

OpenConference

Generado por Doxygen 1.5.8

Thu Aug 13 19:00:28 2009

Índice general

1. Índice de clases	1
1.1. Lista de clases	1
2. Documentación de las clases	3
2.1. Referencia de la Clase Centinela	3
2.1.1. Descripción detallada	4
2.1.2. Documentación de las funciones miembro	4
2.1.2.1. Centinela	4
2.1.2.2. check	4
2.1.2.3. getClave	5
2.1.2.4. getId	5
2.1.2.5. getNick	5
2.1.2.6. getNivel	5
2.1.2.7. login	5
2.1.2.8. logout	6
2.2. Referencia de la Clase Noticias	7
2.2.1. Descripción detallada	7
2.2.2. Documentación de las funciones miembro	7
2.2.2.1. borrar	7
2.2.2.2. comentar	8
2.2.2.3. comentarios	8
2.2.2.4. datos_noticia	8
2.2.2.5. editar	8
2.2.2.6. index	8
2.2.2.7. modificar	8
2.2.2.8. Noticias	9
2.2.2.9. publicar	9
2.3. Referencia de la Clase Noticias_model	10

2.3.1.	Descripción detallada	10
2.3.2.	Documentación de las funciones miembro	10
2.3.2.1.	Borrar	10
2.3.2.2.	Comentar	11
2.3.2.3.	Comentarios	11
2.3.2.4.	Datos	11
2.3.2.5.	Editar	11
2.3.2.6.	Listado	12
2.3.2.7.	Noticias_model	12
2.3.2.8.	Publicar	12
2.4.	Referencia de la Clase Upload	13
2.4.1.	Descripción detallada	13
2.4.2.	Documentación de las funciones miembro	13
2.4.2.1.	cargar	13
2.4.2.2.	descargar	13
2.4.2.3.	index	13
2.4.2.4.	Upload	14
2.5.	Referencia de la Clase Upload_model	15
2.5.1.	Descripción detallada	15
2.5.2.	Documentación de las funciones miembro	15
2.5.2.1.	Cargar	15
2.5.2.2.	Descargar	15
2.5.2.3.	Listado_descargas	16
2.5.2.4.	Upload_model	16
2.6.	Referencia de la Clase Usuarios	17
2.6.1.	Descripción detallada	17
2.6.2.	Documentación de las funciones miembro	17
2.6.2.1.	index	17
2.6.2.2.	login	17
2.6.2.3.	logout	17
2.6.2.4.	registrar	18
2.6.2.5.	Usuarios	18
2.7.	Referencia de la Clase Usuarios_model	19
2.7.1.	Descripción detallada	19
2.7.2.	Documentación de las funciones miembro	19
2.7.2.1.	_check_user	19

2.7.2.2. Inicio	19
2.7.2.3. Login	20
2.7.2.4. Registro	20
2.7.2.5. Usuarios_model	20

Capítulo 1

Índice de clases

1.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Centinela (Utilizada para crear y destruir sesiones de usuario, y para restringir el acceso de los mismos)	3
Noticias (Utilizada para listar, publicar, editar y borrar noticias)	7
Noticias_model (Utilizada para recoger la información que utilizada el sistema de noticias) . . .	10
Upload (Utilizada para listar descargas, cargar archivos al servidor y descargarlos)	13
Upload_model (Utilizada para recoger la información que utilizada el sistema de Upload) . . .	15
Usuarios (Utilizada para registrar usuarios, loguearlos y de-loguearlos)	17
Usuarios_model (Utilizada para recoger la información que utilizada el sistema de usuarios) . .	19

Capítulo 2

Documentación de las clases

2.1. Referencia de la Clase Centinela

Utilizada para crear y destruir sesiones de usuario, y para restringir el acceso de los mismos.

Métodos públicos

- **Centinela** (\$auto=TRUE)
- **getId** ()
Obtiene el identificador de la sesión iniciada por el usuario.
- **getNick** ()
Obtiene el nick del usuario que ha iniciado la sesión.
- **getClave** ()
Obtiene el password del usuario que ha iniciado la sesión.
- **getNivel** ()
Obtiene el nivel de acceso que tiene el usuario que ha iniciado la sesión.
- **login** (\$nick="", \$clave="")
Permite que el usuario pueda iniciar sesión en el sistema.
- **logout** ()
Permite que el usuario cierre la sesión con total seguridad.
- **check** (\$nivel=0, \$estricto=TRUE)
Permite o restringe el acceso a ciertas zonas del sistema, dependiendo del nivel de acceso del usuario.

Atributos públicos

- **\$_id** = 0
- **\$_nick** = ""

- `$_clave = ""`
- `$_nivel = ""`
- `$_auth = FALSE`

2.1.1. Descripción detallada

Utilizada para crear y destruir sesiones de usuario, y para restringir el acceso de los mismos.

Mediante esta clase se llevan a cabo diversas operaciones que luego nos facilitan el trabajo a la hora de controlar el inicio de sesión del usuario, así como garantizar su correcta salida de la aplicación. Además esta clase dispone de funciones para restringir el acceso a ciertas zonas del sistema donde el usuario no pueda entrar. Se dispone de varios niveles de acceso: visitante, registrado, ponente, administrador.

2.1.2. Documentación de las funciones miembro

2.1.2.1. Centinela::Centinela (`$ auto = TRUE`)

Constructor

Se encarga de crear una instancia de [Centinela](#), a partir de la cuál se trabajará. Básicamente crea una sesión e inicializa las variables de sesión con los datos del usuario que accede al sistema, con lo cuál se crea una sesión específica para él que se mantendrá hasta que el usuario cierre la sesión o borre las cookies de su navegador.

Nota:

La clase session de Codeigniter trabaja en realidad con cookies.

2.1.2.2. Centinela::check (`$ nivel = 0, $ estricto = TRUE`)

Permite o restringe el acceso a ciertas zonas del sistema, dependiendo del nivel de acceso del usuario.

Esta función comprueba si el usuario tiene el suficiente nivel como para acceder a la zona solicitada. Como ejemplo se podrán considerar algunos niveles como:

- Nivel 0: El más básico, es el que tienen todos los usuarios que entren a la web.
- Nivel 1: Este nivel es para usuarios registrados.
- Nivel 3: Este nivel es para ponentes.
- Nivel 4 en adelante: Administradores.

Parámetros:

nivel Nivel de acceso que se debe tener para acceder a la zona solicitada.

estricto Si el valor es verdadero, únicamente podrán acceder los de ese nivel, y nadie más. En cambio si su valor es falso, podrán acceder también los niveles superiores.

Valores devueltos:

TRUE Si es posible acceder a la zona.

FALSE Si se restringe el acceso a la zona.

2.1.2.3. Centinela::getClave ()

Obtiene el password del usuario que ha iniciado la sesión.

Devuelve:

Password del usuario.

2.1.2.4. Centinela::getId ()

Obtiene el identificador de la sesión iniciada por el usuario.

Devuelve:

Identificador de sesión.

2.1.2.5. Centinela::getNick ()

Obtiene el nick del usuario que ha iniciado la sesión.

Devuelve:

Nick del usuario.

2.1.2.6. Centinela::getNivel ()

Obtiene el nivel de acceso que tiene el usuario que ha iniciado la sesión.

Devuelve:

Nivel de acceso al sistema que tiene el usuario.

2.1.2.7. Centinela::login (\$ nick = "", \$ clave = "")

Permite que el usuario pueda iniciar sesión en el sistema.

Esta función realiza todo el trabajo de autenticación del usuario. Comprueba que el usuario ya está registrado y posteriormente crea las variables de sesión correspondientes con los datos del usuario que inicia sesión.

Parámetros:

nick Nick del usuario que accede al sistema.

clave Password del usuario que accede al sistema.

Valores devueltos:

TRUE Si ha sido posible iniciar sesión.

FALSE Si no ha sido posible iniciar sesión.

2.1.2.8. Centinela::logout ()

Permite que el usuario cierre la sesión con total seguridad.

Esta función simplemente destruye las variables de sesión del usuario y lo saca del sistema de manera segura.

La documentación para esta clase fue generada a partir del siguiente fichero:

- system/application/libraries/Centinela.php

2.2. Referencia de la Clase Noticias

Utilizada para listar, publicar, editar y borrar noticias.

Métodos públicos

- [Noticias \(\)](#)
- [index \(\)](#)
Llama a la función Listado() para mostrar todas las noticias publicadas.
- [editar \(\\$id\)](#)
Recupera datos de una noticia y los modifica.
- [publicar \(\)](#)
Muestra un formulario para escribir noticias.
- [datos_noticia \(\)](#)
Publica noticias en el sistema.
- [modificar \(\\$id\)](#)
Edita una noticia ya publicada.
- [borrar \(\\$id\)](#)
Borra una noticia ya publicada.
- [comentarios \(\\$id\)](#)
Muestra los comentarios que se han hecho sobre una noticia.
- [comentar \(\)](#)
Publica un comentario en una noticia del sistema existente.

2.2.1. Descripción detallada

Utilizada para listar, publicar, editar y borrar noticias.

Esta clase dispone de funciones para listar las noticias (en grupos de 4), para publicar noticias, así como para editarlas y borrarlas. Estas tres últimas funcionalidades sólo están disponibles para el administrador del sistema.

2.2.2. Documentación de las funciones miembro

2.2.2.1. Noticias::borrar (\$ id)

Borra una noticia ya publicada.

Llama a la función Borrar() del modelo, pasándole el identificador de la noticia que se quiere borrar.

Parámetros:

id Identificador de la noticia a borrar

2.2.2.2. Noticias::comentar ()

Publica un comentario en una noticia del sistema existente.

Llama a la función Comentar() del modelo, pasándole los datos del formulario de envío de comentarios, la cuál deja el comentario sobre la noticia elegida.

2.2.2.3. Noticias::comentarios (\$ id)

Muestra los comentarios que se han hecho sobre una noticia.

Recopila los comentarios que se han dejado en una determinada noticia, agrupándolos y mostrándolos en cada noticia. También pone a disposición el formulario para escribir el comentario. Al menos hay que ser usuario registrado para dejar un comentario.

Parámetros:

id Identificador de la noticia de la que se deben mostrar los comentarios

2.2.2.4. Noticias::datos_noticia ()

Publica noticias en el sistema.

Llama a la función Publicar(), que crea la noticia en el sistema, pasándole los datos recibidos del formulario.

2.2.2.5. Noticias::editar (\$ id)

Recupera datos de una noticia y los modifica.

Esta función es la encargada de poner a disposición del administrador los datos de la noticia en cuestión, para así poder modificarlos y reenviarlos.

Parámetros:

id Identificador de la noticia a editar.

2.2.2.6. Noticias::index ()

Llama a la función Listado() para mostrar todas las noticias publicadas.

Es la que se ejecuta por defecto al entrar en el apartado de noticias, y se encarga de llamar a la función Listado() del modelo para ir mostrando en grupos de 4 todas las noticias publicadas hasta la fecha.

2.2.2.7. Noticias::modificar (\$ id)

Edita una noticia ya publicada.

Llama a la función Editar(), pasándole el identificador de la noticia a editar, y todos los datos de la noticia que van a ser modificados.

Parámetros:

id Identificador de la noticia a editar

2.2.2.8. Noticias::Noticias ()

Constructor

Se encarga de crear una instancia de [Noticias](#), y llama automáticamente al modelo de noticias.

2.2.2.9. Noticias::publicar ()

Muestra un formulario para escribir noticias.

En el formulario, el administrador escribe el artículo o noticia a publicar.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `system/application/controllers/noticias.php`

2.3. Referencia de la Clase Noticias_model

Utilizada para recoger la información que utilizada el sistema de noticias.

Métodos públicos

- [Noticias_model \(\)](#)
- [Listado \(\)](#)
Recopila todas las noticias publicadas en el sistema y las muestra.
- [Datos \(\\$id\)](#)
Recupera datos que cumplan cierta condición.
- [Editar \(\\$id, \\$datos\)](#)
Modifica los datos de una noticia ya publicada.
- [Publicar \(\\$datos\)](#)
Publica una noticia en el sistema.
- [Borrar \(\\$id\)](#)
Borra una noticia publicada en el sistema.
- [Comentarios \(\\$id\)](#)
Muestra los comentarios que se han hecho sobre una noticia.
- [Comentar \(\\$datos\)](#)
Publica un comentario en una determinada noticia (elegida por el usuario).

2.3.1. Descripción detallada

Utilizada para recoger la información que utilizada el sistema de noticias.

Mediante esta clase se llevan a cabo diversas operaciones para buscar y recuperar información de la base de datos y mandársela al controlador, el cuál ya hará las operaciones oportunas.

2.3.2. Documentación de las funciones miembro

2.3.2.1. Noticias_model::Borrar (\$ id)

Borra una noticia publicada en el sistema.

Esta función recibe el identificador de la noticia que se quiere eliminar, y posteriormente la busca en la base de datos y la elimina de la misma.

Parámetros:

id Identificador de la noticia a borrar.

2.3.2.2. Noticias_model::Comentar (\$ datos)

Publica un comentario en una determinada noticia (elegida por el usuario).

La función recibe un array con todos los datos del comentario (nick del autor, e-mail, comentario...) y posteriormente los inserta en la base de datos, teniendo como clave foránea el identificador de la noticia comentada.

Parámetros:

datos Es un array que contiene los datos del comentario a publicar.

2.3.2.3. Noticias_model::Comentarios (\$ id)

Muestra los comentarios que se han hecho sobre una noticia.

La función recibe el identificador de la noticia, y busca en la base de datos todos los comentarios que se han hecho sobre dicha noticia para mostrárselos al usuario.

Parámetros:

id Identificador de la noticia cuyos comentarios se desean ver.

Devuelve:

Array con los comentarios hechos sobre una noticia.

2.3.2.4. Noticias_model::Datos (\$ id)

Recupera datos que cumplan cierta condición.

La función recibe el identificador de la noticia seleccionada, y devuelve todos los datos de la misma: el autor, el título, la categoría, etc...

Parámetros:

id Identificador de la noticia de la que se quieren recuperar datos.

Devuelve:

Información de una determinada noticia.

2.3.2.5. Noticias_model::Editar (\$ id, \$ datos)

Modifica los datos de una noticia ya publicada.

Esta función recibe el identificador de la noticia a editar, así como los nuevos datos a publicar, y actualiza la información de la base de datos.

Parámetros:

id Identificador de la noticia a editar.

datos Es un array con los datos ya modificados.

2.3.2.6. Noticias_model::Listado ()

Recopila todas las noticias publicadas en el sistema y las muestra.

La función se encarga de extraer de la base de datos todas las noticias que se hayan publicado y paginarlas en grupos de 3, por cuestión de estilo y navegabilidad.

Devuelve:

Array con todas las noticias publicadas en el sistema.

2.3.2.7. Noticias_model::Noticias_model ()

Constructor

Crea una instancia de [Noticias_model](#), a partir de la cuál se trabajará.

2.3.2.8. Noticias_model::Publicar (\$ datos)

Publica una noticia en el sistema.

Esta función recibe los datos a partir del formulario de noticias, y los inserta en la base de datos de noticias.

Parámetros:

datos Es un array que contiene todos los datos de la noticia a publicar (autor, título, categoría, etc...).

La documentación para esta clase fue generada a partir del siguiente fichero:

- `system/application/models/noticias_model.php`

2.4. Referencia de la Clase Upload

Utilizada para listar descargas, cargar archivos al servidor y descargarlos.

Métodos públicos

- [Upload \(\)](#)
- [index \(\)](#)

LLama a la función Listado_descargas() para mostrar todos los archivos a descargar.

- [descargar \(\\$id_fichero\)](#)

Permite descargar el archivo seleccionado directamente al pc, o abrirlo en el navegador.

- [cargar \(\)](#)

Carga archivos desde el pc al servidor.

2.4.1. Descripción detallada

Utilizada para listar descargas, cargar archivos al servidor y descargarlos.

Esta clase dispone de funciones para listar todo los archivos disponibles para descargar, también dispone de una función que permite descargar el archivo a tu ordenador o abrirlo en el navegador, así como una función para cargar archivos desde el pc al servidor.

2.4.2. Documentación de las funciones miembro

2.4.2.1. Upload::cargar ()

Carga archivos desde el pc al servidor.

Llama a la función Cargar() del modelo, pasándole los datos del formulario de subida de ficheros. La función almacenará el archivo en el servidor, junto con otros datos, y pondrá un enlace en la sección correspondiente para poder descargarlo.

2.4.2.2. Upload::descargar (\$ id_fichero)

Permite descargar el archivo seleccionado directamente al pc, o abrirlo en el navegador.

Llama a la función Descargar() del modelo, pasándole el identificador del archivo que se quiere descargar.

Parámetros:

id_fichero Identificador del fichero a descargar.

2.4.2.3. Upload::index ()

LLama a la función Listado_descargas() para mostrar todos los archivos a descargar.

Es la que se ejecuta por defecto al entrar en el apartado de descargas, y se encarga de llamar a la función Listado_descargas() del modelo para mostrar todos los archivos que se pueden descargar desde la página actualmente.

2.4.2.4. Upload::Upload ()

Constructor

Se encarga de crear una instancia de [Upload](#), y llama automáticamente al modelo de upload.

La documentación para esta clase fue generada a partir del siguiente fichero:

- system/application/controllers/upload.php

2.5. Referencia de la Clase Upload_model

Utilizada para recoger la información que utilizada el sistema de [Upload](#).

Métodos públicos

- [Upload_model \(\)](#)
- [Listado_descargas \(\)](#)
Recopila todas los archivos disponibles para descargar y los muestra.
- [Descargar \(\\$id_fichero\)](#)
Permite descargar archivos desde el servidor al pc del usuario.
- [Cargar \(\\$fichero\)](#)
Muestra un formulario para que el usuario suba archivos al servidor.

2.5.1. Descripción detallada

Utilizada para recoger la información que utilizada el sistema de [Upload](#).

Mediante esta clase se llevan a cabo diversas operaciones para buscar y recuperar información de la base de datos y mandársela al controlador, el cuál ya hará las operaciones oportunas.

2.5.2. Documentación de las funciones miembro

2.5.2.1. Upload_model::Cargar (\$fichero)

Muestra un formulario para que el usuario suba archivos al servidor.

La función recibe un array que contiene datos sobre el fichero a descargar (tamaño, tipo, nombre, etc..) y después de comprobar que el fichero cumple los requisitos, se realiza la consulta para insertar toda la información en la base de datos de archivos.

Parámetros:

fichero Es un array que contiene información sobre el fichero (tamaño, nombre, tipo, etc...).

2.5.2.2. Upload_model::Descargar (\$id_fichero)

Permite descargar archivos desde el servidor al pc del usuario.

La función recibe el identificador del fichero que se quiere descargar, comprueba si es posible descargar dicho fichero, y posteriormente le da la posibilidad al usuario de abrirlo desde el navegador o guardarlo en su pc.

Parámetros:

id_fichero Identificador del fichero que se va a descargar.

2.5.2.3. Upload_model::Listado_descargas ()

Recopila todas los archivos disponibles para descargar y los muestra.

La función se encarga de extraer de la base de datos todos los archivos que se hayan subido al servidor y que puedan ser descargados, y se los muestra al usuario.

Devuelve:

Array con todas los archivos descargables.

2.5.2.4. Upload_model::Upload_model ()

Constructor

Crea una instancia de [Upload_model](#), a partir de la cuál se trabajará.

La documentación para esta clase fue generada a partir del siguiente fichero:

- system/application/models/upload_model.php

2.6. Referencia de la Clase Usuarios

Utilizada para registrar usuarios, loguearlos y de-loguearlos.

Métodos públicos

- [Usuarios \(\)](#)
- [index \(\)](#)

Llama a la función Inicio() del modelo para mostrar la página principal.

- [registrar \(\)](#)

Permite al usuario registrarse en el sistema.

- [login \(\)](#)

Permite al usuario acceder a su cuenta del sistema.

- [logout \(\)](#)

Permite al usuario cerrar sesión en el sistema con total seguridad.

2.6.1. Descripción detallada

Utilizada para registrar usuarios, loguearlos y de-loguearlos.

Esta clase dispone de funciones para registrar nuevos usuarios mediante un formulario de ingreso, también posee funciones para que los usuarios registrados accedan con su cuenta al sistema, así como para que salgan del sistema sin fallos.

2.6.2. Documentación de las funciones miembro

2.6.2.1. Usuarios::index ()

Llama a la función Inicio() del modelo para mostrar la página principal.

Es la que se ejecuta por defecto al entrar en el sistema, y se encarga de llamar a la función Inicio() del modelo que muestra un menú principal en el que el usuario podrá acceder con su cuenta si es un usuario registrado, o si aún no lo es, registrarse.

2.6.2.2. Usuarios::login ()

Permite al usuario acceder a su cuenta del sistema.

Llama a la función Login() del modelo, que pone a disposición del usuario un formulario para introducir los datos de su cuenta y acceder al sistema y a las zonas que requieren que el usuario esté registrado para poder verlas.

2.6.2.3. Usuarios::logout ()

Permite al usuario cerrar sesión en el sistema con total seguridad.

Esta función destruye la sesión iniciada por el usuario, y limpia todas las variables del registro para asegurar una correcta salida del sistema.

2.6.2.4. Usuarios::registrar ()

Permite al usuario registrarse en el sistema.

Llama a la función Registro() de modelo, donde se muestra el formulario de registro mediante el cuál el usuario podrá registrarse dando un nick y una clave (además de otros datos).

2.6.2.5. Usuarios::Usuarios ()

Constructor

Se encarga de crear una instancia de [Usuarios](#), y llama automáticamente al modelo de usuarios.

La documentación para esta clase fue generada a partir del siguiente fichero:

- system/application/controllers/usuarios.php

2.7. Referencia de la Clase Usuarios_model

Utilizada para recoger la información que utilizada el sistema de usuarios.

Métodos públicos

- [Usuarios_model \(\)](#)
- [Inicio \(\)](#)
Recopila todos los usuarios registrados en el sistema y los muestra (para administradores).
- [Registro \(\)](#)
Permite al usuario registrarse en el sistema.
- [_check_user \(\\$nick\)](#)
Comprueba si el usuario ya existe.
- [Login \(\)](#)
Permite al usuario loguearse en el sistema.

2.7.1. Descripción detallada

Utilizada para recoger la información que utilizada el sistema de usuarios.

Mediante esta clase se llevan a cabo diversas operaciones para buscar y recuperar información de la base de datos y mandársela al controlador, el cuál ya hará las operaciones oportunas.

2.7.2. Documentación de las funciones miembro

2.7.2.1. Usuarios_model::_check_user (\$ nick)

Comprueba si el usuario ya existe.

La función recibe el nick del usuario que se desea registrar y comprueba si éste ya está utilizado.

Parámetros:

nick Sobrenombre que ha escogido el usuario que se registra.

Devuelve:

Valor booleano que depende de si el nick ya está o no ocupado por otro usuario.

2.7.2.2. Usuarios_model::Inicio ()

Recopila todos los usuarios registrados en el sistema y los muestra (para administradores).

La función se encarga de extraer de la base de datos todos los usuarios que se hayan registrado en el sistema y se los muestra al administrador.

Devuelve:

Array con todos los usuarios registrados.

2.7.2.3. Usuarios_model::Login ()

Permite al usuario loguearse en el sistema.

La función muestra al usuario un formulario de acceso, comprueba que los datos recibidos son correctos y también que el usuario esté registrado en el sistema, y posteriormente crea una sesión específica para el usuario.

Devuelve:

FALSE si hay error de validación o es la primera vez que se entra.

2.7.2.4. Usuarios_model::Registro ()

Permite al usuario registrarse en el sistema.

Esta función muestra un formulario de registro, comprueba que sus datos son válidos y posteriormente los inserta en la base de datos, dando de alta al usuario en el sistema.

Devuelve:

FALSE si los datos no validan.

2.7.2.5. Usuarios_model::Usuarios_model ()

Constructor

Crea una instancia de [Usuarios_model](#), a partir de la cuál se trabajará.

La documentación para esta clase fue generada a partir del siguiente fichero:

- system/application/models/usuarios_model.php

Índice alfabético

- [_check_user](#)
 - [Usuarios_model, 19](#)
- [Borrar](#)
 - [Noticias_model, 10](#)
- [borrar](#)
 - [Noticias, 7](#)
- [Cargar](#)
 - [Upload_model, 15](#)
- [cargar](#)
 - [Upload, 13](#)
- [Centinela, 3](#)
 - [Centinela, 4](#)
 - [check, 4](#)
 - [getClave, 4](#)
 - [getId, 5](#)
 - [getNick, 5](#)
 - [getNivel, 5](#)
 - [login, 5](#)
 - [logout, 5](#)
- [check](#)
 - [Centinela, 4](#)
- [Comentar](#)
 - [Noticias_model, 10](#)
- [comentar](#)
 - [Noticias, 7](#)
- [Comentarios](#)
 - [Noticias_model, 11](#)
- [comentarios](#)
 - [Noticias, 8](#)
- [Datos](#)
 - [Noticias_model, 11](#)
- [datos_noticia](#)
 - [Noticias, 8](#)
- [Descargar](#)
 - [Upload_model, 15](#)
- [descargar](#)
 - [Upload, 13](#)
- [Editar](#)
 - [Noticias_model, 11](#)
- [editar](#)
 - [Noticias, 8](#)
- [getClave](#)
 - [Centinela, 4](#)
- [getId](#)
 - [Centinela, 5](#)
- [getNick](#)
 - [Centinela, 5](#)
- [getNivel](#)
 - [Centinela, 5](#)
- [index](#)
 - [Noticias, 8](#)
 - [Upload, 13](#)
 - [Usuarios, 17](#)
- [Inicio](#)
 - [Usuarios_model, 19](#)
- [Listado](#)
 - [Noticias_model, 11](#)
- [Listado_descargas](#)
 - [Upload_model, 15](#)
- [Login](#)
 - [Usuarios_model, 19](#)
- [login](#)
 - [Centinela, 5](#)
 - [Usuarios, 17](#)
- [logout](#)
 - [Centinela, 5](#)
 - [Usuarios, 17](#)
- [modificar](#)
 - [Noticias, 8](#)
- [Noticias, 7](#)
 - [borrar, 7](#)
 - [comentar, 7](#)
 - [comentarios, 8](#)
 - [datos_noticia, 8](#)
 - [editar, 8](#)
 - [index, 8](#)
 - [modificar, 8](#)
 - [Noticias, 8](#)
 - [publicar, 9](#)
- [Noticias_model, 10](#)
 - [Borrar, 10](#)
 - [Comentar, 10](#)

- Comentarios, [11](#)
- Datos, [11](#)
- Editar, [11](#)
- Listado, [11](#)
- Noticias_model, [12](#)
- Noticias_model, [12](#)
- Publicar, [12](#)

- Publicar
 - Noticias_model, [12](#)
- publicar
 - Noticias, [9](#)

- registrar
 - Usuarios, [18](#)
- Registro
 - Usuarios_model, [20](#)

- Upload, [13](#)
 - cargar, [13](#)
 - descargar, [13](#)
 - index, [13](#)
 - Upload, [13](#)
- Upload_model, [15](#)
 - Cargar, [15](#)
 - Descargar, [15](#)
 - Listado_descargas, [15](#)
 - Upload_model, [16](#)
 - Upload_model, [16](#)
- Usuarios, [17](#)
 - index, [17](#)
 - login, [17](#)
 - logout, [17](#)
 - registrar, [18](#)
 - Usuarios, [18](#)
- Usuarios_model, [19](#)
 - _check_user, [19](#)
 - Inicio, [19](#)
 - Login, [19](#)
 - Registro, [20](#)
 - Usuarios_model, [20](#)
 - Usuarios_model, [20](#)