



Dokeos 1.5

Manual del Desarrollador



Roan Embrechts & Hugues Peeters, April 2004
Distributed under Gnu Free Documentation License
© Dokeos & Universiteit Gent, Université catholique de Louvain

Traducción: Santiago Márquez Solís, Agosto 2005



Dokeos 1.5 Manual del Desarrollador

Tabla de Contenidos

1. Introducción.....	3
2. Estructura del código de Dokeos.....	4
3. El API de Dokeos.....	6
4. Estructura de la base de datos de Doleos.....	8
5. Convenciones de código propuestas.....	9
6. Plantilla para nuevos desarrollos (add-ons).....	13
7. Método de trabajo & metodología de diseño.....	15
8. Comunicación entre desarrolladores.....	16
9. Desarrollo del Software en general.....	17
10. Diseño de la Interface de Usuario.....	18
11. Entrada del Usuario.....	19
12. Promoción de Dokeos.....	20
13. Enlaces de Interes.....	21
14. GNU Free Documentation License.....	22
15. Modificar este manual.....	27



1. Introducción

Hay varios tipos diferentes de manuales disponibles de Dokeos en función del perfil del usuario:

- estudiantes
- profesores
- desarrolladores (programadores)
- y Administradores

Los manuales del Profesor y del Estudiante han sido traducidos a varios idiomas. En la dirección <http://www.dokeos.com> puedes encontrar la última versión de los mismos.

El manual que tienes entre manos, es un manual de introducción para desarrolladores de Dokeos, la plataforma gratuita de e-learning y administración de cursos. El manual no está completo ni es perfecto, por lo que cualquier sugerencia o contribución será bienvenida. Dokeos es un desarrollo basado en el software de e-learning original Caroline (versión 1.4.2); Dokeos 1.5 y sucesivos está basado en este código y corrige muchos errores, además de incluir nuevas mejoras y características. El software se distribuye bajo licencia GNU General Public License.



2. Estructura del código de Dokeos

Dokeos consiste en diferentes piezas de funcionalidad que reciben el nombre de módulos o herramientas: por ejemplo la agenda, los documentos, los enlaces, el foro... El código de cada uno de estos elementos está situado en una carpeta separada del resto. El código de Dokeos es mayormente procedural. Lentamente está evolucionando y convirtiéndose en un código más maduro. Las funciones están siendo usadas más y más en el nuevo código, se piensa en encapsulación y en el diseño por capas... no obstante, no se utilizan objetos muy a menudo. Todo el código que puede ser compartido entre las diferentes herramientas se encuentra recopilado en la carpeta 'inc'.

Inc por include

La carpeta 'inc'- abreviatura de incluye (inclusión) - es un directorio de propósito general que es utilizado por todos los módulos estándar, y proporciona todas las variables globales de Dokeos, los encabezados y pies estándar, y muchas otras funciones útiles. Está compuesta de ficheros y algunos directorios. En la carpeta **inc/lib** podemos encontrar el código de las librerías, en estos ficheros se proporciona la funcionalidad que es común, como por ejemplo: funciones para la administración de ficheros, consulta de base de datos, funciones generales... Juntos, todos estos ficheros proporcionan una Interface de Programación de Aplicaciones (*Application Programming Interface* o API).

En el directorio **inc/conf** tenemos los ficheros de configuración para algunas herramientas. Fíjate que algunos de estos ficheros son tentativos, es decir, aparecen pero no interactúan con ninguna de las herramientas. Todas estas carpetas están actualmente en progreso y evolución.

NOTE: La carpeta 'inc' se llamaba 'include' en versiones de Dokeos anteriores a la 1.5.

A partir de la versión 1.4 de Claroline/Dokeos, la creación de tus propios módulos es un poco más fácil: solamente hay que usar tres sentencias de inclusión, y obtenemos el encabezado, el pie y las variables globales de Dokeos.

Más adelante puedes ver una plantilla para empezar a desarrollar tu propio código. Te recomendamos que utilices una plantilla para hacer un nuevo desarrollo. Además, respeta las convenciones de [código utilizadas](#); haciéndolo conseguiremos que el código de Dokeos sea consistente y más legible para todo el mundo.

Por favor, mantén tu código nuevo tan modular como sea posible. Divídelo en diferentes secciones, usa funciones, comenta a menudo para explicar lo que haces y procura dividirlo en piezas lógicas. Utiliza funciones siempre que sea posible, ya que realmente



ayudará a mantener la modularidad del código. Si deseas ir un paso más lejos, puedes usar programación orientada a objetos, PHP tiene algo de soporte para ella (y está mejorado en PHP 5). Y por favor, escribe una o dos líneas al principio de cada fichero de script explicando su propósito y finalidad.

NOTA: Si eres nuevo en el uso de funciones en PHP: ten en cuenta que no funcionan del modo en que lo hacen en otros lenguajes de programación. Este es el origen de muchos errores y de un montón de frustración

Código global y código específico de un curso

El código de la aplicación Dokeos es global. Los cambios se aplican a todos los cursos del servidor. Sin embargo, hay maneras de escribir código y usarlo solamente para un curso específico. Esta es una solución para requerimientos especiales que no son compartidos por todos los cursos, además resulta más práctico probar y depurar una personalización en condiciones de “entorno real” antes de aplicarlas globalmente. La manera más fácil de hacerlo es “Añadir un enlace en la página de inicio”, seleccionando la segunda posibilidad “Enlace a un sitio”, y proporcionando la URL del código. Por ejemplo, podrías almacenar este código en una carpeta denominada plugin.



3. El API de Dokeos

Hay varias funciones que pueden ser útiles para los desarrolladores. En el futuro intentaremos tener la documentación del API disponible, usando por ejemplo [PhpDocumentor](http://pear.php.net/package/PhpDocumentor) (<http://pear.php.net/package/PhpDocumentor>). Las funciones de librería se encuentran en la carpeta 'inc' - abreviatura de include -. Antes de empezar la programación, por favor, examina el contenido de esta carpeta, y de las subcarpeta inc/lib y inc/conf - para que te puedas hacer una idea de lo que Dokeos te puede ofrecer desde el principio.

Una de las últimas incorporaciones es el fichero **claro_main.lib.php**, el cual se encuentra en la carpeta inc/lib/ que tiene unas cuantas funciones de carácter general, y el fichero **main_db.lib.php**, el cual contiene funciones de consulta a la base de datos (y que se encuentran sin testear en este momento). Estos ficheros se están ampliando constantemente. Hay también otras librerías por ejemplo para manipulación de texto, para el trabajo con ficheros...

Por ejemplo, hay una función general para mostrar la título de herramientas, hay una función para mostrar la opción [vista de estudiante / vista del administrador del curso], para mostrar una caja de dialogo en html, etc.... No hay funciones de base de datos específicas todavía, solo hay algunas de propósito general, pero es un área del desarrollo en el cual estamos trabajando para mejorarla desde ahora. Si trabajas en Dokeos, y necesitas una consulta que devuelve, por ejemplo, las propiedades de un grupo, deberías añadir una función `getGroupProperties` (con los parámetros que necesitase) en el fichero `claro_main.lib.php` y hacer a posteriori la llamada a la función. Esto representa un poco más de trabajo al principio, pero hará que el código sea más pequeño y más fácil de mantener. Además esto también significa que solo hay un lugar en donde habrá que cambiar la consulta si es necesario realizar algún cambio en ella no tendremos que actualizar 700 copias de la misma consulta.

Librería Principal (Main)

Las siguientes funciones están disponibles dentro del fichero `claro_main.lib.php`. Este fichero se encuentra además incluido en cada fichero de Dokeos. Para ser más precisos, se incluye cuando incluyes el fichero `claro_init_global.inc.php`.

Funciones en el API de ficheros:

```
//DATABASE - BASE DE DATOS
```

```
function claro_sql_query($sqlQuery, $dbHandler = '#')
```

```
function claro_sql_query_fetch_all($sqlQuery, $dbHandler = '#')
```

```
//FAILURE MANAGEMENT - ADMINISTRACIÓN DE ERRORES/FALLOS
```

```
function claro_set_failure($failureType)
```

```
class claro_failure -> function set_failure($failureType)
```

```
class claro_failure -> function get_last_failure()
```



```
//TOOL FUNCTIONS - FUNCIONES DE HERRAMIENTAS  
function claro_disp_tool_title($titleElement)  
function claro_disp_tool_view_option($isStudentView = false)  
function claro_disp_message_box($message)  
function is_allowed_to_edit()
```

4. Estructura de la base de datos de Doleos

Dokeos puede ser instalado usando una única base de datos o bien varias. Normalmente la opción de base de datos múltiple es la opción utilizada. Hay dos bases de datos principales, una para mantener la información de carácter general como usuarios, cursos y facultades; y otra para las estadísticas. Por otro lado, cada curso dispone también de su base de datos.

Cualquier persona podría dibujar el modelo E-R, yo no he encontrado tiempo para hacerlo todavía. Por ahora, puedes aprender mucho examinando la estructura de la base de datos usando PhpMyAdmin.

Un comentario adicional que hay que hacer es sobre algunas palabras que aparecen en Dokeos: las versiones iniciales fueron desarrolladas por personas que hablaban francés, este es el motivo por el cual hay algunas que aparecen aquí y allí (cours en vez de course, faculte en vez de faculty...)

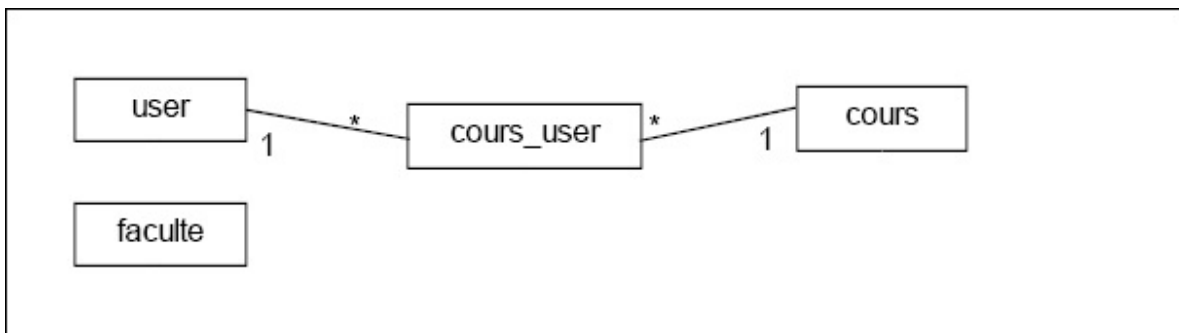
Hay un rumor que dice que la última versión de PhpMyAdmin puede crear diagramas E/R automáticamente. Lo investigaremos...

Un usuario puede seguir varios cursos.

Un curso puede tener muchos usuarios.

Tenemos definida una tabla extra que relaciona usuarios y cursos: cours_user.

Un curso pertenece exactamente a una facultad, y una facultad puede tener muchos cursos.





5. Convenciones de código propuestas

Lenguaje

El código, comentarios y las instrucciones de desarrollo deberán de estar escritas en inglés y no en el lenguaje nativo del desarrollador (por ejemplo español o francés). Es una gran idea que los manuales se encuentren disponibles en muchos lenguajes pero el código con los comentarios y variables deben estar en inglés, el francés o el español son más complicados de comprender.

Espacios en blanco

Añadir una línea en blanco entre líneas de código para crear párrafos y ayudar a visualizarlo. Nadie querría leer un libro de miles de páginas y líneas y sin separaciones entre estas, del mismo modo a nadie le gusta leer código escrito de esta manera.

Igualmente, usa espacios alrededor de los operadores y después de la coma, ya que puede ayudar a hacer el código más legible.

```
function something (param a, paramb)
if ( ( a && c) == !b)
a = b + c;
```

Indentacion

Utiliza los tabuladores para indentar. Generalmente, un tabulador equivale a cuatro espacios en blanco.

Llaves

Por favor, utiliza esta notación

```
if ($wwwww > $jjjjj)
{
código...
}
```

En vez de esta otra

```
if ($wwwww > $jjjjj) {
código ...
}
```



Variables y funciones

- Usaremos el guión bajo para separar palabras en vez de poner en mayúsculas la primera letra de cada palabra después de la primera.
- Las variables y las funciones se nombran de este modo:

```
$my_variable  
function test_database_api()
```

y no de este otro:

```
$myVariable  
function testDatabaseApi()
```

- Si es posible, intenta limitar el nombre de las variables a no más de tres palabras.
- Intenta nombrar las variables utilizando los nombres en singular en vez de en plural y teniendo en cuenta el contexto.

Espacios de nombres (Namespaces)

Por favor intenta evitar el uso de variables globales directamente. PHP carece hasta la fecha del concepto de espacios de nombres (namespaces): cada variable usada es global y puede ser vista en el programa entero (excepto en las funciones). Para evitar problemas, tienes que pensar en el diseño de tu programa. Haz un mayor uso de las funciones y/o objetos, e intenta evitar el uso de variables globales directamente en tú código. No utilices 700 veces la variable `$global_some_var`, mejor utiliza la función `get_global_var()`. De este modo el programa será más fácil de mantener y de resolver errores.

Constantes

- Las constantes deben de aparecer en MAYUSCULAS y si es aplicable `UN_NOMBRE_LARGO_IRA_ASI`

Clases

- Los nombres de las clases deben empezar con la primera letra en mayúsculas (Usuario, y no usuario).



Comentarios

Los comentarios deben de hacer el código más fácil de leer, no hay que explicar lo que es obvio, se asume que el lector dispone de un conocimiento del lenguaje de programación.

- usar el estilo de Perl como comentarios #
- Siempre usar el estilo del C como comentario (`/* */`)
- Usar el estilo de C++ cuando el comentario sea muy pequeño (`//`). Estos comentarios deberían aparecer cerca del código al que se refieren, siempre que sea posible.

Cada fichero de debe de empezar con el siguiente encabezado, si contribuyes a un fichero, añade tu nombre a la lista de contribuidores de ese fichero

```
/*
-----
-
Dokeos - elearning and course management software
Copyright (c) 2004 Dokeos S.A.
Copyright (c) 2003 University of Ghent (UGent)
Copyright (c) 2001 Universite catholique de Louvain (UCL)
Copyright (c) Jean Programmer (jean@somewhere.net)
For a full list of contributors, see "credits.txt".
The full license can be read in "license.txt".
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
See the GNU General Public License for more details.
Contact: Dokeos, 181 rue Royale, B-1000 Brussels, Belgium,
info@dokeos.com
-----
*/
```

Las secciones están estructuradas acorde a la convención siguiente:

```
/*
=====
LEVEL 1
=====
*/
/*
-----
LEVEL 2
-----
*/
```



Para las clases y las interfaces de las funciones, usa las convenciones de documentación de Java, así podremos obtener estos comentarios más tarde cuando realicemos la construcción de la documentación con el programa PHP Documentor (<http://pear.php.net/package/PhpDocumentor>).

Los detalles de la documentación Java pueden encontrarse en el documento:
How to Write Doc Comments for the Javadoc Tool
<http://java.sun.com/j2se/javadoc/writingdoccomments/index.html>

Un ejemplo de comentario en formato JavaDoc:

```
/**
 * Draws as much of the specified image as is currently available
 * with its northwest corner at the specified coordinate (x, y).
 * This method will return immediately in all cases, even if the
 * entire image has not yet been scaled, dithered and converted
 * for the current output device.
 *
 * If the current output representation is not yet complete then
 * the method will return false and the indicated {@link ImageObserver}
 * object will be notified as the conversion process progresses.
 *
 * @author Sami Shaio
 * @author Arthur van Hoff
 *
 * @param img the image to be drawn
 * @param x the x-coordinate of the northwest corner of the
 * destination rectangle in pixels
 * @param y the y-coordinate of the northwest corner of the
 * destination rectangle in pixels
 * @param observer the image observer to be notified as more of the
 * image is converted. May be null
 * @return true if the image is completely loaded and was
 * painted successfully;
 * false otherwise.
 * @see Image
 * @see ImageObserver
 * @see imageWrite
 * @since JDK1.0
 * @deprecated imageWrite performs it faster
 */
function drawImage($img, $x, $y, $observer);
```



6. Plantilla para nuevos desarrollos (add-ons)

Utiliza la siguiente plantilla para iniciar un nuevo desarrollo con tu propio código. Esta plantilla puedes encontrarla dentro de la carpeta “plugin” en el directorio donde se encuentre instalado Dokeos. Los extraños símbolos que aparecen al principio “# \$Id\$” son directrices para el CVS - esto significa que el gestor de código puede añadir versiones e información en este punto.

```
<?php # $Id$
/*
-----
Dokeos - elearning and course management software
Copyright (c) 2004 Dokeos S.A.
Copyright (c) 2003 University of Ghent (UGent)
Copyright (c) 2001 Universite catholique de Louvain (UCL)
Copyright (c) Jean Programmer (jean@somewhere.net)
For a full list of contributors, see "credits.txt".
The full license can be read in "license.txt".
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
See the GNU General Public License for more details.
Contact: Dokeos, 181 rue Royale, B-1000 Brussels, Belgium,
info@dokeos.com
-----
*/
/*=====
DOKEOS INIT
=====*/
// name of the lang file which needs to be included
// 'inc.php' is automatically appended to the file name
// $langFile = "langFile";
@include("../inc/claro_init_global.inc.php"); // settings initialisation
/*
Optional extra http header
If you need to add some HTTP/HTML headers code
www.dokeos.com 9
*/
// $httpHeadXtra[] = "";
// $httpHeadXtra[] = "";
// ...
//
// $htmlHeadXtra[] = "";
// $htmlHeadXtra[] = "";
// ...
$nameTools = ""; // title of the page (comes from the language file)
$query_string=''; // used for the breadcrumb
```



```
// when one needs to add a parameter after the
filename
@include($includePath."/claro_init_header.inc.php");
/*=====
MAIN CODE
=====*/
// PUT YOUR CODE HERE ...
echo "<h1><center>Hello world!<center></h1>";
/*=====
DOKEOS FOOTER
=====*/
@include($includePath."/claro_init_footer.inc.php");
?>
```

Añadiendo JavaScript a la cabecera Html de Dokeos

Si deseas incluir JavaScript, debes de añadirlo dentro de la variable *\$htmlHeadExtra*, la cual debe de estar establecida antes de incluir la cabecera de Dokeos. Si utilizas comillas dobles en tu JavaScript, es una buena idea utilizar comillas sencillas para tus variables. De otro modo, tendrás que convertir todas las comillas dobles en tu JavaScript añadiendo la barra (backslash) delante de ellas: `\` .

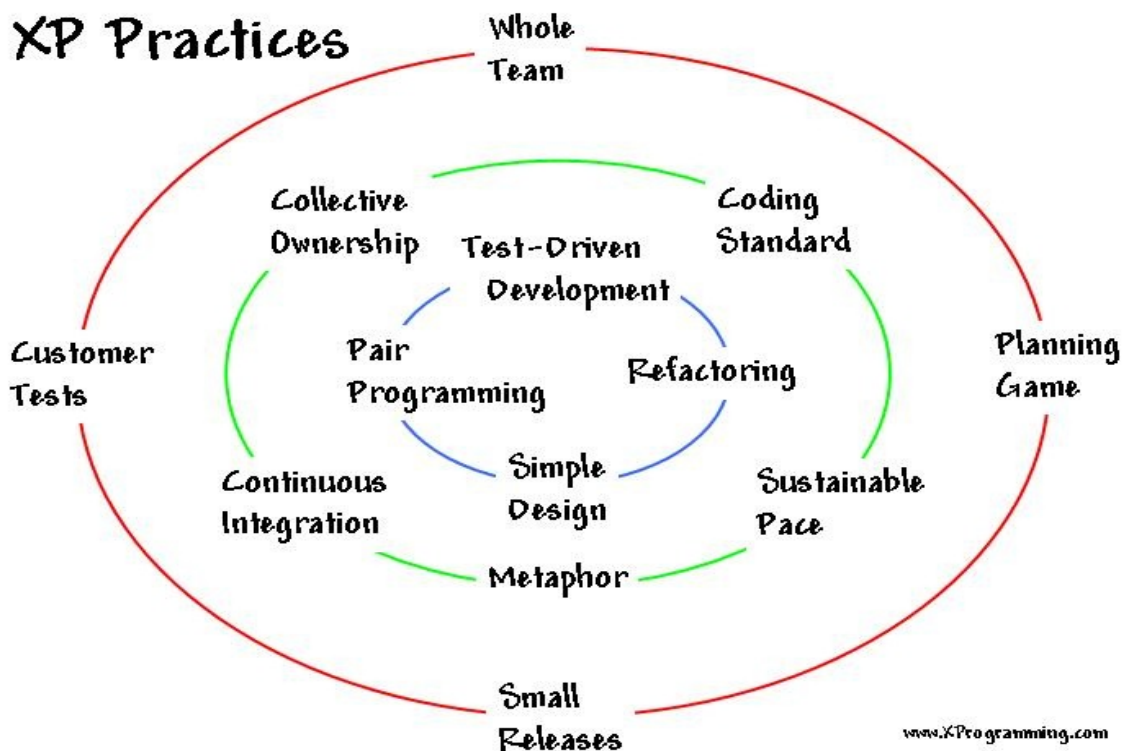
Advertencia: usar comillas simples significa que no podrás usar tus variables php en el código Javascript, como pueden ser las variables del lenguaje. Si las necesitas, entonces tendrás que usar comillas dobles y usar la barra invertida.

```
$htmlHeadXtra[]='<script type="text/javascript" language="JavaScript">
<!-- //Begin my fantastic javascript code
function helloworld(param1, param2) {
[...]}
// End -->
</script>
';
```

7. Método de trabajo & metodología de diseño

El software se desarrolla acorde a una determinada forma de trabajar que se denomina metodología. Una metodología puede respetar una cierta filosofía o conjunto de ideas acerca del desarrollo del software. Algunas metodologías son muy estrictas – las cuales pueden ser necesarias por ejemplo en el desarrollo del software de las naves espaciales – otras sin embargo, son muy simples y flexibles. Dokeos es desarrollado siguiendo la metodología de Programación Extrema (*Extreme Programming* o XP) (para una introducción ver [Qué es XP](#)). Esta metodología es simple y flexible.

Además, Dokeos es un programa de software libre. El software libre y el software Open Source software funciona tan bien porque anima a todo el mundo a participar en él. Para conseguir esto, es necesario comunicarse a menudo; y liberar las modificaciones tan pronto como sea posible: corrección de errores, nuevas piezas de código, una nueva herramienta, nueva documentación... Liberar incluso cuando algo no está completo del todo todavía, puede resumirse en: **liberar pronto, liberar a menudo**. La comunidad de usuarios te ayudará a resolver los problemas. Con el tiempo se puede crear una pieza perfecta de código, por lo que no intentes hacerlo todo solo, con la ayuda de la comunidad conseguiremos este objetivo. Esta aproximación además elimina los errores más rápidamente que otras aproximaciones. Se puede leer más acerca estos temas en e famoso trabajo de Eric S. Raymond's, *The Cathedral and the Bazaar*.





8. Comunicación entre desarrolladores

La cooperación entre los desarrolladores puede hacer o romper el desarrollo del software. La comunicación, es un elemento importante de la cooperación, y es más difícil cuando los desarrolladores están geográficamente separados. Por tanto, una buena cooperación requiere de un buen modelo de comunicación.

En el software libre, y en la programación extrema, el código no pertenece a nadie, y todos debemos aceptar que su código puede ser cambiado y criticado. Dokeos tiene un proyecto en sourceforge, y se utiliza para el CVS. CVS es un sistema para almacenar ficheros de código, junto con un histórico de las versiones realizadas, por lo que se hace necesaria una buena cooperación. Cada desarrollador tiene acceso de lectura al CVS, y un pequeño “núcleo” de desarrolladores tiene acceso de escritura.

¿Como se comunican los desarrolladores? Los desarrolladores actuales están de acuerdo en utilizar un foro web como medio de comunicación. El uso de un foro web tiene muchas ventajas: es fácilmente accesible, toda la información está cuidadosamente organizada, y todo el mundo puede discutir sobre el desarrollo y otros aspectos que se quieran. La concentración de toda la información en único medio tiene la ventaja de no causar los problemas que aparecen cuando se usa por ejemplo un foro, un grupo de noticias o el email: algunas personas solo miran en un único sitio, y no miran suficientemente en los otros...

La comunicación con el mundo exterior puede ocurrir de diferentes formas, aunque las preferidas suelen ser el foro y el sitio web.



9. Desarrollo del Software en general

Tenemos muy pocos conceptos importantes sobre el desarrollo del software. Estos han sido establecidos después de años de práctica donde las estrategias claves que han tenido éxito o han fracasado han sido evaluadas. Estas ideas pueden ayudar en la elaboración de un software mejor.

Ocultar la información

La encapsulación de la información es quizás el aspecto más importante en el desarrollo del software. Los lenguajes orientados a objetos son los mejores para hacer esto, pero es posible también hacerlo en otros lenguajes. La idea es que trates de ocultar todas las cosas que no se deben de ver desde el exterior.

Por ejemplo, las variables locales en una función no pueden ser vistas desde fuera de la función en donde están definidas. Las variables de clase (también llamadas variables miembros) no pueden ser vistas desde fuera de los objetos de la clase.

Diseño en capas

El código debe de estar diseñado en varias capas. En las aplicaciones de escritorio (standalone) típicas a menudo vemos un diseño de tres capas: una capa con todo el código relativo a la base de datos, the user interface and the code that does the real work, the “business” layer. Esto ayuda a mejorar y mantener el código a posteriori. Si queremos cambiar la apariencia que muestra el programa, no hay que hacer cambios en la capa de la base de datos, de igual modo, si cambiamos algún cálculo, no tenemos que realizar modificaciones a nivel del interface gráfico. Dokeos está también dividido en tres capas: el código del núcleo con el corazón de Dokeos, el código de herramientas, y el - muy joven - capa API la cual enlace entre las herramientas y el núcleo (kernel) y proporcionar una manera a las herramientas para hablar con el código del núcleo.

Cuando aplicarlo

Las técnicas de desarrollo de software deberían de verse como “una herramienta, no un martillo”. Es decir, hay que usarlas cuando sean aplicables a una cierta situación y cuando obtengamos una ventaja de su uso. No hay que usarlas por el mero hecho de decir que lo estamos haciendo. Aparte de los aspectos filosóficos, los cuales son muy interesantes, el desarrollo de software se resume al final en una única cosa: - software que funciona.



10. Diseño de la Interface de Usuario

Unas pocas reglas al respecto:

- **Una interface de usuario está bien diseñada cuando el programa se comporta exactametne como el usuario piensa que lo hará.**
- A más opciones, más complejo el interface de usuario llega a ser.
- El programa debe de verse y sentirse elegante, además de proporcionar todas las herramientas para que el trabajo pueda ser realizado.

En aplicaciones del tipo standalone, a menudo se pueden poner opciones en una barra de herramientas y poner el resto en un menú. Sin embargo, la mayoría de las aplicaciones web no tienen una barra de menú, por lo que hay que pensar sobre que características son las que llegarán a ser importantes.

Cada vez que se proporciona una opción, estamos preguntando al usuario para que tome una decisión. Esto significa que tendrá que pensar sobre ello y decidir que hacer, aunque no es realmente una “mala” cosa, en general se debe tratar de minimizar el número de decisiones que la gente tenga que tomar.

Esto no significa eliminar toda opción. Hay muchas decisiones que el usuario tendrá que tomar de todos modos: la apariencia de sus documentos, o el comportamiento que el sitio web tendrá, y todos los demás aspectos que están integrados con el trabajo que el usuario está haciendo. — *Joel Spolsky, [User Interface Design for Programmers](#), Chapter 3, 2000*

11. Entrada del Usuario

Últimamente, son los usuarios quienes deciden en que dirección debe Doleos continuar. Esto, por supuesto no excluye las sugerencias y puntos de vista de los desarrolladores, pero el software se hace para los usuarios no para los programadores (Aunque en el caso del software libre, a menudo se da el caso de que el usuario es también el programador). **El objetivo último de un entorno de e-learning como Doleos es ayudar a las personas ha aprender mejor.**



12. Promoción de Dokeos

Dokeos es una plataforma gratuita de e-learning y administración de cursos. En pocas palabras: Dokeos es bueno, cada vez mejor, fácil de usar y adaptable a tus propias necesidades. Cualquiera que sea la razón, es una buena idea estar seguros que la base de usuarios de Dokeos es tan amplia como sea posible. A más usuarios, mayor feedback para los desarrolladores. Indicaciones de errores, mejoras para el usuario, nuevas características... cada contribución cuenta.

Cuanto más programadores, más estable será el código y su proceso de mantenimiento. Todo el mundo puede echar un vistazo al código, modificarlo y redistribuirlo. El desarrollo mediante Open Source es superior a los viejos desarrollos cerrados.

Una de las grandes ventajas del software libre es que es capaz de ayudar a la gente a cooperar en la consecución de un objetivo común. Si piensas que Dokeos es, o puede ser, una buena herramienta, no vaciles en informar a instituciones educativas, directores de e-learning, desarrolladores de software... de nuestra existencia, y dirigirlos a nuestra página web.



13. Enlaces de Interes

General

[PHP.net](http://www.php.net) y el manual de [PHP](http://www.php.net/docs.php) (<http://www.php.net/docs.php>)

The [Pear](http://pear.php.net/) collection of reusable PHP components (<http://pear.php.net/>)

Software Libre & Opensource

[GNU.org](http://www.gnu.org)

The [Cathedral and the Bazaar](http://catb.org/~esr/writings/cathedral-bazaar/) (<http://catb.org/~esr/writings/cathedral-bazaar/>)

Dokeos

[Dokeos.com](http://www.dokeos.com)

[Forum](http://www.dokeos.net/forum/) (<http://www.dokeos.net/forum/>)

[SourceForge](http://www.sourceforge.net/projects/dokeos/) (<http://www.sourceforge.net/projects/dokeos/>)

Introducción a la XP

[What is XP](#)

Patrones de Diseño

[Portland Pattern Repository](#)

[Patterns FAQ](#)

Christopher Alexander es un arquitecto (de ciudades y edificios) fue quien invento los patrones. Para más información consulta: [Notes on Christopher Alexander](#).



14. GNU Free Documentation License

NOTA IMPORTANTE: La licencia GNU Free Documentation License no dispone de traducción oficial al idioma español, por lo que a efectos de este manual se deja la versión original en inglés que es la que dispone de la validez jurídica adecuada. No obstante, si se desea una traducción no oficial de la misma, se puede consultar la siguiente dirección: <http://gugs.sindominio.net/licencias/>

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language. A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections.

If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this



License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only. The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3. You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts:

Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.



If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item skating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.



M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work. In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.



If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail. If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may use any version ever published (not as a draft) by the Free Software Foundation.

[How to use this License for your documents](#)

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this: with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

15. Modificar este manual

Si quieres realizar alguna modificación o mejora en este manual necesitaras una copia del mismo en formato OpenOffice o Ms-Word, puedes solicitarlos en la siguiente dirección:

info@dokeos.com
Dokeos
181 rue Royale
B-1210 Brussels
Belgium, Europe
Tel. 0032 2 227 00 26