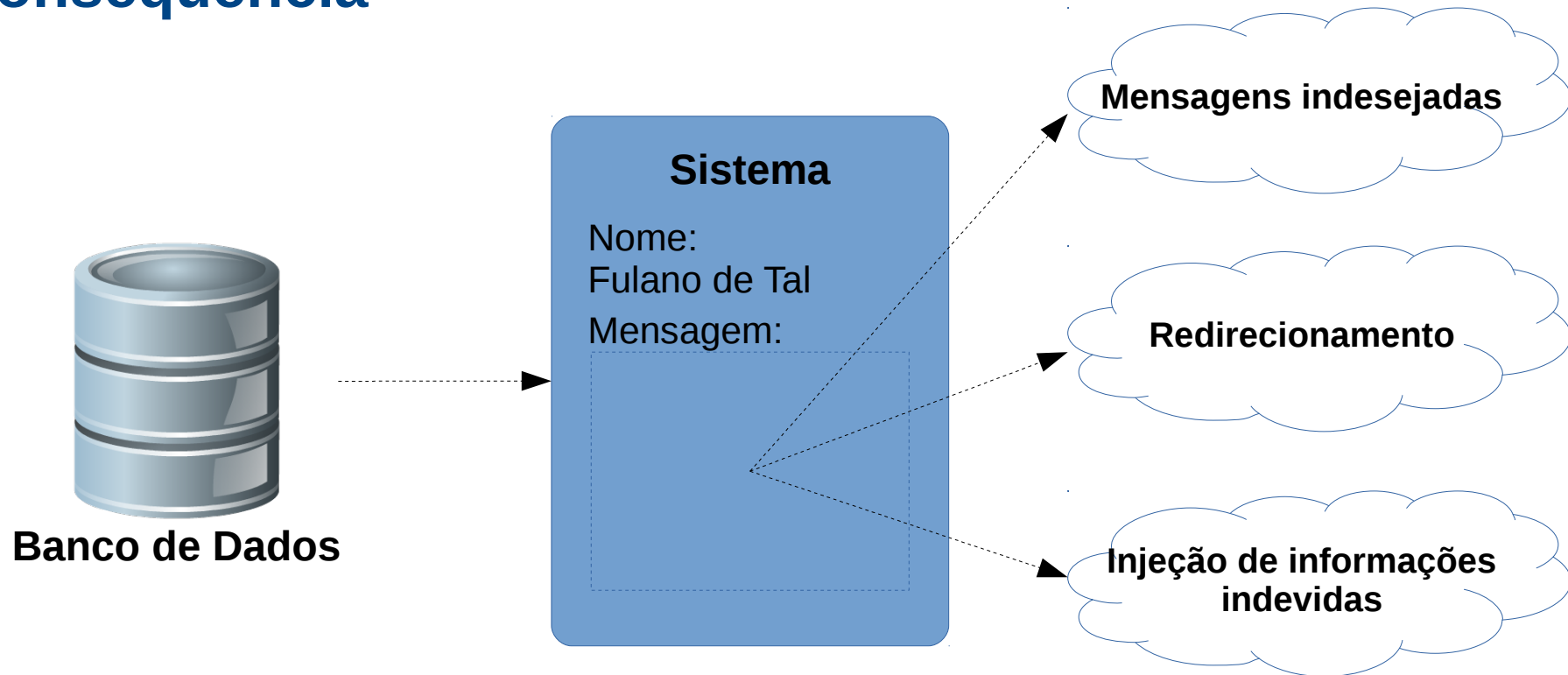
### A causa do problema



# Segurança de Aplicações Web

## Cross-site scripting

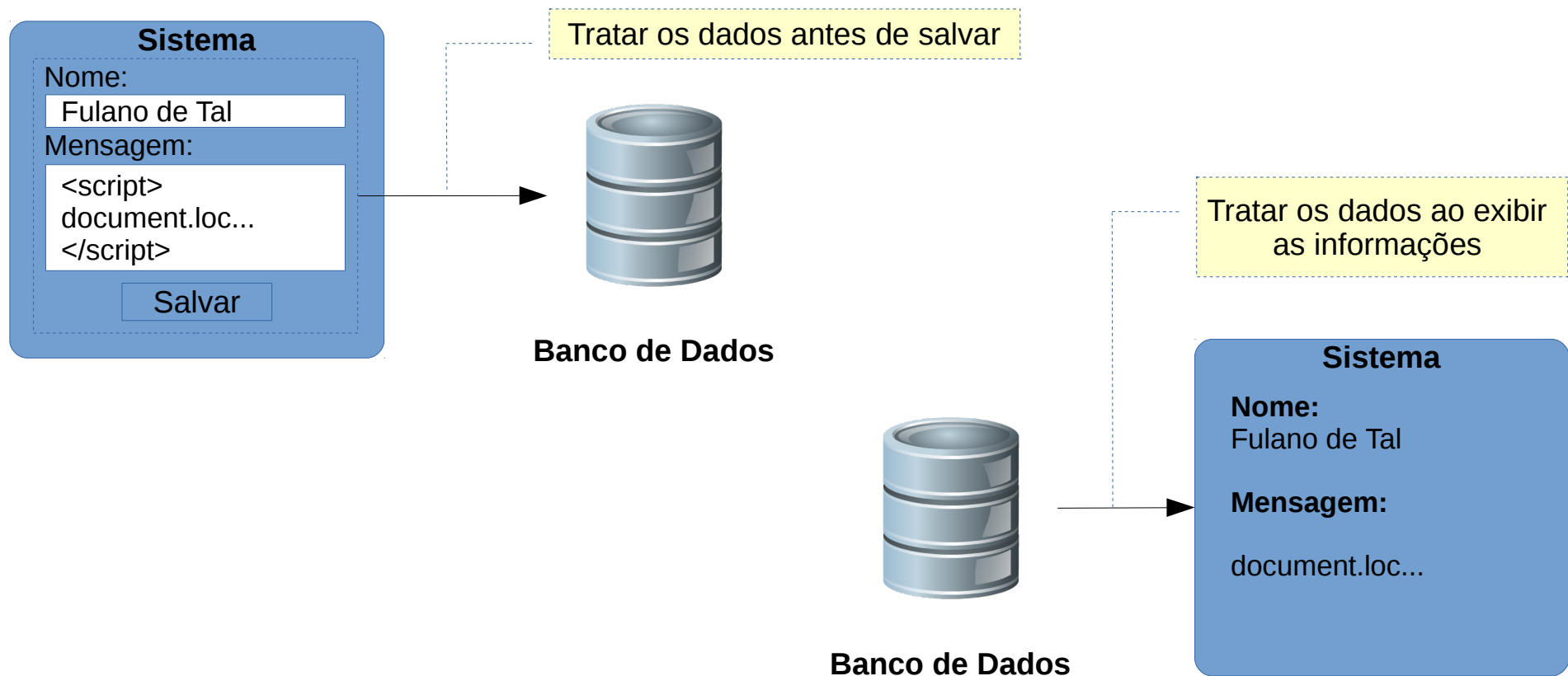
### A consequência



# Segurança de Aplicações Web

## Cross-site scripting

### As soluções



# Segurança de Aplicações Web

---

## Cross-site request forgery

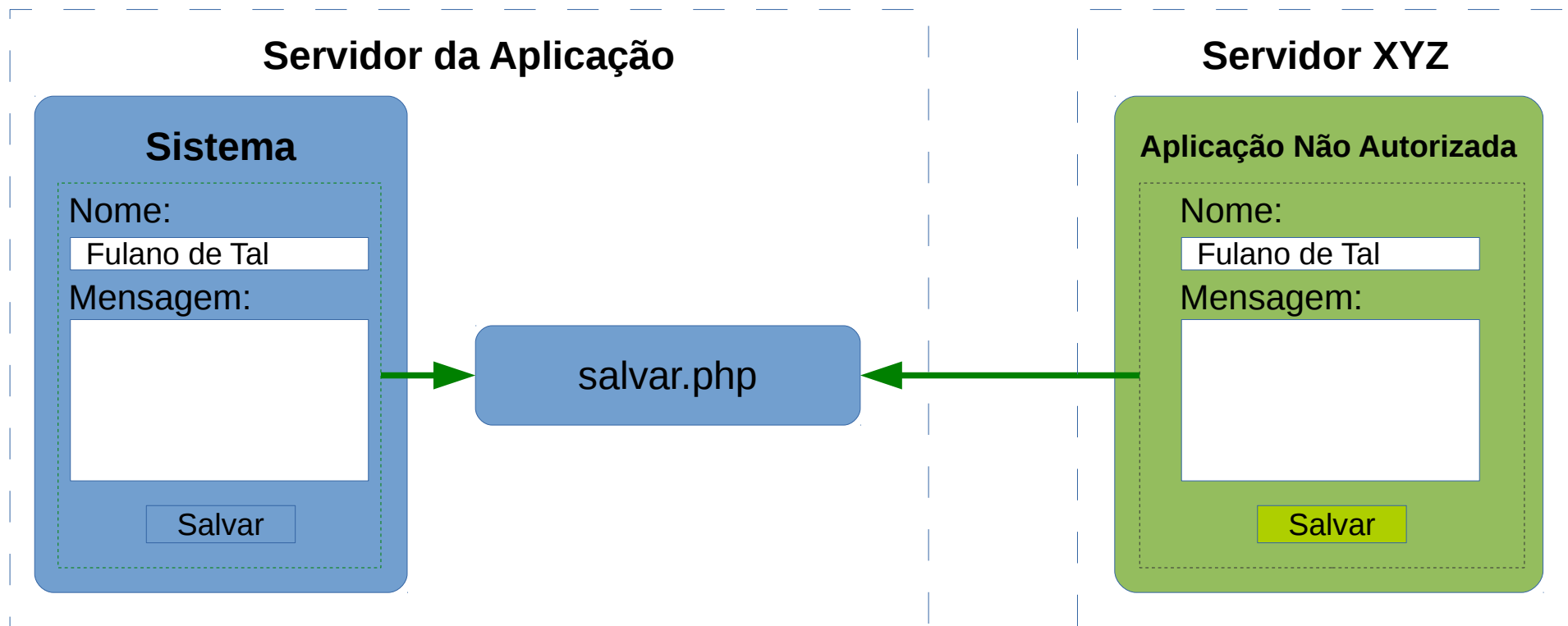
### O que é?

- Cross-site request forgery, também conhecido como **CSRF**, é o ataque que explora a relação de confiança entre aplicativos web e seu usuário legítimo;
- Basicamente uma aplicação não autorizada consegue realizar requisições e executar processos em outra aplicação, como se estas requisições estivessem sendo feitas pela própria aplicação.

# Segurança de Aplicações Web

## Cross-site request forgery

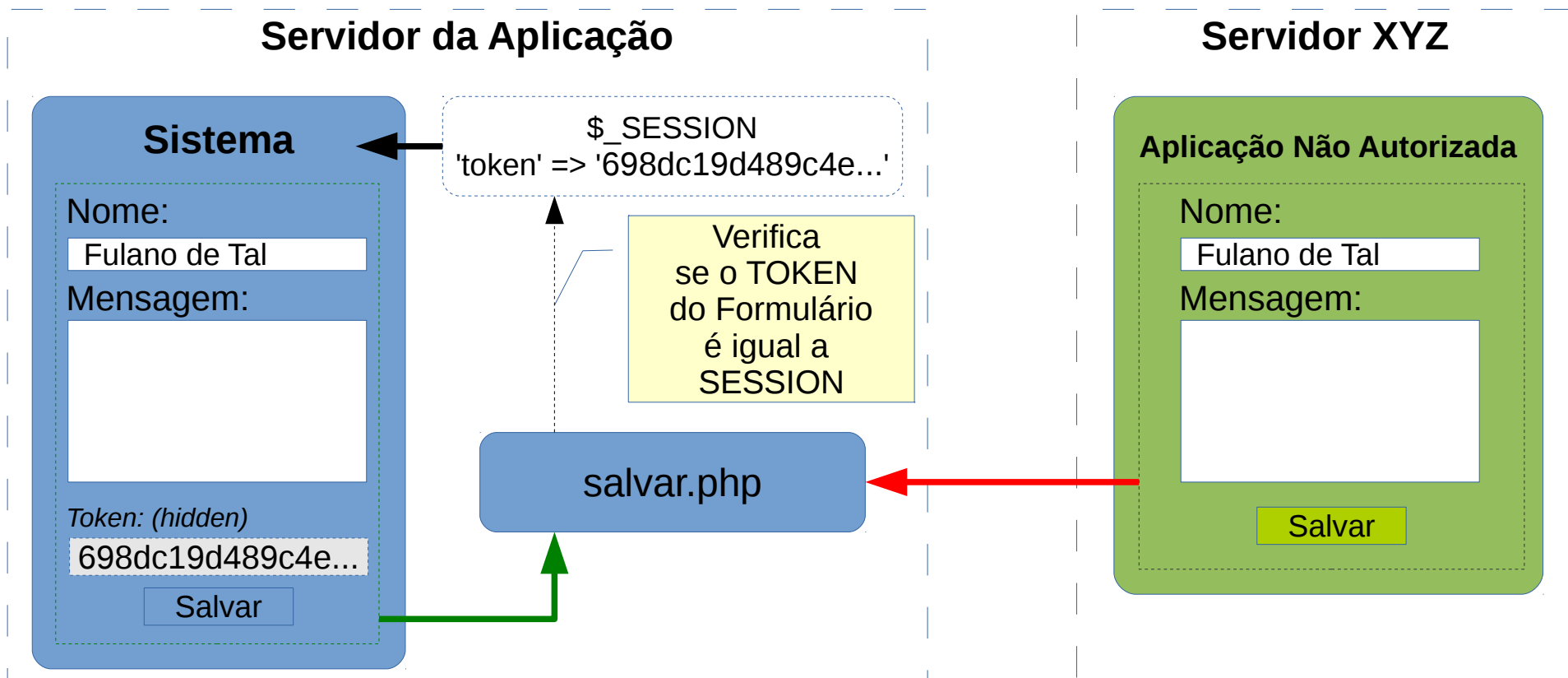
### Entendendo o problema



# Segurança de Aplicações Web

## Cross-site request forgery

### Solucionando o problema



# Segurança de Aplicações Web

---

## SQL Injection

### O que é?

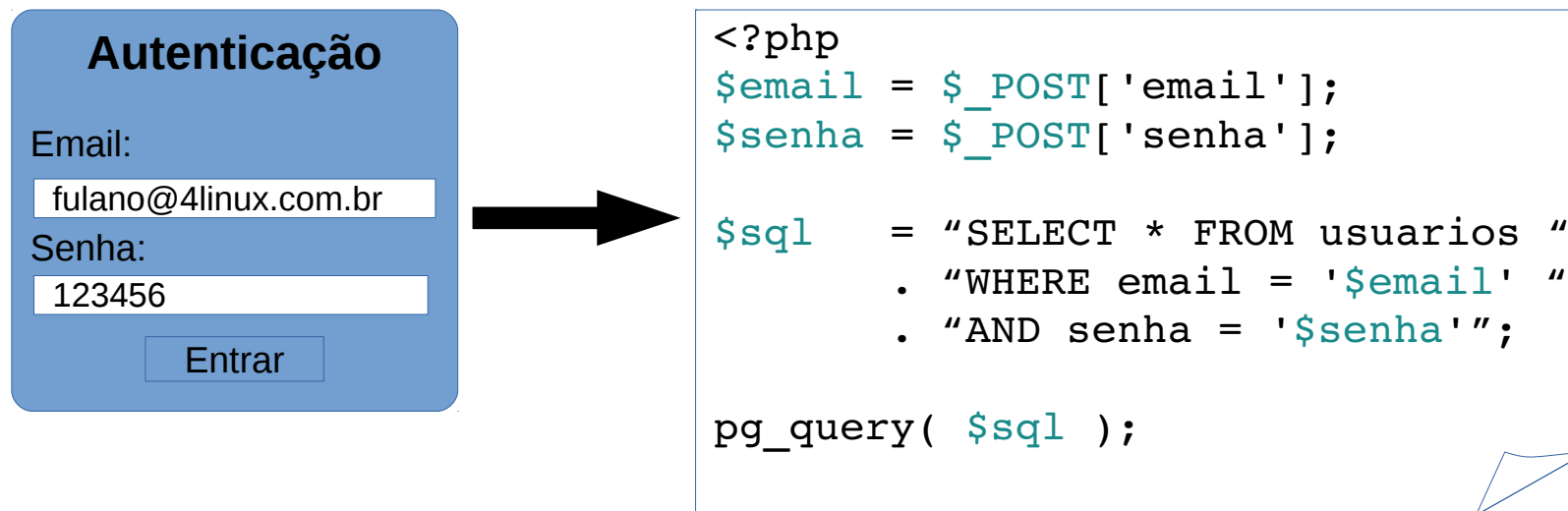
- Uma SQL injection, ou injeção de SQL, é um tipo de ataque que explora uma vulnerabilidade na camada de comunicação com banco de dados em aplicações web;
- Dependendo da maneira como as consultas da aplicação foram construídas e como a entrada do usuário foi utilizada nessas consultas, um usuário pode manipular o comportamento da consulta.



# Segurança de Aplicações Web

## SQL Injection

### Entendendo o problema: sem injection

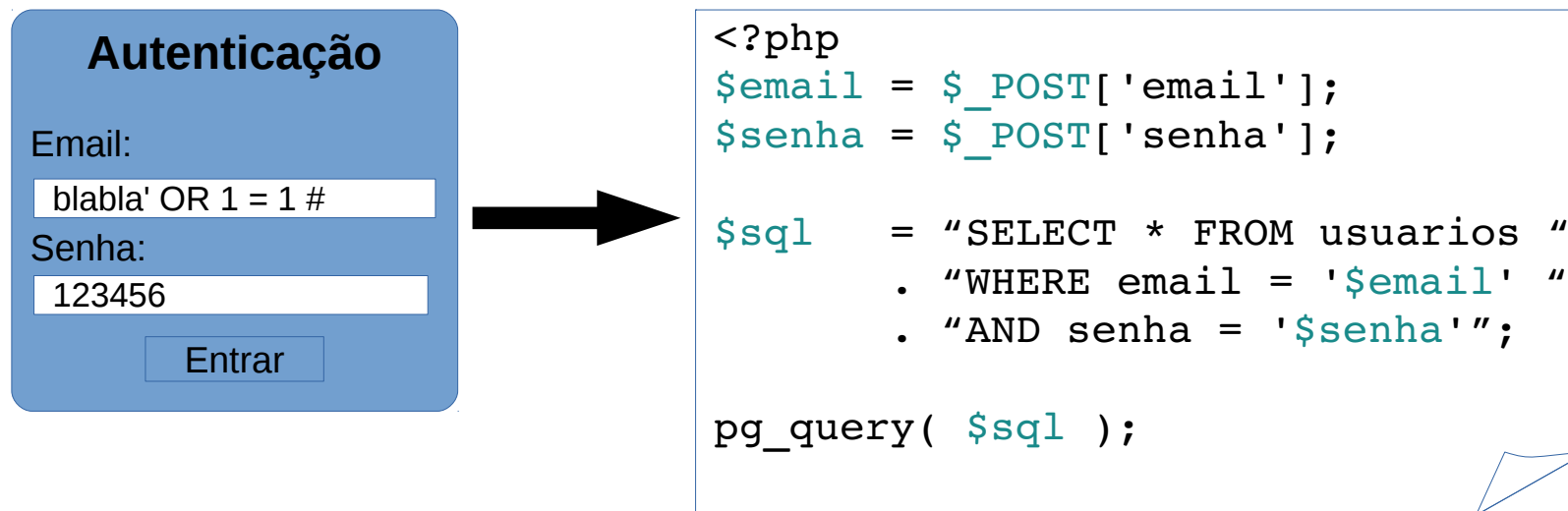


```
SELECT * FROM usuarios WHERE email = 'fulano@4linux.com.br' AND senha = '123456'
```

# Segurança de Aplicações Web

## SQL Injection

### Entendendo o problema: com injection



```
SELECT * FROM usuarios WHERE email = 'blabla' OR 1 = 1 #' AND senha = '123456'
```

# Segurança de Aplicações Web

---

## SQL Injection

### A solução: Prepared Statements

- Prepared statements são comandos SQL pré-construídos, que podem ser manipulados utilizando parâmetros variáveis;
- Seu uso garante que nenhuma query que foi preparada possa sofrer um ataque de SQL injection.

# Segurança de Aplicações Web

## SQL Injection

### A solução

**Autenticação**

Email:

Senha:



```
<?php
$email = $_POST['email'];
$senha = $_POST['senha'];

$sql = "SELECT * FROM usuarios "
      . "WHERE email = '?' "
      . "AND senha = '?'";

$pdo = new PDO('...');

$stmt = $pdo->prepare( $sql );
$stmt->execute( array( $email, $senha ) );
```

# Segurança de Aplicações Web

---

## Remote File Inclusion

### O que é?

- A Remote File Inclusion (ou **RFI**), popularmente conhecida também como **Remote Injection**, é um tipo de vulnerabilidade que explora o mau uso de inclusão de arquivos em uma aplicação web;
- Dependendo da forma como a aplicação foi criada, ela aceita a entrada do usuário para inclusão de arquivos sem filtrar a entrada, gerando uma vulnerabilidade.

# Segurança de Aplicações Web

## Remote File Inclusion

### Entendendo o problema: requisição prevista

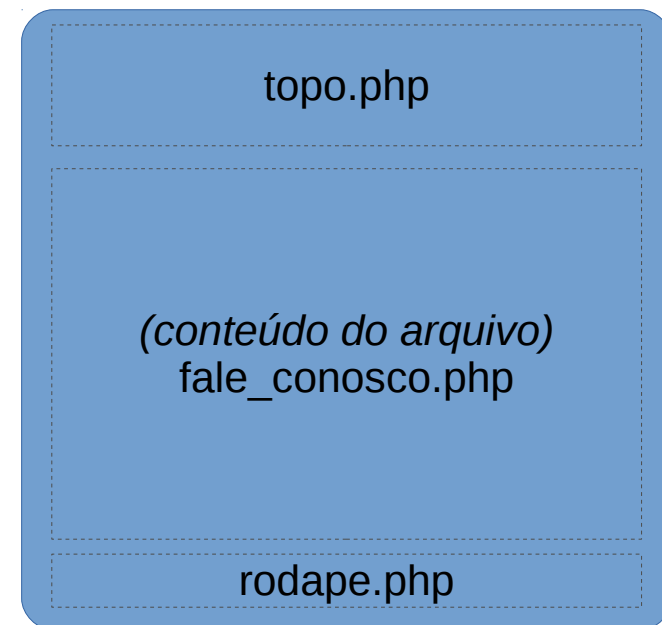
URL: [http://www.seusite.com.br/index.php?pagina=fale\\_conosco.php](http://www.seusite.com.br/index.php?pagina=fale_conosco.php)



```
<?php
include( 'topo.php' );

include( $_GET[ 'pagina' ] );

include( 'rodape.php' );
```



# Segurança de Aplicações Web

## Remote File Inclusion

### Entendendo o problema: requisição maliciosa

URL: <http://www.seusite.com.br/index.php?pagina=http://conteudo.com/pagina.txt>



```
<?php
include( 'topo.php' );

include( $_GET[ 'pagina' ] );

include( 'rodape.php' );
```



# Segurança de Aplicações Web

## Remote File Inclusion

### A solução: verificar a origem do arquivo

URL: <http://www.seusite.com.br/index.php?pagina=http://conteudo.com/pagina.txt>



```
<?php
include( 'topo.php' );

if( file_exists($_GET['pagina'])) {
    include( $_GET[ 'pagina' ] );
} else {
    echo 'Arquivo inválido!';
}

include( 'rodape.php' );
```





# Segurança de Aplicações Web

---

## E-mail Injection

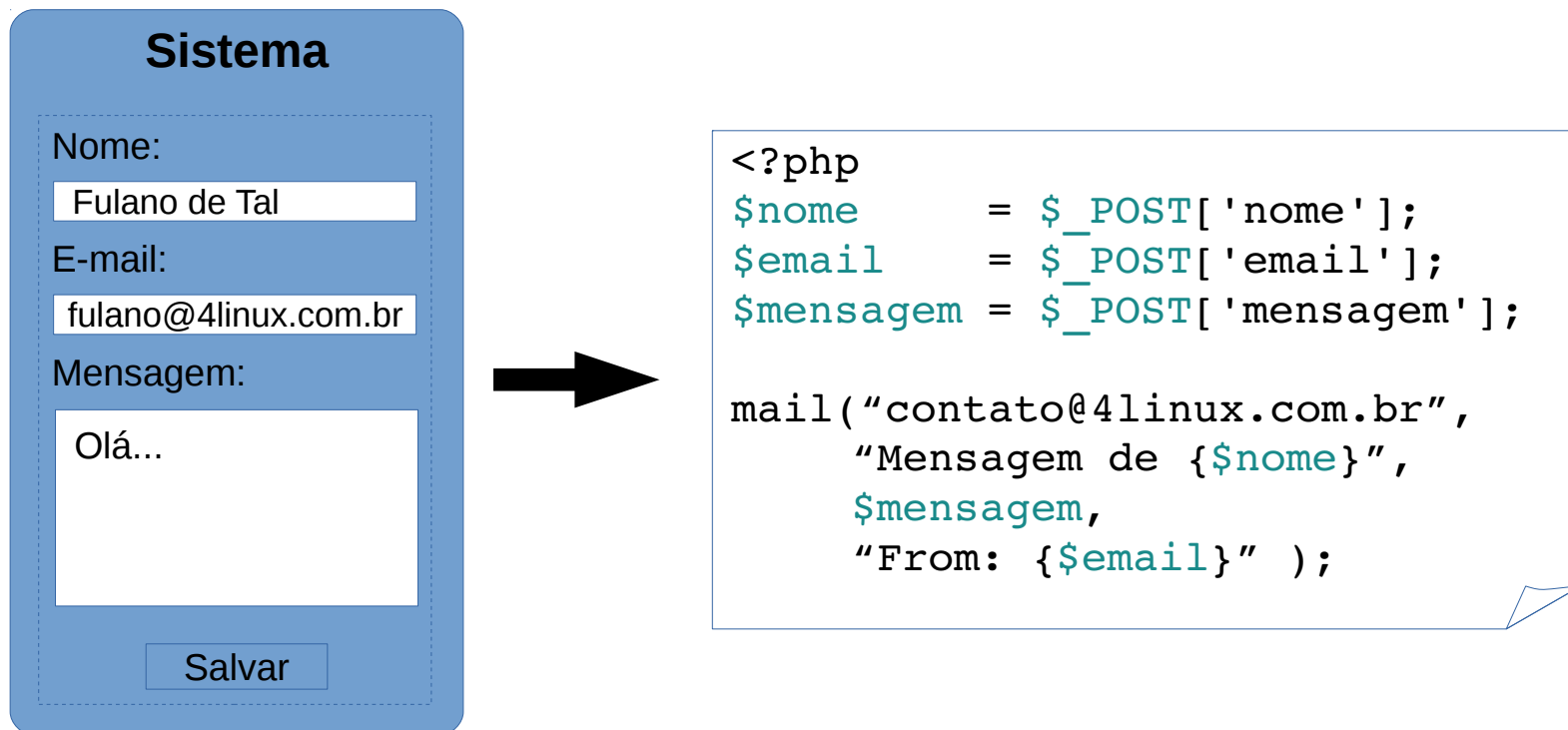
### O que é?

E-mail Injection é um tipo de vulnerabilidade que ocorre em aplicações web que permite o envio de mensagens de e-mail e utiliza entrada de dados do usuário para montar a mensagem.

# Segurança de Aplicações Web

## E-mail Injection

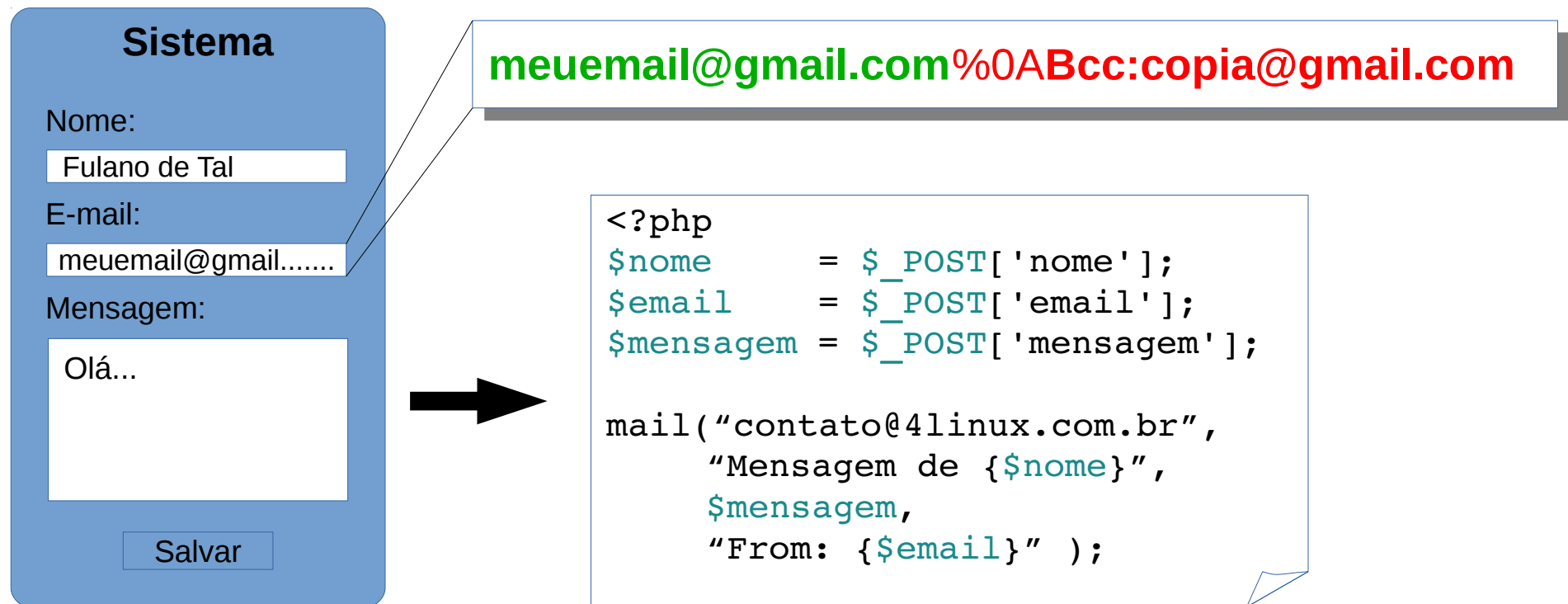
### Entendendo o problema: processamento esperado



# Segurança de Aplicações Web

## E-mail Injection

### Entendendo o problema: processamento malicioso



# Segurança de Aplicações Web

---

## Filtros de Entrada e Saída

### O que são?

- Como pudemos ver, o maior risco durante o desenvolvimento de uma aplicação web é a falta de atenção e cuidado durante o processamento de dados inseridos pelo usuário;
- **Jamais** devemos confiar nas informações vindas de fora da aplicação, independente da fonte: POST, GET, Webservice, RSS, etc;

# Segurança de Aplicações Web

---

## Filtros de Entrada e Saída

### O que são?

- Em outras palavras, se para determinado processamento você espera um valor inteiro, você precisa se certificar que o valor informado pelo usuário é inteiro.

# Segurança de Aplicações Web

---

## Filtros de Entrada e Saída

### Afinal, o que fazer?

- Podemos aplicar expressões regulares, como já conhecemos, para filtrar e validar as informações;
- Mas o PHP possui recursos para filtragem e validação de informações, poupando-nos de ter que desenvolver expressões regulares para tal aplicação;
- Para isso, faremos uso da função **filter\_var**.

# Segurança de Aplicações Web

---

## Filtros de Entrada e Saída

**Função: `filter_var( <valor de entrada>, FILTRO )`**

A função é responsável por aplicar o FILTRO no valor de entrada.

### **Parâmetros:**

valor de entrada: o valor a ser verificado;

FILTRO: constante com o filtro desejado.

# Segurança de Aplicações Web

---

## Filtros de Entrada e Saída

### Exemplos de filtros existentes

- **FILTER\_SANITIZE\_EMAIL**: remove todos os caracteres, exceto letras, dígitos `!#$%&'*+,-/=/?^_{}~@[];`
- **FILTER\_SANITIZE\_NUMBER\_INT**: remove todos os caracteres, exceto dígitos e sinais de '+' e 'i';
- **FILTER\_SANITIZE\_STRING**: remove tags



# Segurança de Aplicações Web

---

## Validadores

### Exemplos de validadores

- **FILTER\_VALIDATE\_EMAIL**: retorna true se o endereço de e-mail for válido;
- **FILTER\_VALIDATE\_INT**: retorna true se o valor for um número inteiro;
- **FILTER\_VALIDATE\_BOOLEAN**: retorna TRUE para valores '1', 'true', 'on' e 'yes'. Retorna FALSE para qualquer outro valor;

# Segurança de Aplicações Web

---

## Criptografia

- Simplificadamente, usamos encriptação para converter dados sensíveis para dados ininteligíveis. Isto ajuda a proteger a privacidade quando mandando dados de um remetente à um destinatário. Do lado do destinatário, os dados podem ser descriptografados para voltarem à sua forma original;
- Tipos de criptografia:
  - **Hash Functions, Chaves públicas/privadas, Hash functions + Salt**

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Segurança;
- Responder as questões do **Teste de Conhecimento**.

**Mãos à obra!**

# A Certificação Zend

---



# Objetivos da Aula

---

- Sobre a Certificação
- Porque se certificar?
- Como fazer a prova?
- Como a prova é aplicada?
- O que esperar após a prova?

# A Certificação Zend

---

## Porque se certificar?

### Seja um profissional com destaque no mercado!

- Sim, você ganhará muito mais que um simples título, você será um profissional com destaque no mercado;
- Como um Zend Certified PHP Engineer você fará parte de uma comunidade que está crescendo a cada dia, além de mostrar que você sabe do que está falando e sabe desenvolver aplicações de qualidade para web;
- As empresas buscam profissionais assim!

# A Certificação Zend

---

## Como fazer a prova?

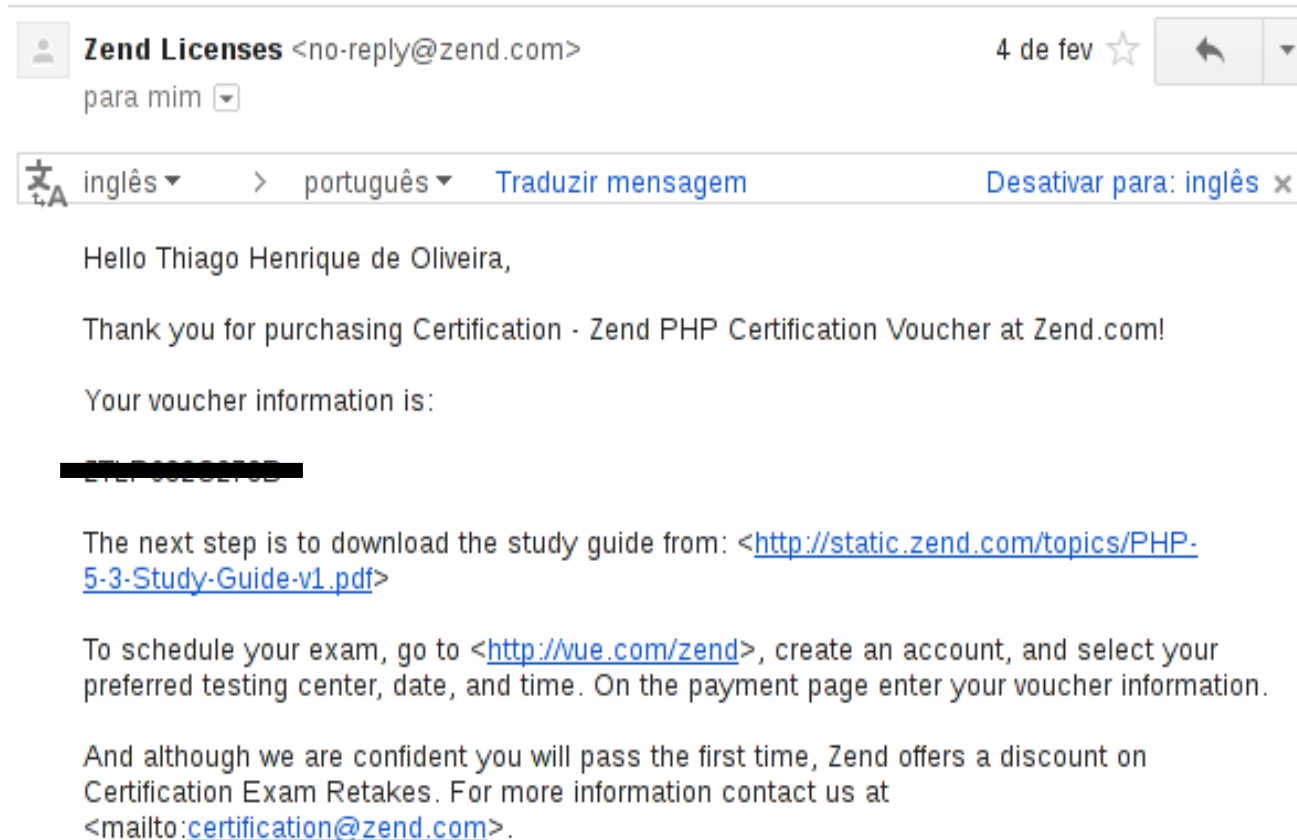
### 1º Passo: A compra do Voucher

- Onde comprar: <http://www.zend.com>
- Valor: \$195,00
- Validade: após a compra você tem 1 ano para fazer a prova.

# A Certificação Zend

## Como fazer a prova?

### 1º Passo: A compra do Voucher



Código do Voucher



# A Certificação Zend

---

## Como fazer a prova?

### 2º Passo: Criando conta na PearsonVue

- Onde: <http://www.pearsonvue.com>
- Ao criar a conta, caso você ainda não possua, será solicitado o número do Voucher adquirido no site da Zend;
- Após criar a conta, será solicitado que aguarde ~ 24 horas para a conta ser ativada;
- Assim que a conta estiver ativada, será enviado um e-mail com usuário e senha de acesso ao site da PearsonVue.

# A Certificação Zend

---

## Como fazer a prova?

### 3º Passo: Agendando a prova

- Onde: <http://www.pearsonvue.com>
- Autentique-se com o usuário e senha enviado pela PearsonVue;
- O agendamento é online, selecione o exame que desejará fazer, e em seguida será apresentado uma consulta dos Centros credenciados para aplicar a prova;
- Após escolha do local, será apresentado os dias e horários disponíveis no Centro escolhido.

# A Certificação Zend

---

## Como fazer a prova?

### 4º Passo: O dia da prova

- Leve junto com você, **dois** documentos com foto!

*Não se esqueça disso, é uma exigência da PearsonVue; sem os dois documentos você não será autorizado a fazer a prova.*

# A Certificação Zend

---

## Como a prova é aplicada?

### Características da prova

- Duração: 90 minutos
- Total de questões: 70
- Idioma: Inglês

# A Certificação Zend

---

## Como a prova é aplicada?

### Características da prova

- Tipos de questões:
  - **Dissertativas:** não exigem mais do que nome de funções, ou ainda, resultado da execução de algum trecho de código;
  - **Múltipla-escolha:** questões com 1, 2 até 3 alternativas corretas, fique tranquilo, a questão indica quantas questões é esperada para ser selecionada. Ex.: “(Choose 2)”

# A Certificação Zend

---

## Como a prova é aplicada?

### Características da prova

- NÃO é permitido uso de quaisquer tipo de fonte de pesquisa;
- A única ferramenta disponibilizada, é uma pequena lousa e uma caneta para auxiliar na resolução das questões;
- E o resultado sai na hora! (*Ufa!*)

# A Certificação Zend

---

## O que esperar após a prova?

- PHP BASICS;
- DATA FORMATS AND TYPES;
- STRINGS;
- INPUT / OUTPUT;
- OBJECT-ORIENTED PROGRAMMING;
- DATABASES;
- SECURITY;
- WEB FEATURES;

# A Certificação Zend

---

## ➤ **Maior Ênfase**

- **PHP BASICS;**
- **SECURITY;**
- **OOP**

## ➤ **Ênfase Média;**

- **FUNCTIONS;**
- **WEB FEATURES;**
- **ARRAYS;**
- **STRINGS & PATTERNS**

## ➤ **Ênfase baixa**

- **DATABASES;**
- **DATA FORMATS & TYPES;**
- **INPUT / OUTPUT**



# A Certificação Zend

---

## ➤ PHP BASICS

- Sintaxe;
- Operadores;
- Variáveis;
- Estrutura de Controle;
- Funções;
- Constantes;
- Namespaces;

- Extensões;
- Configuração;
- Performance;

# A Certificação Zend

---

What is the output of the following code?

```
<?php
```

```
$a = 1;
```

```
++ $a;
```

```
$a *= $a;
```

```
echo $a --;
```

A: 4 ✓

B: 3

C: 5

D: 0

E: 1

# A Certificação Zend

---

**When PHP is running on a command line, what super-global will contain the command line arguments specified?**

- A: `$_SERVER` ✓
- B: `$_ENV`
- C: `$GLOBALS`
- D: `$_POST`
- E: `$_ARGV`

# A Certificação Zend

---

What is the name of the error level constant that is used to designate PHP code that will not work in future versions?

**E\_DEPRECATED**

# A Certificação Zend

---

## ➤ Data Formats & Types

- XML Básico;
- XML Extension;
- Simple XML;
- DOM;
- SOAP;
- REST;
- Json / Ajax;

- DATE / TIME;

# A Certificação Zend

---

What is wrong with this XML document?

```
<?xml version="1.0" encoding="UTF-8"?>
<node>
<?var type="string" ?>
<leaf>Value</leaf>
</node>
```

A: The encoding is only required for non-western languages

B: <?var is not a valid node type


C: <?var is  ssing a closing tag

D: Nothing

# A Certificação Zend

---

**Which of the following is a feature that does NOT exist in the DateTime extension?**

- A: The ability to list time zones by continent
- B: The ability to modify date data
- C: The ability to generate dates between two date periods
- D: The ability to parse  dates in the cookie format
- E: None of the above

# A Certificação Zend

---

**What is the name of the method that allows xpath expressions in SimpleXML?**

A: There is no such method

B: XPath

C: xpathExpression ✓

D: xpath




# A Certificação Zend

---

## What is JSON?

A: A way of serializing any PHP type in order to exchange it with different programming languages and systems

B: A portable XML representation of the data using PHP's  
`serialize($value, true)`

C: A format to represent any PHP type, except a resource, that  can be used later on in JavaScript or other languages

# A Certificação Zend

---

## ➤ Strings

- Quoting;
- HEREDOC / NOWDOC;
- Matching;
- Searching;
- Replacing;
- PCRE;
- Encoding;


# A Certificação Zend

---

**What is a good rule to follow when quoting string data?**

A: Use double quotes because you might want to use variable interpolation at a later time.

B: Use single quotes unless you are using variable interpolation because single quotes are faster.

C: Use single quotes unless you have a ' in your string or  you are doing variable interpolation because it declares whether you want variables to be interpolated

# A Certificação Zend

---

What will the following code print out?

```
$str = 'aletra';  
if(strpos($str, 'a')){  
    echo 'Found the letter a';  
}else{  
    echo 'Could not find the letter a';  
}
```

Could not find the letter a.

# A Certificação Zend

---

What will this code do?

```
$var = 2;  
$str = 'aabbccddeeaabbccdd';  
echo str_replace('a', 'z', $str, $var);
```

A: Replace all of the 'a' characters with 'z' characters and put the replacement count in \$var. ✓

B: Replace up to 2 of the 'a' characters with a 'z' character.

C: 2 is a flag which, when passed to str\_replace, will remove all characters \_except\_ those listed

# A Certificação Zend

---

## ➤ **ARRAYS**

- **Enumerated Arrays;**
- **Associative Arrays;**
- **Multi-Dimensional;**
- **Array Iteration;**
- **Array Functions;**
- **SPL / Objects as Arrays;**

# A Certificação Zend

---

What is the output of the following code ?

```
$a = array(  
    "1" => "A", 1 => "B", "C", 2 => "D" );  
  
echo count($a);
```

**Resposta : 2**

# A Certificação Zend

---

**Read carefully: Which interface can be used to allow an object to be executed in a foreach loop?**

- A:  
ArrayObject ✓
- B: Iterator
- C: ArrayList
- D: Hashtable



# A Certificação Zend

---

Given the following PHP code, which of these answers creates a valid associative array in PHP?

```
$one = [ 'one', 'two', 'three' ];  
$two = [ 1, 2, 3 ];
```

- A: array\_combine(\$one, two) ✓
- B: array\_merge(\$one, two)
- C: array\_values(\$two)
- D: array\_flip(\$one)

# A Certificação Zend

---

## ➤ INPUT / OUTPUT

- Files;
- FileSystem Functions;
- Streams;
- Contexts;
- Reading;
- Writing;

# A Certificação Zend

---

Which function can be used to read and parse data from a CSV file?

**fgetcsv**

# A Certificação Zend

---

**What is the output of the following function call (assuming that foo.txt exists and contains text)?**

A: A file handle that can be used in subsequent calls such as fread

B: True if the file could successfully be read, false if not

C: A string containing the contents of foo.txt



D: An array where every entry is a line from the file foo.txt

E. True if the file exists, false if not

# A Certificação Zend

---

Consider the following snippet of code. What is the name of the function that needs to be inserted in the placeholder?

**readdir**

# A Certificação Zend

---

## ➤ FUNCTIONS

- Syntax;
- Arguments;
- Variables;
- References;
- Returns;
- Variable Scope;
- Anonymous Functions(Closures);

# A Certificação Zend

---


Take a look at the following code...

```
function myFunction($a)
{
    $a ++;
}

$b = 1;

myFunction($b);
```

**What code do you need to replace so that \$b has the value 2 at the end of the script?**

- A: Line 02: Replace \$a with &\$a
- B: Line 03: Replace \$a++ with \$a +=2;
- C: Line 03: Replace \$a++ with  \$a \*=2;
- D: Line 06: Replace \$b with &\$b

# A Certificação Zend

---

What is the output of the following code?

```
function increment($a){  
    return ++$a;  
}
```

```
echo increment(1);
```

**Resposta : 2**



# A Certificação Zend

---

**What is the best way to test if \$param is an anonymous function in a method?**

- A: Use `method_exists($param, ')`
- B: Use `is_callable($param)` ✓
- C: Use the type-hint Closure on the signature
- D: Use `is_executable($param)`

# A Certificação Zend

---

## ➤ OOP

- Instatiation;
- Instance Methods & Properties;
- Class Definitions;
- Modifiers / Inheritance Abstracts;
- Interfaces;
- Exceptions;
- Static Methods & Properties;

# A Certificação Zend

---

## ➤ OOP

- Autoload;
- Reflection;
- Type Hinting;
- Class Constants;
- Late Static Binding;
- Magic Methods;
- SPL / Traits;

# A Certificação Zend

---

**What is the relationship between classes and objects?**

A: A class is a collection of objects

B: A class is a template from which objects are made ✓

C: Objects are distinguished from one another by assigning them to a class

# A Certificação Zend

---

**Which of the following statements about static functions is true?**

- A: Static functions can only access static properties of the class. ✓
- B: Static functions cannot be called from non static functions.
- C: Static functions cannot be abstract.
- D: Static functions cannot be inherited.

# A Certificação Zend

---

**Which of the following CANNOT be a part of the class definition?**

A: Constant

B: Variable

C: Function

D: Interface ✓

# A Certificação Zend

---

**Which is the correct syntax to define a class constant for the class MyClass?**

A: `const $NAME = "value";`

B: `define("MyClass::NAME", "value");`

C: `const NAME = 'value';` ✓

D: `static final $NAME = 'value';`

# A Certificação Zend

---

## ➤ DATABASES

- SQL;
- JOINS;
- Analyzing Queries;
- Prepared Statements;
- Transactions;
- PDO;



# A Certificação Zend

---

**Which is the most secure approach for handling dynamic data in SQL queries?**

A: Enable `magic_quotes_gpc`.

B: Use prepared statements if supported by the database library, data-specific escaping functions otherwise. ✓

C: Use stored procedures.

D: Use `addslashes()`.

# A Certificação Zend

---

**When comparing prepared statements and regular, application-constructed SQL statements, which of the following is true?**

- A: Prepared statements are faster
- B: None of the above
- C: Prepared statements are easier to develop
- D: Prepared statements are more secure ✓
- E: Prepared statements are always shorter

# A Certificação Zend

---

**Some databases support the LIMIT clause. Is a method to ensure that...**

A: only certain users can access the database.

B: only certain rows are deleted in DELETE queries.

C: only a defined subset of rows are read in SELECT queries. ✓

# A Certificação Zend

---

## ➤ SECURITY

- Configuration;
- Session Security;
- Cross-Site Scripting;
- Cross-Site Request Forgeries;
- SQL Injection;
- Remote Code Injection;
- Email Injection;

# A Certificação Zend

---

## ➤ SECURITY

- Input Filtering;
- Escape Output;
- Password Hashing API;
- File Uploads;
- Data Storage;
- SSL;

# A Certificação Zend

---

**What is the recommended setting for error\_reporting for production servers?**

- A: E\_ALL & ~E\_DEPRECATED & ~E\_STRICT ✓
- B: E\_ALL & ~E\_NOTICE
- C: E\_STRICT
- D: OFF

# A Certificação Zend

---

**Which of the following measures provides good protection against Cross-Site Request Forgery?**

- A: Relying on HTTP POST only
- B: Relying on HTTP reference header
- C: Relying on a one-time token ✓
- D: Relying on the user agent

# A Certificação Zend

---

**Escaping output may help protect from which common security vulnerabilities? (Choose 2)**

- A: Clickjacking
- B: Cross-Site Scripting ✓
- C: Cross-Site Request Forgery
- D: SQL Injection ✓



# A Certificação Zend

---

**What does the `max_file_uploads` configuration option contain?**

- A: The maximum number of file uploads per session
- B: The maximum number of file uploads per request ✓
- C: The maximum number of file uploads per user
- D: The maximum number of file uploads before the web service process is restarted

# A Certificação Zend

---

## ➤ WEB FEATURES

- Sessions;
- Forms;
- GET and POST Data;
- File Uploads;
- Cookies;
- HTTP Headers and Code;

# A Certificação Zend

---

**What is the default timeout of a PHP session cookie?**

- A: Depends on the web server
- B: 10 minutes
- C: 20 minutes
- D: Until the browser is closed ✓

# A Certificação Zend

---

**If a form's action attribute is set to an empty string, where is data usually sent to?**

A: /

B: the current URI 

C: index.php

D: the default page of the current directory

# A Certificação Zend

---

**Which class of HTTP status codes is used for error conditions?**

A: 1XX

B: 3XX

C: 5XX 

# A Certificação Zend

---

## O que esperar após a prova?

### Após alcançar o sonhado “*Congratulations*”

- Você receberá um documento informando que passou;
- Poderá fazer uso do logo ZCE / ZCPE em projetos, cartões de visita, etc;
- Em até 48 horas, será liberado seu perfil na YellowPage no site da Zend (o *hall* da fama dos programadores PHP);
- Em até 48 horas, receberá um e-mail com uma licença perpétua do ZendStudio;

# A Certificação Zend

---

## O que esperar após a prova?

### Após alcançar o sonhado “*Congratulations*”

- E por fim, em 1 mês, aproximadamente, receberá uma correspondência com o Certificado impresso.

# Próximos Passos

---

Para que você tenha um melhor aproveitamento do curso, participe das seguintes atividades disponíveis no Netclass:

- Fazer a leitura da **Apostila** Certificação;
- Pegue o **Simulado** e conheça os tipos de questões que poderão cair.

**Mãos à obra!**



# 4LINUX

OPEN SOFTWARE SPECIALISTS



**ESPECIALISTA EM "JUNTAR AS PEÇAS" DO MUNDO OPEN SOURCE**

[WWW.4LINUX.COM.BR](http://WWW.4LINUX.COM.BR)