Documentacion phpKROND

Contents

Package phpkrond.modulos.calendario Procedural Elements

calendario.inc.php

* **Package** phpkrond.modulos.calendario

_MOD_CALENDARIO_CLASS = 1 *[line 31]*

Package phpkrond.modulos.calendario
Classes

Class calendario_admin

[line 918]

Summary: Clase que gestiona las administración del Calendario.
La administración del calendario controla los eventos y las citas del mismo.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor calendario_admin::calendario_admin([$operacion = ""]) *[line 926]*

Function Parameters:

* *$operacion* **$operacion** String con la operación a realizar.

Summary: Contructor de la clase.
Ejecuta la operación especificada en el parámetro.

Array calendario_admin::calcula_array_graficos() *[line 1516]*

Summary: Devuelve un array con los graficos disponibles

String calendario_admin::calcula_select_evento([$idEvento = 0]) *[line 1539]*

Function Parameters:

* *$idTema* **$idEvento** Identificador del evento seleccionado por defecto.

Summary: Calcula el campo select de un formulario con los Eventos.
Los eventos se encuentran en la tabla 'calendario_eventos'.

String calendario_admin::calcula_select_graficos([$grafico = ""]) *[line 1496]*

Function Parameters:

* *$grafico* **$grafico** Grafico seleccionado por defecto.

Summary: Calcula el campo select de un formulario con los graficos disponibles.
Los graficos se encuentran todos en 'modulos/calendario/graficos/'.

String calendario_admin::form_cabecera() *[line 1052]*

Summary: Cabecera del Módulo de Administración de Calendario
Muestra una cabecera con el título de Administración del Módulo de Calendario.

String calendario_admin::form_cita_nueva([$idEvento = 0]) *[line 1240]*
Summary: Muestra el formulario para crear una nueva cita.

String calendario_admin::form_cita_propiedades($idCita) *[line 1341]*
Function Parameters:

* *$idCita* **$idCita** Identificador de la cita.

Summary: Muestra el formulario para cambiar propiedades de una cita.

String calendario_admin::form_config() *[line 1449]*
Summary: Muestra el formulario de configuración del Módulo.
Desde este formulario se pueden cambiar las variables de configuración 'nombre_pagina_mostrar'. Estas variables se guardan en la tabla 'calendario_config'.

String calendario_admin::form_evento_nuevo() *[line 1085]*
Summary: Muestra el formulario para crear un nuevo evento.
Informa de los eventos disponibles y permite hacer las operacion de administración de los mismos.

String calendario_admin::form_evento_propiedades($idEvento) *[line 1158]*
Function Parameters:

* *$idEvento* **$idEvento** Identificador de evento.

Summary: Muestra el formulario para cambiar propiedades de un evento.
Lista tambien las citas que tiene el evento.

String calendario_admin::form_general([$fecha = ""]) *[line 1066]*
Summary: Muestra el formulario general de la administración del calendario.
Este formulario muestra los tipos de eventos que clasifican las citas en el calendario

String calendario_admin::form_grafico_nuevo() *[line 1404]*
Summary: Muestra el formulario para añadir un nuevo grafico de eventos.
Lista los graficos disponibles en el directorio 'modulos/calendario/graficos/'.

Class calendario_bloque

[line 809]

Summary: Clase que muestra un pequeño bloque de calendario
El bloque muestra el calenario correspondiente al mes actual.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor calendario_bloque::calendario_bloque() *[line 818]*

Summary: Constructor de la clase para mostrar el bloque pequeño del calendario.
Devuelve un bloque con un calendario pequeño. El mes que toma del calendario es de la fecha actual del sistema.

Class calendario_cita

[line 312]

Summary: Clase que gestiona las Citas en el calendario.

Las citas son las fechas del calendario a recordar. La tabla que recoge esta información en la base de datos se llama 'calendario_citas'.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed calendario_cita::$db = *[line 332]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed calendario_cita::$idCita = *[line 321]*

Summary: Atributo de la clase que guarda la clave de la Cita.

El atributo de la clase $idCita es el identificador utilizado como clave primaria en la tabla 'calendario_citas'.

Constructor calendario_cita::calendario_cita([$idCita = 0]) *[line 343]*

Function Parameters:

* *$idCita* **$idCita** Identificador de la Cita.

Summary: Contructor de la clase.

Inicializa el atributo $idCita de la clase. Este atributo se utiliza como clave primaria en la tabla 'calendario_citas'.

boolean calendario_cita::borrar() *[line 574]*

Summary: Borra una cita del calendario

Elimina la entrada de la cita que se encuentra en la tabla 'calendario_citas'.

boolean calendario_cita::check_datos($datos) *[line 544]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos
En principio solo comprueba que los datos son distintos de "". Util para comprobar la fecha
* la descripción de una cita.

boolean calendario_cita::check_fecha($fecha) *[line 557]*

Function Parameters:

* *$fecha* **$fecha** String que la fecha a comprobar.

Summary: Verifica que la fecha es correcta.
Comprueba que la fecha dada es correcta. La fecha se pasa en formato 'AAAA-MM-DD'.

boolean calendario_cita::crear($fecha, $nombre, $descripcion, $idEvento) *[line 520]*

Function Parameters:

* *$fecha* **$fecha** Fecha de la nueva cita.
* *$nombre* **$nombre** Nombre de la cita.
* *$descripcion* **$descripcion** Descripción de la nueva cita.
* *$idEvento* **$idEvento** Tipo de evento de la cita.

Summary: Crea una nueva cita en el calendario.
Crea una nueva entrada en la tabla 'calendario_citas' inicializando todos los campos con los datos de la nueva cita.

boolean calendario_cita::existe() *[line 497]*

Summary: Verifica si la cita existe.
Comprueba que el identificador de la cita es correcto.

Array calendario_cita::get_all() *[line 437]*

Summary: Devuelve las citas del calendario.
Los citas estan ordenadas por fecha en orden decreciente.

Array calendario_cita::get_citas_mes([$fecha = ""]) *[line 460]*

Function Parameters:

* *$fecha* **$fecha** Fecha en la que mirar las citas. De esa fecha se toma el mes y el año para consultarlo con el sistema.

Summary: Devuelve un array con los dias del mes y las citas en ellos.
El array esta indexado por dias y si tiene citas en un dia particular, contiene un array de objetos 'calendario_cita' con las citas que se dan en ese dia.

String calendario_cita::get_descripcion() *[line 404]*

Summary: Devuelve la descripción de la cita.
La descripción contiene toda la información sobre la cita. Se almacena en el campo 'descripcion' de la tabla 'calendario_citas'.

calendario_evento calendario_cita::get_evento() *[line 420]*

Summary: Devuelve el tipo de Evento de la cita.
Las citas se clasificas en tipos de eventos.

String calendario_cita::get_fecha() *[line 368]*

Summary: Devuelve la fecha de la cita.
La fecha esta en formato 'AAAA-MM-DD'.

Integer calendario_cita::get_id() *[line 357]*

Summary: Devuelve el identificador de la cita.
El identificador de la cita se guarda en la variable $this->idCita de la clase. Se utiliza como clave primaria en la tabla 'calendario_citas'.

String calendario_cita::get_nombre() *[line 386]*

Summary: Devuelve el nombre de la cita.
La nombre de la cita es una referencia corta de la cita. Se almacena en el campo 'nombre_cita' de la tabla 'calendario_citas'.

Boolean calendario_cita::set($fecha, $nombre, $descripcion, $idEvento) *[line 597]*

Function Parameters:

* *$fecha* **$fecha** Nueva fecha de la cita.
* *$nombre* **$nombre** Nuevo nombre de la cita.
* *$descripcion* **$descripcion** Nueva descripción de la cita.
* *$idEvento* **$idEvento** Nuevo tipo de evento de la cita.

Summary: Modifica las propiedades de una cita.
Modifica la descripción, fecha y el tipo de evento asociado a la misma. Comprueba que los nuevos datos son correctos.

Class calendario_config

[line 1568]

Summary: Clase que gestiona las variables de configuración del módulo.
Las variables de configuración se encuentran almacenadas en la tabla 'calendario_config'.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed calendario_config::$db = *[line 1578]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor calendario_config::calendario_config() *[line 1587]*

Summary: Constructor de la clase.
Inicializa el atributo $db que contiene el objeto de la conexino a la base de datos. Este objeto es de tipo 'newADOconnection'.

String calendario_config::get_variable($nombreVar) *[line 1601]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Recupera el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'calendario_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

calendario_config::set_variable($nombreVar, $nuevoValor) *[line 1620]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable a modificar.
* *$nuevoValor* **$nuevoValor** Nuevo valor de la variable.

Summary: Modifica el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'calendario_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

Class calendario_error

[line 1636]

Summary: Clase para mostrar mensajes del módulo calendario.
Todos los mensajes del módulo se consideran mensajes de error.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor calendario_error::calendario_error([$tituloError = "Calendario - Error "], [$msgError = ""]) *[line 1645]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class calendario_evento

[line 45]

Summary: Clase que gestiona los tipos de Eventos del calendario.

El Calendario se compone de tipos de Eventos, que son una forma de clasificar las Citas (fechas importantes) La tabla que recoge esta información en la base de datos se llama 'calendario_eventos'.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed calendario_evento::$db = *[line 65]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed calendario_evento::$idEvento = *[line 54]*

Summary: Atributo de la clase que guarda la clave del tipo de Evento.

El atributo de la clase $idEvento es el identificador utilizado como clave primaria en la tabla 'calendario_eventos'.

Constructor calendario_evento::calendario_evento([$idEvento = 0]) *[line 77]*

Function Parameters:

* *$idEvento* **$idEvento** Identificador del Tipo de Evento.

Summary: Contructor de la clase.

Inicializa el identificador del tipo de Evento. Este identificador es utilizado como clave primaria en la tabla 'calendario_eventos' que contiene la información y descripción del tipo de evento.

boolean calendario_evento::borrar($vaciar) *[line 243]*

Function Parameters:

* *$vaciar* **$vaciar** Si true entonces sólo borra las citas. En caso contrario, comprueba que el evento esta vacio (no tiene citas) y borra el evento.

Summary: Borra un evneto del calendario.

Elimina la entrada del evento que se encuentra en la tabla 'calendario_eventos' siempre y cuando el evento este vacio.

boolean calendario_evento::check_datos($datos) *[line 226]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el nombre de un evento o de un grafico.

boolean calendario_evento::crear($nombre, $grafico) *[line 203]*

Function Parameters:

* *$nombre* **$nombre** Nombre del Evento.
* *$grafico* **$grafico** Grafico del Evento.

Summary: Crea un nuevo tipo de evento en el calendario.

Crea una nueva entrada en la tabla 'calendario_eventos' inicializando todos los campos con los datos del nuevo tema.

boolean calendario_evento::existe() *[line 182]*

Summary: Verifica si el enveto existe.

Comprueba que el identificador de evento es correcto.

Array calendario_evento::get_all() *[line 163]*

Summary: Devuelve los Eventos del calendario.

Devuelve un array de objetos 'calendario_evento'.

Array calendario_evento::get_citas([$comienzo = 0]) *[line 144]*

Function Parameters:

* *$comienzo* **$comienzo** Cita por la que comenzar a listar. Se utiliza para paginar las citas.

Summary: Devuelve las citas de un evento.

Un evento (tipo de evento) contiene diversos diversas citas en distintas fechas. Con esto se ofrece una forma de mantener un orden lógico en las citas que se pueden ver en el módulo. Devuelve un array de objetos calendario_cita ordenados por fecha en orden decreciente.

String calendario_evento::get_grafico() *[line 120]*

Summary: Devuelve el nombre del grafico asociado al tipo de evento.

El nombre del fichero que contiene el icono se encuentra en el campo 'grafico_evento' de la tabla 'calendario_eventos'.

Integer calendario_evento::get_id() *[line 91]*

Summary: Devuelve el identificador del evento.

El identificador del evento se guarda en la variable de la clase $this->idEvento y es el que se utiliza como clave primaria en la tabla 'calendario_eventos'.

String calendario_evento::get_nombre() *[line 103]*

Summary: Devuelve el nombre del evento.

El nombre del evento se encuentra en el campo 'nombre' de la tabla 'calendario_eventos'.

Boolean calendario_evento::set($nombre, $grafico) *[line 288]*

Function Parameters:

* *$nombre* **$nombre** Nuevo nombre del tema.
* *$grafico* **$grafico** Nuevo grafico del tema.

Summary: Modifica las propiedades del evento.

Modifica el nombre del tema y el grafico asociado. El grafico asociado no se borra. Comprueba que los nuevos datos son correctos.

Class calendario_mostrar

[line 622]

Summary: Clase que muestra el calendario
Muestra el mes seleccionado y las citas marcadas en el mismo.

* **Package** phpkrond.modulos.calendario
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor calendario_mostrar::calendario_mostrar([$operacion = ""]) *[line 630]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase para mostrar el calendario.
Realiza la operación indicada por el parametro.

String calendario_mostrar::mostrar([$fecha = ""]) *[line 654]*

Function Parameters:

* *$fecha* **$fecha** Fecha a mostrar el calendario

Summary: Formulario del calendario.
Muestra un cuadro con el mes seleccionado en la fecha y las citas disponibles para ese mes.

Package phpkrond.modulos.contenidos Procedural Elements

contenidos.inc.php

* **Package** phpkrond.modulos.contenidos

_MOD_CONTENIDOS_CLASS = 1 *[line 31]*

Package phpkrond.modulos.contenidos
Classes

Class contenidos_admin

[line 1115]

Summary: Clase para administrar el módulo de contenidos.
Se utiliza para crear las secciones y articulos que ofrece el módulo. Tambien permite llevar un poco de gestión sobre los miemos, modificandolos
* borrandolos.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor contenidos_admin::contenidos_admin([$operacion = ""]) *[line 1123]*

Function Parameters:

* *$operacion* **$operacion** String con la operación a realizar.

Summary: Contructor de la clase.
Ejecuta la operación especificada en el parámetro.

String contenidos_admin::form_articulo_crear($idSeccion) *[line 1595]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion Padre

Summary: Muestra el formulario para crear un nuevo articulo en la seccion.
Recoge los datos del articulo, titulo y contenidos, ademas de la referencia a la seccion que contendre el articulo.

String contenidos_admin::form_articulo_propiedades($idArticulo) *[line 1648]*

Function Parameters:

* *$idArticulo* **$idArticulo** Identificador del Articulo.

Summary: Muestra el formulario con las propiedades de un articulo.
Permite editar el titulo y el contenido de un articulo.

String contenidos_admin::form_cabecera() *[line 1229]*

Summary: Cabecera del Módulo de Administración de Contenidos

Muestra una cabecera con el título de Administración del Módulo de Contenidos.

String contenidos_admin::form_config() *[line 1409]*

Summary: Muestra formuario de configuración del módulo.

Permite modificar algunas variables de configuración que controlan el comportamiento de las clases del modulo. 'nombre_pagina_mostrar', destina a albergar el nombre de la pagina que contiene la llamada al objeto que muesta los contenidos.

String contenidos_admin::form_general([$idSeccion = 0]) *[line 1246]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion

Summary: Muestra el formulario general de la administración de contenidos.

Este formulario muestra el contenido de una seccion de los contenidos (subcategorias y articulos), un subformulario para cambiar propiedades de la categoria, añadir categorias y añadir articulos.

String contenidos_admin::form_seccion_borrar($idSeccion) *[line 1553]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion

Summary: Muestra el formulario para borrar una seccion.

Pregunta por el metodo de borrado. Si es recursivo elimina las secciones y articulos de la seccion. Si no es recuersivo solo eliminara la seccion si esta vacia (no tiene subsecciones ni articulos).

String contenidos_admin::form_seccion_crear($idSeccion) *[line 1506]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion Padre

Summary: Muestra el formulario para añadir un nueva subSeccion a la seccion actual

Este formulario permitira crear secciones en el sistema.

String contenidos_admin::form_seccion_propiedades($idSeccion) *[line 1464]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion

Summary: Muestra el formulario con las propiedades de la seccion.
Este formulario muestra las propiedades y permite modificarlas.

contenidos_admin::select_secciones([$secInicio = 0], [$secMarcar = 0], [$separacion = "   "]) *[line 1731]*

Class contenidos_articulo

[line 459]

Summary: Clase que gestiona los articulos de los contenidos.

Las articulos de los contenidos se clasifican en diferentes seccionespara permitir una organización. Cada articulo tiene un titulo, contenido del articulo propiamente dicho fecha de creación y la referencia a la sección a la que pertenece.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed contenidos_articulo::$db = *[line 479]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed contenidos_articulo::$idArticulo = *[line 468]*

Summary: Atributo de la clase que guarda la clave del articulo.

El atributo de la clase $idArticulo es el identificador utilizado como clave primaria en la tabla 'contenidos_articulos'.

Constructor contenidos_articulo::contenidos_articulo([$idArticulo = 0]) *[line 490]*

Function Parameters:

* *$idArticulo* **$idArticulo** Identificador del articulo.

Summary: Constructor de la clase que inicializa el atributo $idArticulo.

El atributo de la clase $idArticulo es el identificador utilizado como clave primaria en la tabla 'contenidos_articulos'.

boolean contenidos_articulo::borrar() *[line 692]*

Summary: Borra un articulo de los contenidos.

Elimina la entrada del articulo que se encuentra en la tabla 'contenidos_articulos'.

boolean contenidos_articulo::check_datos($datos) *[line 680]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el titulo

* el contenido de un articulo.

boolean contenidos_articulo::crear($titulo, $contenido, $idSeccion) *[line 651]*

Function Parameters:

* *$nombre* **$titulo** Titulo del articulo.
* *$contenido* **$contenido** Contenido del articulo.
* *$idSeccion* **$idSeccion** Identificador de la sección.

Summary: Crea un nuevo articulo en los contenidos.

Crea una nueva entrada en la tabla 'contenidos_articulos' inicializando todos los campos con los datos del nuevo articulo.

boolean contenidos_articulo::existe() *[line 629]*

Summary: Verifica si el articulo existe.

Comprueba que el identificador de articulo es correcto.

String contenidos_articulo::get_contador() *[line 574]*

Summary: Devuelve el contador de lecturas del articulo.

El contador de lecturas del artículo se encuentra guardado en la tabla 'contenidos_articulos' en el campo 'titulo_articulo'.

String contenidos_articulo::get_contenido() *[line 536]*

Summary: Devuelve el contenido del articulo.

El contenido del artículo se encuentra guardado en la tabla 'contenidos_articulos' en el campo 'contenido_articulo'.

String contenidos_articulo::get_fecha() *[line 556]*

Summary: Devuelve la fecha de creación del articulo.

La fecha de creación del artículo se encuentra guardada en la tabla 'contenidos_articulos' en el campo 'fecha_articulo'. Esta en formato 'AAAA-MM-DD'

Integer contenidos_articulo::get_id() *[line 504]*

Summary: Devuelve el identificador del articulo.

El identificador del articulo es la clave primaria de la tabla 'contenidos_articulos' que contiene la información del articulo de los contenidos.

Integer contenidos_articulo::get_posicion() *[line 594]*

Summary: Devuelve la posicion del articulo.

La posicion de un articulo es una forma de mantener ordenados los articulo de una seccion a gusto del usuario. La posicion se guarda en el campo 'posicion_articulo' dentro de la tabla 'contenidos_articulos'.

contenidos_articulos_Seccion contenidos_articulo::get_seccion() *[line 613]*

Summary: Devuelve la sección del articulo

Todos los articulos pertenecen a una seccion. La información de la seccion a la que pertenece se se encuentra en el campo 'id_seccion' de la tabla 'contenidos_articulos'.

String contenidos_articulo::get_titulo() *[line 517]*

Summary: Devuelve el título del articulo.

El título del artículo se encuentra guardado en la tabla 'contenidos_articulos' en el campo 'titulo_articulo'.

contenidos_articulo::inc_contador() *[line 782]*

Summary: Incrementa el contador de lecturas del articulo.

El contador de lecturas del articulo se incrementa en 1 cada vez que se muestra el articulo. Ese contador se encuentra en el campo 'contador_articulo' de la tabla 'contenidos_articulos'.

boolean contenidos_articulo::intercambiar($idArt2) *[line 755]*

Function Parameters:

* *$idArt2* **$idArt2** Identificador del articulo con el que se intercambia la posicion.

Summary: Intercambia las posicion de dos articulos.

Los articulos que se intercambian deben pertenecer a la misma seccion padre.

Boolean contenidos_articulo::set($titulo, $contenido, $contador, $idSeccion) *[line 718]*

Function Parameters:

* *$titulo* **$titulo** Nuevo titulo del articulo.
* *$contenido* **$contenido** Nuevo contenido del articulo.
* *$contador* **$contador** Contador de lecturas del articulo.
* *$idSeccion* **$idSeccion** Nueva seccion a la que pertenece el articulo.

Summary: Modifica las propiedades de un articulo.

La propiedad que modifica son el titulo y el contenido del mismo, que se encuentran en la tabla 'contenidos_articulos'. Tambien permite modificar la seccion a la que pertenece el articulo. Comprueba que los nuevos datos son correctos.

Class contenidos_bloque

[line 1019]

Summary: Clase que muestra un pequeño bloque informatico de las secciones.

Este bloque es navegable y muestra las secciones de la seccion actual y los articulos presentes en ella.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor contenidos_bloque::contenidos_bloque([$idSeccion = 0]) *[line 1028]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion a mostrar.

Summary: Contructor de la clase que obtiene el bloque de los contenidos.

Muestra las subsecciones y los articulos. En principio estan ordenados por el atributo posición.

Class contenidos_config

[line 1766]

Summary: Clase que gestiona las variables de configuración del módulo.
Las variables de configuración se encuentran almacenadas en la tabla 'contenidos_config'.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed contenidos_config::$db = *[line 1776]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor contenidos_config::contenidos_config() *[line 1784]*

Summary: Contructor de la clase.
Inicializa el atributo $this->db que contiene el enlace con la base de datos.

String contenidos_config::get_variable($nombreVar) *[line 1798]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Recupera el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'contenids_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

contenidos_config::set_variable($nombreVar, $nuevoValor) *[line 1817]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable a modificar.
* *$nuevoValor* **$nuevoValor** Nuevo valor de la variable.

Summary: Modifica el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'contenidos_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

Class contenidos_error

[line 1833]

Summary: Clase para mostrar mensajes del modulo contenidos.
Todos los mensajes del modulo se consideran mensajes de error.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor contenidos_error::contenidos_error([$tituloError = "phpKROND - Error "], [$msgError = ""]) *[line 1842]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class contenidos_mostrar

[line 799]

Summary: Clase que muestras las secciones y articulos del módulo.

Muestra las secciones y los articulos que tienen. Permitiendo navegar por ellos como si de un sistema de ficheros se tratase.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor contenidos_mostrar::contenidos_mostrar([$operacion = ""]) *[line 807]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase para mostrar los contenidos.

Realiza la operación indicada por el parametro.

String contenidos_mostrar::form_articulo_mostrar($idArticulo) *[line 960]*

Function Parameters:

* *$idArticulo* **$idArticulo** Identificador del articulo a mostrar.

Summary: Muestra el articulo.

Muestar el articulo y toda la información relacionada con él (titulo, fecha, lecturas y contenido). Tambien muestra una ruta completa para llegar al articulo. Incrementa el contador de lecuras del articulo.

String contenidos_mostrar::get_ruta([$idSeccion = 0]) *[line 936]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de seccion actual.

Summary: Devuelve la ruta completa de una seccion a partir de la raiz.

Cómo las secciones pueden anidarse, esto devuelve la ruta completa desde la seccion raiz hasta la seccion actual.

String contenidos_mostrar::mostrar([$idSeccion = 0]) *[line 843]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de Seccion a mostrar.

Summary: Muestra la seccion de los contenidos.
Muestra las subsecciones y los articulos. En principio estan ordenados por el atributo posición.

Class contenidos_seccion

[line 47]

Summary: Clase que gestiona las secciones de los contenidos.

Las articulos de los contenidos se clasifican en diferentes seccionespara permitir una organización. Una sección de contenidos puede a su vez contener otros secciones (subsecciones). La información de una sección se encuentra en la tabla 'contenidos_secciones'.

* **Package** phpkrond.modulos.contenidos
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed contenidos_seccion::$db = *[line 67]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed contenidos_seccion::$idSeccion = *[line 56]*

Summary: Atributo de la clase que guarda la clave de la sección.

El atributo de la clase $idSeccion es el identificador utilizado como clave primaria en la tabla 'contenidos_secciones'.

Constructor contenidos_seccion::contenidos_seccion([$idSeccion = 0]) *[line 78]*

Function Parameters:

* *$idSeccion* **$idSeccion** Identificador de la seccion.

Summary: Constructor de la clase que inicializa el atributo $idSeccion.

El atributo de la clase $idSeccion es el identificador utilizado como clave primaria en la tabla 'contenidos_secciones'.

boolean contenidos_seccion::borrar($recursivo) *[line 341]*

Function Parameters:

* *$recursivo* **$recursivo** Indica si se borra de forma recursiva (=true) o no.

Summary: Borra una sección de los contenidos.

Elimina la entrada de la sección que se encuentra en la tabla 'contenidos_secciones'.

Existen dos modos de funcionamiento de esta funcion. Borrar de forma recursiva para cuando la seccion tiene subsecciones y articulos, y borrar de forma no recursiva que solo borrara la sección si esta vacia (carece de articulos y no tiene subsecciones).

boolean contenidos_seccion::check_datos($datos) *[line 324]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el nombre de la sección.

boolean contenidos_seccion::crear($nombre, $idSecPadre) *[line 297]*

Function Parameters:

* *$nombre* **$nombre** Nombre de la sección.
* *$idSecPadre* **$idSecPadre** Identificador de la sección padre.

Summary: Crea una nueva seccion en los contenidos.

Crea una nueva entrada en la tabla 'contenidos_secciones' inicializando todos los campos con los datos de la nueva sección.

boolean contenidos_seccion::existe() *[line 272]*

Summary: Verifica si la seccion existe.

Comprueba que el identificador de seccion es correcto.

Array contenidos_seccion::get_articulos() *[line 187]*

Summary: Devuelve los artículos de la sección.

Una seccion contiene diversos articulos para mostrar. Con esto se ofrece una forma de mantener un orden lógico en los articulos que se pueden ver en el módulo. Devuelve un array de objetos contenidos_articulo.

Integer contenidos_seccion::get_id() *[line 96]*

Summary: Devuelve el identificador de sección.

El identificador de sección es la clave primaria de la tabla 'contenidos_secciones' que contiene la información de la sección de los contenidos. Existe una sección, padre de todas las demas y cuyo identificador es '0'. La información de esa sección 'nombre' se almacena en la configuración de este módulo.

String contenidos_seccion::get_nombre() *[line 137]*

Summary: Devuelve el nombre de la sección.

El nombre de la sección se encuentra guardado en la tabla 'contenidos_secciones' en el campo 'nombre_seccion'. Para la seccion de id = 0, (seccion raiz que da nombre al modulo de contenidos), el nombre se encuentra en la configuración del modulo, en la variable 'contenidos_nombre_raiz' dentro de la tabla 'contenidos_config'.

integer contenidos_seccion::get_num_articulos() *[line 252]*

Summary: Devuelve el numero de articulos que estan en la seccion.

Si la seccion tiene subsecciones, calcula el numero de articulos de las subsecciones y se los suma.

integer contenidos_seccion::get_num_secciones() *[line 228]*

Summary: Devuelve el numero de subSecciones que estan en la seccion.

No calcula el numero de subSecciones de forma recursiva. Sólo da el numero de secciones que contiene en el primer nivel.

Integer contenidos_seccion::get_posicion() *[line 163]*

Summary: Devuelve la posicion de la sección.

La posicion de una seccion es una forma de mantener ordenados las subsecciones de una seccion a gusto del usuario. La posicion se guarda en el campo 'posicion_seccion' dentro de la tabla 'contenidos_secciones'.

Integer contenidos_seccion::get_prev() *[line 111]*

Summary: Devuelve la seccion previa (padre) de la seccion actual.

Las secciones de contenidos pueden contener otras secciones (subsecciones). Esta información se guarda en la tabla 'contenidos_secciones' dentro del campo 'prev_seccion' y es el identificador de dicha sección.

Array contenidos_seccion::get_secciones() *[line 208]*

Summary: Devuelve las secciones (subsecciones) de la seccion.

Una sección puede tener a su vez diversas secciones

* subsecciones dentro de ella. Devuelve un array de
* bjetos contenidos_seccion.

boolean contenidos_seccion::intercambiar($idSec2) *[line 426]*

Function Parameters:

* *$idSec2* **$idSec2** Identificador de seccion con la que se intercambia la posicion.

Summary: Intercambia las posicion de dos secciones.

Las secciones que se intercambian deben pertenecer a la misma seccion padre.

Boolean contenidos_seccion::set($nombre) *[line 399]*

Function Parameters:

* *$nombre* **$nombre** Nuevo nombre de la sección.

Summary: Modifica las propiedades de una sección.

La propiedad que modifica es el nombre, que se encuentran en la tabla 'contenidos_secciones'. Comprueba que los nuevos datos son correctos.

Package phpkrond Procedural Elements

mainfunctions.php

* **Package** phpkrond

CAJA_TIPO1 = 1 *[line 63]*
CAJA_TIPO2 = 2 *[line 64]*
CAJA_TIPO3 = 3 *[line 65]*
CAJA_VACIA = *[line 62]*
OBJ_CONTENEDOR = 2 *[line 61]*
OBJ_DINAMICO = 1 *[line 60]*
OBJ_ESTATICO = *[line 59]*
PAGINA_NORMAL = *[line 66]*
PAGINA_SISTEMA = 1 *[line 67]*
_PHPKROND_CLASS = 1 *[line 31]*
require_once("adodb/adodb.inc.php") *[line 54]*

Package phpkrond Classes

Class krond_cfg

[line 2398]

Summary: Clase que gestiona la configuración del sistema.
Inicializa la conexión con la base de datos.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_cfg::$basedatos = *[line 2409]*

Summary: Contiene el nombre de la base de datos.

mixed krond_cfg::$db = *[line 2469]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_cfg::$locale = *[line 2458]*

Summary: Tipo de local de la página.
Se utiliza para configurar el programa según las preferencias del país. Ajustando los formatos de fecha y hora. Ej: Para sistemas windows 'sp', 'spanish' -> Español 'en', 'english' -> Ingles 'us' -> USA En sistemas unix se rigen por los nombres de locales tradicionales. 'es_ES' -> Español de España 'en_ES' -> Ingles 'en_US' -> Ingles de EEUU

mixed krond_cfg::$motorSql = *[line 2441]*

Summary: Tipo de servidor de BD.
Los tipos de servidores se pueden consultar en los tipos de servidores adminitidos por ADODB: mysql, pgsql, odbc y otros.

mixed krond_cfg::$password = *[line 2423]*

Summary: Password del usuario para conectarse con la BD.
Esta clave no esta encriptada.

mixed krond_cfg::$servidor = *[line 2431]*

Summary: Nombre del servidor de BD.
Puede ser el nombre DNS del servidor o su dirección IP.

mixed krond_cfg::$username = *[line 2415]*

Summary: Nombre del usuario con permisos para conectarse con la BD.

mixed krond_cfg::$version = "200" *[line 2403]*

Summary: Atributo de la clase que guarda la versión del programa.

Constructor krond_cfg::krond_cfg($bd, $user, $pwd, $server, $motorSql, $locale) *[line 2484]*

Function Parameters:

* *$bd* **$bd** Nombre de la base de datos que alberga el sistema.
* *$user* **$user** Nombre del usuario de conexión con la BD.
* *$pwd* **$pwd** Password del usuario de conexión con la BD.
* *$server* **$server** Servidor de Base de Datos.
* *$motorSql* **$motorSql** Tipo de Servidor de BD.

Summary: Contructor de la clase para inicializar los atributos.

Inicializa los atributos de la clase y llama a la función que realiza la conexion con la base de datos. Estos valores se pueden poner en el fichero 'config.inc.php'.

String krond_cfg::get_variable($nombreVar) *[line 2538]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Devuelve el valor de un variable de configuración.

Esta funcion se utiliza para devolver valores de configuración que se encuentran almencenados en la base de datos. Estas variables se almacenan en la tabla 'krond_config'. El nombre de la variable se encuentra en el campo 'nombre' y su valor en campo 'valor'.

krond_cfg::init() *[line 2501]*

Summary: Realiza la conexion con la base de datos.

Inicializa el atributo $this->db que contiene la conexión con la base de datos. Tambien configura el locale del sistema.

krond_cfg::toString() *[line 2516]*

Summary: Muestra las variables de la configuración del sistema.

Las variables de configuración del sistema dan información sobre la versión, nombre de la base de datos, usuario de de conexion y clave del mismo. Nombre del servidor y tipo de

servidor.

Class krond_contenedor

[line 1628]

Summary: Clase que gestiona los Objetos Contenedores del Sistema.

Controla todas las operaciones que se hacen con el Objeto Contenedor. Añade, elimina e intercambia de posición objetos contenidos en el contenedor. Tambien sir para obtener la lista de objetos contenidos en el contenedor. Esta información del contenedor se encuentra en la tabla 'krond_contenedores'.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_contenedor::$db = *[line 1648]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_contenedor::$idContenedor = *[line 1637]*

Summary: Atributo de la clase que guarda la clave del contenedor.

El atributo de la clase $idContenedor es el identificador utilizado como clave primaria en la tabla 'krond_contenedores'.

Constructor krond_contenedor::krond_contenedor([$idContenedor = 0]) *[line 1659]*

Function Parameters:

* *$idContenedor* **$idContenedor** Integer con el identificador del contenedor.

Summary: Contructor de la clase que inicializa el atributo $idContenedor.

El atributo $idContenedor es la clave primaria de la tabla 'krond_contenedores' para recuperar la información de los objetos padre y contenidos del contenedor.

Boolean krond_contenedor::check_ciclo($objPadre, $objHijo) *[line 1783]*

Summary: Comprueba que no se crea un ciclo recursivo de contenedores

Antes de añadir un objeto a un contenedor, se tiene que comprobar que esa incorporación no formaría un ciclo recursivo, que provocaria un error. Un ejemplo, siendo A, B y C contenedores: A->B->C Si añadimos al contenedor C, el contenedor A, se formaría un ciclo A->B->C->A... y nunca se podria evaluar el objeto puesto que tiene infinitos elementos. El algoritmo busca entre los ID de los objetos contenidos el ID del padre, si lo

encuentra indica que se formaria un ciclo recursivo.

Boolean krond_contenedor::crear($idPadre, $idContenido) *[line 1744]*

Function Parameters:

* *$idPadre* **$idPadre** Identificador del objeto padre.
* *$idContenido* **$idContenido** Identificador del objeto contenido.

Summary: Añade un nuevo un contenedor a un objeto contenedor.

Crea un nueva entrada en la tabla 'krond_contenedores' inicializando los campos 'id_obj_padre' con el identificador del objeto padre y 'id_obj_contenido' con el identificador del objeto contenido. Ambos identificadores son claves ajenas.

krond_contenedor::del() *[line 1834]*

Summary: Elimina un contenedor del sistema.

Borra la entrada de la tabla 'krond_contenedores' del contenedor especificado.

Integer krond_contenedor::get_id() *[line 1672]*

Summary: Recupera el identificador del contenedor.

El identificador de plantilla es un entero y que se utiliza como clave primaria en la tabla 'krond_contenedores'.

krond_objeto krond_contenedor::get_obj_contenido() *[line 1705]*

Summary: Devuelve el objeto contenido del contenedor.

El objeto contenido del contenedor es un objeto. El identificador de este objeto se encuentra en el campo 'id_obj_contenido' de la tabla 'krond_contenedores'. Es una clave ajena de esta tabla.

krond_objeto krond_contenedor::get_obj_padre() *[line 1686]*

Summary: Devuelve el objeto padre del contenedor.

El objeto padre del contenedor es un objeto de tipo contenedor. El identificador de este objeto se encuentra en el campo 'id_obj_padre' de la tabla 'krond_contenedores'. Es una clave ajena de esta tabla.

Integer krond_contenedor::get_posicion() *[line 1723]*

Summary: Devuelve la posición del objeto contenido en el contenedor.

Los objetos contenidos de un contenedor estan ordenados según una posicion. Esta posición se guarda en el campo 'posicion' de la tabla 'krond_contenedores'.

boolean krond_contenedor::intercambia($idCont2) *[line 1848]*

Function Parameters:

* *$idCont2* **$idCont2** Identificador del Contenedor segundo.

Summary: Intercambia los objetos contenidos de dos contenedores.

Cambia los objetos contenido del contenedor actual y del contenedor pasado en el parametro.

Class krond_error

[line 2555]

Summary: Clase para mostrar mensajes del sistema.
Todos los mensajes del sistema se consideran mensajes de error.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor krond_error::krond_error([$tituloError = "phpKROND - Error "], [$msgError = ""]) *[line 2564]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class krond_objeto

[line 1195]

Summary: Clase que gestiona los Objetos del Sistema.

Controla todas las operaciones que se hacen con un Objeto. Crea y borra Objetos en el sistema, establece los atributos de los mismos. También los evulua para luego insertarlos en las variables de plantilla que tienen las páginas. La información de las objetos se encuentra en la tabla 'krond_objetos'.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_objeto::$db = *[line 1215]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_objeto::$idObjeto = *[line 1204]*

Summary: Atributo de la clase que guarda la clave del objeto.

El atributo de la clase $idObjeto es el identificador utilizado como clave primaria en la tabla 'krond_objetos'.

Constructor krond_objeto::krond_objeto([$idObjeto = 1]) *[line 1226]*

Function Parameters:

* *$idObjeto* **$idObjeto** Identificador del Objeto

Summary: Constructor de la clase que inicializa el atributo $idObjeto.

El identificador del Objeto es un entero y que se utiliza como clave primaria en la tabla 'krond_objetos'. Por defecto se inicializa con el objeto nulo.

Boolean krond_objeto::borrar() *[line 1578]*

Summary: Elimina un Objeto del sistema.

Borra el objeto del sistema. Antes de proceder a borrarlo realiza una serie de comprobaciones para ver si puede borrarlo.

* El objeto nulo no se puede borrar.
* Un objeto utilizado en una página no se puede borrar.
* Un objeto utilizado por un contenedor no se puede borrar.
* Un objeto contenedor que no este vacio no se puede borrar.

Boolean krond_objeto::check_caja($caja) *[line 1499]*

Function Parameters:

* *$caja* **$caja** Tipo de Objeto

Summary: Comprueba validez del tipo de caja

boolean krond_objeto::check_datos($datos) *[line 1469]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el nombre de objeto y su contenido

Boolean krond_objeto::check_tipo($tipo) *[line 1479]*

Function Parameters:

* *$tipo* **$tipo** Tipo de Objeto

Summary: Comprueba validez del tipo de objeto

krond_objeto::crear($nombre, $tipo, $contenido, $caja) *[line 1447]*

Function Parameters:

* *$nombre* **$nombre** Nombre del Objeto.
* *$tipo* **$tipo** Tipo del Objeto. Los objetos pueden ser 'OBJ_ESTATICO', 'OBJ_DINAMICO', 'OBJ_CONTENEDOR'.
* *$contenido* **$contenido** Contenido del Objeto.
* *$caja* **$caja** Tipo de Caja del Objeto. Los tipos de cajas son 'CAJA_VACIA', 'CAJA_TIPO1', 'CAJA_TIPO2', 'CAJA_TIPO3'.

Summary: Crea un objeto en el sistema.

Crea un nueva entrada en la tabla 'krond_objetos' inicializando los campos con los parámetros de la función.

String krond_objeto::evalua() *[line 1393]*

Summary: Devuelve el resultado de la evaluzación del objeto.

Evalua el objeto sea sea el tipo que tenga. Si es un objeto estático devuelve el contenido del objeto. Si es un objeto dinamico evalua el objeto (utilizando la funcion eval de PHP) y devuelve el resultado de dicha evaluación. Por ultimo, si el objeto es de tipo contenedor, va evaluando los objetos que contiene para ir concatenando las diferentes evaluaciones de esos objetos. Esta función utiliza Output Buffering.

Boolean krond_objeto::existe() *[line 1522]*

Summary: Comprueba si un objeto existe en el sistema.

La verificación se realiza consultado en la tabla 'krond_objetos' si existe el identificador del objeto.

Array krond_objeto::get_all() *[line 1366]*

Summary: Obtiene los Objetos en el Sistema

Devuelve un array de krond_objetos con todos los objetos presentes en el sistema.

Array krond_objeto::get_contenedores() *[line 1327]*

Summary: Devuelve un array con los objetos contenedores del objeto.

La información se recoge de la tabla 'krond_contenedores'.

String krond_objeto::get_contenido() *[line 1311]*

Summary: Devuelve el contenido del objeto.

La información se recoge del campo 'contenido_objeto' de la tabla 'krond_objetos'.

Integer krond_objeto::get_id() *[line 1239]*

Summary: Recupera el identificador del objeto.

El identificador del Objeto es un entero y que se utiliza como clave primaria en la tabla 'krond_objetos'.

String krond_objeto::get_nombre() *[line 1293]*

Summary: Devuelve el nombre del objeto.

La información se recoge del campo 'nombre_objeto' de la tabla 'krond_objetos'.

Array krond_objeto::get_paginas() *[line 1346]*

Summary: Devuelve un array con las página que utilizan el objeto.

La información se recoge de la tabla 'krond_sustituciones'.

Integer krond_objeto::get_tipo() *[line 1256]*

Summary: Recupera el tipo del objeto.

La información se recoge del campo 'tipo_objeto' de la tabla 'krond_objetos'. En el sistema los objetos son de 3 tipos diferentes:

* Objetos Estáticos que son objetos fijos que no varian.
* Objetos Dinámicos que son evaluables en tiempo de ejecución.
* Objetos Contenedores que contienen otros objetos.

Integer krond_objeto::get_tipo_caja() *[line 1275]*

Summary: Recupera el tipo de caja que enmarca al objeto.

La información se recoge del campo 'tipo_caja_objeto' de la tabla 'krond_objetos'.

Boolean krond_objeto::set($nombre, $tipo, $contenido, $caja) *[line 1548]*

Function Parameters:

* *$nombre* **$nombre** Nombre del Objeto.
* *$tipo* **$tipo** Tipo del Objeto. Los objetos pueden ser 'OBJ_ESTATICO', 'OBJ_DINAMICO', 'OBJ_CONTENEDOR'.
* *$contenido* **$contenido** Contenido del Objeto.
* *$caja* **$caja** Tipo de Caja del Objeto. Los tipos de cajas son 'CAJA_VACIA', 'CAJA_TIPO1', 'CAJA_TIPO2', 'CAJA_TIPO3'.

Summary: Cambia los valores de los atributos de un objeto.

El cambio se realiza en los campos de la tabla 'krond_objetos'. Concretamente los campos son 'nombre_objeto', 'tipo_objeto', 'contenido_objeto' y 'tipo_caja_objeto'.

Class krond_pagina

[line 82]

Summary: Clase que gestiona las páginas del Sistema.

Controla todas las operaciones que se hacen con una página. Crea y borra páginas en el sistema, establece los atributos de las mismas, verifica la autorización para mostrarla, etc. La información de las páginas se encuentra en la tabla 'krond_paginas'.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_pagina::$db = *[line 102]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_pagina::$idPagina = *[line 91]*

Summary: Atributo de la clase que guarda la clave de la página.

El atributo de la clase $idPagina es el identificador utilizado como clave primaria en la tabla 'krond_paginas'.

Constructor krond_pagina::krond_pagina([$idPagina = ""]) *[line 113]*

Function Parameters:

* *$idPagina* **$idPagina** Identificador de Página.

Summary: Contructor de la clase.

Inicializa el identificador de página. Si el identificador es nulo, se utiliza el nombre del script para determinar cuál es su identificador.

Boolean krond_pagina::borrar() *[line 380]*

Summary: Borra la página del Sistema.

Si la página es del sistema no se borrará, primero se debe cambiar a página normal para poder borrarlas. El proceso de borrado de la página es el siguiente:

* Borrar sustituciones.
* Borrar entrada en la tabla 'krond_paginas' de la BD.
* Eliminar archivo en sistema de ficheros.

String krond_pagina::calcula() *[line 176]*

Summary: Función que calcula el contenido de la página.

Se encarga de solicitar el contenido de la plantilla, obtener las variables de la misma y objetos asociados a las sustituciones de la página para luego evaluarlos y reemplazar las variables por el resultado de la evaluacion.

boolean krond_pagina::check_auth() *[line 539]*

Summary: Checkea usuario de administración mediante HTTP.

Si el usuario no se ha validado, el navegador muestra un cuadro de diálogo solicitando login y passwd para validar al usuario.

boolean krond_pagina::check_cookie() *[line 500]*

Summary: Checkea la cookie de administracion.

La cookie de administración es $_COOKIE['krond_admin'] que guarda el login de administración y la passwd encriptada con md5. Comprueba que esos datos son correctos contrastandolos con la BD.

Boolean krond_pagina::check_nombre($nombre) *[line 462]*

Function Parameters:

* *$nombre* **$nombre** String con el nombre de la página a crear.

Summary: Checkea que el nombre la página sea correcto.

Un nombre de página es correcto si:

* No existe el nombre de la página en el sistema.
* No existe un fichero en el sistema con el mismo nombre.
* Se puede crear el fichero en el sistema con ese nombre.

Boolean krond_pagina::crear($nombre, $idPlantilla, $tipo) *[line 414]*

Function Parameters:

* *$nombre* **$nombre** String con el nombre la página.
* *$idPlantilla* **$idPlantilla** Identificador de plantilla.
* *$tipo* **$tipo** Boolean con el tipo de página.

Summary: Crea una página en el sistema.

El proceso de creación de una página es el siquiente:

* Insertar entrada de página en la tabla de la base de datos 'krond_paginas'.
* Obtener identificador de página.
* Inicializar sustituciones al objeto NULO.
* Crear fichero y escribir contenido de inicializaciónEl nombre de la página es checkeado para comprobar su validez.

boolean krond_pagina::es_sistema() *[line 593]*

Summary: Función para indicar que la página es del sistema.

Se utiliza la funcion get_tipo para saberlo.

Array krond_pagina::get_all() *[line 355]*

Summary: Obtiene las páginas existentes en el sistema.

Devuelve un array de krond_paginas con todas las páginas activas del sistema.

Boolean krond_pagina::get_autorizacion() *[line 317]*

Summary: Obtiene la autorización para mostrar la página.

Si la página es del sistema entonces se comprobará si el usuario que solicita la página tiene autorización. La autorización de comprueba mirando la variable de session $_SESSION['KROND_AUTH']. En el caso de que no este definida, se comprobará checkeando la cookie. En último caso se pide autorización HTTP.

Integer krond_pagina::get_id() *[line 257]*

Summary: Devuelve el identificador de la página.

El identificador de página es un entero y que se utiliza como clave primaria en la base de datos.

String krond_pagina::get_nombre() *[line 198]*

Summary: Devuelve el nombre de la página.

El nombre de la página se recupera de la base de datos aunque también existe un fichero con su mismo nombre. El nombre se encuentra en la tabla 'krond_paginas' en el campo 'nombre_pagina'.

krond_plantilla krond_pagina::get_plantilla() *[line 272]*

Summary: Devuelve la Plantilla de la página.

Se obtiene un objeto de la clase Plantilla con la plantilla a la que pertenece la página. Se consulta el campo 'id_plantilla' de la tabla 'krond_paginas' para saber el identificador de la plantilla y construir el objeto.

Array krond_pagina::get_sustituciones() *[line 293]*

Summary: Devuelve un array de Sustituciones de variables de la página.

Se consulta el campo 'id_sustitucion' de la tabla 'krond_sustituciones' para saber las sustituciones de la página.

Boolean krond_pagina::get_tipo() *[line 224]*

Summary: Devuelve el tipo de la página.

El tipo de la página se encuentra en la tabla 'krond_paginas' en el campo 'sistema'. Las páginas son de 2 tipos diferentes:

* PAGINA_SISTEMA que requieren autorización para verla. Solo los administradores pueden verlas.
* PAGINA_NORMAL que cualquiera puede verlas.

krond_pagina::inicializar_sustituciones($idPagina, $idPlantilla) *[line 438]*

Function Parameters:

* *$idPagina* **$idPagina** Identificador de página para la sustitución.
* *$idPlantilla* **$idPlantilla** Identificador de plantilla para la sustitución.

Summary: Inicializa las sustituciones de la página con el objeto NULO.

Las sustiticiones se encuentran en la tabla 'krond_sustituciones'.

krond_pagina::mostrar() *[line 137]*

Summary: Muestra la página.

Devuelve al servidor web el contenido de la página. Si la página es del Sistema, pide la autorización para mostrarla.

krond_pagina::set_tipo($tipo) *[line 243]*

Function Parameters:

* *$tipo* **$tipo** Tipo de página. 'PAGINA_SISTEMA' o 'PAGINA_NORMAL'

Summary: Establece el tipo de la página.

Las páginas pueden ser de 2 tipos diferentes:

* 'PAGINA_SISTEMA' que requieren autorización para verla. Solo los administradores pueden verlas.
* PAGINA_NORMAL' que cualquiera puede verlas.

Class krond_plantilla

[line 614]

Summary: Clase que controla las plantillas del Sistema.

Las plantillas del sistema son módelos de páginas con etiquetas del tipo {nombreEtiqueta} que son variables de sustitución. La información de las plantillas se encuentra en la tabla 'krond_plantillas'.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_plantilla::$db = *[line 634]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_plantilla::$idPlantilla = *[line 623]*

Summary: Atributo de la clase que guarda la clave de la plantilla.

El atributo de la clase $idPlantilla es el identificador utilizado como clave primaria en la tabla 'krond_plantillas'.

Constructor krond_plantilla::krond_plantilla([$idPlantilla = ""]) *[line 644]*

Function Parameters:

* *integer* **$idPlantilla** Identificador de plantilla.

Summary: Contructor de la clase que inicializa el atricuto $idPlantilla.

El atributo $idPlantilla es la clave primaria para recuperar la información de la plantilla en la tabla 'krond_plantillas'.

String krond_plantilla::add_llaves($cadena) *[line 1174]*

Function Parameters:

* *$cadena* **$cadena** Texto sobre el que se realiza la sustitución.

Summary: Sustituye las entidades HTML '{', '}' por su correspondiente carácter.

Esta función es útil cuando se ha editado una plantilla y se quieren deshacer los cambios

de {} para que aparezcan de nuevo en el contenido de la plantilla. Hace lo contrario que la función strip_llaves.

Boolean krond_plantilla::borrar() *[line 859]*

Summary: Borra la plantilla del Sistema.

Si la plantilla no tiene dependencias, es decir, la plantilla no esta siendo utilizada por otras páginas, entonces se se puede borrar sin problemas. Elimina las variables de la plantilla que estan en la la tabla 'krond_variables' y la entrada en la tabla 'krond_plantillas'.

Array krond_plantilla::buscar_variables($contenido) *[line 1131]*

Function Parameters:

* *$contenido* **$contenido** String con el contenido de la plantilla.

Summary: Devuelve las variables de plantilla que contiene la cadena $contenido.

Las variables de plantilla son cadenas de la forma {nombreVariable}. Las expresión regular que busca estas variables es "[{][a-zA-Z_0-9]+[}]". Esta función es util para inicializar las variables de plantilla cuando se crea una nueva plantilla.

String krond_plantilla::caja_evalua($tipoCaja, $contenido) *[line 798]*

Function Parameters:

* *$tipoCaja* **$tipoCaja** Constante para seleccionar el tipo de caja de la plantilla. Las constantes son 'CAJA_VACIA', 'CAJA_TIPO1', 'CAJA_TIPO2', 'CAJA_TIPO3'.
* *$contenido* **$contenido** String con el contenido a enmarcar.

Summary: Devuelve un texto enmarcado con el tipo de caja seleccionado de la plantilla

El texto se enmarca según el tipo de caja que se pasa por los paramteros de la función y sustituye a la variable {cuerpo} del contenido de la caja por el texto enviado.

Boolean krond_plantilla::check_caja($caja) *[line 1114]*

Function Parameters:

* *$caja* **$caja** String con el contenido de la caja.

Summary: Checkea el contenido de la caja de la plantilla.

Comprueba que el contenido de la caja de la plantilla contiene la variable de sustitución {cuerpo}

Boolean krond_plantilla::check_contenido($contenido) *[line 1096]*

Function Parameters:

* *$contenido* **$contenido** String con el contenido de la plantilla.

Summary: Checkea el contenido de la plantilla.

Comprueba la validez del contenido de la plantilla. Deberia verificar que el contneido contiene variables de sustitución del tipo {nombreVariable}.

Boolean krond_plantilla::check_dependencias() *[line 1076]*

Summary: Checkea las dependencias de una plantilla.

Comprueba si la plantilla tiene páginas que la usan. Una plantilla no se puede borrar siempre que tenga dependencias de uso de páginas.

Boolean krond_plantilla::check_existe($nombre) *[line 1057]*

Function Parameters:

* *$nombre* **$nombre** String con el nombre de la plantilla.

Summary: Checkea si existe una plantilla con el mismo nombre.

Comprueba si existe una plantilla que tenga el mismo nombre. No pueden existir plantillas con el mismo nombre.

Boolean krond_plantilla::check_nombre($nombre) *[line 1044]*

Summary: Checkea el nombre de la plantilla.

Comprueba la validez del nombre de la plantilla. Función no implementada.

Boolean krond_plantilla::crear($nombre, $contenido, $caja1, $caja2, $caja3) *[line 893]*

Function Parameters:

* *$nombre* **$nombre** Nombre de la plantilla a crear.
* *$contenido* **$contenido** Contenido de la plantilla.
* *$caja1* **$caja1** Contenido del tipo de Caja 1 de la plantilla
* *$caja2* **$caja2** Contenido del tipo de Caja 2 de la plantilla
* *$caja2* **$caja3** Contenido del tipo de Caja 3 de la plantilla

Summary: Crea una nueva plantilla en el sistema.

Las plantillas son modelos de páginas. Los datos de la plantilla se encuentran en dos tablas, 'krond_plantillas' y 'krond_variables'. El contenido de una plantilla contiene, además

de texto html, variables del tipo {nombreVariable} que serán sustituidas por las los objetos del sistema en las sustituciones de páginas. Tambien contiene 3 módelos de cajas difierentes para enmarcar textos adecuándolos a la plantilla. Los modelos de cajas contienen en su interior la variable {cuerpo}.

Boolean krond_plantilla::existe() *[line 837]*

Summary: Comprueba si el identificador de plantilla existe en el sistema.

La verificación se realiza consultado en la tabla 'krond_plantilla' si existe el identificador de plantilla.

Array krond_plantilla::get_all() *[line 817]*

Summary: Obtiene las plantillas del sistema.

Devuelve un array de krond_plantillas con todas las plantillas del sistema.

String krond_plantilla::get_caja($tipoCaja) *[line 758]*

Function Parameters:

* *$tipoCaja* **$tipoCaja** Constante para seleccionar el tipo de caja de la plantilla. Las constantes son 'CAJA_VACIA', 'CAJA_TIPO1', 'CAJA_TIPO2', 'CAJA_TIPO3'.

Summary: Devuelve el contenido del tipo de caja de una plantilla.

Se consulta el campo 'tipo_caja1' o 'tipo_caja2' o 'tipo_caja13' de la tabla 'krond_plantillas'. El tipo de caja es un modelo de plantilla especial para generar Cajas de texto. En su interior contiene una variable de de sustitucion de nombre {cuerpo} que será sustituida con la funcion caja_evalua ().

String krond_plantilla::get_contenido() *[line 690]*

Summary: Devuelve el contenido de la plantilla.

El contenido de la plantilla se recupera de la tabla 'krond_plantillas' del campo 'texto_plantilla'. Este contenido es un módelo de página HTML que contiene etiquetas del tipo {nombrEtiqueta} que luego son utilizadas como variables para ser sustituidas por los objetos en las sustituciones de páginas.

Integer krond_plantilla::get_id() *[line 657]*

Summary: Recupera el identificador de la plantilla.

El identificador de plantilla es un entero y que se utiliza como clave primaria en la base de datos.

String krond_plantilla::get_nombre() *[line 669]*

Summary: Devuelve el nombre de la plantilla.

El nombre de la plantilla se recupera de la tabla 'krond_plantillas' del campo

'nombre_plantilla'.

Array krond_plantilla::get_paginas() *[line 707]*

Summary: Devuelve un array de Paginas que usan la plantilla.

Se consulta el campo 'id_pagina' de la tabla 'krond_paginas' para saber las páginas que la utilizan.

Array krond_plantilla::get_variables() *[line 731]*

Summary: Devuelve un array de Variables que tiene la plantilla.

Se consulta el campo 'id_variable' de la tabla 'krond_variables' para saber las variables que tiene. Las variables se han dado de alta en la tabla cuando se creo la plantilla. Las variables son cadenas de caracteres que estan en el contenido de la plantilla del tipo {nombreVariable}.

Boolean krond_plantilla::set($nombre, $contenido, $caja1, $caja2, $caja3) *[line 943]*

Function Parameters:

* *$nombre* **$nombre** Nombre de la plantilla a crear.
* *$contenido* **$contenido** Contenido de la plantilla.
* *$caja1* **$caja1** Contenido del tipo de Caja 1 de la plantilla
* *$caja2* **$caja2** Contenido del tipo de Caja 2 de la plantilla
* *$caja2* **$caja3** Contenido del tipo de Caja 3 de la plantilla

Summary: Modifica los atributos de una plantilla.

Modifica todos los atributos de la plantilla. Si se detecta que se han modificado las variables de plantilla que estan en $contenido, (aparecen nuevas variables de plantilla o se han eliminado algunas), entonces esta funcion reconstruye todas las asignaciones que puedieran tener las páginas que usen esta plantilla. Eliminando las sustituciones que corresponen a variables que han desaparecido e inicializando las sustituciones (con el objeto nulo) de las variables nuevas.

String krond_plantilla::strip_llaves($cadena) *[line 1157]*

Function Parameters:

* *$cadena* **$cadena** Texto sobre el que se realiza la sustitución.

Summary: Sustituye las llaves '{', '}' por su correspondiente entidad HTML.

Esta función es útil cuando se esta editando una plantilla y no se quiere que el motor de la aplicación realiza sustituciones sobre el propio código de la plantilla. Hace lo contrario que la funcion add_llaves.

Class krond_sustitucion

[line 1875]

Summary: Clase que gestiona las Sustituciones de Objetos del Sistema.
Controla todas las operaciones que se hacen con una Sustitución.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_sustitucion::$db = *[line 1895]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_sustitucion::$idSustitucion = *[line 1884]*

Summary: Atributo de la clase que guarda la clave de la sustitución.
El atributo de la clase $idSustitucion es el identificador utilizado como clave primaria en la tabla 'krond_sustituciones'.

Constructor krond_sustitucion::krond_sustitucion($idSustitucion) *[line 1905]*

Function Parameters:

* *$idObjeto* **$idSustitucion** Identificador de la Sustitución

Summary: Constructor de la clase que inicializa el atributo $idSustitucion.
El identificador del Sustitución es un entero y que se utiliza como clave primaria en la tabla 'krond_sustituciones'.

Integer krond_sustitucion::get_id() *[line 1919]*

Summary: Recupera el identificador de la Sustitución.
El identificador de la Sustitución es un entero y que se utiliza como clave primaria en la tabla 'krond_sustituciones'.

krond_objeto krond_sustitucion::get_objeto() *[line 1930]*

Summary: Devuelve el Objeto de la sustitución
Todas las sustituciones las realizan objetos.

krond_pagina krond_sustitucion::get_pagina() *[line 1950]*

Summary: Devuelve la Pagina de la sustitución.

Todas las sustituciones se realizan en páginas que sustituyen las variables de plantilla por los objetos.

krond_variable krond_sustitucion::get_variable() *[line 1970]*

Summary: Devuelve la Variable de la sustitución.

Todas las sustituciones se realizan en variables de plantillas.

krond_sustitucion::set($idObjeto) *[line 1991]*

Function Parameters:

* *$idObjeto* **$idObjeto** Identificador del objeto de sustitución.

Summary: Modifica el objeto de sustitución.

La modificación se hace en el campo 'id_objeto' de la tabla 'krond_sustituciones'. $idObjeto es una clave ajena.

Class krond_useradm

[line 2122]

Summary: Clase que gestiona los usuarios de administración del sistema.
Controla todas las operaciones que se hacen con los Usuarios de Administración.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_useradm::$db = *[line 2142]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_useradm::$idAdmin = *[line 2131]*

Summary: Atributo de la clase que guarda la clave del usuario.
El atributo de la clase $idAdmin es el identificador utilizado como clave primaria en la tabla 'krond_admin_users'.

Constructor krond_useradm::krond_useradm([$idAdmin = 0]) *[line 2153]*

Function Parameters:

* *$idAdmin* **$idAdmin** Identificador de Usuario de Administración.

Summary: Constructor de la clase que inicializa el atributo $idAdmin.
El identificador de Usuario de Administración es un entero y que se utiliza como clave primaria en la tabla 'krond_admin_users'.

boolean krond_useradm::borrar() *[line 2363]*

Summary: Elimina el usuario de administración.
Borra si el usuario de administración referenciado por el atributo de la clase $idAdmin. Verifica que el usuario de administración se ha logado de forma correcta y que puede borrarlo (bien porque sea superadministrador o porque sea él mismo).

boolean krond_useradm::check_admin($login, $passwd) *[line 2285]*

Function Parameters:

* *$login* **$login** Login del Administrador.

* *$passwd* **$passwd** Password del Administrador. Esta clave se entrega en texto llano.

Summary: Checkea que el usuario de administración es correcto.
Comprueba que el login y passwd del usuario de administración son correctos contrastandolos con la base de datos y luego, en el caso de ser correctos, inicializa el atributo '$idAdmin' con el identificador del administracor obtenido.

Boolean krond_useradm::check_datos($cadena) *[line 2307]*

Function Parameters:

* *$cadena* **$cadena** Datos a chequear su corrección.

Summary: Checkea que los datos entegrados no contienen caracteres no validos.
En el login ni en el password del administrador se permiten los caracteres "%|#|=|'|,|_" para evitar SQL-inyecttion.

boolean krond_useradm::crear($login, $passwd, $nombre, [$super = 0]) *[line 2340]*

Function Parameters:

* *$login* **$login** Login del nuevo usuario de administración.
* *$passwd* **$passwd** Password del nuevo usuario de administración, no encriptada.
* *$nombre* **$nombre** Nombre completo del usuario de administración.
* *$super* **$super** boolean para indicar si el usuario de SuperAdministrador.

Summary: Crea un nuevo usuario de Administración en el sistema.
Inicializa una nueva entrada en la tabla 'krond_admin_users'.

boolean krond_useradm::es_super() *[line 2262]*

Summary: Indica si el usuario de administración es super usuario.
El super usuario es el primer usuario que se crea en el sistema. Este administrador puede crear y borrar a otros usuarios de administración. Los restantes usuarios no pueden borrar usuarios de administración. La información se encuentra en el campo 'super' de la tabla 'krond_admin_users'.

boolean krond_useradm::existen() *[line 2321]*

Summary: Comprueba si hay al menos un usuario de administración.
Si no existe ningun usuario de administración devuelve false.

Array krond_useradm::get_all() *[line 2238]*

Summary: Obtiene los Usuarios de Administración en el Sistema

Devuelve un array de krond_useradm con todos los usuarios de administración del sistema.

Integer krond_useradm::get_id() *[line 2167]*

Summary: Recupera el identificador del usuario de administración.

El identificador de Usuario de Administración es un entero y que se utiliza como clave primaria en la tabla 'krond_admin_users'.

String krond_useradm::get_login() *[line 2201]*

Summary: Devuelve el login del Administrador.

El login del administración se encuentra en la tabla 'krond_admin_users' en el campo 'login_admin' y es el nombre usado para hacer el login en el sistema por ese administrador

String krond_useradm::get_nombre() *[line 2181]*

Summary: Devuelve el nombre del Administrador.

El nombre del administración se encuentra en la tabla 'krond_admin_users' en el campo 'nombre_admin' y es el nombre completo del administración. Este nombre no se utiliza para hacer el login.

String krond_useradm::get_passwd() *[line 2220]*

Summary: Devuelve la password del Administrador.

La password del adminstrador se encuentra en el campo 'passwd_admin' en la tabla 'krond_admin_users'. Esta password se encuentra encriptada con md5.

Class krond_variable

[line 2010]

Summary: Clase que gestiona las Variables de Plantilla.
Controla todas las operaciones que se hacen con las variables de plantilla.

* **Package** phpkrond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed krond_variable::$db = *[line 2030]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed krond_variable::$idVariable = *[line 2019]*

Summary: Atributo de la clase que guarda la clave de la variable.
El atributo de la clase $idVariable es el identificador utilizado como clave primaria en la tabla 'krond_variables'.

Constructor krond_variable::krond_variable($idVariable) *[line 2040]*

Function Parameters:

* *$idObjeto* **$idVariable** Identificador de la Variable

Summary: Constructor de la clase que inicializa el atributo $idVariable.
El identificador de Variable es un entero y que se utiliza como clave primaria en la tabla 'krond_variables'.

krond_variable::borrar() *[line 2102]*

Summary: Elimina la variable de plantilla
Esta funcion borra una variable de plantilla (eso puede suceder cuando se modifica el contenido de una plantilla) Tambien elimina las posibles sustituciones que tenga en las páginas de la plantilla.

Integer krond_variable::get_id() *[line 2054]*

Summary: Recupera el identificador de la Variable.
El identificador de la Variable es un entero y que se utiliza como clave primaria en la tabla 'krond_variables'.

String krond_variable::get_nombre() *[line 2066]*

Summary: Devuelve el nombre de la Variable.

El nombre de la Variable se recupera de la tabla 'krond_variables' del campo 'nombre_variable'.

String krond_variable::strip_llaves() *[line 2086]*

Summary: Sustituye las llaves '{', '}' del nombre de la variable por su correspondiente entidad HTML.

Las variables empiezan por '{' y acaban con '}'. Útil para evitar sustituciones inadecuadas en el contenido de de página.

Package phpkrond.modulos.descargas Procedural Elements

descargas.inc.php

* **Package** phpkrond.modulos.descargas

_MOD_DESCARGAS_CLASS = 1 *[line 31]*

Package phpkrond.modulos.descargas
Classes

Class descargas_admin

[line 1061]

Summary: Clase que administra las descargas del modulo.
Permite realizar todas las operaciones de las categorías y los ficheros de las descargas.

* **Package** phpkrond.modulos.descargas
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor descargas_admin::descargas_admin([$operacion = ""]) *[line 1069]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase de administración de las descargas.
Realiza la operación indicada por el parámetro.

String descargas_admin::form_cabecera() *[line 1208]*

Summary: Cabecera del Módulo de Administración de Descargas
Muestra una cabecera con el título de Administración del Módulo de Descargas.

String descargas_admin::form_categoria_crear([$idCat = 0]) *[line 1517]*

Function Parameters:

* *$idCat* **$idCat** Identificador de la categoria padre.

Summary: Muestra el formulario para crear una nueva categoria.
Si la categoria padre $idCat no existe, la nueva categoria se creara dentro de la categoria raiz /.

String descargas_admin::form_categoria_propiedades([$idCat = 0]) *[line 1443]*

Function Parameters:

* *$idCat* **$idCat** Identificador de la categoria.

Summary: Muestra el formulario con las propiedades de una categoría.
Si la categoria es la raiz, muestra las variables de configuración del modulo de descargas que contienen dichas propiedades.

descargas_admin::form_config() *[line 1387]*

Summary: Muestra el formulario de configuración del Módulo.

Desde estr formulario se pueden cambiar las variables de configuración 'nombre_descargas' y 'descripcion_descargas' utilizadas para el nombre y la descripción de la categoria '/' del módulo de descargas.

String descargas_admin::form_fichero_crear_http([$idCat = 0]) *[line 1661]*

Function Parameters:

* *$idCat* **$idCat** Identificador de la categoria del fichero.

Summary: Muestra el formulario para crear un fichero de descarga.

El fichero se sube utilizando HTTP en lugar de subirse de forma manual. El programa se encarga de todas las la tareas en lugar del usuario.

String descargas_admin::form_fichero_crear_manual([$idCat = 0]) *[line 1738]*

Function Parameters:

* *$idCat* **$idCat** Identificador de la categoria del fichero.

Summary: Muestra el formulario para crear un fichero de descarga.

El fichero se sube de forma manual y es responsiblidad del usuario subirla a la ruta especificada en el formulario.

String descargas_admin::form_fichero_propiedades($idFichero) *[line 1574]*

Function Parameters:

* *$idFichero* **$idFichero** Identificador del fichero.

Summary: Muestra el formulario con las propiedades de fichero.

El formulario permite modificar las propiedades del mismo.

String descargas_admin::form_general([$idCat = 0]) *[line 1225]*

Function Parameters:

* *$idCat* **$idCat** Identificador de Categoria

Summary: Muestra el formulario general de la administración de descargas.

Este formulario muestra el contenido de una categoria de descargas (subcategorias y ficheros), un subformulario para cambiar propiedades de la categoria, añadir categorias y añadir ficheros de descargas.

String descargas_admin::select_categorias([$catInicio = 0], [$catMarcar = 0], [$separacion = "   "]) *[line 1810]*

Function Parameters:

* *$catInicio* **$catInicio** Categoria sobre la que se empieza a generar el boton.
* *$catMarcar* **$catMarcar** Categoria que aparecera seleccionada por defecto en el boton.
* *$separacion* **$separacion** Cadena de texto de separación para anidadmientos.

Summary: Genera el contenido html de un boton select con las categorias de descargas.

El boton select generado se puede utilizar luego en diversos formularios para cambiar de categoria padre de un fichero o de una categoria.

Class descargas_categoria

[line 47]

Summary: Clase que gestiona las categorias de descargas.

Las descargas de ficheros se organizan en diferentes categorias para permitir una organización. Una categoría de descarga puede a su vez contener otras categorías. La información de una categoría de descarga se encuentra en la tabla 'descargas_categorias'.

* **Package** phpkrond.modulos.descargas
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed descargas_categoria::$db = *[line 67]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed descargas_categoria::$idCategoria = *[line 56]*

Summary: Atributo de la clase que guarda la clave de la categoria.

El atributo de la clase $idCategoria es el identificador utilizado como clave primaria en la tabla 'descargas_categorias'.

Constructor descargas_categoria::descargas_categoria([$idCategoria = 0]) *[line 78]*

Function Parameters:

* *$idCategoria* **$idCategoria** Identificador de la categoría.

Summary: Constructor de la clase que inicializa el atributo $idCategoria.

El atributo de la clase $idCategoria es el identificador utilizado como clave primaria en la tabla 'descargas_categorias'.

boolean descargas_categoria::borrar($recursivo) *[line 344]*

Function Parameters:

* *$recursivo* **$recursivo** Indica si se borra de forma recursiva (=true) o no.

Summary: Borra un categoria de descargas.

Elimina la entrada de la categoria de descargas que se encuentra en la tabla

'descargas_categorias'. Existen dos modos de funcionamiento de esta funcion. Borrar de forma recursiva para cuando la categoria tiene subcategorias y ficheros y borrar de forma no recursiva que solo borrara la categoria si esta vacia (carece de ficheros y no tiene subcategorias).

boolean descargas_categoria::check_datos($datos) *[line 327]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el nombre del fichero y la descripción.

descargas_categoria::crear($nombre, $descripcion, $idCatPadre) *[line 299]*

Function Parameters:

* *$nombre* **$nombre** Nombre de la categoria.
* *$descripcion* **$descripcion** Descripción de la categoria.
* *$idCatPadre* **$idCatPadre** Identificador de la categoria padre.

Summary: Crea una nueva categoria de descargas.

Crea una nueva entrada en la tabla 'descargas_categorias' inicializando todos los campos con los datos de la nueva categoría. Al crearla la colaca en la última posición de las categorias de $idCatPadre.

boolean descargas_categoria::existe() *[line 273]*

Summary: Verifica si la categoria existe.

Comprueba que el identificador de categoria es correcto.

Array descargas_categoria::get_categorias() *[line 210]*

Summary: Devuelve las categoría (subcategorias) de la categoría.

Una categoria puede tener a su vez diversa categorías

* subcategorias dentro de ella. Devuelve un array de
* bjetos descargas_categoria.

String descargas_categoria::get_descripcion() *[line 162]*

Summary: Devuelve la descripción de la categoría.

La descripción de la categoría se encuentra guardado en la tabla 'descargas_categorias' en el campo 'desc_categoria'. Para la categoria de id = 0, la descripción se encuentra en la configuración del modulo, en la variable 'descripcion_descargas' dentro de la tabla

'descargas_config'.

Array descargas_categoria::get_ficheros() *[line 233]*

Summary: Devuelve los ficheros de la categoría.

Una categoria contiene diversos ficheros para descargas. Con esto se ofrece una forma de mantener un orden lógico en los ficheros que se descargan del sistema. Devuelve un array de objetos descargas_fichero.

Integer descargas_categoria::get_id() *[line 96]*

Summary: Devuelve el identificador de categoria.

El identificador de categoria es la clave primaria de la tabla 'descargas_categorias' que contiene la información de una categoría de descarga. Existe una categoria, padre de todas las demas y cuyo identificador es '0'. La información de esa categoria 'nombre' y 'descripción' se almacena en la configuración de este módulo.

String descargas_categoria::get_nombre() *[line 135]*

Summary: Devuelve el nombre de la categoría.

El nombre de la categoría se encuentra guardado en la tabla 'descargas_categorias' en el campo 'nombre_categoria'. Para la categoria de id = 0, el nombre se encuentra en la configuración del modulo, en la variable 'nombre_descargas' dentro de la tabla 'descargas_config'.

integer descargas_categoria::get_num_ficheros() *[line 253]*

Summary: Devuelve el numero de ficheros que estan en la categoria.

Si la categoria tiene subcategorias, calcula el numero de ficheros de las subcaterias y se los suma.

Integer descargas_categoria::get_posicion() *[line 188]*

Summary: Devuelve la posición de la categoría.

La posición de la categoría se encuentra guardada en la tabla 'descargas_categorias' en el campo 'posicion_categoria'. Está posición sirve para mantener ordenadas las categorias (subCategorias) dentro de una categoria.

Integer descargas_categoria::get_prev() *[line 111]*

Summary: Devuelve la categoria previa (padre) de la categoria actual.

Las categorias de descargas pueden contener otra categorias (subcategorias). Esta información se guarda en la tabla 'categoria_descargas' dentro del campo 'prev_categoria' y es el identificador de dicha categoria.

boolean descargas_categoria::intercambiar($idCat2) *[line 425]*

Function Parameters:

* *$idCat2* **$idCat2** Identificador de categoria con la que se intercambia la posicion.

Summary: Intercambia las posicion de dos categorias.
Las categorias que se intercambian deben pertenecer a la misma categoria padre.

Boolean descargas_categoria::set($nombre, $descripcion) *[line 403]*

Function Parameters:

* *$nombre* **$nombre** Nuevo nombre de la categoria.
* *$descripcion* **$descripcion** Nueva descripcion de la categoria.

Summary: Modifica las propiedades de una categoria.
Las propiedades que modifica son el nombre y la descriptcion que se encuentran en la tabla 'descargas_categorias'. Comprueba que los nuevos datos son correctos.

Class descargas_config

[line 1845]

Summary: Clase que gestiona las variables de configuración del módulo.
Las variables de configuración se encuentran almacenadas en la tabla 'descargas_config'.

* **Package** phpkrond.modulos.descargas
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed descargas_config::$db = *[line 1855]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor descargas_config::descargas_config() *[line 1863]*

Summary: Contructor de la clase.
Inicializa el atributo $this->db que contiene el enlace con la base de datos.

String descargas_config::get_variable($nombreVar) *[line 1877]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Recupera el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'descargas_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

descargas_config::set_variable($nombreVar, $nuevoValor) *[line 1896]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable a modificar.
* *$nuevoValor* **$nuevoValor** Nuevo valor de la variable.

Summary: Modifica el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'descargas_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

Class descargas_error

[line 1912]

Summary: Clase para mostrar mensajes del modulo descargas.
Todos los mensajes del modulo se consideran mensajes de error.

* **Package** phpkrond.modulos.descargas
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor descargas_error::descargas_error([$tituloError = "phpKROND - Error "], [$msgError = ""]) *[line 1921]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class descargas_fichero

[line 458]

Summary: Clase que gestiona los ficheros de las descargas.

Las operaciones típicas que se pueden encontrar son crear (subir un fichero), modificar propiedades, borrar y descargas el fichero. La información de un fichero de descarga se encuentra en la tabla 'descargas_ficheros'.

* **Package** phpkrond.modulos.descargas
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed descargas_fichero::$db = *[line 477]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed descargas_fichero::$idFichero = *[line 466]*

Summary: Atributo de la clase que guarda la clave del fichero.

El atributo de la clase $idFichero es el identificador utilizado como clave primaria en la tabla 'descargas_ficheros'.

Constructor descargas_fichero::descargas_fichero([$idFichero = 0]) *[line 487]*

Function Parameters:

* *$idFichero* **$idFichero** Identificador del fichero.

Summary: Constructor de la clase que inicializa el atributo $idFichero.

El atributo de la clase $idFichero es el identificador utilizado como clave primaria en la tabla 'descargas_ficheros'.

boolean descargas_fichero::borrar($unlink) *[line 720]*

Function Parameters:

* *$unlink* **$unlink** Si es 1 elimina el fichero fisicamente, en otro caso lo deja en el sistema, pero elimina la entrada en la bd.

Summary: Borra el fichero

Elimina la entrada en la tabla 'descargas_ficheros' del fichero. Tambien puede borrar el

fichero fisicamente del sistema.

descargas_fichero::crear($idCat, $nombreFichero, $descFichero, $rutaFichero) *[line 686]*

Function Parameters:

- *$idCat* **$idCat** Categoria a la que pertenece el fichero.
- *$nombreFichero* **$nombreFichero** Nombre del Fichero a descargar.
- *$descFichero* **$descFichero** Descripción del Fichero.
- *$rutaFichero* **$rutaFichero** Ruta del fichero en el servidor.

Summary: Crea un nuevo fichero a descargar.

Añade una nueva entrada en la tabla 'descargas_ficheros' con toda la información del fichero. Las campos que guardan la fecha, tamaño, posicion y contador de descargas son inicializados de forma automatica.

descargas_fichero::download() *[line 827]*

Summary: Descarga el fichero por http.

Lee el contenido del fichero de su ruta original y lo envia al cliente. Luego incrementa el contador de descarga del fichero.

boolean descargas_fichero::existe() *[line 663]*

Summary: Verifica si el fichero existe.

Comprueba que el identificador del fichero es correcto.

descarga_categoria descargas_fichero::get_categoria() *[line 609]*

Summary: Devuelve la categoría del fichero.

Todos los ficheros pertenecen a una categoria. La información de la categoría a la que pertenece se se encuentra en el campo 'id_categoria' de la tabla 'descargas_fichero'.

Integer descargas_fichero::get_contador() *[line 569]*

Summary: Devuelve el contador del numero de descargas del fichero.

El contador del númeor de descargas del fichero se encuentra guardado en la tabla 'descargas_ficheros' en el campo 'contador_descarga'.

String descargas_fichero::get_descripcion() *[line 532]*

Summary: Devuelve la descripción del fichero.

La descripción del fichero se encuentra guardada en la tabla 'descargas_ficheros' en el campo 'desc_fichero'.

String descargas_fichero::get_fecha() *[line 590]*

Summary: Devuelve la Fecha de Subida del del fichero.

La fecha de subida del fichero se encuentra guardada en la tabla 'descargas_ficheros' en el campo 'fecha_fichero'. Esta fecha informa del momento en que añadio la descarga al sistema. La fecha se guarda en formado 'AAAA-MM-DD HH:MM:SS' (año:mes:dia hora:minuto:segundo).

Integer descargas_fichero::get_id() *[line 501]*

Summary: Devuelve el identificador del Fichero.

El identificador de Fichero es la clave primaria de la tabla 'descargas_ficheros' que contiene la información de un fichero de descarga.

String descargas_fichero::get_nombre() *[line 514]*

Summary: Devuelve el nombre del fichero.

El nombre del fichero se encuentra guardado en la tabla 'descargas_ficheros' en el campo 'nombre_fichero'.

Integer descargas_fichero::get_posicion() *[line 551]*

Summary: Devuelve la posición del fichero.

La posición del fichero se encuentra guardada en la tabla 'descargas_ficheros' en el campo 'posicion_fichero'. Es una forma de mantener ordenados los ficheros en una categoria.

Integer descargas_fichero::get_ruta() *[line 647]*

Summary: Devuelve la ruta en el sistema de ficheros del fichero.

La ruta del fichero se encuentra en el campo 'ruta_fichero' dentro de la tabla 'descargas_ficheros'. La ruta del fichero no tiene nada que ver con la cree ver el usuario que descarga el fichero.

Integer descargas_fichero::get_size() *[line 628]*

Summary: Devuelve el tamaño del fichero en bytes.

El tamaño del fichero se encuentra en el campo 'tam_fichero' dentro de la tabla 'descarga_ficheros'. Este tamaño se calcula en el momento de crear la descarga en el sistema.

descargas_fichero::inc_contador() *[line 815]*

Summary: Incrementa el contador de descargas del fichero.

El contador de descargas del fichero se incrementa en 1 cada vez que es descargado el fichero. Ese contador se encuentra en el campo 'contador_descargar' de la tabla 'descargas_ficheros'.

boolean descargas_fichero::intercambiar($idFichero2) *[line 786]*

Function Parameters:

* *$idFich2* **$idFichero2** Identificador del fichero con el que se intercambia la posicion.

Summary: Intercambia las posicion de dos ficheros.
Las ficheros que se intercambian deben pertenecer a la misma categoria padre.

boolean descargas_fichero::set($idCat, $nombre, $descripcion, $contador) *[line 749]*

Function Parameters:

* *$idCat* **$idCat** Identificador de la nueva categoria a la que pertenecera el fichero.
* *$nombre* **$nombre** Nuevo nombre del fichero.
* *$descripcion* **$descripcion** Nueva descripcion del fichero.
* *$contador* **$contador** Nuevo valor del contador de descargas.

Summary: Modifica las propiedades del fichero.
Las propiedades del fichero que modifican son El nombre que utiliza (no el nombre real que tiene en el sistema de fichero), la descripción, contador de descargas y categoria a la que pertenece.

boolean descargas_fichero::verifica($idCat, $nombre, $descripcion, $ruta) *[line 860]*

Function Parameters:

* *$idCat* **$idCat** Identificador de Categoria del fichero.
* *$nombre* **$nombre** Nombre del fichero
* *$descripcion* **$descripcion** Descripción del Fichero.
* *$ruta* **$ruta** Ruta del fichero en el servidor.

Summary: Comprueba que los datos del fichero son correctos.
Los datos que verifica son el identificador de categoria y que el nombre, la descripcion y la ruta del fichero son válidos.

Class descargas_mostrar

[line 878]

Summary: Clase que muestras las descargas del modulo.

Muestra los categorias y los ficheros que tienen. Permitiendo navegar por ellos como si de un sistema de ficheros se tratase.

* **Package** phpkrond.modulos.descargas
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor descargas_mostrar::descargas_mostrar([$operacion = ""]) *[line 886]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase para mostrar las descargas.

Realiza la operación indicada por el parametro.

String descargas_mostrar::get_ruta([$idCat = 0]) *[line 1036]*

Summary: Devuelve la ruta completa de una categoria a partir de la ruta raiz.

Cómo las categorías pueden anidarse, esto devuelve la ruta completa desde la categoría raiz hasta la categoria actual.

String descargas_mostrar::mostrar([$idCat = 0]) *[line 918]*

Function Parameters:

* *$idCat* **$idCat** Identificador de Categoria a mostrar.

Summary: Muestra la categoria de descarga.

Muestra las subcategorias de los ficheros de descargas. En principio estan ordenados por el atributo posición.

Package phpkrond.modulos.forum
Procedural Elements

forum.inc.php

* **Package** phpkrond.modulos.forum

_MOD_FORUM_CLASS = 1 *[line 31]*

require_once("modulos/users/users.inc.php") *[line 34]*

Package phpkrond.modulos.forum Classes

Class forum_admin

[line 1424]

Summary: Clase que administra el Módulo del Forum.

Permite manipular la configuración y los temas y foros presentes en el forum. Tambien puede llevar a cabo labores de control de los mensajes y respuestas de los foros.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor forum_admin::forum_admin([$operacion = ""]) *[line 1432]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase de administración del Forum.

Realiza la operación indicada por el parámetro.

String forum_admin::form_cabecera() *[line 1563]*

Summary: Cabecera del Módulo de Administración del Forum

Muestra una cabecera con el título de Administración del Módulo del Forum.

String forum_admin::form_config() *[line 1579]*

Summary: Muestra el formulario de configuración del Módulo.

Desde este formulario se pueden cambiar las variables de configuración 'titulo_forum', 'permitir_anonimo' y 'nombre_user_anonimo'. Estas variables se guardan en la tabla 'forum_config'.

String forum_admin::form_foro_mostrar($idForo) *[line 1968]*

Summary: Formulario que muestra el contenido del Foro.

Muestra los mensajes que aparecen el foro y ademas ofrece las operaciones básicas de mantenimiento del foro.

String forum_admin::form_foro_nuevo() *[line 1813]*

Summary: Formulario crear un numero Foro.

Muestra el formulario para crear un nuevo Foro. Recoge el titulo del Foro, una descripción del mismo y a que tema va a pertenecer.

String forum_admin::form_foro_propiedades($idForo) *[line 1886]*

Summary: Formulario que muestra las propiedades del Foro.

Muestra un formulario con las propiedades actuales del Foro y ofrece la posibilidad de cambiarlas. Permite cambiar el titulo, descripción y tema al que pertenece el foro.

String forum_admin::form_general() *[line 1642]*

Summary: Muestra el formulario general del Módulo

En este formulario se muestra todas los temas y los foros que estan dentro de ellos. Ofrece la posibilidad de realizar las operaciones de mantenimiento necesarios del Forum.

String forum_admin::form_mensaje_mostrar($idMensaje) *[line 2058]*

Summary: Formulario que muestra el contenido de un Mensaje

Muestra el mensaje y las respuestas a el mismo, ademas ofrece las operaciones básicas de mantenimiento del foro.

String forum_admin::form_tema_nuevo() *[line 1724]*

Summary: Formulario crear un numero Tema.

Muestra el formulario para crear un nuevo Tema. Recoge el Titulo del Tema.

String forum_admin::form_tema_propiedades($idTema) *[line 1764]*

Summary: Formulario que muestra las propiedades del Tema.

Muestra un formulario con las propieadades actuales del Tema y ofrece la posibilidad de cambiarlas.

Class forum_config

[line 2689]

Summary: Clase que gestiona las variables de configuración del módulo.
Las variables de configuración se encuentran almacenadas en la tabla 'forum_config'.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed forum_config::$db = *[line 2699]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor forum_config::forum_config() *[line 2707]*

Summary: Contructor de la clase.
Inicializa el atributo $this->db que contiene el enlace con la base de datos.

String forum_config::get_variable($nombreVar) *[line 2721]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Recupera el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'forum_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

forum_config::set_variable($nombreVar, $nuevoValor) *[line 2740]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable a modificar.
* *$nuevoValor* **$nuevoValor** Nuevo valor de la variable.

Summary: Modifica el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'forum_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

Class forum_error

[line 2756]

Summary: Clase para mostrar mensajes del modulo del Forum.
Todos los mensajes del modulo se consideran mensajes de error.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor forum_error::forum_error([$tituloError = "phpKROND - Error "], [$msgError = ""]) *[line 2765]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class forum_foro

[line 261]

Summary: Clase que controla los foros presentes en el sistema.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed forum_foro::$db = *[line 281]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed forum_foro::$idForo = *[line 270]*

Summary: Atributo de la clase que guarda la clave del foro.

El atributo de la clase $idForo es el identificador utilizado como clave primaria en la tabla 'forum_foros'.

Constructor forum_foro::forum_foro([$idForo = 0]) *[line 296]*

Function Parameters:

* *$idForo* **$idForo** Identificador del Foro.

Summary: Contructor de la clase para gestionar los Foros del Forum

El Módulo del Forum contiene diversos Foros que se agrupan en distintos Temas. La información sobre los Foros se guarda en la tabla 'forum_foros'. La clave primaria de dicha tabla es el identificador del Foro. El contructor inicializa el atributo del objeto, ademas de el atributo que contiene la conexión con la base de datos.

forum_foro::actualiza() *[line 659]*

Summary: Actualiza fecha del Foro con la fecha actual.

La fecha del foro se actualiza con cualquier operación de borrado/creación de mensajes/respuestas del foro.

Boolean forum_foro::borrar() *[line 532]*

Summary: Borra el Foro del tema.

Elimina la entrada en la tabla 'forum_foros' del Foro Borra tambien los mensajes y respuestas del Foro.

Boolean forum_foro::crear($idTema, $titulo, $descripcion) *[line 510]*

Function Parameters:

* *$idTema* **$idTema** Identificador del Tema del Foro.
* *$titulo* **$titulo** Titulo del nuevo Foro.
* *$descripcion* **$descripcion** Descripcion del Foro.

Summary: Crea un nuevo Foro.

Añade un nuevo Foro en el Forum, insertando una nueva entrada en la tabla 'forum_foros'.

forum_foro::dec_mensajes() *[line 619]*

Summary: Decrementa contador de mensajes del foro.

El contador de mensaje se encuentra en el campo 'num_mensajes' en la tabla 'forum_foros'. Es una optimización para calcular el número de mensajes más rapidamente.

forum_foro::dec_respuestas() *[line 647]*

Summary: Decrementa contador de respuestas del foro.

El contador de respuestas se encuentra en el campo 'num_respuestas' en la tabla 'forum_foros'. Es una optimización para calcular el número de respuestas más rapidamente.

Boolean forum_foro::existe() *[line 489]*

Summary: Devuelve un booleano indicando si el foro existe.

Esta función es útil para realizar otras operaciones con los foros (borrar, modificar).

String forum_foro::fecha_calcula_dif() *[line 375]*

Summary: Devuelve la diferencia entre la fecha de modificación del foro y la fecha actual.

La diferencia la devuelve formateado en dias, horas y minutos.

String forum_foro::get_descripcion() *[line 338]*

Summary: Devuelve la Descripción del Foro.

La descripción del foro se encuentra en el campo 'desc_foro' de la tabla 'forum_foros'.

String forum_foro::get_fecha() *[line 359]*

Summary: Devuelve la Fecha de la última modificación del foro.

La fecha de la última modificación del foro se encuentra en el campo 'fecha_foro' de la tabla 'forum_foros'. Esta fecha se actualiza con cualquier nuevo mensaje/respuesta en el foro. Se encuentra en formato 'AAAA-MM-DD HH:MM:SS'

Integer forum_foro::get_id() *[line 309]*

Summary: Devuelve el Identificador del Foro.

El identificador del Foro es utilizado como clave primaria en la tabla 'forum_foros'.

Array forum_foro::get_mensajes() *[line 403]*

Summary: Devuelve un array de objetos con todos los mensajes del foro.

El array de objetos que devuelve es de tipo 'forum_mensaje'. Los mensajes se obtienen ordenador por orden decreciente de fecha. Primero aparecen los enviados más recientemente.

Integer forum_foro::get_num_mensajes() *[line 429]*

Summary: Devuelve el número de mensajes del Foro.

El número de mensajes del foro se encuentra en el campo 'num_mensajes' de la tabla 'forum_foros'. Se guarda este dato para mejorar el rendimiento a la hora de calcular los mensajes que existen en el foro. Eso obliga a que operaciones de borrado de mensajes o creación de nuevos mensajes decrementen/incremente dicho valor.

Integer forum_foro::get_num_respuestas() *[line 454]*

Summary: Devuelve el número de respuestas del Foro.

El número de respuestas del foro se encuentra en el campo 'num_respuestas' de la tabla 'forum_foros'. Se guarda este dato para mejorar el rendimiento a la hora de calcular las respuestas que existen en el foro. Eso obliga a que operaciones de borrado de respuestas o creación de nuevas respuestas decrementen/incremente dicho valor. (incluso hay que tenerlo en cuenta cuando se borra un mensaje).

forum_tema forum_foro::get_tema() *[line 472]*

Summary: Devuelve el Tema al que pertence el Foro.

El Tema al que pertenece el foro se encuentra en el campo 'id_tema' de la tabla 'forum_foros'.

String forum_foro::get_titulo() *[line 321]*

Summary: Devuelve el Titulo del foro.

El titulo del Foro se encuentra en el campo 'titulo_foro' de la tabla 'forum_foros'.

forum_foro::inc_mensajes() *[line 605]*

Summary: Incrementa contador de mensajes del foro.

El contador de mensaje se encuentra en el campo 'num_mensajes' en la tabla 'forum_foros'. Es una optimización para calcular el número de mensajes más rapidamente.

forum_foro::inc_respuestas() *[line 633]*

Summary: Incrementa contador de respuestas del foro.

El contador de respuestas se encuentra en el campo 'num_respuestas' en la tabla 'forum_foros'. Es una optimización para calcular el número de respuestas más rapidamente.

Boolean forum_foro::set($titulo, $descripcion, $idTema) *[line 583]*

Function Parameters:

* *$titulo* **$titulo** Nuevo titulo del Foro.
* *$descripcion* **$descripcion** Nueva descripcion del Foro.
* *$idTema* **$idTema** Nuevo Identificador del Tema del Foro.

Summary: Cambia los atributos del Foro.

Los atributos que cambia estan almacenados en la la tabla 'forum_foro'. Permite modificar el titulo, descripcion y el tema al que pertenece.

Boolean forum_foro::vaciar() *[line 557]*

Summary: Elimina los mensajes (y sus repuestas) del foro.

Al eliminar todos los mensajes del foro se tienen que actualizar los contadores del numero de mensajes y del numero de respuestas.

Class forum_mensaje

[line 673]

Summary: Clase que gestiona los mensajes de los usuarios a los foros.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed forum_mensaje::$db = *[line 693]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed forum_mensaje::$idMensaje = *[line 682]*

Summary: Atributo de la clase que guarda la clave del mensaje.

El atributo de la clase $idMensaje es el identificador utilizado como clave primaria en la tabla 'forum_mensajes'.

Constructor forum_mensaje::forum_mensaje([$idMensaje = 0]) *[line 706]*

Function Parameters:

* *$idMensaje* **$idMensaje** Identificador del Mensaje.

Summary: Contructor de la clase para gestionar los Mensajes en los Foros.

La información sobre los Mensajes se guarda en la tabla 'forum_mensajes'. La clave primaria de dicha tabla es el identificador del Mensaje. El contructor inicializa el atributo del objeto, ademas de el atributo que contiene la conexión con la base de datos.

forum_mensaje::actualiza() *[line 1115]*

Summary: Actualiza fecha del Mensaje con la fecha actual.

La fecha del mensaje se actualiza con cualquier operación de borrado/creación de respuestas del mensaje.

Boolean forum_mensaje::borrar() *[line 1014]*

Summary: Borra el Mensaje del Foro.

Elimina la entrada en la tabla 'forum_mensaje' del Mensaje Borra tambien las respuestas del mensaje. Decrementa el contador de mensajes del foro.

Array forum_mensaje::buscar($texto) *[line 1135]*

Function Parameters:

* *$texto* **$texto** Texto a buscar en el mensaje.

Summary: Busca entre los mensajes el texto especificado.
La busqueda se puede limitar a varios campos. Búsqueda por autor, titulo o contenido. Pudiendo combinar ambos tipos de búsquedas.

Boolean forum_mensaje::crear($idForo, $titulo, $contenido, $autor) *[line 965]*

Function Parameters:

* *$idForo* **$idForo** Identificador del Foro.
* *$titulo* **$titulo** Titulo del nuevo Mensaje.
* *$contenido* **$contenido** Contenido del nuevo Mensaje.
* *$autor* **$autor** Nombre del Autor del Mensaje.

Summary: Crea un nuevo Mensaje.
Añade un nuevo Mensajeen el Foro, insertando una nueva entrada en la tabla 'forum_mensajes'.

forum_mensaje::dec_respuestas() *[line 1099]*

Summary: Decrementa contador de respuestas del mensaje.
El contador de respuestas se encuentra en el campo 'num_respuestas' en la tabla 'forum_mensaje'. Es una optimización para calcular el número de respuestas más rapidamente.

Boolean forum_mensaje::existe() *[line 943]*

Summary: Devuelve un booleano indicando si el mensaje existe.
Esta función es útil para realizar otras operaciones con los mensaje (borrar, modificar).

String forum_mensaje::fecha_calcula_dif() *[line 783]*

Summary: Devuelve la diferencia entre la fecha de modificación del mensaje y la fecha actual.
La diferencia la devuelve formateado en dias, horas y minutos.

String forum_mensaje::get_autor() *[line 832]*

Summary: Devuelve el nombre del autor del mensaje.
La nombre del autor del mensaje se encuentra en el campo 'autor_mensaje' de la tabla 'forum_mensajes'.

String forum_mensaje::get_contenido() *[line 748]*

Summary: Devuelve el Contenido del Mensaje.

El contenido del Mensaje se encuentra en el campo 'contenido_mensaje' de la tabla 'forum_mensajes'.

String forum_mensaje::get_fecha() *[line 767]*

Summary: Devuelve la Fecha de envio del mensaje.

La fecha de la creación del mensaje se encuentra en el campo 'fecha_mensaje' de la tabla 'forum_mensajes'. Se encuentra en formato 'AAAA-MM-DD HH:MM:SS'

forum_foro forum_mensaje::get_foro() *[line 904]*

Summary: Devuelve el foro al que pertence el mensaje.

El foro al que pertenece el mensaje se encuentra en el campo 'id_foro' de la tabla 'forum_mensajes'.

Integer forum_mensaje::get_id() *[line 719]*

Summary: Devuelve el Identificador del Mensaje.

El identificador del Mensaje es utilizado como clave primaria en la tabla 'forum_mensajes'.

String forum_mensaje::get_ip() *[line 850]*

Summary: Devuelve la ip del autor del mensaje.

La IP del autor del mensaje se encuentra en el campo 'ip_autor' de la tabla 'forum_mensajes'.

String forum_mensaje::get_modificado() *[line 814]*

Summary: Devuelve la Fecha de modificación del mensaje.

La fecha de la modificación del mensaje se encuentra en el campo 'fecha_modificado' de la tabla 'forum_mensajes'. Se encuentra en formato 'AAAA-MM-DD HH:MM:SS'. Esta fecha se actualiza si hay respuesta al mensaje aunque en principio coincide con la fecha de creación del mensaje.

Integer forum_mensaje::get_num_lecturas() *[line 868]*

Summary: Devuelve el número de lecturas del mensaje.

El número de lecturas del mensaje se encuentra en el campo 'num_lecturas' de la tabla 'forum_mensajes'.

Integer forum_mensaje::get_num_respuestas() *[line 886]*

Summary: Devuelve el número de respuestas del mensaje.

El número de respuestas del mensaje se encuentra en el campo 'num_respuestas' de la tabla 'forum_mensajes'.

Array forum_mensaje::get_respuestas() *[line 923]*

Summary: Devuelve un array de objetos con todos las respuestas del mensaje.

El array de objetos que devuelve es de tipo 'forum_respuesta'. Las Respuestas se obtienen ordenadas fecha en orden ascendente (primero aparecen las primeras respuestas).

String forum_mensaje::get_titulo() *[line 731]*

Summary: Devuelve el Titulo del Mensaje.

El titulo del Mensaje se encuentra en el campo 'titulo_mensaje' de la tabla 'forum_mensajes'.

forum_mensaje::inc_lecturas() *[line 1064]*

Summary: Incrementa contador de lecturas del mensaje.

El contador de lecturas se encuentra en el campo 'num_lecturas' en la tabla 'forum_mensaje'.

forum_mensaje::inc_respuestas() *[line 1081]*

Summary: Incrementa contador de respuestas del mensaje.

El contador de respuestas se encuentra en el campo 'num_respuestas' en la tabla 'forum_mensaje'. Es una optimización para calcular el número de respuestas más rapidamente.

String forum_mensaje::obtiene_ip() *[line 996]*

Summary: Obtiene la Ip del Cliente.

Obtiene la IP real del cliente.

Boolean forum_mensaje::vaciar() *[line 1043]*

Summary: Vacia las respuestas del Mensaje del Foro.

Elimina todas las respuestas que se han enviado al foro en contestación al mensaje. El mensaje permanece.

Class forum_mostrar

[line 2135]

Summary: Clase que Muestra el Módulo del Forum.
Muestra los foros y los contenidos de los mismos, mensajes de los usuarios y respuestas. Los foros estan organizados en Temas.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor forum_mostrar::forum_mostrar([$operacion = ""]) *[line 2143]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase que muestra el Forum.
Realiza la operación indicada por el parámetro.

String forum_mostrar::form_buscar() *[line 2675]*

Summary: Muestra la formulario para busca entre los mensajes y respuestas.
El formulario recoge los campos que se usuran en la busqueda y el tipo de busqueda que se realizara.

String forum_mostrar::form_foro_mostrar($idForo) *[line 2292]*

Function Parameters:

* *$idForo* **$idForo** Identificador del Foro a mostrar

Summary: Muestra la formulario principal de un Foro.
Lista los mensajes enviados al foro.

String forum_mostrar::form_mensaje_mostrar($idMensaje) *[line 2476]*

Function Parameters:

* *$idMensaje* **$idMensaje** Identificador del Mensaje

Summary: Devuelve el formulario que muestra un mensaje.
El formulario lista el contenido del mensaje y las respuestas que se han enviado en

contestación de dicho mensaje.

String forum_mostrar::form_mensaje_nuevo($idForo) *[line 2399]*

Function Parameters:

* *$idForo* **$idForo** Identificador del Foro donde se enviará el nuevo mensaje.

Summary: Muestra la formulario para enviar un nuevo Mensaje.
El formulario recoge los campos que el usario rellenara: autor, titulo, contenido (y foro).

String forum_mostrar::form_mostrar() *[line 2216]*

Summary: Muestra la formulario principal del Forum
Este formulario lista los Temas y Foros que estan en el Forum.

String forum_mostrar::form_respuesta_nuevo($idMensaje) *[line 2598]*

Function Parameters:

* *$idMensaje* **$idMensaje** Identificador del Mensaje que se contesta.

Summary: Muestra la formulario para enviar una nueva Respuesta.
El formulario recoge los campos que el usuario rellenara: autor, titulo, contenido (y foro).

Class forum_respuesta

[line 1146]

Summary: Clase que controla las respuestas de los usuarios a los foros.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed forum_respuesta::$db = *[line 1166]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed forum_respuesta::$idRespuesta = *[line 1155]*

Summary: Atributo de la clase que guarda la clave de la respuestas.

El atributo de la clase $idRespuesta es el identificador utilizado como clave primaria en la tabla 'forum_respuestas'.

Constructor forum_respuesta::forum_respuesta([$idRespuesta = 0]) *[line 1179]*

Function Parameters:

* *$idRespuesta* **$idRespuesta** Identificador de la Respuesta.

Summary: Contructor de la clase para gestionar las Respuestas a los Mensajes.

La información sobre las Respuestas se guarda en la tabla 'forum_mensajes'. La clave primaria de dicha tabla es el identificador de la Respuesta. El contructor inicializa el atributo del objeto, ademas de el atributo que contiene la conexión con la base de datos.

Boolean forum_respuesta::borrar() *[line 1331]*

Summary: Borra la Respuesta del Mensaje en el Foro.

Elimina la entrada en la tabla 'forum_respuesta' del Mensaje Decrementa el contador de respuestas del mensaje y del foro.

Array forum_respuesta::buscar($texto) *[line 1408]*

Function Parameters:

* *$texto* **$texto** Texto a buscar en la respuesta.

Summary: Busca entre las respuestas el texto especificado.
La busqueda se puede limitar a varios campos. Búsqueda por autor, titulo o contenido. Pudiendo combinar ambos tipos de búsquedas.

Boolean forum_respuesta::crear($idMensaje, $titulo, $contenido, $autor) *[line 1357]*

Function Parameters:

* *$idMensaje* **$idMensaje** Identificador del Mensaje.
* *$titulo* **$titulo** Titulo de la nueva Respuesta.
* *$contenido* **$contenido** Contenido de la nueva Respuesta.
* *$autor* **$autor** Nombre del Autor de la Respuesta.

Summary: Crea una nueva Respuesta.
Añade una nuevo Respuesta a el Mensaje, insertando una nueva entrada en la tabla 'forum_respuestas'.

Boolean forum_respuesta::existe() *[line 1312]*

Summary: Devuelve un booleano indicando si la respuesta existe.
Esta función es útil para realizar otras operaciones con las respuestas (borrar, modificar).

String forum_respuesta::get_autor() *[line 1259]*

Summary: Devuelve el nombre del autor de la respuesta.
La nombre del autor de la respuesta se encuentra en el campo 'autor_respuesta' de la tabla 'forum_respuestas'.

String forum_respuesta::get_contenido() *[line 1222]*

Summary: Devuelve el Contenido de la Respuesta.
El contenido de la Respuesta se encuentra en el campo 'contenido_respuesta' de la tabla 'forum_respuestas'.

String forum_respuesta::get_fecha() *[line 1241]*

Summary: Devuelve la Fecha de envio de la respuesta.
La fecha de la creación de la Respuesta se encuentra en el campo 'fecha_respuesta' de la tabla 'forum_respuestas'. Se encuentra en formato 'AAAA-MM-DD HH:MM:SS'

Integer forum_respuesta::get_id() *[line 1192]*

Summary: Devuelve el Identificador de la Respuesta.
El identificador de la Respuesta es utilizado como clave primaria en la tabla 'forum_respuestas'.

String forum_respuesta::get_ip() *[line 1277]*

Summary: Devuelve la ip del autor de la respuesta.

La IP del autor de la respuesta se encuentra en el campo 'ip_autor' de la tabla 'forum_respuestas'.

forum_mensaje forum_respuesta::get_mensaje() *[line 1295]*

Summary: Devuelve el Mensaje al que pertence la respuesta.

El mensaje al que pertenece el mensaje se encuentra en el campo 'id_mensaje' de la tabla 'forum_respuestas'.

String forum_respuesta::get_titulo() *[line 1204]*

Summary: Devuelve el Titulo de la Respuesta.

El titulo de la Respuesta se encuentra en el campo 'titulo_respuesta' de la tabla 'forum_respuestas'.

String forum_respuesta::obtiene_ip() *[line 1389]*

Summary: Obtiene la Ip del Cliente.

Obtiene la IP real del cliente.

Class forum_tema

[line 49]

Summary: Clase que gestiona los diferentes temas de los foros.

Un ejemplo: Tema 'Configuración de Apache' Foros de este tema, 'Apache en win32', 'Apache en Linux' Tema 'Lenguajes de programación' Foros 'C/C++', 'Java', 'PHP', 'Perl', 'Visual C++'.

* **Package** phpkrond.modulos.forum
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed forum_tema::$db = *[line 69]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed forum_tema::$idTema = *[line 58]*

Summary: Atributo de la clase que guarda la clave del tema.

El atributo de la clase $idTema es el identificador utilizado como clave primaria en la tabla 'forum_temas'.

Constructor forum_tema::forum_tema([$idTema = 0]) *[line 83]*

Function Parameters:

* *$idTema* **$idTema** Identificador del Tema.

Summary: Contructor de la clase para gestionar los Temas del Forum

El Módulo del Forum esta compuesto de diversos Temas en donde se agrupan los distintos Foros del módulo. Esta información se guarda en la tabla 'forum_temas'. La clave primaria de dicha tabla es el identificador del tema. El contructor inicializa el atributo del objeto, ademas de el atributo que contiene la conexión con la base de datos.

Boolean forum_tema::borrar() *[line 205]*

Summary: Borra el tema del Forum.

Elimina la entrada en la tabla 'forum_temas' del tema Si el tema no esta vacio no se puede borrar.

Boolean forum_tema::crear($titulo) *[line 185]*

Function Parameters:

* *$titulo* **$titulo** Titulo del nuevo tema.

Summary: Crea un numero tema.
Añade un nuevo tema en el Forum, insertando una nueva entrada en la tabla 'forum_temas'.

Boolean forum_tema::existe() *[line 243]*

Summary: Devuelve un booleano indicando si el tema existe.
Esta función es útil para realizar otras operaciones con los temas (borrar, modificar).

Array forum_tema::get_all() *[line 164]*

Summary: Devuelve un array con todos los temas.
El array contiene objetos forum_tema con todos los temas que existen en el módulo.

String forum_tema::get_fecha() *[line 126]*

Summary: Devuelve la fecha de creación del tema.
La fecha de creación del tema se encuentra en el campo 'fecha_tema' de la tabla 'forum_temas'. La fecha esta en formato 'AAAA-MM-DD HH:MM:SS'

Array forum_tema::get_foros() *[line 144]*

Summary: Devuelve un array de objetos con todos los foros del tema.
Un Tema contiene varios foros. Los foros se obtienen ordenados alfabeticamente en orden ascendente por titulo del foro.

Integer forum_tema::get_id() *[line 96]*

Summary: Devuelve el Identificador del Tema.
El identificador del Tema es utilizado como clave primaria en la tabla 'forum_temas'.

String forum_tema::get_titulo() *[line 108]*

Summary: Devuelve el Titulo del tema.
El titulo del tema se encuentra en el campo 'titulo_tema' de la tabla 'forum_temas'.

Boolean forum_tema::set($titulo) *[line 224]*

Function Parameters:

* *$titulo* **$titulo** Nuevo titulo del tema.

Summary: Cambia el atributo Titulo del tema.
El atributo se almacena en el campo 'titulo_tema' de la tabla 'forum_temas.'

Package phpkrond.modulos.krond
Procedural Elements

krond.inc.php

* **Package** phpkrond.modulos.krond

_MOD_KROND_CLASS = 1 *[line 31]*

Package phpkrond.modulos.krond Classes

Class krond_admin

[line 44]

Summary: Clase que gestiona la Administración del Sistema.

Permita gestionar todo el sistema. Desde la manipulación de Plantillas, Paginas y Objetos hasta la de Usuarios de Administración.

* **Package** phpkrond.modulos.krond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor krond_admin::krond_admin([$operacion = ""]) *[line 52]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Contructor de la clase.

Recoge la operación ha realizar en el sistema.

krond_admin::form_admins() *[line 1165]*

Summary: Muestra el Formulario del Menu de Usuarios de Administración.

Lista todas los Usuarios de Administración y muestra un formulario para crear nuevos formularios. En el listado de usuarios se pueden acceder a las propiedades de los mismos y a otras operaciones.

krond_admin::form_menu() *[line 377]*

Summary: Muestra el Formulario del Menu Principal.

En el menu principal se muestran las operaciones principales que se pueden hacer con las Plantillas, Páginas, Objetos y Usuarios de Administración. Tambien muestra la opción de salir del sistema.

krond_admin::form_objetos() *[line 393]*

Summary: Muestra el Formulario del Menu de Objetos.

Lista todos los Objetos y muestra un formulario para crear nuevos objetos. En el listado de objetos se pueden acceder a las propiedades de los mismos y otra operaciones.

krond_admin::form_objeto_contenedor($idObjeto) *[line 625]*

Function Parameters:

* *$idObjeto* **$idObjeto** Identificador del Contenedor.

Summary: Muestra las propiedades adicionales de un contenedor

Muestra el formulario con los objetos contenidos en el el contenedor y permite añadir nuevos objetos, cambiar posiciones de los mismos o borrar los que existen.

krond_admin::form_objeto_propiedades($idObjeto) *[line 511]*

Function Parameters:

* *$id* **$idObjeto** Identificador del objeto.

Summary: Muestra las propiedades de un objeto.

Muestra el formulario con las propiedades del objeto y permite modificarlas desde ese formulario. Si el objeto es un contenedor llama al formulario que muestra las propiedades adicionales del contenedor.

krond_admin::form_paginas() *[line 775]*

Summary: Muestra el Formulario del Menu de Páginas.

Lista todas las Páginas y muestra un formulario para crear nuevas páginas. En el listado de páginas se pueden acceder a las propiedades de las mismas y a otras operaciones.

krond_admin::form_pagina_propiedades($id) *[line 859]*

Function Parameters:

* *$id* **$id** Identificador de la página.

Summary: Muestra las propiedades de una página.

Muestra el formulario con las propiedades de la página y permite modificarlas desde ese formulario. Las propiedades que permite modificar son las sustituciones que tiene definida la página.

krond_admin::form_plantillas() *[line 937]*

Summary: Muestra el Formulario del Menu de Plantillas.

Lista todas las Plantillas y muestra un formulario para crear nuevas plantillas. En el listado de plantillas se pueden acceder a las propiedades de las mismas y a otras operaciones.

krond_admin::form_plantilla_dependencias($id) *[line 1100]*

Function Parameters:

* *$id* **$id** Identificador de Plantilla.

Summary: Muestra las dependencias de la plantilla.
Muestra las página que utilizan la plantilla.

krond_admin::form_plantilla_propiedades($id) *[line 1023]*

Function Parameters:

* *$id* **$id** Identificador de la plantilla.

Summary: Muestra las propiedades de una plantilla
Muestra el formulario con las propiedades de la plantilla y permite modificarlas desde ese formulario.

krond_admin::logout() *[line 1231]*

Summary: Hace logout del sistema.
El logout del sistema se hace borrando la cookie de administración $_COOKIE['krond_admin'] y la variable de session $_SESSION['KROND_AUTO'].

krond_admin::select_objetos([$idSeleccionado = 1]) *[line 740]*

Function Parameters:

* *$idSeleccionado* **$idSeleccionado** Identificador del objeto seleccionado por defecto. String Contenido HTML del campo select.

Summary: Devuele un select con los objetos del presentes.
Devuelve el codigo HTML de un campo select con los objetos presentes en el sistema. En ese select aparecerá seleccionado el objeto $idSeleccionado.

Class paginas_activas

[line 1255]

Summary: Clase para mostrar las páginas activas en el sistema.

Obtiene las páginas activas en el sistema, tanto las páginas normales como las del sistema de administración.

* **Package** phpkrond.modulos.krond
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed paginas_activas::$db = *[line 1265]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor paginas_activas::paginas_activas() *[line 1274]*

Summary: Constructor de la clase.

Inicializa el atributo $db que contiene la conexion a la base de datos.

Array paginas_activas::get_paginas([$tipoPagina = PAGINA_NORMAL]) *[line 1357]*

Function Parameters:

* *$tipoPagina* **$tipoPagina** Tipo de Página a devolver.

Summary: Devuelve un array de Paginas activas del sistema.

Puede devolver las paginas normales o las del sistema.

Array paginas_activas::get_pnormales() *[line 1334]*

Summary: Devuelve un array de paginas normales activas en el sistema.

El array que devuelve es de objetos krond_paginas.

Array paginas_activas::get_psistemas() *[line 1345]*

Summary: Devuelve un array de paginas del sistema activas en el sistema.

El array que devuelve es de objetos krond_paginas.

String paginas_activas::mostrar() *[line 1288]*

Summary: Devuelve el codigo html con las paginas activas.

Las paginas que estan activas son tanto las paginas del sistema (que necesitan login y pass) como las paginas publicas.

Package phpkrond.modulos.noticias Procedural Elements

noticias.inc.php

* **Package** phpkrond.modulos.noticias

_MOD_NOTICIAS_CLASS = 1 *[line 31]*

Package phpkrond.modulos.noticias
Classes

Class noticias_admin

[line 951]

Summary: Clase para administrar el módulo de noticias.
Se utiliza para crear articulos y temas. Tambien permite llevar un poco de gestión sobre los mismos, modificandolos o borrandolos.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor noticias_admin::noticias_admin([$operacion = ""]) *[line 959]*

Function Parameters:

* *$operacion* **$operacion** String con la operación a realizar.

Summary: Contructor de la clase.
Ejecuta la operación especificada en el parámetro.

String noticias_admin::calcula_select_graficos([$grafico = ""]) *[line 1451]*

Function Parameters:

* *$grafico* **$grafico** Grafico seleccionado por defecto.

Summary: Calcula el campo select de un formulario con los graficos disponibles.
Los graficos se encuentran todos en 'modulos/noticias/graficos/'.

String noticias_admin::calcula_select_temas([$idTema = 0]) *[line 1475]*

Function Parameters:

* *$idTema* **$idTema** Identificador del tema seleccionado por defecto.

Summary: Calcula el campo select de un formulario con los temas.
Los temas se encuentran en la tabla 'noticias_temas'.

String noticias_admin::form_articulo_propiedades($idArticulo) *[line 1382]*

Summary: Muestra el formulario con las propiedades de un Articulo.
Este formulario muestra las propiedades del articulo.

String noticias_admin::form_cabecera() *[line 1090]*

Summary: Cabecera del Módulo de Administración de Noticias

Muestra una cabecera con el título de Administración del Módulo de Noticias.

String noticias_admin::form_config() *[line 1502]*

Summary: Muestra el formulario de configuración del Módulo.

Desde este formulario se pueden cambiar las variables de configuración 'nombre_pagina_mostrar'. Estas variables se guardan en la tabla 'noticias_config'.

String noticias_admin::form_general() *[line 1105]*

Summary: Muestra el formulario general de la administración de las noticias.

Este formulario muestra los temas que clasifican las noticias y las noticias principales que aparecen en la seccion principal.

String noticias_admin::form_tema_propiedades($idTema) *[line 1300]*

Summary: Muestra el formulario con las propiedades de un Tema.

Este formulario muestra las propiedades del temas y los articulos que tiene.

Class noticias_anteriores

[line 892]

Summary: Clase que gestiona las noticias anteriores.
Sirve para obtener un cuadro con las noticias que no salen en la seccion principal.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor noticias_anteriores::noticias_anteriores() *[line 900]*

Summary: Contructor de la clase.
Devuelve un bloque con el numero de noticias anteriores definido en la configuración del modulo.

Class noticias_articulo

[line 315]

Summary: Clase que gestiona las noticias (articulos).

Las noticas son articulos. Los forma de tratarlo es como una especie de weblog y donde las noticias estan clasificadas según los temas. La información de un articulo se encuentra en la tabla 'noticias_articulos'.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed noticias_articulo::$db = *[line 335]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed noticias_articulo::$idArticulo = *[line 324]*

Summary: Atributo de la clase que guarda la clave de la noticia.

El atributo de la clase $idArticulo es el identificador utilizado como clave primaria en la tabla 'noticias_articulos'.

Constructor noticias_articulo::noticias_articulo([$idArticulo = 0]) *[line 346]*

Function Parameters:

* *$idSeccion* **$idArticulo** Identificador de la seccion.

Summary: Constructor de la clase que inicializa el atributo $idArticulo.

El atributo de la clase $idArticulo es el identificador utilizado como clave primaria en la tabla 'noticias_articulos'.

boolean noticias_articulo::borrar() *[line 616]*

Summary: Borra un articulo de las noticias.

Elimina la entrada del articulo que se encuentra en la tabla 'noticias_articulos'.

boolean noticias_articulo::check_datos($datos) *[line 604]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el titulo de un articulo o el contenido.

boolean noticias_articulo::crear($titulo, $contenido, $idTema) *[line 579]*

Function Parameters:

* *$titulo* **$titulo** Titulo del articulo.
* *$contenido* **$contenido** Contenido del articulo.
* *$idTema* **$idTema** Tema del articulo.

Summary: Crea un nuevo articulo en las noticias.

Crea una nueva entrada en la tabla 'noticias_articulos' inicializando todos los campos con los datos del nuevo articulo.

boolean noticias_articulo::existe() *[line 557]*

Summary: Verifica si el articulo existe.

Comprueba que el identificador del articulo es correcto.

Array noticias_articulo::get_all([$comienzo = 0]) *[line 491]*

Function Parameters:

* *$comienzo* **$comienzo** Articulo por el comenzar a listar. Se utiliza para paginar los articulos.

Summary: Devuelve los articulos de las noticias.

No Devuelve todas las noticias que hay disponibles sino devuelve una cantidad limitada. Los articulos estan ordenados por fecha en orden decreciente.

Array noticias_articulo::get_all_paginar([$comienzo = 0], [$limite = 0]) *[line 521]*

Function Parameters:

* *$comienzo* **$comienzo** Articulo por el comenzar a listar. Se utiliza para paginar los articulos.
* *$limite* **$limite** Ultimo Articulo por que listar.

Summary: Devuelve un subconjunto de articulos de las noticias.

No Devuelve todas las noticias que hay disponibles sino devuelve una cantidad limitada. Los articulos estan ordenados por fecha en orden decreciente. Util para paginar los articulos.

String noticias_articulo::get_contenido() *[line 392]*

Summary: Devuelve el contenido del articulo.

El contenido del artículo se encuentra guardado en la tabla 'noticias_articulos' en el campo 'contenido_articulo'.

String noticias_articulo::get_fecha() *[line 412]*

Summary: Devuelve la fecha de creación del articulo.

La fecha de creación del artículo se encuentra guardada en la tabla 'noticias_articulos' en el campo 'fecha_articulo'. Esta en formato 'AAAA-MM-DD hh:mm:ss'

Integer noticias_articulo::get_id() *[line 360]*

Summary: Devuelve el identificador de articulo.

El identificador de articulo es la clave primaria de la tabla 'noticias_articulo' que contiene la información de la noticia.

String noticias_articulo::get_resumen() *[line 430]*

Summary: Obtiene un resumen del contenido del articulo.

El resumen son las primeras letras (definidas por la constante de configuracion del modulo 'noticias_caracteres_resumen').

contenidos_tema_Tema noticias_articulo::get_tema() *[line 471]*

Summary: Devuelve el tema del articulo

Todos los articulos pertenecen a una tema. La información del tema a la que pertenece se se encuentra en el campo 'id_tema' de la tabla 'noticias_articulos'.

String noticias_articulo::get_titulo() *[line 373]*

Summary: Devuelve el título del articulo.

El título del artículo se encuentra guardado en la tabla 'noticias_articulos' en el campo 'titulo_articulo'.

Integer noticias_articulo::get_total_articulos() *[line 543]*

Summary: Devuelve el numero de articulos del sistema.

El numero de articulos de todos los temas.

Boolean noticias_articulo::set($titulo, $contenido, $idTema) *[line 638]*

Function Parameters:

* *$titulo* **$titulo** Nuevo titulo del articulo.
* *$contenido* **$contenido** Nuevo contenido del articulo.
* *$idTema* **$idTema** Nuevo tema del articulo.

Summary: Modifica las propiedades del articulo.

Modifica el titulo y el contenido asociado. Tambien permite cambiarlo de tema. Comprueba que los nuevos datos son correctos.

Class noticias_aviso

[line 664]

Summary: Clase que gestiona los avisos.

Los avisos son noticias especiales, que tienen un tratamiento diferente a una noticias normal. La información de un articulo se encuentra

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Class noticias_config

[line 1578]

Summary: Clase que gestiona las variables de configuración del módulo.
Las variables de configuración se encuentran almacenadas en la tabla 'noticias_config'.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed noticias_config::$db = *[line 1588]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor noticias_config::noticias_config() *[line 1596]*

Summary: Contructor de la clase.
Inicializa el atributo $this->db que contiene el enlace con la base de datos.

String noticias_config::get_variable($nombreVar) *[line 1610]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Recupera el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'noticias_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

noticias_config::set_variable($nombreVar, $nuevoValor) *[line 1629]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable a modificar.
* *$nuevoValor* **$nuevoValor** Nuevo valor de la variable.

Summary: Modifica el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'noticias_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

Class noticias_error

[line 1645]

Summary: Clase para mostrar mensajes del modulo noticias.
Todos los mensajes del modulo se consideran mensajes de error.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor noticias_error::noticias_error([$tituloError = "phpKROND - Error "], [$msgError = ""]) *[line 1654]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class noticias_mostrar

[line 683]

Summary: Clase que muestras los últimos articulos del modulo.

Obtiene los 'n' articulos del modulo ordenados por fecha en orden decreciente, dando un pequeño resumen del mismo y ofreciendo información adicional. El numero de articulos que muestra y la forma en que los muestra estan determinados por las variables de configuración del modulo 'noticias_num_cols' y 'noticas_articulos_cols'. Tambien puede mostrar un articulo en concreto.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor noticias_mostrar::noticias_mostrar([$operacion = ""]) *[line 691]*

Function Parameters:

* *$operación* **$operacion** Cadena que indica la operación.

Summary: Constructor de la clase para mostrar los contenidos.

Realiza la operación indicada por el parametro.

String noticias_mostrar::form_articulo_corto($idArticulo, [$alinear = 0]) *[line 793]*

Function Parameters:

* *$idArticulo* **$idArticulo** Identificador del articulo
* *$alinear* **$alinear** Tipo de alineacion (true, false)

Summary: Muestra el formato corto de un articulo.

En este formato se muestra solo el resumen del articulo.

String noticias_mostrar::form_articulo_largo($idArticulo, [$alinear = 0]) *[line 845]*

Function Parameters:

* *$idArticulo* **$idArticulo** Identificador del articulo
* *$alinear* **$alinear** Tipo de alineacion (true, false)

Summary: Muestra el formato largo de un articulo.

En este formato se muestra el contenido completo del articulo.

String noticias_mostrar::mostrar_portada([$inicio = 0]) *[line 729]*

Function Parameters:

* *$inicio* **$inicio** Indice del articulo a mostrar.

Summary: Muestra las ultimas noticias (articulos)

Los ordena segun la fecha en orden decreciente y ordenados en columnas (segun la variable de configuracion que dice cuando columnas se muestran).

Class noticias_tema

[line 45]

Summary: Clase que gestiona los temas de las noticias.

Las temas de las noticias clasifican las noticias en diferentes secciones para permitir una organización. La información de un tema se encuentra en la tabla 'noticias_temas'.

* **Package** phpkrond.modulos.noticias
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed noticias_tema::$db = *[line 65]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed noticias_tema::$idTema = *[line 54]*

Summary: Atributo de la clase que guarda la clave del tema.

El atributo de la clase $idTema es el identificador utilizado como clave primaria en la tabla 'noticias_temas'.

Constructor noticias_tema::noticias_tema([$idTema = 0]) *[line 76]*

Function Parameters:

* *$idTema* **$idTema** Identificador del tema.

Summary: Constructor de la clase que inicializa el atributo $idTema.

El atributo de la clase $idTema es el identificador utilizado como clave primaria en la tabla 'noticias_temas'.

boolean noticias_tema::borrar($vaciar) *[line 244]*

Function Parameters:

* *$vaciar* **$vaciar** Si true entonces sólo borra las noticias. En caso contrario, comprueba que el tema esta vacio (notiene noticias) y borra el tema.

Summary: Borra un tema de las noticias.

Elimina la entrada del tema que se encuentra en la tabla 'noticias_temas' siempre y cuando el tema este vacio.

boolean noticias_tema::check_datos($datos) *[line 227]*

Function Parameters:

* *$datos* **$datos** String que los datos a comprobar.

Summary: Verifica que los datos son correctos

En principio solo comprueba que los datos son distintos de "". Util para comprobar el nombre de un tema o de un grafico.

boolean noticias_tema::crear($nombre, $grafico) *[line 204]*

Function Parameters:

* *$nombre* **$nombre** Nombre del tema.
* *$grafico* **$grafico** Grafico del tema.

Summary: Crea un nuevo tema en las noticias.

Crea una nueva entrada en la tabla 'noticias_temas' inicializando todos los campos con los datos del nuevo tema.

boolean noticias_tema::existe() *[line 183]*

Summary: Verifica si el tema existe.

Comprueba que el identificador de tema es correcto.

Array noticias_tema::get_all() *[line 164]*

Summary: Devuelve los temas que clasifican las noticias.

Devuelve un array de objetos 'noticias_tema'.

Array noticias_tema::get_articulos([$comienzo = 0]) *[line 145]*

Function Parameters:

* *$comienzo* **$comienzo** Articulo por el comenzar a listar. Se utiliza para paginar los articulos.

Summary: Devuelve los artículos de un tema.

Un tema contiene diversos articulos para mostrar. Con esto se ofrece una forma de mantener un orden lógico en los articulos que se pueden ver en el módulo. Devuelve un array de objetos noticias_articulo ordenados por fecha en orden decreciente.

String noticias_tema::get_grafico() *[line 121]*

Summary: Devuelve el nombre del archivo del grafico del tema.

El nombre de dicho archivo se encuentra guardado en la tabla 'noticias_temas' en el campo 'grafico_tema'. Dicho archivo se guarda en 'modulos/noticias/graficos/'.

Integer noticias_tema::get_id() *[line 90]*

Summary: Devuelve el identificador de tema.

El identificador de tema es la clave primaria de la tabla 'noticias_temas' que contiene la información del tema de las noticias.

String noticias_tema::get_nombre() *[line 102]*

Summary: Devuelve el nombre del tema.

El nombre del tema se encuentra guardado en la tabla 'noticias_temas' en el campo 'nombre_tema'.

Boolean noticias_tema::set($nombre, $grafico) *[line 289]*

Function Parameters:

* *$nombre* **$nombre** Nuevo nombre del tema.
* *$grafico* **$grafico** Nuevo grafico del tema.

Summary: Modifica las propiedades del tema.

Modifica el nombre del tema y el grafico asociado. El grafico asociado no se borra. Comprueba que los nuevos datos son correctos.

Package phpkrond.modulos.users
Procedural Elements

users.inc.php

* **Package** phpkrond.modulos.users

_MOD_USERS_CLASS = 1 *[line 32]*

Package phpkrond.modulos.users Classes

Class users_admin

[line 523]

Summary: Clase para administrar los usuarios registrados.

Se utiliza para ofrecer algunas utilidades a los administradores de la web. Borrar registros, Altas Modificaciones de datos y se puede extender a más cosas como envio de emails e informaciones varias.

* **Package** phpkrond.modulos.users
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor users_admin::users_admin([$operacion = ""]) *[line 531]*

Function Parameters:

* *$operacion* **$operacion** String con la operación a realizar.

Summary: Contructor de la clase.

Ejecuta la operación especificada en el parámetro.

String users_admin::form_cabecera() *[line 554]*

Summary: Cabecera del Módulo de Administración de Usuarios

Muestra una cabecera con el título de Administración del Módulo de Usuarios Registrados.

String users_admin::form_general() *[line 568]*

Summary: Muestra el Formulario general de Administración de los usuarios.

Este formulario lista los usuarios registrados en el sistema.

Class users_bloque

[line 612]

Summary: Clase que muestra un pequeño formulario de login del usuario.
Ese formulario sirve para logarse o invitar a los usuarios a que se registren en el sistema.

* **Package** phpkrond.modulos.users
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor users_bloque::users_bloque([$operacion = ""]) *[line 621]*

Function Parameters:

* *$operacion* **$operacion** String con la operacion a realizar

Summary: Constructor de la clase.
El bloque de usuarios es un modulo para logarse en usuario en la web.

String users_bloque::form_login() *[line 658]*

Summary: Formulario para logarse los usuarios.
Si el usuario esta registrado muestra información para salir del sistema. Si el usuario no se ha logado y no ha se ha dado de alta en el sistema, se le muestra la opcion para que lo haga.

String users_bloque::form_logout() *[line 724]*

Summary: Mensaje para salir del sistema.
Informa al usuario de que ha salido del sistema.

Class users_config

[line 994]

Summary: Clase que gestiona las variables de configuración del módulo.
Las variables de configuración se encuentran almacenadas en la tabla 'users_config'.

* **Package** phpkrond.modulos.users
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed users_config::$db = *[line 1004]*

Summary: Contiene la conexion de la base de datos.
Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

Constructor users_config::users_config() *[line 1012]*

Summary: Contructor de la clase.
Inicializa el atributo $this->db que contiene el enlace con la base de datos.

String users_config::get_variable($nombreVar) *[line 1026]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable.

Summary: Recupera el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'users_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

users_config::set_variable($nombreVar, $nuevoValor) *[line 1045]*

Function Parameters:

* *$nombreVar* **$nombreVar** Nombre de la variable a modificar.
* *$nuevoValor* **$nuevoValor** Nuevo valor de la variable.

Summary: Modifica el valor de la variable de configuración especificada.
Las variables de configuración se encuentran en la tabla 'descargas_config'. El campo 'nombre' guarda el nombre de la variable y el campo 'valor' su valor.

Class users_error

[line 1061]

Summary: Clase para mostrar mensajes del modulo Usuarios Registrados.
Todos los mensajes del modulo se consideran mensajes de error.

* **Package** phpkrond.modulos.users
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor users_error::users_error([$tituloError = "phpKROND - Error "], [$msgError = ""]) *[line 1070]*

Function Parameters:

* *$tituloError* **$tituloError** Titulo del Mensaje de Error.
* *$msgError* **$msgError** Contenido del Mensaje de Error.

Summary: Constructor de la clase.
Muestra el mensaje de error indicado.

Class users_mostrar

[line 741]

Summary: Clase que muestra las opciones principales de registro de usuarios.
Permite registrarse a los nuevos usuarios y tambien cambiar las preferencias de los mismos. Tambien da opcion a borrarse del sistema de registros.

* **Package** phpkrond.modulos.users
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

Constructor users_mostrar::users_mostrar([$operacion = ""]) *[line 749]*

Function Parameters:

* *$operacion* **$operacion** String con la operación a realizar.

Summary: Contructor de la clase.
Ejecuta la operación especificada en el parámetro.

String users_mostrar::form_general() *[line 810]*

Summary: Muestra el formulario de registro de usuarios.
Si el usuario esta logado muestra el formulario de preferencias. De lo contrario muestra el formulario de registro.

String users_mostrar::form_user_nuevo() *[line 831]*

Summary: Muestra el formulario de registro de usuario.
Pide nombre de login, email y nombre completo del usuario para registrar en el sistema.

String users_mostrar::form_user_preferencias() *[line 887]*

Summary: Muestra el formulario de cambio de opciones del usuario.
Permite cambiar el email del usuario y la clave del mismo. Tambien ofrece la opcion de darse de baja en el sistema.

Class users_registro

[line 73]

Summary: Clase que gestiona los usuarios registrados.

Especificación para identificar usuarios registrados: La forma de saber si un usuario registrado esta validado se hace mediante la variable de session krond_users_register. Si esta variable es igual a 'true' entonces el usuario se ha validado correctamente.

Si la variable de session krond_users_register no esta definida o esta puesta a 'false' el usuario no se ha validado o no esta registrado.

Para validar al usuario (que debe haberse registrado) se mira la cookie KROND_USERS, en esta cookie estan los valores del id_usuario y password (encriptada).

Si la password coincide con la que hay almacenada en la BBDD, entonces se define la variable de session krond_users_register = 'true'. De lo contrario, el usuario no se valida.

La tabla que guarda toda la información sobre el usuario se llama 'users_registros'.

$_COOKIE["krond_users"] $_SESSION["krond_users_register"]

Existe un usuario anonimo que no necesita validarse.

Información que se guarda en la cookie:

* id de usuario
* clave encriptada de usuario

* **Package** phpkrond.modulos.users
* **Version** 200
* **Author** Francisco José Sánchez Sánchez

mixed users_registro::$db = *[line 93]*

Summary: Contiene la conexion de la base de datos.

Este atributo es del tipo newADOconnection que pertenece a ADODB. Automaticamente es inicializado recogiendolo de la objeto $GLOBALS['KROND_CFG']. Este objeto tiene un atributo, $db, que inicializa la conexion con la base de datos.

mixed users_registro::$idUser = *[line 82]*

Summary: Atributo de la clase que guarda la clave del usuario.

El atributo de la clase $idUser es el identificador utilizado como clave primaria en la tabla 'users_registros'.

Constructor users_registro::users_registro([$idUser = 0]) *[line 103]*

Function Parameters:

* *$idUser* **$idUser** Identificador de usuario.

Summary: Constructor de la clase

Iniciliza el identificador de usuarios y la conexion con la base de datos.

Boolean users_registro::borrar() *[line 278]*

Summary: Elimina el usuario registrado

Para eliminarlo el usuario debe estar logado.

Boolean users_registro::check_datos($datos) *[line 389]*

Function Parameters:

* *$datos* **$datos** String con los datos a chequear.

Summary: Comprueba que no se han utilizado caracteres raros en los datos.

Es una medida de seguridad para evitar SQL-Inyection.

Boolean users_registro::check_email($email) *[line 408]*

Function Parameters:

* *$email* **$email** Direccion de correo electronico.

Summary: Comprueba que la direccion de correo es correcta.

Se consideran direcciones de correo correctas si son del tipo siguiente. $expEmail = "^([A-Za-z0-9_]|\\-|\\.)+" . "@" . "(([A-Za-z0-9_]|\\-)+\\.)+" . "[a-zA-z]{2,4}$";

Boolean users_registro::crear($login, $email, $nombreCompleto) *[line 476]*

Function Parameters:

* *$login* **$login** Login del Usuario.
* *$email* **$email** Email del Usuario.
* *$nombreCompleto* **$nombreCompleto** Nombre completo del usuario.

Summary: Crea un nuevo usuario registrado.

Recoge los datos de login, email y nombre completo y si son correctos, añade una entrada en la tabla 'users_registros' con los datos de registro. Esta función inicializa la clave del usuario con una clave aleatoria que se le manda al usuario por correo a su dirección de correo electronico.

Boolean users_registro::enviar_correo($destino_nombre, $destino_email, $origen_nombre, $origen_email, $titulo, $mensaje) *[line 446]*

Function Parameters:

* *$destino_nombre* **$destino_nombre** Nombre del Receptor del correo.
* *$destino_email* **$destino_email** Email del Receptor.
* *$origen_nombre* **$origen_nombre** Nombre del Remitente.
* *$origen* **$origen_email** email del Remitente.
* *$titulo* **$titulo** Titulo del correo (Subject).
* *$mensaje* **$mensaje** Mensaje del correo.

Summary: Envia un correo al email del usuario.
Esta función se limita a mandar un correo, todos los datos del mismo se le tienen que pasar en los parametros.

Boolean users_registro::esta_registrado() *[line 113]*
Summary: Comprueba que el usuario se ha registrado.

String users_registro::generar_clave() *[line 424]*
Summary: Genera una clave aleatoria para el usuario.
La clave aleatoria tiene una longitud de 12 caracteres.

Array users_registro::get_all() *[line 213]*
Summary: Devuelve un array con todos los usuarios registrados.
El array es de objetos users_registro.

String users_registro::get_email() *[line 170]*
Summary: Devuelve el email de usuario.
El email del usuario se encuentra almacenado en la tabla 'users_registros' dentro del campo 'email'.

String users_registro::get_fecha() *[line 194]*
Summary: Devuelve la fecha de registro del usuario.
La fecha de registro del usuario se encuentra almacenado en el campo 'fecha_user' de la tabla 'users_registros'. La fecha esta en formato 'AAAA-MM-DD'.

Integer users_registro::get_id() *[line 134]*
Summary: Devuelve el identificador de usuario.
El identificaodr de usuario es la clave primaria utilizada en la tabla 'usuarios_registros' para almacenar los datos de los usuarios registrados.

Integer users_registro::get_nombre() *[line 147]*
Summary: Devuelve el identificador de usuario.
El identificador de usuario es la clave primaria utilizada en la tabla 'usuarios_registros'

para almacenar los datos de los usuarios registrados.

users_registro::logout() *[line 297]*

Summary: Hace logout del usuario.

Elimina la cookie y las variables de sesion del usuario. La proxima vez que el usuario quiera ser reconocido por el sistema debera introducir los datos de login y password.

Boolean users_registro::set_email($nuevoEmail) *[line 235]*

Function Parameters:

* *$nuevoEmail* **$nuevoEmail** Nuevo email del usuario.

Summary: Modifica email del usuario.

La modificación solo es posible si el email no existe y es correcto. Para cambiarlo el usuario debe estar logado.

Boolean users_registro::set_passwd($nuevoPasswd) *[line 260]*

Function Parameters:

* *$nuevoPasswd* **$nuevoPasswd** Nueva clave del suaurio

Summary: Modifica passwd del usuario.

Envia por correo la nueva passwd al usuario. Para cambiarla el usuario debe estar logado.

users_registro::validar_cookie() *[line 317]*

Summary: Valida la cookie de registro del usuario.

$_COOKIE['krond_users'] contiene el identificador de usuario y la clave encriptada con md5 del mismo. Comprueba que corresponden a un usuario en el sistema. Si la cookie es correcta inicializa las variables de sesion del usuario.

return Boolean True si la cookie es correcta.

users_registro::validar_login($login, $passwd) *[line 358]*

Function Parameters:

* *$login* **$login** Nombre de login del usuario.
* *$passwd* **$passwd** Clave del usuario. return Boolean True si el usuario se ha validado.

Summary: Valida los datos de login de un usuario registrado.

Recoge el login del usuario y la clave (en texto llano) para compararla con la clave

almacenada en la bd. La almacenada en el tabla se encuentra encriptada con md5 por lo que es necesario encriptar la clave de usuario para comprobar que son iguales. Tambien crea $_COOKIE['krond_users'] y las variables $_SESSION['krond_users_register'] y $_SESSION['krond_users_id'].

Appendices

Appendix A - Class Trees

Package phpkrond

krond_cfg
* krond_cfg

krond_contenedor
* krond_contenedor

krond_error
* krond_error

krond_objeto
* krond_objeto

krond_pagina
* krond_pagina

krond_plantilla
* krond_plantilla

krond_sustitucion
* krond_sustitucion

krond_useradm
* krond_useradm

krond_variable
* krond_variable

Package phpkrond.modulos.calendario

calendario_admin
* calendario_admin

calendario_bloque
* calendario_bloque

calendario_cita
* calendario_cita

calendario_config
* calendario_config

calendario_error
* calendario_error

calendario_evento
* calendario_evento

calendario_mostrar
* calendario_mostrar

Package phpkrond.modulos.contenidos

contenidos_admin
* contenidos_admin

contenidos_articulo
* contenidos_articulo

contenidos_bloque
* contenidos_bloque

contenidos_config

* contenidos_config

contenidos_error

* contenidos_error

contenidos_mostrar

* contenidos_mostrar

contenidos_seccion

* contenidos_seccion

Package phpkrond.modulos.descargas

descargas_admin

* descargas_admin

descargas_categoria

* descargas_categoria

descargas_config

* descargas_config

descargas_error

* descargas_error

descargas_fichero

* descargas_fichero

descargas_mostrar

* descargas_mostrar

Package phpkrond.modulos.forum

forum_admin
* forum_admin

forum_config
* forum_config

forum_error
* forum_error

forum_foro
* forum_foro

forum_mensaje
* forum_mensaje

forum_mostrar
* forum_mostrar

forum_respuesta
* forum_respuesta

forum_tema
* forum_tema

Package phpkrond.modulos.krond

krond_admin
* krond_admin

paginas_activas
* paginas_activas

Package phpkrond.modulos.noticias

noticias_admin
* noticias_admin

noticias_anteriores
* noticias_anteriores

noticias_articulo
* noticias_articulo

noticias_aviso
* noticias_aviso

noticias_config
* noticias_config

noticias_error
* noticias_error

noticias_mostrar
* noticias_mostrar

noticias_tema
* noticias_tema

Package phpkrond.modulos.users

users_admin
* users_admin

users_bloque

* users_bloque

users_config

* users_config

users_error

* users_error

users_mostrar

* users_mostrar

users_registro

* users_registro

Index