

# Documentación de SVCMS



# Contents

<a href="#">Package svcms1 Procedural Elements</a>	2
<a href="#">remoteFunctions.php</a>	2
<a href="#">Function includeCommon</a>	2
<a href="#">Function includeCommonClass</a>	2
<a href="#">setup.php</a>	3
<a href="#">fuente-SimpleEmailEdit.php</a>	4
<a href="#">save.php</a>	5
<a href="#">saveAll.php</a>	6
<a href="#">SimpleEmail.php</a>	7
<a href="#">SV-detalle-articulo.php</a>	8
<a href="#">Search.php</a>	9
<a href="#">searchSuggest.php</a>	10
<a href="#">USAGE.php</a>	11
<a href="#">Package svcms1 Classes</a>	12
<a href="#">Class funciones</a>	12
<a href="#">Method borrar_imagenes</a>	12
<a href="#">Method borrar_imagenes2</a>	12
<a href="#">Method limpiarUrl</a>	12
<a href="#">Method limpiarUrlBD</a>	12
<a href="#">Method LoadCalendario</a>	13
<a href="#">Method stripaccents</a>	13
<a href="#">Class shoppingCart</a>	13
<a href="#">Var \$guardadoEnCookies</a>	13
<a href="#">Var \$productos</a>	14
<a href="#">Constructor construct</a>	14
<a href="#">Method agregarProducto</a>	14
<a href="#">Method cambiarCantidadProducto</a>	14
<a href="#">Method eliminarProducto</a>	15
<a href="#">Method obtenerCookie</a>	15
<a href="#">Method obtenerShoppingCart</a>	15
<a href="#">Method salvar</a>	15
<a href="#">Method saveCookie</a>	15
<a href="#">Method saveDB</a>	15
<a href="#">Method show</a>	15
<a href="#">Class widget</a>	15
<a href="#">Var \$formatoPresentacion</a>	16
<a href="#">Var \$formatoSinResultados</a>	16
<a href="#">Var \$rutaBase</a>	16
<a href="#">tools.inc</a>	17
<a href="#">Function destruct</a>	17
<a href="#">Class database</a>	17
<a href="#">Var \$dbc</a>	17
<a href="#">Var \$nreg</a>	17

<a href="#">Var \$result</a>	18
<a href="#">Var \$ultimoID</a>	18
<a href="#">Constructor construct</a>	18
<a href="#">Method abrir transaccion</a>	18
<a href="#">Method autoconexion</a>	19
<a href="#">Method borrar</a>	19
<a href="#">Method campos query</a>	19
<a href="#">Method cerrar</a>	19
<a href="#">Method cerrar transaccion</a>	19
<a href="#">Method conectar</a>	19
<a href="#">Method insertar</a>	20
<a href="#">Method insertar2</a>	20
<a href="#">Method liberar</a>	20
<a href="#">Method numero filas afecc</a>	20
<a href="#">Method query</a>	20
<a href="#">Method reciclar conexion</a>	21
<a href="#">Method total registros</a>	21
<a href="#">Method ultimo id</a>	21
<a href="#">Method update</a>	21
<a href="#">Class formulario</a>	21
<a href="#">Constructor construct</a>	22
<a href="#">Method getvar</a>	22
<a href="#">Method insert data</a>	22
<a href="#">Method update data</a>	23
<a href="#">Class tools</a>	23
<a href="#">Constructor construct</a>	23
<a href="#">Method abreviar</a>	24
<a href="#">Method array query</a>	24
<a href="#">Method array query2</a>	24
<a href="#">Method burbuja</a>	24
<a href="#">Method buscar estructdb</a>	25
<a href="#">Method combo array</a>	25
<a href="#">Method combo db</a>	26
<a href="#">Method combo dir</a>	26
<a href="#">Method delete directory</a>	26
<a href="#">Method eliminar de</a>	27
<a href="#">Method estructura db</a>	27
<a href="#">Method format txt</a>	27
<a href="#">Method javavisio</a>	28
<a href="#">Method limpiar array</a>	28
<a href="#">Method listar archivos</a>	28
<a href="#">Method llenar array</a>	28
<a href="#">Method obtenerArticulos</a>	29
<a href="#">Method obtenerCategorias</a>	29
<a href="#">Method obtenerConfigPages</a>	29
<a href="#">Method obtenerConfiguracionProductos</a>	29
<a href="#">Method obtenerProductos</a>	30
<a href="#">Method redirect</a>	30
<a href="#">Method repetidos</a>	30

Method seencuentra	30
Method simple_db	31
Method suma_array	31
Method unicos	31
Method upload_file	32
<b>Package svcms2 Classes</b>	33
Class articulo	33
Var \$activos	33
Var \$funciones	33
Var \$rutaBase	33
Constructor__construct	33
Method obtenerArticulos	33
Method obtenerArticulosByQuery	34
Method setRutaBase	34
Class categoria	34
Var \$funciones	34
Var \$nivel	34
Var \$rutaBase	34
Constructor__construct	34
Method obtenerCategoria	34
Method obtenerCategorias	35
Method obtenerCategoriasByQuery	35
Method obtenerSubCategorias	35
Method setRutaBase	35
Class producto	35
Var \$activos	36
Var \$empaquetado	36
Var \$fecha_mod	36
Var \$funciones	36
Var \$id	36
Var \$imagen	36
Var \$link	37
Var \$medidas	37
Var \$nombre	37
Var \$peso	37
Var \$precio	37
Var \$resumen	37
Var \$rutaBase	38
Var \$stock	38
Constructor__construct	38
Constructor__construct	38
Method comprar	38
Method obtenerProductos	38
Method obtenerProductosByQuery	39
Method setRutaBase	39
<b>Package svcms2 Procedural Elements</b>	41
SimpleList.inc	41
Class simpleList	41

<a href="#">Var \$articulo</a>	42
<a href="#">Var \$autorArticulo</a>	42
<a href="#">Var \$categoria</a>	42
<a href="#">Var \$formatoPresentacionHoja</a>	42
<a href="#">Var \$formatoTitulo</a>	42
<a href="#">Var \$funciones</a>	42
<a href="#">Var \$maxElementos</a>	42
<a href="#">Var \$producto</a>	43
<a href="#">Var \$tagInterno</a>	43
<a href="#">Var \$titulo</a>	43
<a href="#">Constructor construct</a>	43
<a href="#">Method articulosDestacados</a>	43
<a href="#">example: Ejemplo de uso de los articulos Destacados</a>	43
<a href="#">Method articulosEnCategoria</a>	44
<a href="#">example: Muestra los articulos de una categoria</a>	44
<a href="#">Method articulosMasVistos</a>	44
<a href="#">Method articulosRelacionados</a>	44
<a href="#">Method categoriasDeArticulos</a>	44
<a href="#">example: Agregar las categorias a una barra</a>	44
<a href="#">Method categoriasDeProductos</a>	45
<a href="#">Method categoriasEnCategoria</a>	46
<a href="#">Method elementosEnCategorias</a>	46
<a href="#">Method listarArticulos</a>	46
<a href="#">Method listarArticulosByQuery</a>	46
<a href="#">Method listarProductos</a>	47
<a href="#">Method listarProductosByQuery</a>	47
<a href="#">Method productosDestacados</a>	47
<a href="#">Method productosEnCategoria</a>	47
<a href="#">Method productosMasVendidos</a>	48
<a href="#">Method productosMasVistos</a>	48
<a href="#">Method productosRelacionados</a>	48
<a href="#">Method subCategoriasDeArticulos</a>	48
<a href="#">Method subCategoriasDeProductos</a>	48
<a href="#">Method subSubCategoriasDeArticulos</a>	49
<a href="#">Method subSubCategoriasDeProductos</a>	49
<a href="#">Method traer_item_categoria</a>	49
<a href="#">Appendices</a>	50
<a href="#">Appendix A - Class Trees</a>	51
<a href="#">svcms1</a>	51
<a href="#">svcms2</a>	51



# Package svcms1 Procedural Elements

## remoteFunctions.php

- **Package** svcms1

*void function includeCommon(\$url) [line 8]*

***Function Parameters:***

- **\$url**

*void function includeCommonClass(\$className) [line 2]*

***Function Parameters:***

- **\$className**

## setup.php

\*\*\*\*\* PARAMETROS MODIFICABLES\*\*\*\*\*

- **Package** svcms1

include ['dbconfig.php'](#)*[line 17]*

require ['redirect.php'](#)*[line 16]*

\*\*\*\*\* NO ALTERE ESTE BLOQUE \*\*\*\*\*



# fuentes-SimpleEmailEdit.php

- **Package** svcms1

include **\$presentacionEmail** *[line 22]*

include\_once **'../SVsystem/class/tools.php'** *[line 1]*

include\_once **'../SVsystem/class/funciones.php'** *[line 3]*

include\_once **'../SVsystem/config/dbconfig.php'** *[line 2]*

# save.php

- **Package** svcms1

include "../SVsystem/config/setup.php" *[line 4]*

include "../SVsystem/class/tools.php" *[line 5]*

# saveAll.php

- **Package** svcms1

include "../SVsystem/config/setup.php" *[line 4]*

include "../SVsystem/class/tools.php" *[line 5]*

# SimpleEmail.php

- **Package** svcms1

include\_once '**SVsystem/class/tools.php**' *[line 1]*

include\_once '**SVsystem/class/email.php**' *[line 2]*

include\_once '[SVsystem/config/dbconfig.php](#)' *[line 3]*

include\_once '**SVsystem/class/classes.php**' *[line 4]*

include\_once '**SVsystem/class/funciones.php**' *[line 5]*

# SV-detalle-articulo.php

- **Package** svcms1

include 'SVsystem/class/funciones.php' [line 4]

include 'SVsystem/config/dominio.php' [line 5]

include "SVsystem/class/email.php" [line 6]

include 'SVsystem/config/dbconfig.php' [line 2]

include 'SVsystem/class/formulario.php' [line 3]

include "SVwidgets/Email/SimpleEmail.php" [line 33]

\*\*\*\*\* EMAIL DEL COMENTARIO \*\*\*\*\*

include 'SVG-banners.php' [line 395]

include 'SVG-footer.php' [line 402]

include "http://www.paginainteligente.net/sign.php" [line 404]

include 'SVG-header.php' [line 137]

\*\*\*\*\* FIN EMAIL DEL COMENTARIO \*\*\*\*\*

include 'SVG-menu-categorias.php' [line 141]

# Search.php

- **Package** svcms1

include\_once '**SVsystem/class/tools.php**' *[line 1]*

# searchSuggest.php

- **Package** svcms1

`include_once '../SVsystem/config/dbconfig.php' [line 3]`

`include_once '../SVsystem/class/clases.php' [line 4]`

`include_once '../SVsystem/class/funciones.php' [line 5]`

# USAGE.php

- **Package** svcms1

include "SVwidgets/SimpleList/fuente-categoria-producto.php" *[line 2]*

include "SVwidgets/SimpleList/fuente-categoria-articulo.php" *[line 7]*

include "SVwidgets/SimpleList/fuente-elementos-en-categorias.php" *[line 33]*



# Package svcms1 Classes

## Class funciones

*[line 3]*

- **Package** svcms1

*void* function funciones::borrar\_imagenes(\$imagen) *[line 67]*

**Function Parameters:**

- **\$imagen**

*void* function funciones::borrar\_imagenes2(\$imagen) *[line 77]*

**Function Parameters:**

- **\$imagen**

*void* function funciones::limpiarUrl(\$url) *[line 93]*

**Function Parameters:**

- **\$url**

*void* function funciones::limpiarUrlBD(\$nombreCampo) *[line 119]*

**Function Parameters:**

- **\$nombreCampo**

*void* function funciones::LoadCalendario(\$CEtiqueta, \$Cnombreld, \$CValues, \$CVisible, \$CEjeX, \$CEjeY, \$CImagenClick, \$clase\_t, \$clase\_in) *[line 6]*

**Function Parameters:**

- **\$CEtiqueta**
- **\$Cnombreld**
- **\$CValues**
- **\$CVisible**
- **\$CEjeX**
- **\$CEjeY**
- **\$CImagenClick**
- **\$clase\_t**
- **\$clase\_in**

*void* function funciones::stripaccents(\$string) *[line 88]*

**Function Parameters:**

- **\$string**

## Class shoppingCart

*[line 10]*

**Clase para representar al carrito de compras. Permite manipularlo y trabajarlo.**

Acepta dos tipos de comportamiento, puede ser guardado en cookies o guardado en BD solo para registrados

- **Package** svcms1
- **Author** Luis Poggi

**shoppingCart::\$guardadoEnCookies**

*boolean = [line 24]*

**Indica si el carrito se almacena en cookies o no.**

**shoppingCart::\$productos**

*number\_array = [line 17]*

**Productos almacenados en el carrito con sus cantidades.**

Solo se guardan los ID para que cuando se muestre siempre tenga la información mas exacta.

Constructor *void* function `shoppingCart::__construct([$cookie = true])` *[line 31]*

**Function Parameters:**

- *\$cookie*, **\$cookie** Variable que indica si el carrito de compras es guardado en cookie o no. Valor por defecto = true

**Constructor de la clase. El comportamiento por defecto es que se guarda en cookies**

*void* function `shoppingCart::agregarProducto($idProducto, $cantidad, $idProducto,, $cantidad,)` *[line 41]*

**Function Parameters:**

- *Number \$idProducto*, Identificador del producto a agregar
- *Number \$cantidad*, cantidad a agregar
- **\$idProducto**
- **\$cantidad**

**Permite agregar un producto al carrito de compras**

*void* function `shoppingCart::cambiarCantidadProducto($idProducto, $cantidad, $idProducto,, $cantidad,)` *[line 60]*

**Function Parameters:**

- *Number \$idProducto*, producto a cambiar su cantidad
- *Number \$cantidad*, nueva cantidad, sobrescribe la anterior
- **\$idProducto**
- **\$cantidad**

**Cambia la cantidad de producto de un carrito**

*void function shoppingCart::eliminarProducto(\$idProducto, \$idProducto,) [line 50]*

**Function Parameters:**

- *Number \$idProducto*, Identificador del producto a eliminar
- **\$idProducto**

## Saca un producto del carrito

*void function shoppingCart::obtenerCookie() [line 107]*

**Enter description here ...**

*void function shoppingCart::obtenerShoppingCart() [line 96]*

**Enter description here ...**

*void function shoppingCart::salvar() [line 69]*

**Guarda el carrito. Es afectado por la variable guardadoEnCookies, según esta indique se almacena en BD o en cookies**

*void function shoppingCart::saveCookie() [line 88]*

**Guarda el carrito en la sesion**

*void function shoppingCart::saveDB() [line 81]*

**Guarda el carrito en Base de datos**

*void function shoppingCart::show() [line 116]*

**Enter description here ...**

## Class widget

*[line 9]*

**Clase que representa los widgets. Todos los widgets heredan las propiedades y métodos aquí definidos**

- **Package** svcms1
- **Author** Luis Poggi

#### **widget::\$formatoPresentacion**

*String = [line 14]*

**Formato de la presentación, sirve para mostrar los resultados del widget**

#### **widget::\$formatoSinResultados**

*String = [line 20]*

**Formato para colocar el widget sin resultados.**

Todos los widgets que presentan resultados consideran que puede no venir ninguno, para ello se usa este formato

#### **widget::\$rutaBase**

*String = [line 26]*

**Ruta Base donde se encuentran las imágenes. Próximo a ser deprecado**

Mientras se mantengan las imagenes en el servidor centralizado permanecerá, luego desaparecerá

## tools.inc

- **Package** svcms1
- **Sub-Package** database

*void function \_\_destruct() [line 823]*

## Class database

*[line 12]*

**clase que crea todas las funciones necesarias para trabajr con bases de datos desde su coneccion hasta cierre de la conexion**

- **Package** svcms1
- **Sub-Package** database
- **Author** Desconocido

**database::\$dbc**

*unknown\_type = [line 38]*

**variable de conexion**

- **Access** public

**database::\$nreg**

*unknown\_type = [line 48]*

**cantidad de registros insertados**

- **Access** public

**database::\$result**

*unknown\_type = [line 43]*

**atributo que tiene el resultado del query**

- **Access** public

**database::\$ultimoID**

*unknown\_type = [line 53]*

**ultimo id insertado (auto incremental)**

- **Access** public

Constructor *void* function database::\_\_construct() *[line 58]*

**constructor de la clase**

*void* function database::abrir\_transaccion() *[line 294]*

**Enter description here ...**

- **Access** public

*void* function database::autoconexion() [*line 92*]

**autoconexion discreta segun los parametros y la base dedatos especificadas en el archivo de conf**

*void* function database::borrar(\$dbase) [*line 182*]

**Function Parameters:**

- **\$dbase**

*string* function database::campos\_query() [*line 280*]

**devuelve un arreglo con el nombre de los campos consultados enpezando desde la pos 0**

*void* function database::cerrar() [*line 74*]

**metodo para cerrar la conexion con la base de datos. En este momento esta sin uso por conflictos en las páginas**

- **Deprecated**

*void* function database::cerrar\_transaccion([\$result = true]) [*line 305*]

**Function Parameters:**

- *unknown\_type* **\$result**

**Enter description here ...**

- **Access public**

*unknown* function database::conectar(\$host, \$user, \$pwd, \$db, [\$persiste = false]) [*line 133*]

**Function Parameters:**



- *unknown\_type* \$host
- *unknown\_type* \$user
- *unknown\_type* \$pwd
- *unknown\_type* \$db
- *unknown\_type* \$persiste

**propiedad que cambia la conexion a una nueva database o server con solo invocarla se cambia la conexion el ultimo parametro determina si la conexion es persistente o no.**

*void function database::insertar(\$tabla, \$campos, \$valores, [\$block = false]) [line 194]*

**Function Parameters:**

- *unknown\_type* \$tabla
- *unknown\_type* \$campos
- *unknown\_type* \$valores
- *unknown\_type* \$block

**funcion que realiza un insert simple tabla, campos, valores y boqueo true o false (opcional)**

*void function database::insertar2(\$tabla, \$campos, \$vector, [\$block = false]) [line 221]*

**Function Parameters:**

- *unknown\_type* \$tabla
- *unknown\_type* \$campos
- *unknown\_type* \$vector
- *unknown\_type* \$block

**funcion que realiza un insert simple de los valores ingresados en un vector**

tabla, campos, valores y boqueo true o false (opcional) esta funcion es una mejora de insertar 1 ya que previene errores si los valores poseen ,

*void function database::liberar() [line 85]*

**metodo para liberar el objeto que retorna el valor del query**

*void function database::numero\_filas\_afecc() [line 165]*

*void function database::query(\$sql, [\$lib = false]) [line 116]*

**Function Parameters:**

- *unknown\_type* \$sql
- *unknown\_type* \$lib

**metodo que ejecuta un query de creacion o modificacion o devuelve un error el segundo parametro es para liberar el recurso de una vez**  
utilizado ideal para sentencias insert, update, delete etc

*void function database::recliclar\_conexion(\$link) [line 105]*

**Function Parameters:**

- *unknown\_type* \$link

**permite utilizar la conexion a una database ya creada sin tener que reconectar;**

*void function database::total\_registros() [line 173]*

*void function database::ultimo\_id() [line 159]*

*void function database::update(\$tabla, \$campos, \$vector, \$where, [\$block = false]) [line 251]*

**Function Parameters:**

- *unknown\_type* \$tabla
- *unknown\_type* \$campos
- *unknown\_type* \$vector
- *unknown\_type* \$where
- *unknown\_type* \$block

**funcion que realiza un update simple de los valores y campos ingresados en un vector**  
tabla, campos(array), valores(array) where (condicion) y boqueo true o false (opcional)  
previene errores si los valores poseen ,

## Class formulario

*[line 18]*

**clase para la construccion de formularios y volcamiento de data proveniente de los**

## **mismos**

IMPORTANTE: por defecto se asume que se esta conectado a la base de datos en caso de que desee llamar herramientas del tipo database

- **Package** svcms1
- **Sub-Package** database
- **Author** Desconocido

Constructor *void* function formulario::\_\_construct([\$connect = ""]) [*line 25*]

### ***Function Parameters:***

- **\$connect**

\*\*\*\*\* CONSTRUCTOR

*void* function formulario::getvar(\$var, [\$only = ""]) [*line 44*]

### ***Function Parameters:***

- **\$var**
- **\$only**

- **Access** public

*void* function formulario::insert\_data(\$pref, \$sep, \$tabla, \$variables) [*line 75*]

### ***Function Parameters:***

- **\$pref**
- **\$sep**
- **\$tabla**
- **\$variables**

- **Access** public

*void* function formulario::update\_data(\$pref, \$sep, \$tabla, \$variables, [\$where = ""]) [*line 111*]

**Function Parameters:**

- **\$pref**
- **\$sep**
- **\$tabla**
- **\$variables**
- **\$where**

- **Access** public

## Class tools

[*line 16*]

**clase que cosntruye diversas herramientas fopara usar en la construccion de forms**

IMPORTANTE: por defecto se asume que se esta conectado a la base de datos en caso de que desee llamar herramientas del tipo database

- **Package** svcms1
- **Sub-Package** database
- **Author** Desconocido

Constructor *void* function tools::\_\_construct([\$connect = ""]) [*line 19*]

**Function Parameters:**

- **\$conect**

*void function tools::abreviar(\$texto, \$cmax) [line 463]*

**Function Parameters:**

- **\$texto**
- **\$cmax**

- **Access public**

*void function tools::array\_query(\$query) [line 211]*

**Function Parameters:**

- **\$query**

- **Access public**

*void function tools::array\_query2(\$query) [line 228]*

**Function Parameters:**

- **\$query**

- **Access public**

*void function tools::burbuja(\$array, [\$modo = 0]) [line 330]*

**Function Parameters:**

- **\$array**
- **\$modo**

- **Access** public

*void* function tools::buscar\_estructdb(\$matriz, \$clave, \$elemento, \$consultar) [*line 96*]

**Function Parameters:**

- **\$matriz**
- **\$clave**
- **\$elemento**
- **\$consultar**

- **Access** public

*void* function tools::combo\_array(\$id, \$array, \$array2, [\$select = false], [\$seleccion = false], [\$onchange = false], [\$noreg = ""], [\$multiple = false], [\$desabilita = false], [\$estilo = false]) [*line 172*]

**Function Parameters:**

- **\$id**
- **\$array**
- **\$array2**
- **\$select**
- **\$seleccion**
- **\$onchange**
- **\$noreg**
- **\$multiple**
- **\$desabilita**
- **\$estilo**

- **Access** public

*void* function tools::combo\_db(\$id, \$query, \$option, \$value, [\$select = false], [\$seleccion = false], [\$onchange = false], [\$noreg = ""], [\$multiple = false], [\$desabilita = false], [\$estilo = false]) *[line 120]*

**Function Parameters:**

- **\$id**
- **\$query**
- **\$option**
- **\$value**
- **\$select**
- **\$seleccion**
- **\$onchange**
- **\$noreg**
- **\$multiple**
- **\$desabilita**
- **\$estilo**

- **Access** public

*void* function tools::combo\_dir(\$id, \$path, [\$select = false], [\$seleccion = false], [\$submit = false], \$noreg, [\$multiple = false], [\$desabilita = false]) *[line 584]*

**Function Parameters:**

- **\$id**
- **\$path**
- **\$select**
- **\$seleccion**
- **\$submit**
- **\$noreg**
- **\$multiple**
- **\$desabilita**

- **Access** public

*void* function tools::delete\_directory(\$dir) *[line 625]*

**Function Parameters:**

- **\$dir**

- **Access** public

*void* function tools::eliminar\_de(&\$array, \$element) [*line 393*]

**Function Parameters:**

- **&\$array**
- **\$element**

- **Access** public

*void* function tools::estructura\_db(\$query) [*line 38*]

**Function Parameters:**

- **\$query**

- **Access** public

*void* function tools::format\_txt(\$texto, [\$textarea = false]) [*line 447*]

**Function Parameters:**

- **\$texto**
- **\$textarea**



- **Access** public

*void* function tools::javaviso(\$aviso, [\$url = false]) *[line 429]*

***Function Parameters:***

- **\$aviso**
- **\$url**

- **Access** public

*void* function tools::limpiar\_array(\$array, \$element) *[line 279]*

***Function Parameters:***

- **\$array**
- **\$element**

- **Access** public

*void* function tools::listar\_archivos([\$dirname = "."]) *[line 567]*

***Function Parameters:***

- **\$dirname**

- **Access** public

*void* function tools::llenar\_array(\$ELEMENTOS, [\$sep = ',']) *[line 314]*

***Function Parameters:***

- **\$ELEMENTOS**
- **\$sep**

- **Access public**

*void* function tools::obtenerArticulos(\$nivel, \$id) [*line 812*]

***Function Parameters:***

- **\$nivel**
- **\$id**

*void* function tools::obtenerCategorias() [*line 653*]

- **Access public**

*void* function tools::obtenerConfigPages(\$idPage) [*line 647*]

***Function Parameters:***

- **\$idPage**

- **Access public**

*void* function tools::obtenerConfiguracionProductos(\$config) [*line 640*]

***Function Parameters:***

- **\$config**

- **Access public**

*void function tools::obtenerProductos(\$nivel, \$id) [line 806]*

***Function Parameters:***

- **\$nivel**
- **\$id**

*void function tools::redirect(\$url, [\$parent = ""]) [line 410]*

***Function Parameters:***

- **\$url**
- **\$parent**

- **Access public**

*void function tools::repetidos(\$array) [line 387]*

***Function Parameters:***

- **\$array**

- **Access public**

*void function tools::seencuentra(\$buscar, \$en) [line 260]*

***Function Parameters:***

- **\$buscar**
- **\$en**

- **Access** public

*void* function tools::simple\_db(\$query) [*line 70*]

***Function Parameters:***

- **\$query**

- **Access** public

*void* function tools::suma\_array(\$array) [*line 295*]

***Function Parameters:***

- **\$array**

- **Access** public

*void* function tools::unicos(\$estos) [*line 364*]

***Function Parameters:***

- **\$estos**

- **Access** public

*void* function tools::upload\_file(\$origen, [\$destino = false], [\$tmax = false], [\$tipo = false], [\$overwrite = true]) [*line*  
495]

***Function Parameters:***

- **\$origen**
- **\$destino**
- **\$tmax**
- **\$tipo**
- **\$overwrite**

- **Access** public

# Package svcms2 Classes

## Class articulo

[line 13]

Clase que representa a los articulos

- **Package** svcms2
- **Sub-Package** elementos
- **Author** Luis Poggi

### articulo::\$activos

*mixed* = [line 14]

### articulo::\$funciones

*mixed* = [line 15]

### articulo::\$rutaBase

*mixed* = [line 16]

Constructor *void* function articulo::\_\_construct() [line 18]

*void* function articulo::obtenerArticulos([\$condiciones = ""], [\$orderBy = ""], [\$maxResults = 0]) [line 27]

#### **Function Parameters:**

- **\$condiciones**
- **\$orderBy**
- **\$maxResults**

*void function articulo::obtenerArticulosByQuery(\$query, \$maxResults) [line 48]*

**Function Parameters:**

- **\$query**
- **\$maxResults**

*void function articulo::setRutaBase(\$ruta) [line 23]*

**Function Parameters:**

- **\$ruta**

## Class categoria

*[line 13]*

**Clase que permite administrar las categorias**

- **Package** svcms2
- **Sub-Package** elementos
- **Author** Luis Poggi

**categoria::\$funciones**

*mixed = [line 14]*

**categoria::\$nivel**

*mixed = [line 16]*

**categoria::\$rutaBase**

*mixed = [line 15]*

Constructor *void function categoria::\_\_construct() [line 18]*

*void function categoria::obtenerCategoria(\$catId, \$catNivel) [line 28]*

**Function Parameters:**

- **\$catId**

- **\$catNivel**

*void function categoria::obtenerCategorias([\$condiciones = ""], [\$nivel = 1], [\$orderBy = ""], [\$maxResults = 0]) [line 34]*

**Function Parameters:**

- **\$condiciones**
- **\$nivel**
- **\$orderBy**
- **\$maxResults**

*void function categoria::obtenerCategoriasByQuery(\$query, \$maxResults) [line 62]*

**Function Parameters:**

- **\$query**
- **\$maxResults**

*void function categoria::obtenerSubCategorias(\$catId, [\$nivel = 1]) [line 84]*

**Function Parameters:**

- **\$catId**
- **\$nivel**

*void function categoria::setRutaBase(\$ruta) [line 24]*

**Function Parameters:**

- **\$ruta**

## Class producto

*[line 13]*

**Clase que representa las acciones que se pueden ejecutar sobre un producto**



- **Package** svcms2
- **Sub-Package** elementos
- **Author** Luis Poggi

#### producto::\$activos

*number = [line 81]*

**Esta variable permite configurar si se buscan TODOS los productos o sólo aquellos con stock > 0. Admite los siguiente valores**

<li> 1, para obtener los productos con stock > 0 </li> 0, para obtener todos los productos

#### producto::\$empaque

*String = [line 43]*

**Tipo de empaque del producto. No posee ninguna validación**

#### producto::\$fecha\_mod

*DateTime = [line 67]*

**Última fecha de modificación del producto. Este valor se actualiza automáticamente con cada modificación al producto desde el admin**

#### producto::\$funciones

*funciones = [line 87]*

**Clase utilitaria. Esta deprecada y en proceso de eliminación**

#### producto::\$id

*Number = [line 18]*

**Id del producto**

#### producto::\$imagen

*String = [line 73]*

**Imagenes del producto, si existe, la imagen principal se encuentra en \$imagen[0].**

- **Var Array**

**producto::\$link**

*String = [line 99]*

**Link del articulo en el site**

**producto::\$medidas**

*String(20) = [line 49]*

**Medidas del producto. No posee ninguna validación**

**producto::\$nombre**

*String = [line 24]*

**Nombre del producto, considera los acentos y caracteres especiales ya transformados**

**producto::\$peso**

*String(20) = [line 55]*

**Peso del producto escrito. No posee ninguna validación**

**producto::\$precio**

*float(9,2) = [line 37]*

**Precios del producto con dos decimales.**

Los precios se guardan según la moneda del mismo

**producto::\$resumen**

*String = [line 30]*

**Resumen descriptivo del producto**

**producto::\$rutaBase**

*unknown\_type = [line 93]*

**Ruta a partir de la que se encuentran las imagenes del producto. Esta en proceso de ser deprecada**

**producto::\$stock**

*Number = [line 61]*

**Cantidad de productos en el inventario. Este valor se actualiza automáticamente con cada compra**

Constructor *void* function producto::\_\_construct(\$id, \$rutaBase) *[line 106]*

**Function Parameters:**

- *\$id*, **\$id** identificador del producto a obtener
- *\$rutaBase*, **\$rutaBase** Ruta Base para las imagenes... En proceso de ser deprecado

## Constructor

Constructor *void* function producto::\_\_construct() *[line 153]*

**Constructor Base para operaciones masivas de productos o creación de un producto nuevo**

*void* function producto::comprar(\$id, \$cantidad) *[line 142]*

**Function Parameters:**

- *unknown\_type* **\$id**
- *unknown\_type* **\$cantidad**

**Enter description here ...**

*void* function producto::obtenerProductos([\$condiciones = ""], [\$orderBy = ""], [\$maxResults = 0]) *[line 173]*

**Function Parameters:**

- *\$condiciones*, **\$condiciones** condiciones del query a ejecutar para obtener los productos, si no se coloca trae todos los productos
- *\$orderBy*, **\$orderBy** condiciones de ordenamiento de los productos, si no se coloca se ordena por nombre e id de los productos
- *\$maxResults*, **\$maxResults** número máximo de resultados a obtener, 0 significa infinitos

**Funcion para obtener un conjunto de productos dadas un conjunto de condiciones.  
Esta función es afectada por el parámetro de la clase "activos"**

*void function producto::obtenerProductosByQuery(\$query, \$maxResults) [line 200]*

***Function Parameters:***

- *\$query*, **\$query** query a ejecutar
- *\$maxResults*, **\$maxResults** número máximo de resultados a obtener. 0 Significa infinitos

**Función para obtener un conjunto de productos a partir de un query específico a la Base de datos**

*void function producto::setRutaBase(\$ruta) [line 162]*

***Function Parameters:***

- *\$ruta* **\$ruta**

**Permite setear la ruta base para las imagenes.... Está en proceso de ser deprecado**



# Package svcms2 Procedural Elements

## SimpleList.inc

### Clase para presentar los listados en la página.

Estan los listados tanto para artículos como para productos y se pueden usar combinados o separados

Cada listado tiene parametros que lo configuran, sin embargo los parametros de la clase, tanto los propios como los heredados de widget aplican para todos los métodos, según el caso en algunos pueden tener o no significado

- **Package** svcms2
- **Sub-Package** widget
- **Author** Luis Poggi

## Class simpleList

*[line 28]*

### Clase para presentar los listados en la página.

Estan los listados tanto para artículos como para productos y se pueden usar combinados o separados

Cada listado tiene parametros que lo configuran, sin embargo los parametros de la clase, tanto los propios como los heredados de widget aplican para todos los métodos, según el caso en algunos pueden tener o no significado

- **Package** svcms2
- **Sub-Package** widget

- **Author** Luis Poggi

#### **simpleList::\$articulo**

*articulo = [line 68]*

#### **Articulo en el widget**

#### **simpleList::\$autorArticulo**

*unknown\_type = [line 48]*

#### **Enter description here ...**

#### **simpleList::\$categoria**

*unknown\_type = [line 73]*

#### **Enter description here ...**

#### **simpleList::\$formatoPresentacionHoja**

*unknown\_type = [line 43]*

#### **Enter description here ...**

#### **simpleList::\$formatoTitulo**

*unknown\_type = [line 33]*

#### **Enter description here ...**

#### **simpleList::\$funciones**

*unknown\_type = [line 53]*

#### **Enter description here ...**

#### **simpleList::\$maxElementos**

*unknown\_type = [line 58]*

#### **Enter description here ...**

**simpleList::\$producto**

*unknown\_type = [line 78]*

**Enter description here ...**

**simpleList::\$tagInterno**

*unknown\_type = [line 63]*

**Enter description here ...**

**simpleList::\$titulo**

*unknown\_type = [line 38]*

**Enter description here ...**

Constructor *void* function simpleList::\_\_construct() *[line 85]*

**Constructor. Permite instanciar el widget.**

*void* function simpleList::articulosDestacados() *[line 170]*

## Ejemplo de uso de los articulos Destacados

```
1 <?php
2
3 ?>
```

**Permite listar los articulos destacados del site. Utiliza las siguientes variables**

titulo, Si no esta configurado toma el valor por defecto "Artículos Destacados",

si se coloca como valor "NO", no se carga ningun valor

autorArticulo, Si no esta configurado toma el valor por defecto "1"

rutaBase, permite configurar la ruta desde donde se toman las imagenes

formatoPresentacion, ubicación del archivo de presentación de los articulos encontrados

formatoTitulo, ubicación del archivo de presentación del titulo de los articulos. Si se coloca como valor "NO", no se muestra el formato

```
1 <?php
2 $widget = new simpleList();
3 $widget-> formatoPresentacion="SVwidgets/SimpleList/formato-articulo.php" ;
4 $widget-> formatoTitulo="SVwidgets/SimpleList/formato-titulo.php" ;
5 $widget-> titulo="Importante..." ;
6 $widget-> rutaBase = $DOMINIOSV;
7 $widget-> articulosDestacados();
8 ?>
```

- **Example**



*void function simpleList::articulosEnCategoria(\$catId, \$nivel, [\$orderBy = ""], [\$maxResults = 0]) [line 185]*

## Muestra los articulos de una categoria

```
1 <?php
2
3
4 ?>
```

### **Function Parameters:**

- *Number* **\$catId** Identificador de la categoria a mostrar
- *Number* **\$nivel** Nivel de la categoria a mostrar
- *String* **\$orderBy** campos separados por coma por los cuales se va a ordenar el resultado, si no se configura ninguno se ordena por orden, titulo, id
- *Number* **\$maxResults** maximo numero de resultados a mostrar. Si se coloca 0 (o no se coloca nada) salen todos los resultados posibles

**Permite obtener los articulos en una categoría específica.**

- **Example**

*void function simpleList::articulosMasVistos() [line 229]*

**Enter description here ...**

*void function simpleList::articulosRelacionados(\$idArticulo) [line 240]*

### **Function Parameters:**

- *unknown\_type* **\$idArticulo**

**Enter description here ...**

*void function simpleList::categoriasDeArticulos([\$nivel = 1], [\$mostrarArticulos = true], [\$incluirGeneral = true]) [line 272]*

## Agregar las categorias a una barra

```
1 <?php includeWidget("SimpleList.SimpleList" );?>
2 <div class="menu">
3 <ul class="menu-1">
```

```

4     <?php
5         $widget = new simpleList();
6         $widget-> formatoPresentacion="SVwidgets/SimpleList/formato-barra-categorias.php" ;
7         $widget-> formatoPresentacionHoja="SVwidgets/SimpleList/formato-barra-articulos-
categorias.php" ;
8         $widget-> formatoTitulo="NO" ;
9         $widget-> rutaBase = $DOMINIOSV;
10        $widget-> categoriasDeArticulos(4,true,false);
11    ?>
12    <?php
13        $widget = new simpleList();
14        $widget-> formatoPresentacion="SVwidgets/SimpleList/formato-barra-categorias.php" ;
15        $widget-> formatoPresentacionHoja="SVwidgets/SimpleList/formato-barra-articulos-
categorias.php" ;
16        $widget-> formatoTitulo="NO" ;
17        $widget-> rutaBase = $DOMINIOSV;
18        $widget-> categoriasDeProductos(4,true);
19    ?>
20    </ul>
21    </div>

```

#### Function Parameters:

- **Number \$nivel** Determina el nivel de categorias que se muestran
- **boolean \$mostrarArticulos** Indica si se muestran los articulos en el nivel final. Acepta los valores true para mostrarlos false para no mostrarlos
- **boolean \$incluirGeneral** La categoria #1

#### Permite encontrar las categorias y subcategorias de articulos que están configuradas.

Según las configuraciones se muestran mas o menos niveles de categorias y los articulos de la misma. Además de los parámetros toma en cuenta lo siguientes parametros de la clase formatoPresentacion, formato para las categorías. Se debe colocar en el formato el tag <element/> para indicar donde se quieren los hijos. formatoPresentacionHoja, formato del último nivel del widget. Si se configura para mostrarse 4 niveles, pero en el nivel 2 alguna categoría no tiene mas hijos, los articulos (si están configurados para salir) se muestran con este formato formatoTitulo, formato del titulo del widget rutaBase, próximo a ser eliminado, indica la ruta base de las imagenes. Se configura con \$DOMINIOSV usualmente En el comportamiento estandar de la función se incluye la revisión de la permisología de las categorías. Si en la sesión no hay usuario logueado solo se muestran las categorias con only\_for=0 Si en la sesión hay un usuario logueado se muestran las categorias con only\_for=0,1 y cualquier grupo al cual el pertenezca.

#### • Example

void function simpleList::categoriasDeProductos([\$nivel = 1], [\$mostrarProductos = true]) [line 524]

#### Function Parameters:

- **unknown\_type \$nivel**

- *unknown\_type* **\$mostrarProductos**

Enter description here ...

*void function simpleList::categoriasEnCategoria(\$catId, \$nivel, [\$orderBy = ""], [\$maxResults = 0]) [line 205]*

**Function Parameters:**

- *unknown\_type* **\$catId**
- *unknown\_type* **\$nivel**
- *unknown\_type* **\$orderBy**
- *unknown\_type* **\$maxResults**

Enter description here ...

*void function simpleList::elementosEnCategorias() [line 862]*

Enter description here ...

*void function simpleList::listarArticulos([\$titulo = "NO"], [\$condiciones = ""], [\$orderBy = ""], [\$maxResults = 0], \$titulo,, \$condiciones,, \$orderBy,, \$maxResults,) [line 131]*

**Function Parameters:**

- *String* **\$titulo**, titulo del widget a ser mostrado en el site
- *String* **\$condiciones**, condiciones del query para la base de datos
- *String* **\$orderBy**, campos de la base de datos por los cuales se va a ordenar el listado
- *Number* **\$maxResults**, número máximo de resultados que se van a presentar, si no se indica el valor se asume valor por defecto = 0 (es decir, muestra todos los resultados que se obtengan)
- **\$titulo**
- **\$condiciones**
- **\$orderBy**
- **\$maxResults**

**Permite mostrar los artículos dadas unas condiciones específicas**

**Su uso no se recomienda, ha sido creado para ser usado por métodos de la clase que enmascaren los queries.**

*void function simpleList::listarArticulosByQuery(\$titulo, \$query, \$maxResults, \$titulo,, \$query,, \$maxResults,) [line 105]*

**Function Parameters:**

- *String* **\$titulo**, título del widget a mostrar en la página
- *String* **\$query**, query directo a la base de datos
- *Number* **\$maxResults**, número máximo de resultados que se van a presentar
- **\$titulo**
- **\$query**
- **\$maxResults**

### **Sirve para obtener un listado de artículos dado un query específico.**

Su uso no se recomienda, ha sido creado para ser usado por métodos de la clase que enmascaren los queries.

*void function simpleList::listarProductos([\$titulo = "NO"], [\$condiciones = ""], [\$orderBy = ""], [\$maxResults = 0]) [line 760]*

#### **Function Parameters:**

- *unknown\_type* **\$titulo**
- *unknown\_type* **\$condiciones**
- *unknown\_type* **\$orderBy**
- *unknown\_type* **\$maxResults**

**Enter description here ...**

*void function simpleList::listarProductosByQuery(\$titulo, \$query, \$maxResults) [line 736]*

#### **Function Parameters:**

- *unknown\_type* **\$titulo**
- *unknown\_type* **\$query**
- *unknown\_type* **\$maxResults**

**Enter description here ...**

*void function simpleList::productosDestacados() [line 796]*

**Enter description here ...**

*void function simpleList::productosEnCategoria(\$catId, \$nivel, [\$orderBy = ""], [\$maxResults = 0]) [line 784]*

#### **Function Parameters:**

- *unknown\_type* **\$catId**

- *unknown\_type* \$nivel
- *unknown\_type* \$orderBy
- *unknown\_type* \$maxResults

**Enter description here ...**

*void function simpleList::productosMasVendidos() [line 805]*

**Enter description here ...**

*void function simpleList::productosMasVistos() [line 814]*

**Enter description here ...**

*void function simpleList::productosRelacionados(\$idProducto) [line 825]*

**Function Parameters:**

- *unknown\_type* \$idProducto

**Enter description here ...**

*number function simpleList::subCategoriasDeArticulos(\$catId, \$nivel, [\$mostrarArticulos = true]) [line 396]*

**Function Parameters:**

- *unknown\_type* \$catId
- *unknown\_type* \$nivel
- *unknown\_type* \$mostrarArticulos

**Enter description here ...**

*number function simpleList::subCategoriasDeProductos(\$catId, \$nivel, [\$mostrarProductos = true]) [line 608]*

**Function Parameters:**

- *unknown\_type* \$catId
- *unknown\_type* \$nivel
- *unknown\_type* \$mostrarProductos

**Enter description here ...**

*void* function simpleList::subSubCategoriasDeArticulos(\$catId, \$nivel, [\$mostrarArticulos = true]) [*line 465*]

**Function Parameters:**

- *unknown\_type* **\$catId**
- *unknown\_type* **\$nivel**
- *unknown\_type* **\$mostrarArticulos**

**Enter description here ...**

*void* function simpleList::subSubCategoriasDeProductos(\$catId, \$nivel, [\$mostrarProductos = true]) [*line 677*]

**Function Parameters:**

- *unknown\_type* **\$catId**
- *unknown\_type* **\$nivel**
- *unknown\_type* **\$mostrarProductos**

**Enter description here ...**

*string* function simpleList::traer\_item\_categoria(\$categoria, \$tabla, \$numero, \$orden) [*line 846*]

**Function Parameters:**

- *unknown\_type* **\$categoria**
- *unknown\_type* **\$tabla**
- *unknown\_type* **\$numero**
- *unknown\_type* **\$orden**

**Enter description here ...**

# Appendices

# Appendix A - Class Trees

## Package svcms1

### database

- [database](#)
  - [tools](#)
    - [formulario](#)

### funciones

- [funciones](#)

### widget

- [widget](#)
  - [shoppingCart](#)
  - [simpleList](#)

## Package svcms2

### articulo

- [articulo](#)



## categoria

- [categoria](#)

## producto

- [producto](#)

## simpleList

- [widget](#) (Different package)
  - [simpleList](#)

# Index

## A

<a href="#">articulo::obtenerArticulosByQuery()</a>	34
<a href="#">articulo::setRutaBase()</a>	34
<a href="#">articulo::obtenerArticulos()</a>	33
<a href="#">articulo::\$rutaBase</a>	33
<a href="#">articulo::\$activos</a>	33
<a href="#">articulo::\$funciones</a>	33
<a href="#">articulo</a>	33

*Clase que representa a los articulos*

## C

<a href="#">categoria::obtenerCategoriasByQuery()</a>	35
<a href="#">categoria::obtenerCategorias()</a>	35
<a href="#">categoria::obtenerCategoria()</a>	34
<a href="#">categoria::obtenerSubCategorias()</a>	35
<a href="#">categoria::setRutaBase()</a>	35
<a href="#">constructor simpleList::__construct()</a>	43
<i>Constructor. Permite instanciar el widget.</i>	
<a href="#">constructor producto::__construct()</a>	38
<i>Constructor Base para operaciones masivas de productos o creación de un producto nuevo</i>	
<a href="#">constructor producto::__construct()</a>	38
<i>Constructor</i>	
<a href="#">constructor categoria::__construct()</a>	34
<a href="#">categoria::\$rutaBase</a>	34
<a href="#">constructor tools::__construct()</a>	23
<a href="#">constructor formulario::__construct()</a>	22

\*\*\*\*\* CONSTRUCTOR

<a href="#">constructor database::__construct()</a>	18
<i>constructor de la clase</i>	
<a href="#">constructor articulo::__construct()</a>	33
<a href="#">categoria</a>	34
<i>Clase que permite administrar las categorias</i>	
<a href="#">categoria::\$nivel</a>	34
<a href="#">categoria::\$funciones</a>	34
<a href="#">constructor shoppingCart::__construct()</a>	14

*Constructor de la clase. El comportamiento por defecto es que se guarda en cookies*

## D

<a href="#">database::numero_filas_afecc()</a>	20
<a href="#">database::liberar()</a>	20

*metodo para liberar el objeto que retorna el valor del query*

<a href="#">database::insertar2()</a>	20
<i>funcion que realiza un insert simple de los valores ingresados en un vector</i>	
<a href="#">database::insertar()</a>	20
<i>funcion que realiza un insert simple tabla, campos, valores y boqueo true o false (opcional)</i>	
<a href="#">database::query()</a>	20
<i>metodo que ejecuta un query de creacion o modificacion o devuelve un error el segundo parametro es para liberar el recurso de una vez</i>	
<a href="#">database::reciclar_conexion()</a>	21
<i>permite utilizar la conexion a una database ya creada sin tener que reconectar;</i>	
<a href="#">database::update()</a>	21
<i>funcion que realiza un update simple de los valores y campos ingresados en un vector</i>	
<a href="#">database::ultimo_id()</a>	21
<a href="#">database::total_registros()</a>	21
<a href="#">database::conectar()</a>	19
<i>propiedad que cambia la conexion a una nueva database o server con solo invocarla se cambia la conexion el ultimo parametro determina si la conexion es persistente o no.</i>	
<a href="#">database::cerrar_transaccion()</a>	19
<i>Enter description here ...</i>	
<a href="#">database::\$ultimoID</a>	18
<i>ultimo id insertado (auto incremental)</i>	
<a href="#">database::\$result</a>	18
<i>atributo que tiene el resultado del query</i>	
<a href="#">database::\$nreg</a>	17
<i>cantidad de registros insertados</i>	
<a href="#">database::\$dbc</a>	17
<i>variable de conexion</i>	
<a href="#">database::abrir_transaccion()</a>	18
<i>Enter description here ...</i>	
<a href="#">database::autoconexion()</a>	19
<i>autoconexion discreta segun los parametros y la base dedatos especificadas en el archivo de conf</i>	
<a href="#">database::cerrar()</a>	19
<i>metodo para cerrar la conexion con la base de datos. En este momento esta sin uso</i>	
<a href="#">database::campos_query()</a>	19
<i>devuelve un arreglo con el nombre de los campos consultados enpezando desde la pos 0</i>	
<a href="#">database::borrar()</a>	19
<a href="#">database</a>	17
<i>clase que crea todas las funciones necesarias para trabajr con bases de datos</i>	

## F

<a href="#">formulario</a>	21
<i>clase para la construccion de formularios y volcamiento de data proveniente de los mismos</i>	
<a href="#">funciones::stripaccents()</a>	13
<a href="#">formulario::getvar()</a>	22
<a href="#">formulario::insert_data()</a>	22
<a href="#">formulario::update_data()</a>	23
<a href="#">funciones::LoadCalendario()</a>	13
<a href="#">funciones::limpiarUrlBD()</a>	12
<a href="#">funciones</a>	12
<a href="#">funciones::borrar_imagenes()</a>	12
<a href="#">funciones::borrar_imagenes2()</a>	12

<a href="#">funciones::limpiarUrl()</a>	12
<a href="#">fuente-SimpleEmailEdit.php</a>	4

## I

<a href="#">includeCommonClass()</a>	2
<a href="#">includeCommon()</a>	2

## P

<a href="#">producto::\$rutaBase</a>	38
<i>Ruta a partir de la que se encuentran las imagenes del producto. Esta en proceso de ser deprecada</i>	
<a href="#">producto::\$resumen</a>	37
<i>Resumen descriptivo del producto</i>	
<a href="#">producto::\$precio</a>	37
<i>Precios del producto con dos decimales.</i>	
<a href="#">producto::\$stock</a>	38
<i>Cantidad de productos en el inventario. Este valor se actualiza automáticamente con cada compra</i>	
<a href="#">producto::comprar()</a>	38
<i>Enter description here ...</i>	
<a href="#">producto::setRutaBase()</a>	39
<i>Permite setear la ruta base para las imagenes.... Está en proceso de ser deprecado</i>	
<a href="#">producto::obtenerProductosByQuery()</a>	39
<i>Función para obtener un conjunto de productos a partir de un query específico a la Base de datos</i>	
<a href="#">producto::obtenerProductos()</a>	38
<i>Funcion para obtener un conjunto de productos dadas un conjunto de condiciones. Esta función es afectada por el parámetro de la clase "activos"</i>	
<a href="#">producto::\$peso</a>	37
<i>Peso del producto escrito. No posee ninguna validación</i>	
<a href="#">producto::\$nombre</a>	37
<i>Nombre del producto, considera los acentos y caracteres especiales ya transformados</i>	
<a href="#">producto::\$fecha_mod</a>	36
<i>Última fecha de modificación del producto. Este valor se actualiza automáticamente con cada modificación al producto desde el admin</i>	
<a href="#">producto::\$empaquetado</a>	36
<i>Tipo de empaque del producto. No posee ninguna validación</i>	
<a href="#">producto::\$activos</a>	36
<i>Esta variable permite configurar si se buscan TODOS los productos o sólo aquellos con stock &gt; 0. Admite los siguiente valores</i>	
<a href="#">producto::\$funciones</a>	36
<i>Clase utilitaria. Esta deprecada y en proceso de eliminación</i>	
<a href="#">producto::\$id</a>	36
<i>Id del producto</i>	
<a href="#">producto::\$medidas</a>	37
<i>Medidas del producto. No posee ninguna validación</i>	
<a href="#">producto::\$link</a>	37
<i>Link del articulo en el site</i>	
<a href="#">producto::\$imagen</a>	36

<a href="#">Imágenes del producto, si existe, la imagen principal se encuentra en \$imagen[0].</a>	35
<a href="#">Clase que representa las acciones que se pueden ejecutar sobre un producto</a>	

## R

<a href="#">remoteFunctions.php</a>	2
-------------------------------------	---

## S

<a href="#">simpleList::categoriasDeProductos()</a>	45
<i>Enter description here ...</i>	
<a href="#">simpleList::categoriasDeArticulos()</a>	44
<i>Permite encontrar las categorías y subcategorías de artículos que están configuradas.</i>	
<a href="#">simpleList::categoriasEnCategoria()</a>	46
<i>Enter description here ...</i>	
<a href="#">simpleList::elementosEnCategorias()</a>	46
<i>Enter description here ...</i>	
<a href="#">simpleList::listarArticulos()</a>	46
<i>Permite mostrar los artículos dadas unas condiciones específicas</i>	
<i>&lt;br&gt;Su uso no se recomienda, ha sido creado para ser usado por métodos de la clase que enmascaren los queries.</i>	
<a href="#">simpleList::articulosRelacionados()</a>	44
<i>Enter description here ...</i>	
<a href="#">simpleList::articulosMasVistos()</a>	44
<i>Enter description here ...</i>	
<a href="#">simpleList::\$tagInterno</a>	43
<i>Enter description here ...</i>	
<a href="#">simpleList::\$producto</a>	43
<i>Enter description here ...</i>	
<a href="#">simpleList::\$titulo</a>	43
<i>Enter description here ...</i>	
<a href="#">simpleList::articulosDestacados()</a>	43
<i>Permite listar los artículos destacados del site. Utiliza las siguientes variables</i>	
<a href="#">simpleList::articulosEnCategoria()</a>	44
<i>Permite obtener los artículos en una categoría específica.</i>	
<a href="#">simpleList::listarArticulosByQuery()</a>	46
<i>Sirve para obtener un listado de artículos dado un query específico.</i>	
<a href="#">simpleList::listarProductos()</a>	47
<i>Enter description here ...</i>	
<a href="#">simpleList::subCategoriasDeProductos()</a>	48
<i>Enter description here ...</i>	
<a href="#">simpleList::subCategoriasDeArticulos()</a>	48
<i>Enter description here ...</i>	
<a href="#">simpleList::subSubCategoriasDeArticulos()</a>	49
<i>Enter description here ...</i>	
<a href="#">simpleList::subSubCategoriasDeProductos()</a>	49
<i>Enter description here ...</i>	
<a href="#">simpleList::traer_item_categoria()</a>	49
<i>Enter description here ...</i>	
<a href="#">simpleList::productosRelacionados()</a>	48

<i>Enter description here ...</i>	48
<a href="#">simpleList::productosMasVistos()</a>	48
<i>Enter description here ...</i>	47
<a href="#">simpleList::listarProductosByQuery()</a>	47
<i>Enter description here ...</i>	47
<a href="#">simpleList::productosDestacados()</a>	47
<i>Enter description here ...</i>	47
<a href="#">simpleList::productosEnCategoria()</a>	47
<i>Enter description here ...</i>	48
<a href="#">simpleList::productosMasVendidos()</a>	48
<i>Enter description here ...</i>	42
<a href="#">simpleList::\$maxElementos</a>	42
<i>Enter description here ...</i>	42
<a href="#">simpleList::\$funciones</a>	42
<i>Enter description here ...</i>	14
<a href="#">shoppingCart::\$productos</a>	14
<i>Productos almacenados en el carrito con sus cantidades.</i>	
<a href="#">shoppingCart::\$guardadoEnCookies</a>	13
<i>Indica si el carrito se almacena en cookies o no.</i>	
<a href="#">shoppingCart::agregarProducto()</a>	14
<i>Permite agregar un producto al carrito de compras</i>	
<a href="#">shoppingCart::cambiarCantidadProducto()</a>	14
<i>Cambia la cantidad de producto de un carrito</i>	
<a href="#">shoppingCart::eliminarProducto()</a>	15
<i>Saca un producto del carrito</i>	
<a href="#">shoppingCart</a>	13
<i>Clase para representar al carrito de compras. Permite manipularlo y trabajarlo.</i>	
<a href="#">searchSuggest.php</a>	10
<a href="#">saveAll.php</a>	6
<a href="#">save.php</a>	5
<a href="#">SimpleEmail.php</a>	7
<a href="#">SV-detalle-articulo.php</a>	8
<a href="#">Search.php</a>	9
<a href="#">shoppingCart::obtenerCookie()</a>	15
<i>Enter description here ...</i>	
<a href="#">shoppingCart::obtenerShoppingCart()</a>	15
<i>Enter description here ...</i>	
<a href="#">simpleList::\$autorArticulo</a>	42
<i>Enter description here ...</i>	
<a href="#">simpleList::\$articulo</a>	42
<i>Articulo en el widget</i>	
<a href="#">simpleList::\$categoria</a>	42
<i>Enter description here ...</i>	
<a href="#">simpleList::\$formatoPresentacionHoja</a>	42
<i>Enter description here ...</i>	
<a href="#">simpleList::\$formatoTitulo</a>	42
<i>Enter description here ...</i>	
<a href="#">simpleList</a>	41
<i>Clase para presentar los listados en la página.</i>	
<a href="#">SimpleList.inc</a>	41
<i>Clase para presentar los listados en la página.</i>	
<a href="#">shoppingCart::salvar()</a>	15
<i>Guarda el carrito. Es afectado por la variable guardadoEnCookies, según esta indique se</i>	

<i>almacena en BD o en cookies</i>	
<a href="#">shoppingCart::saveCookie()</a>	15
<i>Guarda el carrito en la sesion</i>	
<a href="#">shoppingCart::saveDB()</a>	15
<i>Guarda el carrito en Base de datos</i>	
<a href="#">shoppingCart::show()</a>	15
<i>Enter description here ...</i>	
<a href="#">setup.php</a>	3
***** PARAMETROS MODIFICABLES *****	

## T

<a href="#">tools::obtenerConfigPages()</a>	29
<a href="#">tools::obtenerConfiguracionProductos()</a>	29
<a href="#">tools::obtenerCategorias()</a>	29
<a href="#">tools::obtenerArticulos()</a>	29
<a href="#">tools::listar_archivos()</a>	28
<a href="#">tools::llenar_array()</a>	28
<a href="#">tools::obtenerProductos()</a>	30
<a href="#">tools::redirect()</a>	30
<a href="#">tools::unicos()</a>	31
<a href="#">tools::upload_file()</a>	32
<a href="#">tools::suma_array()</a>	31
<a href="#">tools::simple_db()</a>	31
<a href="#">tools::repetidos()</a>	30
<a href="#">tools::seencuentra()</a>	30
<a href="#">tools::limpiar_array()</a>	28
<a href="#">tools::javaviso()</a>	28
<a href="#">tools::burbuja()</a>	24
<a href="#">tools::buscar_estructdb()</a>	25
<a href="#">tools::array_query2()</a>	24
<a href="#">tools::array_query()</a>	24
<a href="#">tools</a>	23
<i>clase que cosntruye diversas herramientas fopara usar en la construccion de forms</i>	
<a href="#">tools::abreviar()</a>	24
<a href="#">tools::combo_array()</a>	25
<a href="#">tools::combo_db()</a>	26
<a href="#">tools::estructura_db()</a>	27
<a href="#">tools::format_txt()</a>	27
<a href="#">tools::eliminar_de()</a>	27
<a href="#">tools::delete_directory()</a>	26
<a href="#">tools::combo_dir()</a>	26
<a href="#">tools.inc</a>	17

## U

<a href="#">USAGE.php</a>	11
---------------------------	----

## W

<a href="#"><u>widget::\$rutaBase</u></a>	16
<i>Ruta Base donde se encuentran las imágenes. Próximo a ser deprecado</i>	
<a href="#"><u>widget::\$formatoSinResultados</u></a>	16
<i>Formato para colocar el widget sin resultados.</i>	
<a href="#"><u>widget::\$formatoPresentacion</u></a>	16
<i>Formato de la presentación, sirve para mostrar los resultados del widget</i>	
<a href="#"><u>widget</u></a>	15
<i>Clase que representa los widgets. Todos los widgets heredan las propiedades y métodos aquí definidos</i>	
<hr/>	
<a href="#"><u>__destruct()</u></a>	17