

A background of red curtains, slightly parted in the center, creating a stage-like atmosphere. The curtains are a deep red color with visible folds and texture.

Da argila ao forte: como desenvolver uma loja virtual

Michael Granados - @dgmike



O ambiente

O ambiente

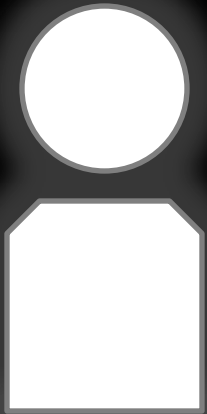
- Sistema Operacional (Windows, Linux, Mac)
- Linguagem (PHP, óbvio)
- Banco de dados (mysql, postgree, sqlite)
- Controle de versão (SVN, GIT, Mercurial)

Controle de Versão

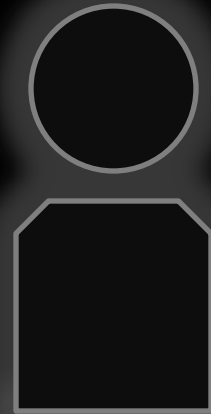


Repositório
FTP

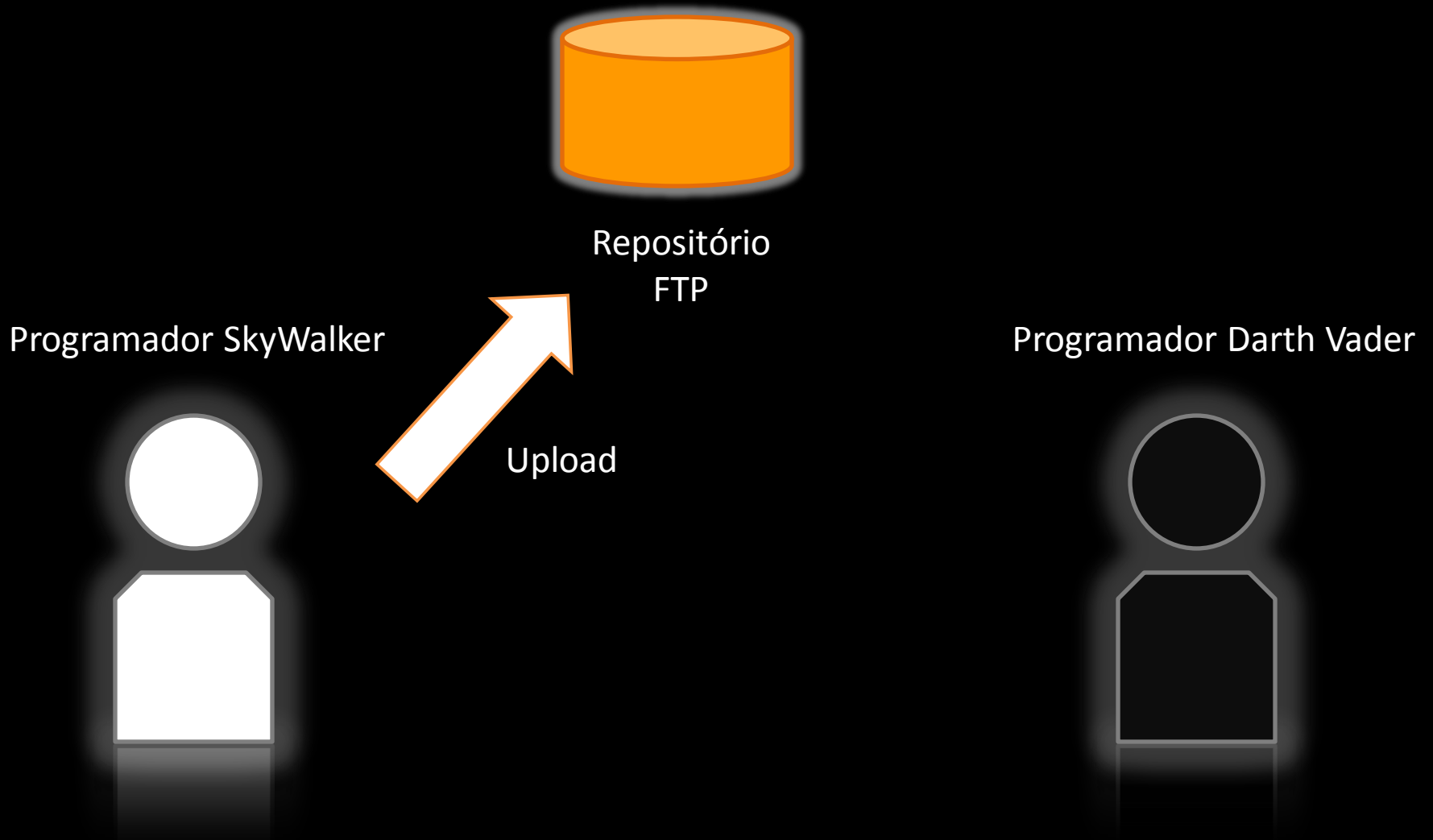
Programador SkyWalker



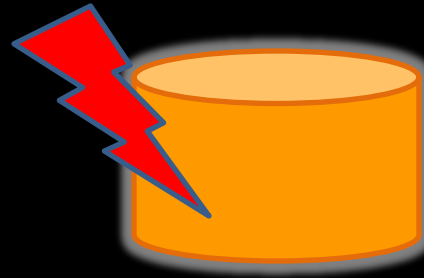
Programador Darth Vader



Controle de Versão



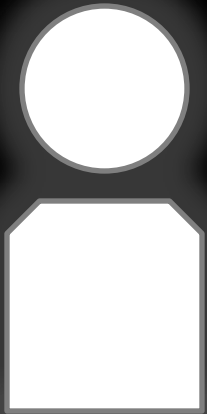
Controle de Versão



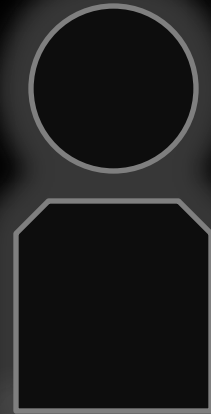
Arruma o bug online

Repositório
FTP

Programador SkyWalker



Programador Darth Vader

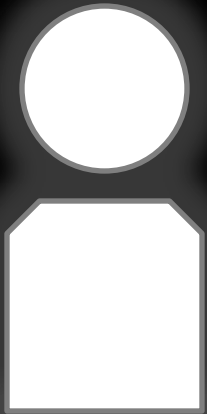


Controle de Versão

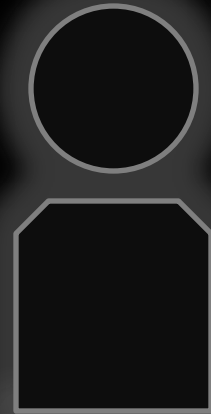


Repositório
FTP

Programador SkyWalker



Programador Darth Vader



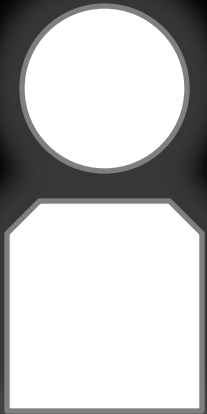
Upload

Controle de Versão

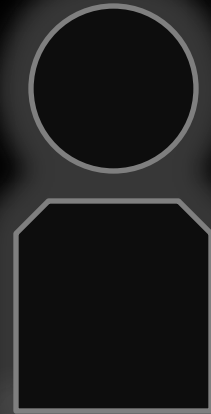


Repositório
SVN

Programador Skywalker



Programador Darth Vader

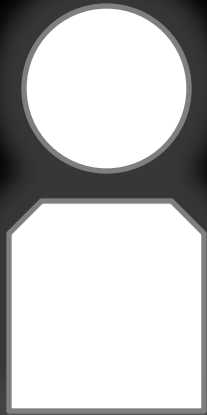


Controle de Versão

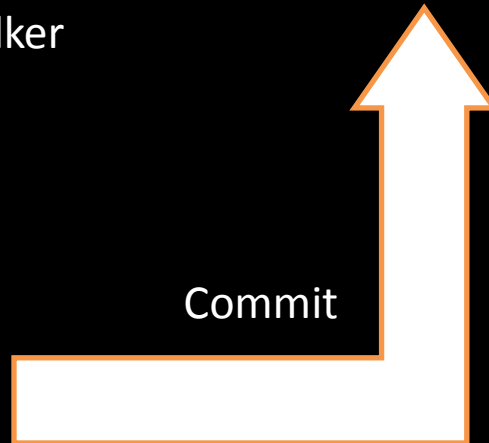
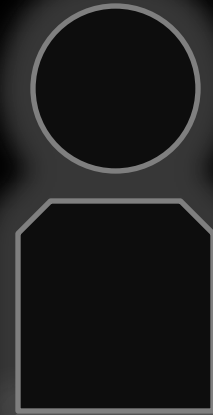


Repositório
SVN

Programador SkyWalker



Programador Darth Vader



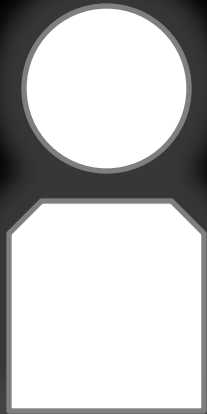
Commit

Controle de Versão

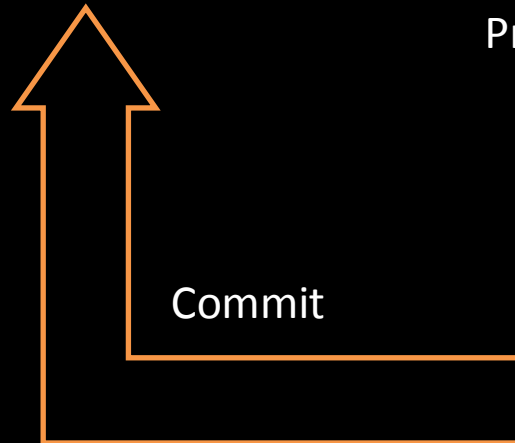
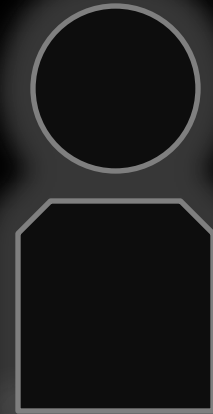


Repositório
SVN

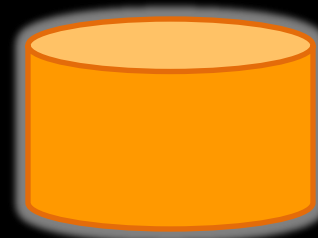
Programador SkyWalker



Programador Darth Vader

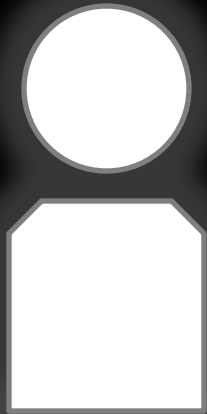


Controle de Versão

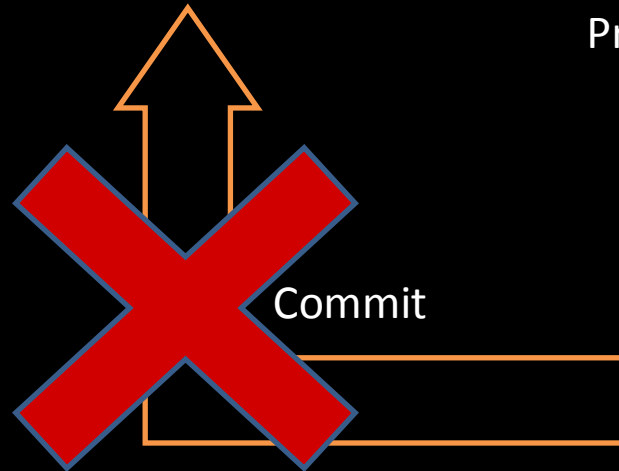
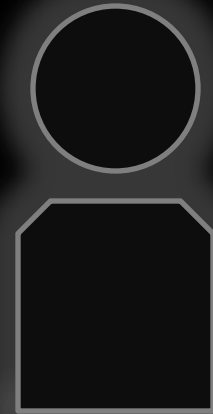


Repositório
SVN

Programador SkyWalker

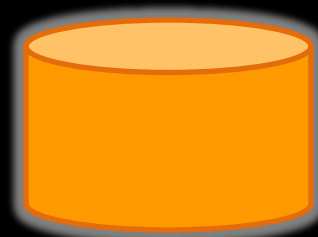


Programador Darth Vader

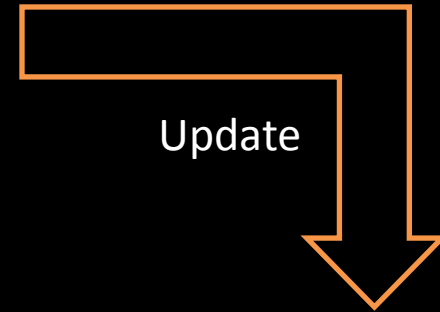


Commit

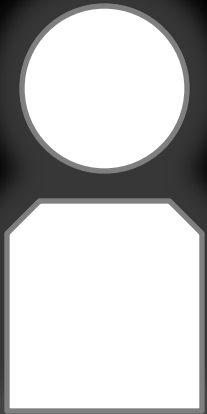
Controle de Versão



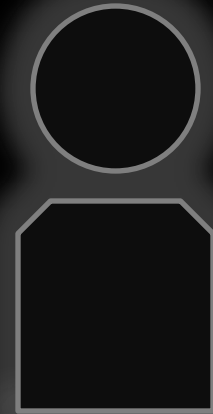
Repositório
SVN



Programador SkyWalker



Programador Darth Vader

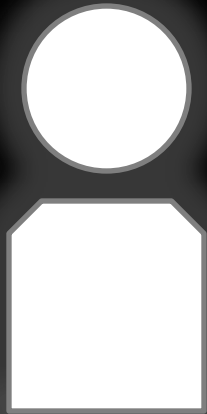


Controle de Versão

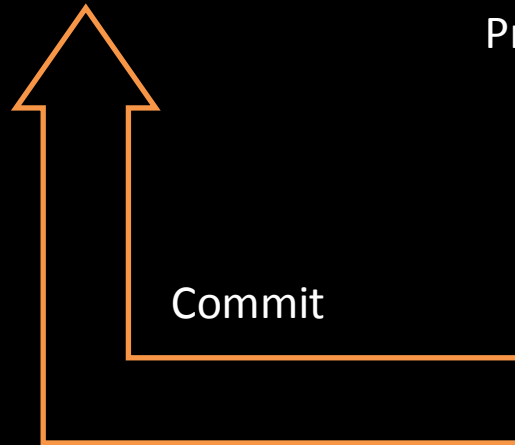
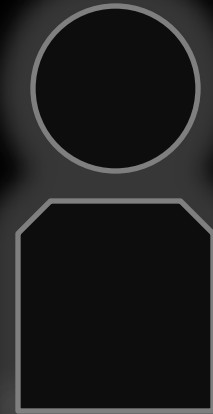


Repositório
SVN

Programador SkyWalker

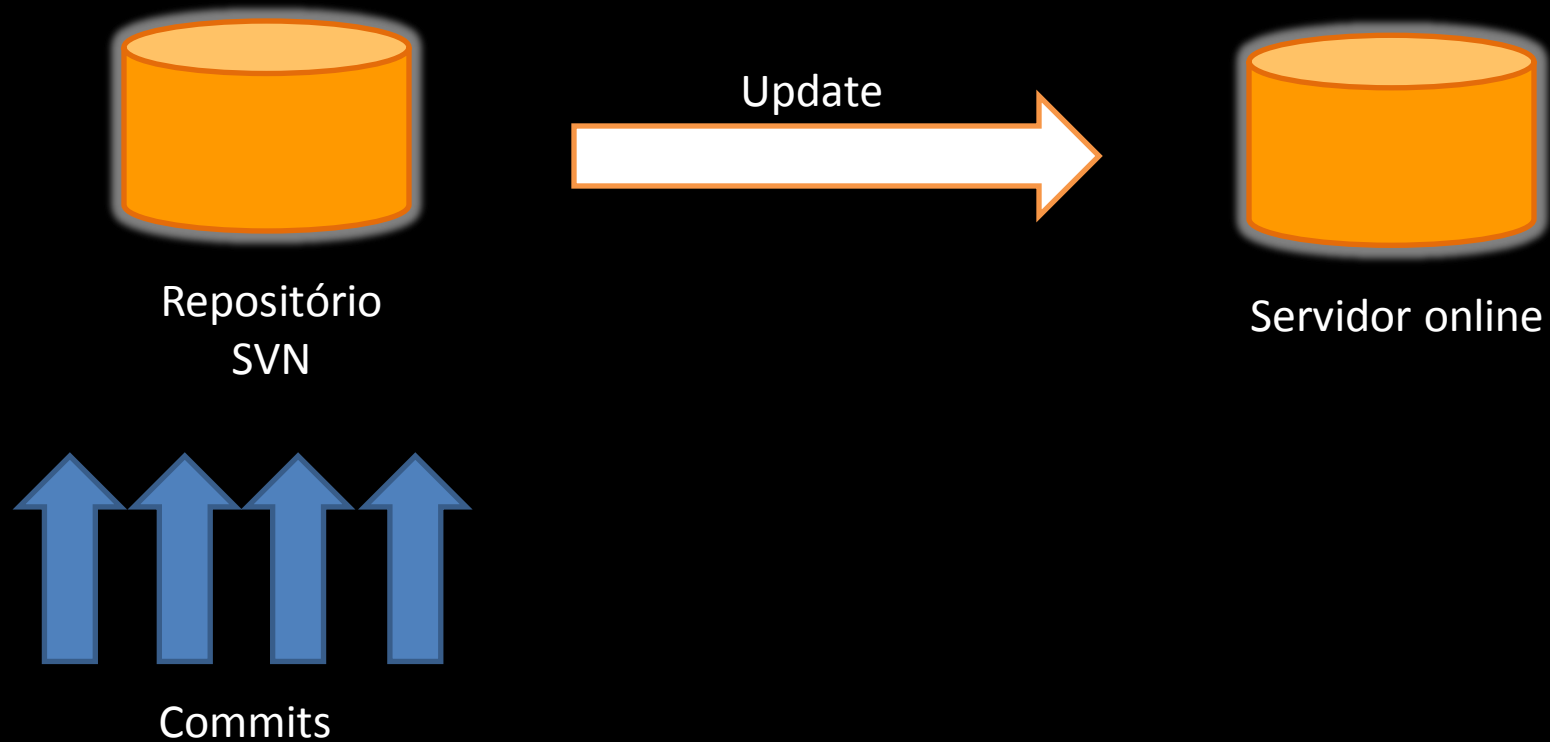


Programador Darth Vader



Commit

Controle de Versão



Ferramenta de Banco: PDO

- Consistente
- Um comando, vários bancos
- Fácil de usar

```
new PDO('mysql:dbname=banco;host=localhost',  
'usuario', 'senha');
```

```
new PDO('sqlite:/opt/database/mydb.sqlite');
```

O MVC

- Model (manipula o banco de dados)
- View (as telas em si)
- Controller (quem faz a coisa funcionar)

Um microframework

- Manipula Banco de dados
- Métodos mágicos definidos por você
- Reaproveitamento de código
- URLs amigáveis
- Segurança
- Fácil de entender
- Simples de dar manutenção

Model

- PDO, claro!

Métodos auxiliares (helpers)

- Funções que serão chamadas quando necessário

```
function error(){  
    die('Page not found.');
```

```
}  
  
function url($url){  
    return 'http://localhost/index.php/'.$url;  
}
```

Mapeamento de URLs

```
$urls = array(  
    '/?' => 'Inicio',  
    '/categoria/(\d+)/?' => 'Categoria',  
    '/produto/(\d+)/?' => 'Produto',  
    '/adiciona/(\d+)/?' => 'Adiciona',  
    '/adiciona/(\d+)(/ajax)/?' => 'Adiciona',  
    '/remove/(\d+)/?' => 'Remove',  
    '/carrinho/?' => 'Carrinho',  
    '/comprar/?' => 'Comprar',  
    '/obrigado/?' => 'Retorno',  
);
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];
```

```
foreach ($urls as $key => $value) {  
    $find = '@^'.$key.'$@';  
    if (preg_match($find, $url, $attrs)) {  
        array_shift($attrs);  
        if (class_exists($value)) {  
            $value = new $value;  
            $method = $_POST ? 'post' : 'get';  
            call_user_method_array($method, $value, $attrs);  
            die();  
        }  
    }  
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}
```

Fazendo a mágica

```
$url = $_SERVER['PATH_INFO'];

foreach ($urls as $key => $value) {
    $find = '@^'.$key.'$@';
    if (preg_match($find, $url, $attrs)) {
        array_shift($attrs);
        if (class_exists($value)) {
            $value = new $value;
            $method = $_POST ? 'post' : 'get';
            call_user_method_array($method, $value, $attrs);
            die();
        }
    }
}

erro();
```

O resultado

```
class Inicio
{
    function get($attr1, $attr2, $attr3)
    {
        // seu controller
    }

    function post()
    { // opcional
    }
}
```

O resultado

```
class Inicio
{
    function get($attr1, $attr2, $attr3)
    {
        // seu controller
    }

    function post()
    { // opcional
    }
}
```

O resultado

```
class Inicio
{
    function get($attr1, $attr2, $attr3)
    {
        // seu controller
    }

    function post()
    { // opcional
    }
}
```

O resultado

```
class Inicio
{
    function get($attr1, $attr2, $attr3)
    {
        // seu controller
    }

    function post()
    { // opcional
    }
}
```

Template

```
class Template {
    public $categorias = false;

    function __construct() {
        $con =& get_con();
        $this->categorias = $con->categorias();
        include('template/topo.php'); // View
    }

    function __destruct() {
        include('template/bottom.php');
    }
}
```

A home

```
class Inicio extends Template
{
    protected $_title = 'Bem vindo';
    function get()
    {
        $con =& get_con();
        $produtos = $con->prodRand();
        include('template/inicio.php');
    }
}
```

Adicionando um produto

```
class Adiciona {
    public function get($id, $ajax = false) {
        $con =& get_con();
        if (!$con->produto($id)) {
            error();
        }
        if (isset($_SESSION['carrinho'][$id])) {
            $_SESSION['carrinho'][$id]++;
        } else {
            $_SESSION['carrinho'][$id] = 1;
        }
        header('Location: '.url('carrinho'));
    }
}
```

E-commerce pronto?!

- Para que o e-commerce esteja pronto de fato, falta apenas um detalhe: o comprador deve comprar

Vamos por a mão na massa!!!

Formas de pagamento

- Boleto bancário
- Cartão de crédito
- Cartões de débito
- Transação entre contas

- ... ou usar o PagSeguro

PagSeguro: como funciona?



PagSeguro: como funciona?

```
1 <form target="pagseguro" action="https://pagseguro.uol.com.br"
2   <input type="hidden" name="email_cobrança" value="mike@vis
3   <input type="hidden" name="encoding" value="utf-8" />
4   <input type="hidden" name="email_cobranca" value="" />
5   <input type="hidden" name="tipo" value="CP" />
6   <input type="hidden" name="moeda" value="BRL" />
7   <input type="hidden" name="item_id_1" value="CAMISA-FLA-G
8   <input type="hidden" name="item_descr_1" value="Camisa do
9   <input type="hidden" name="item_quant_1" value="2" />
10  <input type="hidden" name="item_valor_1" value="2990" />
11  <input type="submit" value="Pague com o PagSeguro" />
12 </form>
```

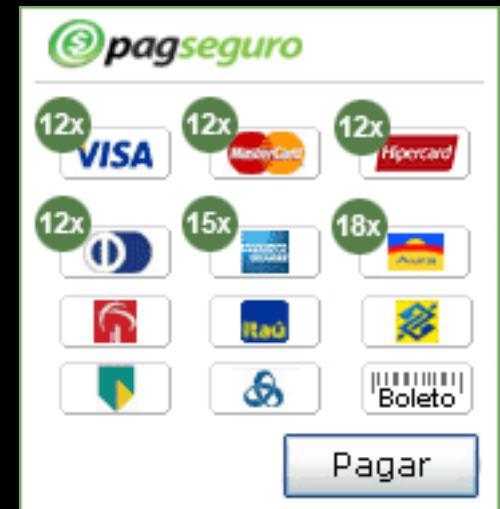
PagSeguro: como funciona?

**Pague com o
PagSeguro**

PagSeguro: como funciona?

Pague com o
PagSeguro

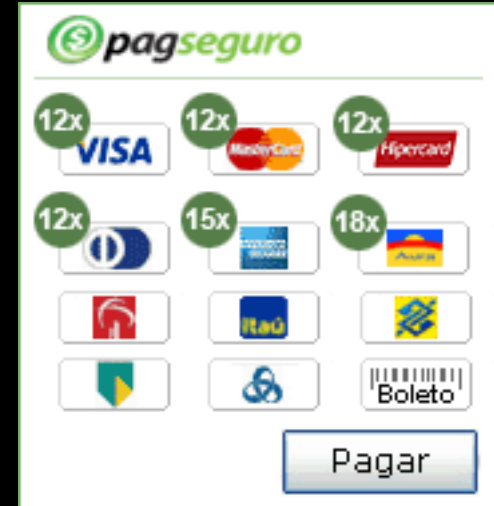
POST



E o tal do retorno?



Usuário - POST



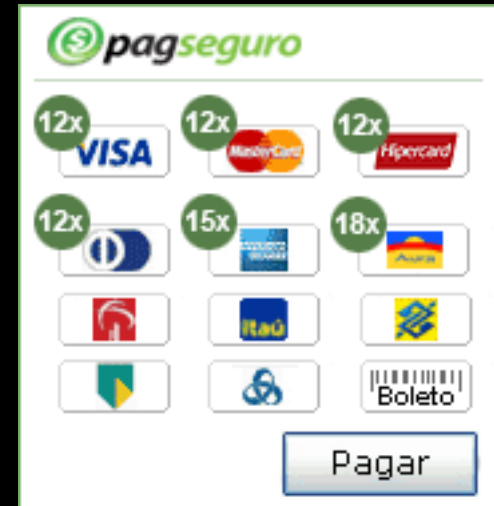
E o tal do retorno?



Usuário - POST



Usuário - GET



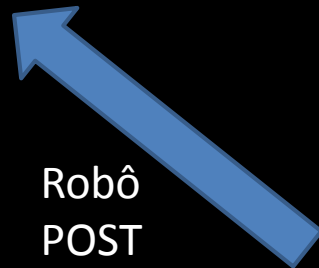
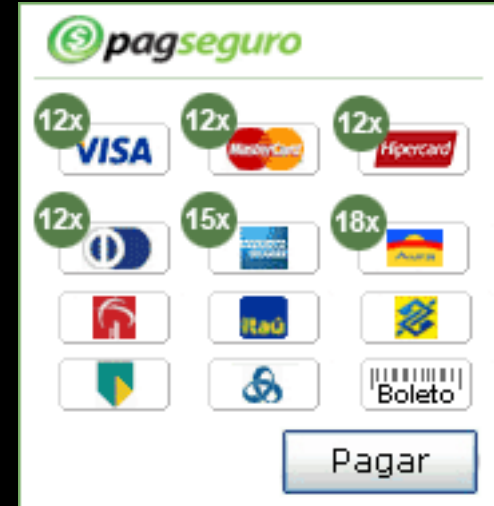
E o tal do retorno?



Usuário - POST



Usuário - GET



Robô
POST



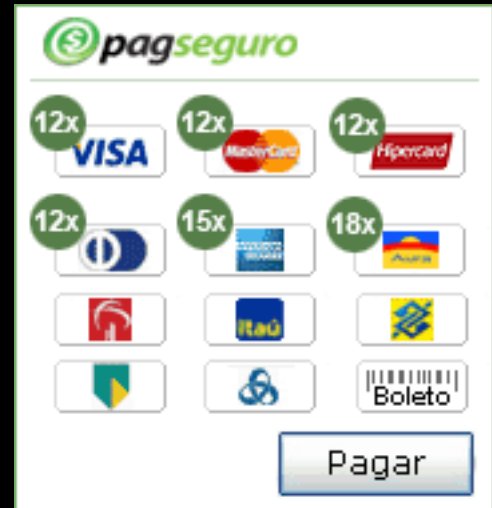
E o tal do retorno?



Usuário - POST



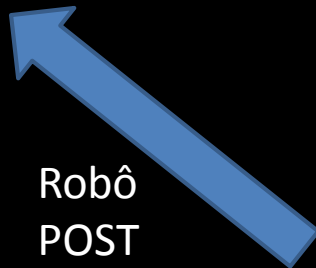
Usuário - GET



Bad Guy
POST?



Robô
POST



E o tal do retorno?



Bad Guy
POST?

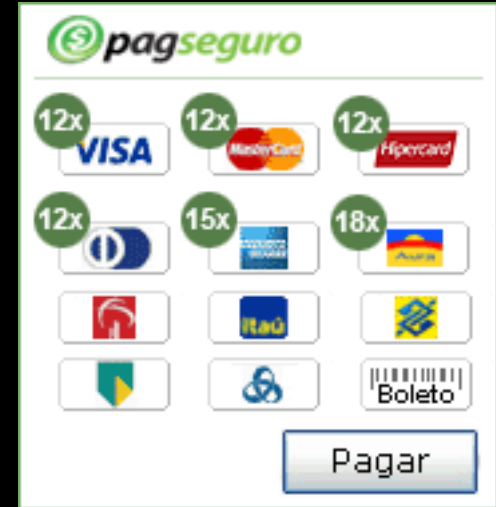
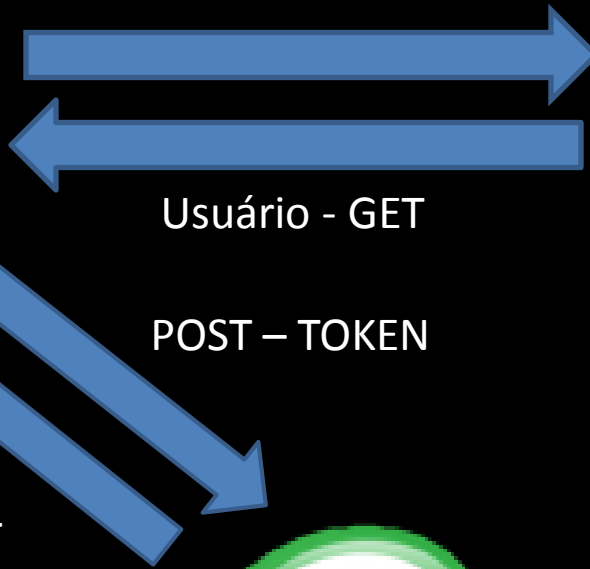


Robô
POST

Usuário - POST

Usuário - GET

POST - TOKEN



VERIFICADO/FALSO

Referências

- <http://php.net/>
- <http://br.php.net/pdo>
- <http://subversion.tigris.org/>
- <http://trac.edgewall.org/>
- <http://visie.com.br/pagseguro>
- <http://www.generatedata.com/>
- <http://tinyurl.com/mkmu58> (W3C: HTML5)

Perguntas?!

<http://meadiciona.com/dgmike>