



# **MANUALZINHO DE SOBREVIVÊNCIA**



## Orientações Gerais

Os documentos devem estar codificados em **ISO-8859-1**.

Fontes de leilões devem estar no formato:

SP\_BIASI.php

Nome: todo em MAIÚSCULAS, sem acentos, com extensão “.php” em minúsculas

SP: Estado

BIASI: Nome do leiloeiro

As fontes de leilões devem ser gravadas no diretório “gurubot/functions”.



## String

Armazena um valor.

Quando informado com aspas, é uma string alfanumérica.

É iniciado pelo caractere "\$".

## Array (Vetor)

[https://www.php.net/manual/pt\\_BR/language.types.array.php](https://www.php.net/manual/pt_BR/language.types.array.php)

Um array no PHP é na verdade um mapa ordenado. Um mapa é um tipo que relaciona valores a chaves.

```
<?php
$array = array("foo", "bar", "hello", "world");
var_dump($array);
?>
```

O exemplo acima irá imprimir:

```
array(4) {
  [0]=>
  string(3) "foo"
  [1]=>
  string(3) "bar"
  [2]=>
  string(5) "hello"
  [3]=>
  string(5) "world"
}
```

`print_r($array)`: mostra os elementos



## http\_get\_curl

Função interna. Lê html da página para armazenamento em uma string.

```
$texto = http_get_curl($url,$agente,$obtemcookies,$limpa,$cookie,$porta,  
$versaossL,$ssl_ignora,$tiraaspas);
```

```
$url='http://www.exemplo.com.br';
```

```
$agente: false (enviará header genérico)
```

```
        'firefox'
```

```
        'ie'
```

```
$obtemcookies: false(padão) ou true
```

```
$limpa: converte caracteres da resposta
```

```
        valores: true, false, 'utf8', 'isolatin1', 'charconvert+utf8'  
ou 'isolatin1+charconvert+utf8'
```

```
$cookie: '' (padrão)
```

```
$porta: '' (padrão) ou codificação para outra porta
```

```
$versaossL: '' (padrão)
```

```
$ssl_ignora: true (padrão) ou false
```

```
$tiraaspas: true (padrão) ou false
```



# Explode

[https://www.php.net/manual/pt\\_BR/function.explode.php](https://www.php.net/manual/pt_BR/function.explode.php)

## Valor Retornado

```
explode(string $delimiter, string $string, int $limit =?): array
```

Retorna uma matriz de strings, cada uma como substring de **string** formada pela divisão dela a partir do **delimiter**.

## Parâmetros

### **delimiter**

O delimitador.

### **string**

A string de entrada.

### **limit**

Se **limit** é definido, o array retornado irá conter o máximo de elementos igual a **limit** com o último elemento contendo o resto da **string**.

Se o parâmetro **limit** é negativo, todos componentes exceto o último **-limit** são retornados.

```
<?php
```

```
// Example 1
```

```
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
```

```
$pieces = explode(" ", $pizza);
```

```
echo $pieces[0]; // piece1
```

```
echo $pieces[1]; // piece2
```



# str\_replace

[https://www.php.net/manual/pt\\_BR/function.str-replace.php](https://www.php.net/manual/pt_BR/function.str-replace.php)

str\_replace — Substitui todas as ocorrências da string de procura com a string de substituição

## Descrição

```
str_replace(  
    mixed $search,  
    mixed $replace,  
    mixed $subject,  
    int &$count = ?  
): mixed
```

```
<?php
```

```
// Fornece: <body text='black'>
```

```
$bodytag = str_replace("%body%", "black", "<body text='%body%'>");
```

```
// Fornece: Hll Wrld f PHP
```

```
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
```

```
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");
```

```
// Fornece: você comeria pizza, cerveja e sorvete todos os dias
```

```
$frase = "você comeria frutas, vegetais, e fibra todos os dias.";
```

```
$saudavel = array("frutas", "vegetais", "fibra");
```

```
$saboroso = array("pizza", "cerveja", "sorvete");
```

```
$novafrase = str_replace($saudavel, $saboroso, $frase);
```



# trim

[https://www.php.net/manual/pt\\_BR/function.trim.php](https://www.php.net/manual/pt_BR/function.trim.php)

trim — Retira espaço no início e final de uma string

## Descrição

```
trim(string $str, string $charlist = ?): string
```



## limpahtml

Função interna. Limpa html página para armazenamento em uma string.

```
$texto = limpahtml($texto,$limpascript);
```

**\$texto:** string com tags html

**\$limpascript:** limpa elementos de estilo também

### Exemplo:

```
<?php
```

```
$texto='<h4>Entidade</h4>
```

```
    <div>Prefeitura Municipal de Capela do Alto</div>
```

```
    <h4>Modalidade</h4>
```

```
    <div>Tomada de Preços</div>';
```

```
$texto_limpo=limpahtml($texto);
```

```
echo $texto_limpo;
```

```
?>
```

### Resultado:

Entidade

Prefeitura Municipal de Capela do Alto

Modalidade

Tomada de Preços





## isolar

Função interna. Isola um trecho específico de uma string.

```
$texto = isolar($texto,$inicio,$inicio2,$fim);
```

**\$texto:** string de entrada

**\$inicio:** texto de busca inicial

**\$inicio2:** (*opcional*) texto de busca após achar o inicial

**\$fim:** texto de busca final

### Exemplo:

```
<?php
```

```
$texto='<h4>Entidade</h4>
```

```
    <div>Prefeitura Municipal de Capela do Alto</div>
```

```
    <h4>Modalidade</h4>
```

```
    <div>Tomada de Preços</div>';
```

```
$entidade=isolar($texto,'</h4',' ','</div');
```

```
echo '
```

```
    Entidade: '.$entidade;
```

```
$modalidade=isolar($texto,'</h4','</h4','</div');
```

```
echo '
```

```
    Modalidade: '.$modalidade;
```

```
?>
```

### Resultado:

```
Entidade: ><div>Prefeitura Municipal de Capela do Alto
```

```
Modalidade: ><div>Tomada de Preços
```



## isolar\_limpo

Função interna. Isola um trecho específico de uma string e limpa o html.

```
$texto = isolar_limpo($texto,$inicio,$inicio2,$fim);
```

**\$texto:** string de entrada

**\$inicio:** texto de busca inicial

**\$inicio2:** (*opcional*) texto de busca após achar o inicial

**\$fim:** texto de busca final

### Exemplo:

```
<?php
```

```
$texto='<h4>Entidade</h4>
```

```
<div>Prefeitura Municipal de Capela do Alto</div>
```

```
<h4>Modalidade</h4>
```

```
<div>Tomada de Preços</div>';
```

```
$entidade=isolar_limpo($texto,'</h4','','</div');
```

```
echo '
```

```
Entidade: '.$entidade;
```

```
$modalidade=isolar_limpo($texto,'</h4','</h4','</div');
```

```
echo '
```

```
Modalidade: '.$modalidade;
```

```
?>
```

### Resultado:

```
Entidade: Prefeitura Municipal de Capela do Alto
```

```
Modalidade: Tomada de Preços
```



## limpa\_sql

Função interna. Formata o texto, preparando para gravar em banco de dados. Exemplo: retira espaços duplos, inclui espaços após pontuações e codifica alguns caracteres.

```
$texto = limpa_sql($texto0);
```



## arrumadata\_robos

Função interna. Formata uma data para o padrão YYYYmmddHHii.

```
$texto = arrumadata_robos($texto,$tipo,$invertemes);
```

**\$texto:** string de entrada

**\$tipo:** *(opcional)*

    '1' (padrão): texto no padrão d/m/Y (numéricos)

    '5': texto no padrão d/m/Y (onde mês está escrito por extenso, como 1/junho/2023)

    '6': texto no padrão d/m/Y (onde mês está abreviado com 3 letras, como 1/JUN/2023)

**\$invertemes:** *(opcional)*

    false (padrão). Data formato d/m/Y

    true. Data formato m/d/Y

### Exemplo:

```
<?php
```

```
$texto='01/6/23';
```

```
$resp['lic_abertura_propostas']=arrumadata_robos($texto);
```

```
echo 'Data formatada: '.$resp['lic_abertura_propostas'];
```

```
?>
```

### Resultado:

Data formatada: 202306010000



## VerificaCadastroLeilao

Função interna. Verifica se o leilão enviado já está no banco de dados. Retorna true se já estiver cadastrada ou false se ainda não estiver no banco de dados.

A verificação deve ser feita enviando um **vetor** com os dados do leilão.

### Exemplo:

```
<?php
    $resp=array();
    $resp['organizador']=SITE_SP_BIASI;
    $resp['ll_pais']='BRASIL';
    $resp['ll_idioma']='pt-BR';
    $resp['ll_moeda']='BRL';
    $resp['ll_lote']='01';
    $resp['ll_descricao']='Imóvel Comercial no Centro de Manaus/AM';
    $resp['ll_cidade']='MANAUS';
    $resp['ll_uf']='AM';

    if (!VerificaCadastroLeilao($resp)){
        // aqui: leilão não estava cadastrado; providencia o cadastro
    }
}
?>
```



## NovoLeilao

Função interna. Cadastra um leilão no banco de dados.

```
$nova_lic = NovoLeilao ($resp,$lote,  
margem_v_icone=0.1,$margem_h_icone=0.1,$margem_v=0,$margem_h=0,$verificamd5=true  
, $precadastro=false);
```

**\$resp:**        vetor de entrada  
outros parâmetros opcionais

**Exemplo:**

```
<?php  
    echo NovoLeilao ($resp,array(),0,0,0,0,false);  
?>
```



## Campos do vetor de cadastro

```
$resp=array();
    $resp=array();
    // Variáveis Obrigatórias
    $resp['organizador']='';
    $resp['ll_resumo']='';
    $resp['ll_comitente']='';
    // Variáveis relevantes
    $resp['ll_extra']='';
    $resp['ll_uf']='';
    $resp['ll_cidade']='';
    $resp['ll_numero']='';
    $resp['ll_edital']='';
    $resp['ll_processo']='';
    $resp['ll_principal']='';
    $resp['ll_natureza']='';
    $resp['ll_data_publicacao']='';
    $resp['ll_inicio_acolhimento_propostas']='';
    $resp['ll_limite_acolhimento_propostas']='';
    $resp['ll_abertura_propostas']='';
    $resp['ll_data_hora_pregao']='';
    $resp['ll_data_considerada']='';
    $resp['ll_lotes']='';
    $resp['ll_temp']='';
    // Variáveis passíveis de mudança
    $resp['ll_data_cadastro']='';
    // Outras variáveis
    $resp['ll_pregoeiro']='';
    $resp['ll_tipo']='';
    $resp['ll_particip_fornecedor']='';
    $resp['ll_prazo_impugnacao']='';
    $resp['ll_situacao']='';
    echo NovaLicitacao ($resp);
```