

Documentation



Contents

Package default Procedural Elements	2
mvc.controller.php	2
Administrador.class.php	3
genero.class.php	4
juego.class.php	5
usuario.class.php	6
index.php	7
recaptchalib.php	8
Package default Classes	9
Class Administrador	9
Method actualizar	9
Method modificarOfertas	9
Class Database	10
Method conectar	10
Method consulta	10
Method desconectar	10
Method fetch_assoc	11
Method numero_filas	11
Class Genero	11
Method buscarGenero	12
Method juegosGenero	12
Method listarGeneros	12
Class Juego	12
Method biblioteca	12
Method buscar	13
Method buscarAvanzado	13
Method favoritos	13
Method mostrarDatos	13
Method mostrarPrecios	13
Method portada	13
Class mvc_controller	13
Method buscador	14
Method genero	14
Method paginaJuego	14
Method perfil	14
Method principal	14
Method registrarse	14
Method resultadoBusqueda	14
Class ReCaptcha	14
Constructor ReCaptcha	15
Method verifyResponse	15
Class ReCaptchaResponse	15
Var \$errorCodes	15

Var \$success	16
Class Usuario	16
Method bibliotecaPerfil	16
Method datosPerfil	16
Method favoritosPerfil	16
Method identificar	16
Method login	17
Method logout	17
Method registrar	17
Method videosPerfil	17
Appendices	18
Appendix A - Class Trees	19
default	19

Package default Procedural Elements

mvc.controller.php

- **Package** default

require **'pageGenerator.php'** *[line 6]*

require ['app/model/genero.class.php'](#) *[line 5]*

require ['app/model/juego.class.php'](#) *[line 4]*

require ['app/model/usuario.class.php'](#) *[line 3]*

Administrador.class.php

- **Package** default

require_once ["juego.class.php"](#)*[line 8]*

require_once ["db.class.php"](#)*[line 7]*

Clase administrador

- **Author** Digital Games
- **Version** 1.0

genero.class.php

- **Package** default

require_once ["db.class.php"](#)*[line 2]*

juego.class.php

- **Package** default

require_once ["db.class.php"](#)*[line 2]*

usuario.class.php

- **Package** default

require_once ["db.class.php"](#)*[line 2]*

index.php

- **Package** default

require ['app/controller/mvc.controller.php'](#) [line 3]

require_once ['recaptchalib.php'](#) [line 4]

recaptchalib.php

This is a PHP library that handles calling reCAPTCHA.

- Documentation and latest version
<https://developers.google.com/recaptcha/docs/php>
- Get a reCAPTCHA API Key
<https://www.google.com/recaptcha/admin/create>
- Discussion group
<http://groups.google.com/group/recaptcha>

- **Package** default
- **Copyright** Copyright (c) 2014, Google Inc.
- **Link** Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Package default Classes

Class Administrador

[line 9]

Clase para interactuar con el usuario registrado

Todas las funciones relacionadas con un usuario se encuentran aquí-

- **Package** default
- **Author** Digital Games
- **Version** 1.0

void function Administrador::actualizar() [line 13]

Función para actualizar la lista de precios y de juegos

void function Administrador::modificarOfertas(\$id, \$precio) [line 165]

Function Parameters:

- *integer* **\$id** identificador del juego cuya oferta se quiere modificar
- *float* **\$precio** nuevo precio a insertar Metodo sin implementar en esta version

Función para actualizar la lista de precios y de juegos

Class Database

[line 8]

Clase que interactua con la base de datos

- **Package** default
- **Author** Digital Games
- **Version** 1.0

array function Database::conectar() [line 17]

Realiza la conexion con la base de datos

- **Access** public

array function Database::consulta(\$sentencia) [line 28]

Function Parameters:

- *string* **\$sentencia** sentencia en formato sql

Realiza una consulta en la base de datos la base de datos

- **Access** public

void function Database::desconectar() [line 49]

Cierra la conexion con la base de datos

- **Access** public

void function Database::fetch_assoc(\$resultado) [line 42]

Function Parameters:

- **\$resultado**

Devuelve los valores de la consulta

- **Access public**

void function Database::numero_filas(\$resultado) [line 36]

Function Parameters:

- **array \$resultado** valor de una consulta

Pide el numero filas obtenido tras una consulta

- **Access public**

Class Genero

[line 8]

Clase que contiene las funciones listar y buscar por g nero

- **Package** default
- **Author** Digital Games
- **Version** 1.0

array function Genero::buscarGenero() [*line 72*]

Busqueda de todos los juegos de un genero

array function Genero::juegosGenero(\$genero) [*line 40*]

Function Parameters:

- *integer* **\$genero** identificador del genero

Metodo que selecciona los tres juegos más populares de un genero concreto

void function Genero::listarGeneros() [*line 13*]

Metodo que lista los nombres de los generos

Class Juego

[*line 9*]

Clase para listar y mostrar toda la información de los juegos

También añade juegos en las tablas favoritos y biblioteca de la base de datos

- **Package** default
- **Author** Digital Games
- **Version** 1.0

void function Juego::biblioteca(\$id, \$sesion) [*line 175*]

Function Parameters:

- *integer* **\$id** identificador del juego

- *integer* **\$sesion** identificador del usuario registrado

Metodo que inserta en la base de datos un juego en la biblioteca del usuario registrado

array function Juego::buscar() [*line 89*]

Metodo que realiza una busqueda por nombre

array function Juego::buscarAvanzado() [*line 125*]

Metodo que busca por una serie de características

void function Juego::favoritos(\$id, \$sesion) [*line 162*]

Function Parameters:

- *integer* **\$id** identificador del juego
- *integer* **\$sesion** identificador del usuario registrado

Metodo que inserta en la base de datos un juego como favorito del usuario registrado

array function Juego::mostrarDatos() [*line 45*]

Carga los datos de un juego

array function Juego::mostrarPrecios() [*line 67*]

Metodo que carga los precios de cada tienda de un juego

array function Juego::portada() [*line 15*]

Metodo que selecciona juegos más nuevos con cierto grado de popular y no gratuitos

Class mvc_controller

[*line 16*]

Clase controlador

Relaciona el modelo con las vistas Es el nucleo de la carga de cada una de las paginas

- **Package** default
- **Author** Digital Games
- **Version** 1.0

void function mvc_controller::buscador() [line 167]

Carga el buscador avanzado

void function mvc_controller::genero() [line 59]

Carga los componentes de genero

void function mvc_controller::paginaJuego() [line 318]

Carga la pagina del juego con sus características y precios

void function mvc_controller::perfil() [line 229]

Carga el perfil del usuario registrado

void function mvc_controller::principal() [line 21]

Carga los componentes de la pagina de inicio

void function mvc_controller::registrarse() [line 298]

Carga el formulario de registro y de inicio de sesión

void function mvc_controller::resultadoBusqueda() [line 180]

Carga el resultado de una búsqueda (por nombre o avanzada)

Class ReCaptcha

[line 40]

- **Package** default

Constructor *void* function ReCaptcha::ReCaptcha(\$secret) [*line 52*]

Function Parameters:

- *string* **\$secret** shared secret between site and ReCAPTCHA server.

Constructor.

ReCaptchaResponse function ReCaptcha::verifyResponse(\$remotelp, \$response) [*line 100*]

Function Parameters:

- *string* **\$remotelp** IP address of end user.
- *string* **\$response** response string from recaptcha verification.

Calls the reCAPTCHA siteverify API to verify whether the user passes CAPTCHA test.

- **Access** public

Class ReCaptchaResponse

[*line 35*]

A ReCaptchaResponse is returned from checkAnswer().

- **Package** default

ReCaptchaResponse::\$errorCodes

mixed = [*line 38*]

- **Access** public

ReCaptchaResponse::\$success

mixed = [line 37]

- **Access** public

Class Usuario

[line 9]

Clase para interactuar con el usuario registrado

Todas las funciones relacionadas con un usuario se encuentran aquí-

- **Package** default
- **Author** Digital Games
- **Version** 1.0

array function Usuario::bibliotecaPerfil() [line 163]

Carga los juegos de la biblioteca del usuario

void function Usuario::datosPerfil() [line 98]

Carga los datos del usuario que tiene iniciada la sesion

array function Usuario::favoritosPerfil() [line 141]

Carga los juegos favoritos del usuario

integer function Usuario::identificar() [line 122]

Carga el identificador del usuario cuya sesion se encuentra iniciada

void function Usuario::login() [line 54]

Inicia una sesion

void function Usuario::logout() [line 91]

Cierra la sesión que se encontraba iniciada

void function Usuario::registrar() [line 13]

Registra un nuevo usuario y se inserta en la base de datos

array function Usuario::videosPerfil() [line 186]

Carga los videos subidos del usuario

Appendices

Appendix A - Class Trees

Package default

Database

- [Database](#)
 - [Genero](#)
 - [Juego](#)
 - [Usuario](#)
 - [Administrador](#)

mvc_controller

- [mvc_controller](#)

ReCaptcha

- [ReCaptcha](#)

ReCaptchaResponse

- [ReCaptchaResponse](#)

Index

A

Administrador::modificarOfertas()	9
<i>Funci3n para actualizar la lista de precios y de juegos</i>	
Administrador::actualizar()	9
<i>Funci3n para actualizar la lista de precios y de juegos</i>	
Administrador	9
<i>Clase para interactuar con el usuario registrado</i>	
Administrador.class.php	3

C

constructor ReCaptcha::ReCaptcha()	15
<i>Constructor.</i>	

D

Database::fetch_assoc()	11
<i>Devuelve los valores de la consulta</i>	
Database::numero_filas()	11
<i>Pide el numero filas obtenido tras una consulta</i>	
Database::desconectar()	10
<i>Cierra la conexi3n con la base de datos</i>	
Database::consulta()	10
<i>Realiza una consulta en la base de datos la base de datos</i>	
Database::conectar()	10
<i>Realiza la conexi3n con la base de datos</i>	
Database	10
<i>Clase que interactua con la base de datos</i>	

G

Genero::listarGeneros()	12
<i>Metodo que lista los nombres de los generos</i>	
Genero::juegosGenero()	12
<i>Metodo que selecciona los tres juegos m3s populares de un genero concreto</i>	
Genero::buscarGenero()	12
<i>Busqueda de todos los juegos de un genero</i>	
Genero	11
<i>Clase que contiene las funciones listar y buscar por g3nero</i>	
genero.class.php	4

I	
index.php	7

J	
Juego::mostrarDatos()	13
<i>Carga los datos de un juego</i>	
Juego::mostrarPrecios()	13
<i>Metodo que carga los precios de cada tienda de un juego</i>	
Juego::portada()	13
<i>Metodo que selecciona juegos mÃ¡s nuevos con cierto grado de popular y no gratuitos</i>	
Juego::favoritos()	13
<i>Metodo que inserta en la base de datos un juego como favorito del usuario registrado</i>	
Juego::buscarAvanzado()	13
<i>Metodo que busca por una serie de características</i>	
Juego	12
<i>Clase para listar y mostrar toda la informaciÃ³n de los juegos</i>	
Juego::biblioteca()	12
<i>Metodo que inserta en la base de datos un juego en la biblioteca del usuario registrado</i>	
Juego::buscar()	13
<i>Metodo que realiza una busqueda por nombre</i>	
juego.class.php	5

M	
mvc_controller::principal()	14
<i>Carga los componentes de la pagina de inicio</i>	
mvc_controller::registrarse()	14
<i>Carga el formulario de registro y de inicio de sesiÃ³n</i>	
mvc_controller::resultadoBusqueda()	14
<i>Carga el resultado de una busqueda (por nombre o avanzada)</i>	
mvc_controller::perfil()	14
<i>Carga el perfil del usuario registrado</i>	
mvc_controller::paginaJuego()	14
<i>Carga la pagina del juego con sus características y precios</i>	
mvc_controller	13
<i>Clase controlador</i>	
mvc_controller::buscador()	14
<i>Carga el buscador avanzado</i>	
mvc_controller::genero()	14
<i>Carga los componentes de genero</i>	
mvc.controller.php	2

R	
ReCaptchaResponse::\$errorCodes	15
ReCaptchaResponse::\$success	16
ReCaptchaResponse	15
<i>A ReCaptchaResponse is returned from checkAnswer().</i>	

ReCaptcha::verifyResponse()	15
<i>Calls the reCAPTCHA siteverify API to verify whether the user passes CAPTCHA test.</i>	
ReCaptcha	14
recaptchalib.php	8
<i>This is a PHP library that handles calling reCAPTCHA.</i>	

U

Usuario::login()	17
<i>Inicia una sesion</i>	
Usuario::logout()	17
<i>Cierra la sesiÃ³n que se encontraba iniciada</i>	
Usuario::registrar()	17
<i>Registra un nuevo usuario y se inserta en la base de datos</i>	
Usuario::videosPerfil()	17
<i>Carga los videos subidos del usuario</i>	
Usuario::identificar()	16
<i>Carga el identificador del usuario cuya sesion se encuentra iniciada</i>	
Usuario::favoritosPerfil()	16
<i>Carga los juegos favoritos del usuario</i>	
Usuario	16
<i>Clase para interactuar con el usuario registrado</i>	
Usuario::bibliotecaPerfil()	16
<i>Carga los juegos de la biblioteca del usuario</i>	
Usuario::datosPerfil()	16
<i>Carga los datos del usuario que tiene iniciada la sesion</i>	
usuario.class.php	6