



DOI: <http://dx.doi.org/10.21501/21454086.2065>



Licencia Creative Commons Atribución-No Comercial-Sin Derivar 4.0 Internacional

Lámpsakos | No. 20 | pp. 40-53 | julio-diciembre | 2018 | ISSN: 2145-4086 | Medellín-Colombia

Frameworks para el desarrollo de prototipos WEB: Un caso de aplicación

Frameworks for the development of WEB prototypes: An application case

Víctor Daniel Gil Vera, MsC

Universidad Católica Luis Amigó
Medellín, Colombia

Victor.gilve@amigo.edu.co

Claudio Rogério Gomes Da Silva, MsC

Universidade Federal do Pará
Belém, Brasil

claudiorogerio001@gmail.com

Juan Carlos Gil Vera

Universidad Nacional de Colombia
Medellín, Colombia

jcgil@unal.edu.co

Jule Teutsch

Universität Passau
Baja Baviera, Alemania

juleteutsch.munchen@gmail.com

(Recibido el 12-12-2016, Aprobado el 18-04-2018, Publicado el 03-07-2018)

Estilo de Citación de Artículo:

V. D. Gil Vera, J. C. Gil Vera, C. R. Gomes, and J. Teutsch, "Frameworks para el desarrollo de prototipos web: un caso de aplicación", Lámpsakos, no. 20, pp 40-53, 2018

DOI: <http://dx.doi.org/10.21501/21454086.2065>

Resumen: Los *Frameworks* para el desarrollo de aplicaciones web facilitan el trabajo de desarrolladores y profesionales de la industria informática. Tareas que requieren conocimientos y técnicas avanzadas de programación pueden realizarse con facilidad y en poco tiempo, suprimiendo la tediosa y repetitiva tarea asociada con patrones comunes de desarrollo web. Este trabajo tiene como objetivo presentar el desarrollo e implementación de un prototipo web que maneja el dominio de los cursos y los docentes del Departamento de Ciencias de la Computación y la Decisión de la Facultad de Minas - Universidad Nacional de Colombia Sede Medellín con el *Framework* PISIS y programación orientada a objetos en PHP. Con este trabajo se concluye que el uso de *Frameworks* ayuda a disminuir errores de invalidez y permiten que los desarrolladores puedan centrarse en la escritura de su aplicación.

Palabras clave: Smarty, Plantilla, Frameworks, Prototipo, Clase, Objeto.

Abstract: The Frameworks for the development of web applications facilitates the work of developers and professionals of the computer industry. Tasks that require knowledge and advanced programming techniques can be done with ease and in a short time by eliminating the tedious and repetitive task associated with common patterns of web development. This paper presents the

development and implementation of a web prototype that manages the domain of the courses and the teachers in the Department of Computer Science of the Faculty of Mines - National University of Colombia - Medellín with the Framework PISIS and object oriented programming in PHP. This paper concluded that the use of frameworks help reduce errors of invalidity and allow developers can focus on writing your application.

Keywords: Smarty, Template, Frameworks, Prototype, Class, Object.

1. INTRODUCCIÓN

Las metodologías ágiles se orientan al desarrollo rápido de aplicaciones de software, apoyándose en prototipos ejecutables que evolucionan hasta la aplicación definitiva, a medida que se descubren e incorporan los requisitos del área de la aplicación. Para facilitar este tipo de desarrollo es necesario contar, por un lado, con una arquitectura para el software que garantice la mantenibilidad, y por el otro, con herramientas automatizadas que den apoyo a la construcción, prueba y depuración del software en elaboración.

Los *Frameworks* han mostrado ser herramientas útiles para dar apoyo al proceso de construcción de software, debido a que impulsan la reutilización del código, al prescribir y soportar una arquitectura estandarizada que garantiza su mantenibilidad. En el grupo de ingeniería del software del Departamento de Ciencias de la Computación y de la Decisión de la Facultad de Minas, Universidad Nacional de Colombia, Sede Medellín, se ha analizado la viabilidad de complementar el método de desarrollo UNC-METODO con metodologías ágiles a través del uso de *Frameworks*.

Con este fin se experimentó con algunos *Frameworks* de uso libre disponibles en la web [1] y se inició la construcción de un prototipo de *Framework* con las características que mejor se adaptaban al método. Al *Framework* desarrollado se le denominó PISIS_FRAMEWORK, el cual ha sido empleado como soporte para la reconstrucción de la página PISIS, que constituye la intra-web del departamento antes referido.

Un aspecto clave del PISIS_FRAMEWORK es que soporta una estrategia de desarrollo que privilegia el prototipado de las interfaces de usuario sobre la definición de la base de datos, en contraste con los *Frameworks* analizados. Es así que para obtener interfaces activas, no es necesario contar previamente con la Base de Datos; retrasando el diseño de esta última, para cuando exista claridad sobre la forma en que será usado el software en el área de aplicación.

En este artículo se presenta la manera como el PISIS-FRAMEWORK facilita la definición de las interfaces de usuario, en el marco de la estructura de navegación de la página web PISIS. El *Framework* utiliza plantillas *smarty* para presentar las vistas de la aplicación por medio de pestañas que facilitan la navegación, este enfoque de diseño enriquece el mantenimiento de la misma al crear un mecanismo de edición para manejar dicho contenido. En el artículo se presentan los siguientes aspectos del *Framework*: estructura, edición de datos, construcción de páginas web, sugerencias para trabajos futuros y por último las conclusiones obtenidas.

2. FRAMEWORKS PARA LA CREACIÓN DE APLICACIONES WEB

Algunas de las definiciones de *Frameworks* para la creación de aplicaciones Web encontradas en la revisión de la literatura hacen referencia a:

- Soluciones de alto nivel para la reutilización de piezas de software, permiten reutilizar funciones comunes y lógica genérica en un dominio de aplicación. Estos garantizan calidad del producto final, ya que una parte importante ha sido construida, probada y validada [2].
- Conjunto de códigos predefinidos o bibliotecas que proporcionan funcionalidad común a toda una clase de aplicaciones [3]. Puede ser visto como una base o esqueleto para construir aplicaciones o sitios web [3].

- Estructuras para la creación de aplicaciones web que permiten al programador desarrollar aplicaciones con interfaces interactivas y también para que la aplicación se ejecute en una máquina que está controlada por el proveedor de la aplicación [4]

Diversos *Frameworks* basados en diferentes lenguajes de programación están disponibles en la web para su libre utilización, como es el caso de: JSF, Strutts, Ruby on Rails, Grails, Cake PHP, Django y Catalyst [5]. A continuación se describe brevemente cada uno de ellos:

- Java Server Faces (JSF): framework para la creación de contenido web dinámico basado en Java, el cual emplea Java Server Pages (JSP) para la visualización de páginas [6].
- Apache Strutts: *framework* Modelo - Vista-Controlador (MVC) de código fuente abierto para la creación de elegantes y modernas aplicaciones web Java [7].
- Ruby on Rails (RoR): software de código abierto que permite a programadores desarrollar aplicaciones utilizando únicamente RoR, sin necesidad de utilizar otros lenguajes de programación tales como JavaScript, C ++ o HTML [8]
- Grails: potente *framework* de código fuente abierto de desarrollo web construido sobre el lenguaje de programación Groovy para la plataforma Java, destinado a multiplicar la productividad de los desarrolladores [9].
- Cake PHP: es uno de varios entornos de aplicaciones de código abierto como Ruby, Zend o symfony que permite la rápida generación de una interfaz gráfica de usuario basada en web para una base de datos SQL, está escrito en PHP y es distribuido bajo la licencia del Instituto tecnológico de Massachusetts MIT [10].

- Django: *framework* web de Python de código fuente abierto de alto nivel que fomenta el rápido desarrollo y el diseño limpio y pragmático [11].
- Catalyst: *framework* para la construcción de aplicaciones web escrito en Perl, el cual soporta la arquitectura Modelo - Vista -Controlador (MVC) [12].
- Symfony: conjunto de componentes de PHP para crear sitios y aplicaciones web basad en el patrón (MVC) [13].

3. ESTRUCTURA DE LA PÁGINA WEB DE PISIS

La página de PISIS está conformada por una serie de vistas que tienen una estructura que se repite de forma sistemática. Las vistas de PISIS tienen una cabecera, un cuerpo principal y un pie de página, Fig 1.



Fig 1. Vista de la Sección de un Docente. Fuente: PISIS

El cuerpo principal presenta los datos que identifican el "objeto base de la vista", en este caso un docente. Luego presenta un conjunto de pestañas correspondientes a información más específica del objeto base. Como son la pestaña "cursos" que lista los cursos a cargo del docente. La lista de cursos contiene una serie de columnas con la información básica de cada curso. La cabecera está conformada por un encabezado que contiene los logos de la

universidad, la facultad y el departamento al que pertenece la información; una sección de identificación para el usuario, que muestra su rol dentro de la página y los campos de autenticación; y finalmente, una sección con enlaces de navegación, que en la figura, señalan a un docente y sus cursos como el contenido del cuerpo principal, Fig. 2.

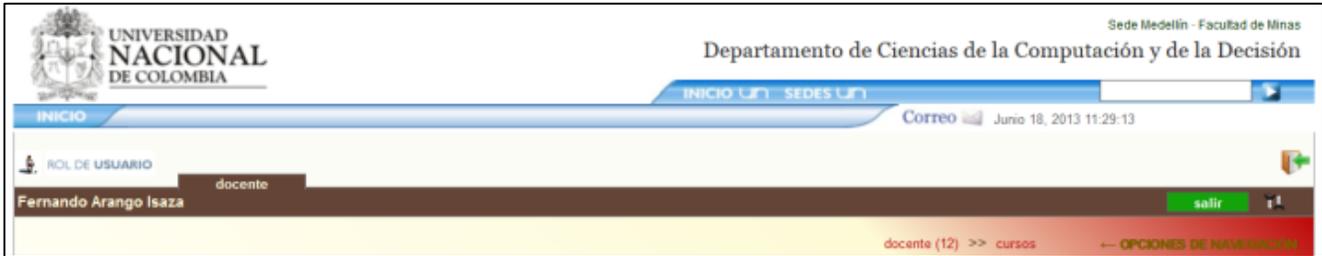


Fig. 2. Cabecera de la página de la Sección de un Docente. Fuente: PISIS

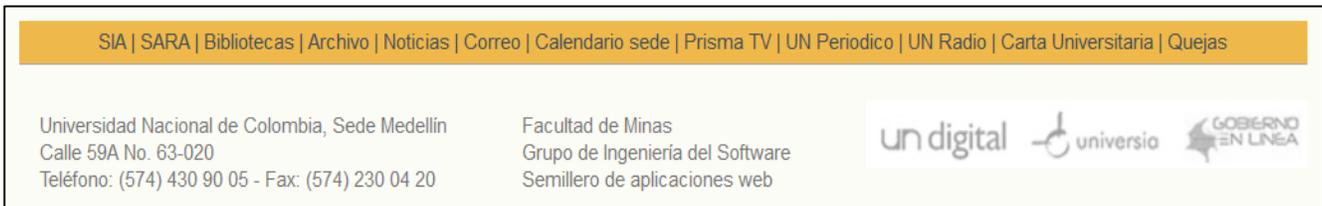


Fig.3. Pie de página de la página de la sección de un Docente. Fuente: PISIS

El pie de página está integrado por un conjunto de links de navegación y de información que aluden a otros sitios de interés en la página de la universidad, Fig. 3.

Desde la lista de cursos se puede navegar a la información de cualquiera de los cursos al presionar doble clic sobre el curso de interés, induciendo el despliegue de la vista que contiene a dicho curso como su objeto base. Al igual que en la vista anterior se presentan primero los datos que identifican el objeto base y luego una serie de pestañas correspondiente a datos más específicos del objeto visualizado. En la Fig. 4, se presenta la selección del curso "Lenguajes Lógicos y Funcionales".

Además de los datos básicos del curso se visualiza la información correspondiente a la pestaña "secciones" que contiene información relativa a las actividades, temas y documentos que componen el curso. Para visualizar otra información diferente del curso se debe dar clic sobre la pestaña correspondiente. A continuación se presenta el contenido de la pestaña "bibliografía", que visualiza los documentos que constituyen el material de lectura del curso.

Similarmente a la lista de cursos, la lista de documentos contiene una serie de columnas con la información básica de cada documento. Los datos de cada columna se denominan "atributos". Al igual que para los cursos, para visualizar de forma más detallada los datos de un documento basta con seleccionar la fila del documento y dar clic. Al navegar por la página e ir avanzando sobre los objetos de la vista, se va cambiando de faceta y la barra de navegación se va actualizando mostrando el camino de navegación. Para devolverse a cualquier lugar del camino es suficiente con dar clic sobre el lugar de navegación al cual se desea retornar.

Para visualizar los datos, el *Framework* crea un árbol de navegación conformado por el objeto raíz del árbol que será el objeto principal desde donde el *Framework* iniciará la navegación y la visualización de la página. La Fig. 6, presenta la jerarquía de los objetos que estructuran la aplicación.

3007743: Lenguajes Lógicos y Funcionales
 2012-3 / Mi-V 10-12 / M8-201

curso secciones bibliografía documentos estudiantes grupos pruebas ← OPCIONES

Datos Propios del Curso

dato	valor
nombre	Lenguajes Lógicos y Funcionales
horario	Mi-V 10-12
lugar	M8-201
matr	
docente	12
asignatura	3007743
grupo	1
año	2012
semestre	3

Fig.4. Vista del Curso Lenguajes Lógicos y Funcionales. Fuente: PISIS

UNIVERSIDAD NACIONAL DE COLOMBIA Facultad de Minas Escuela de Sistemas

Junio 04, 2012 09:32:48

ROL DE USUARIO visitante

[No ha ingresado al sistema] Usuario Contraseña **ingresar**

docentes (8305562) >> cursos (1,3007743,2011,3) >> bibliografía OPCIONES DE NAVEGACION

3007743 : Lenguajes Lógicos y Funcionales
 Semestre: 2011-3 / Horario: Mi-V 10-12 a.m. / Lugar: M8-201

secciones **bibliografía** estudiantes pruebas grupos OPCIONES

Material Bibliográfico del Curso

Id	Autor Año	Titulo	Editorial	Publicacion				Fecha	
				ISBN/ISSN	Pag. Ini	Pag. Fin	Num Vol		Num Ejem
<input type="checkbox"/>	[Abelson1985]	Structure and Interpretation of Computers Programms	The MIT Press	28676					1985
<input type="checkbox"/>	[Arango2008]	Aplicaciones de la Logica al Desarrollo del Software	En preparacion	1232					2008
<input type="checkbox"/>	[Grassmann1996]	Matematica Discreta y Logica	Prentice may, Madrid	84-89660-04-2					1996
<input type="checkbox"/>	[Branch2006]	Hoy dormire mejor que ayer	Revista Dyna		53	58	2	1	2006

SIA | Bibliotecas | Archivo | Agencia de Noticias | Correo | Calendario sede | Prisma TV | UN Periodico | UN Radio | Carta Universitaria | Sistema Quejas

Universidad Nacional de Colombia, Sede Medellín
 Calle 59A No. 63-020
 Teléfono: (574) 430 90 05 - Fax: (574) 230 04 20

Facultad de Minas
 Grupo de Ingeniería del Software
 Semillero de aplicaciones web

un digital universio GOBIERNO EN LINEA

Fig. 5. Vista de la Faceta bibliografía. Fuente: PISIS

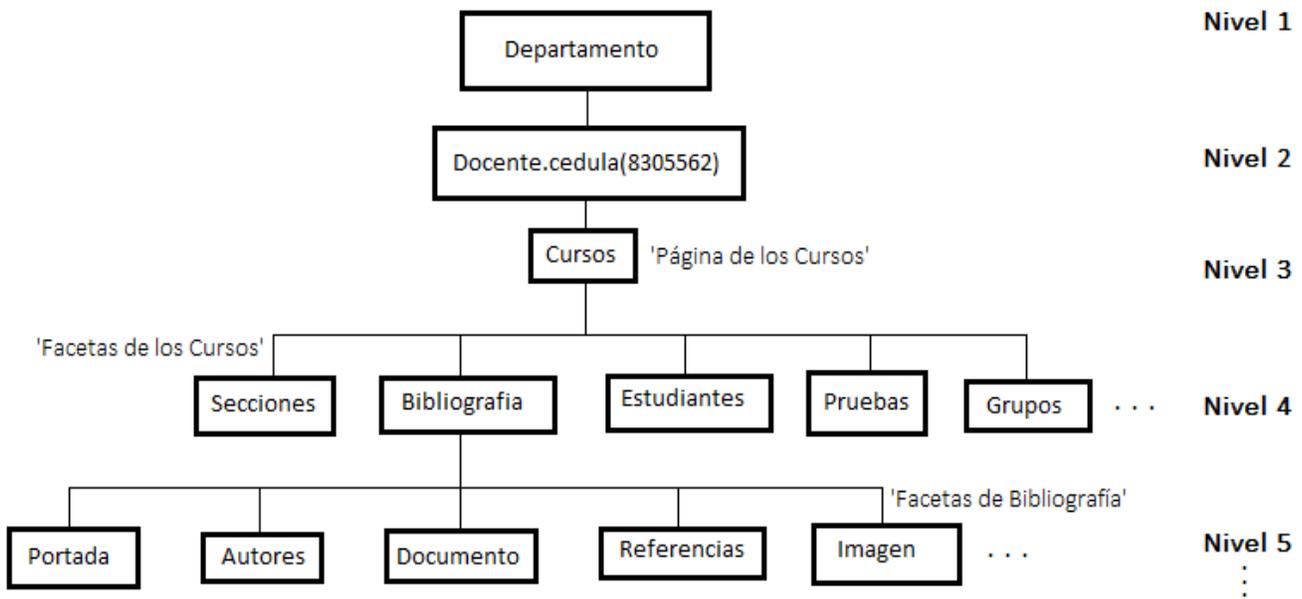


Fig. 6. Árbol de navegación que muestra el camino de navegación de la página a través de las clases. Fuente: elaboración de los autores

El árbol muestra que un objeto "Docente" tendrá una relación con un conjunto de objetos de Curso. De la misma manera un objeto "Curso" tendrá relación con varios conjuntos de objetos: "Secciones", "Bibliografía", "Estudiantes", "Pruebas" y "Grupos de estudiantes" que conforman el curso. A los conjuntos de objetos con los que se relaciona un objeto se denominan "facetas" de dicho objeto. Un objeto "Bibliografía", tiene como facetas "Portada", "Autores", "Documento", "Referencias" e "Imagen".

4. EDICIÓN DE DATOS CON PISIS FRAMEWORK

Una página web desarrollada con el *Framework* puede ser editada si el usuario tiene permisos de edición. Un docente puede ingresar en modo de edición y de consulta a la vista con las secciones de los cursos que tiene a su cargo; y en contraste, un estudiante únicamente puede ingresar a dicha página en modo de consulta. Para pasar del modo de consulta al modo de edición y viceversa, el usuario debe presionar el ícono situado al lado superior derecho de la vista que desea ver o editar. Para realizar una operación de edición en el *Framework*, la vista presenta un conjunto de iconos de edición, Fig. 7.

A continuación se explica la funcionalidad de cada ícono de izquierda a derecha:

1. Borra un objeto seleccionado
2. Modifica los valores de un objeto seleccionado
3. Selecciona como origen de copia a uno ó varios objetos que se desean copiar en otros lugares de la vista
4. Activa una vez se tengan seleccionados uno o varios objetos como origen de copia, y se usa para pegar los objetos origen de copia sobre los objetos seleccionados
- 5 y 6. Introducen nuevos objetos arriba ó debajo de los objetos previamente seleccionados
7. Guarda los cambios realizados
8. Deshace una operación de edición
9. Rehace un cambio antes deshecho
10. Desecha y cancela todos los cambios realizados.

Para editar objetos, la vista de la página presenta a la izquierda de las filas que representan los objetos, una caja de chequeo para seleccionar aquellos sobre los cuales se desea realizar una operación de edición, Fig. 8.

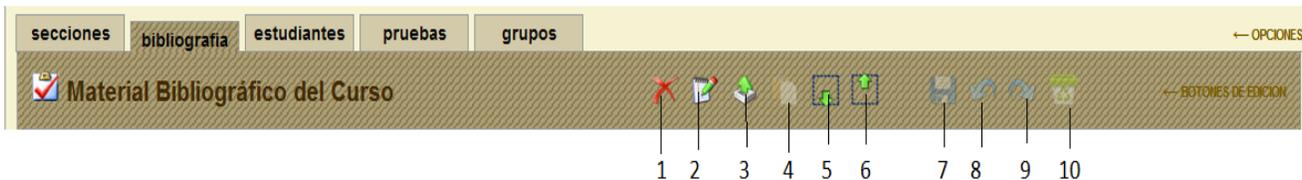


Fig.7. Vista del Menú de Edición con sus Iconos. Fuente: PISIS

Id	Autor Año	Titulo	Editorial	Publicacion					Fecha
				ISBN/ISSN	Pag. Ini	Pag. Fin	Num Vol	Num Ejem	
<input checked="" type="checkbox"/>	[Abelson1985]	Structure and Interpretation of Computers Programms	The MIT Press	28676					1985
<input type="checkbox"/>	[Arango2008]	Aplicaciones de la Logica al Desarrollo del Software	En preparacion	1232					2008
<input type="checkbox"/>	[Grassmann1996]	Matematica Discreta y Logica	Prentice Hall, Madrid	84-89660-04-2					1996

Fig. 8. Edición de objetos. Fuente: PISIS



Fig. 9. Eliminación de objetos. Fuente: PISIS

Para borrar uno o varios objetos, se deben seleccionar las "cajas de chequeo" a la izquierda y a continuación seleccionar el icono de borrar. La Fig. 9 y Fig. 10, presentan como borrar dos objetos de tipo documento:

Para editar uno o varios objetos, primero se deben seleccionar las "caja de chequeo" a la izquierda y a

continuación seleccionamos el icono de editar, lo que permite escribir en un campo de texto sobre el dato que deseamos editar. La Fig. 11, muestra la edición de dos objetos.

Luego de editar los campos deseados se presiona de nuevo el icono. En la Fig. 12, se presentan los resultados después de una operación de edición.

The screenshot shows the same PISIS interface as Fig. 9, but the table now only contains one entry, '[Abelson1985]', indicating that the other two entries have been successfully deleted.

Id	Autor Año	Titulo	Editorial	Publicacion					Fecha
				ISBN/ISSN	Pag. Ini	Pag. Fin	Num Vol	Num Ejem	
<input type="checkbox"/>	[Abelson1985]	Structure and Interpretation of Computers Programms	The MIT Press	28676					1985

Fig. 10. Resultado de la operación de borrar los dos objetos seleccionados después de seleccionar el icono de borrar. Fuente: PISIS

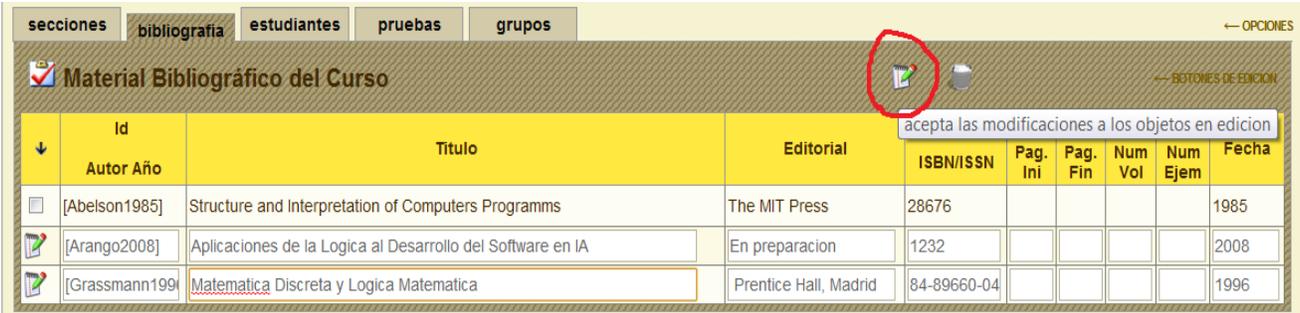


Fig. 11. Edición de los datos en dos objetos al seleccionar el icono de edición. Fuente: PISIS



Fig. 12. Resultado después de editar los títulos de los objetos en edición. Fuente: PISIS

4.1 Incorporación de nuevas facetas en un objeto y de vistas sobre nuevos objetos

En el PISIS_FRAMEWORK, tanto para visualizar nuevas facetas en los objetos de un tipo (del área de la aplicación), como para visualizar los datos y las facetas de los objetos de un nuevo tipo, es suficiente con especificar estos elementos en las clases asociadas con dichos objetos en el marco del FRAMEWORK.

A continuación se muestra primero la estructura de directorios del FRAMEWORK, en el marco del medio ambiente de desarrollo ofrecido por el SDK ECLIPSE para el lenguaje PHP, y luego se describe en detalle los elementos que conforman las clases de la capa del "modelo" del FRAMEWORK.

4.2 Jerarquía de directorios del PISIS - Framework

Una aplicación web desarrollada con el Framework de PISIS presenta la estructura de directorios mostrada en la Fig. 13.

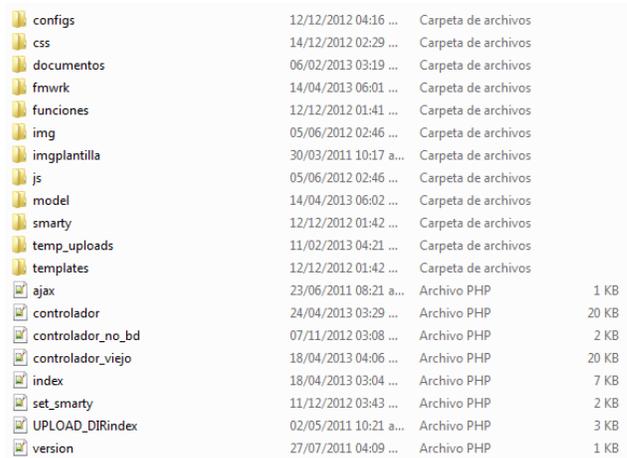


Fig. 13. Estructura de directorios en una aplicación web desarrollada con el Framework de PISIS. Fuente: PISIS

Este conjunto de directorios se encuentra guardado dentro de un directorio principal que se ha denominado "CURSOS", este es el directorio que se debe ubicar en el host del servidor web y que permitirá visualizar la aplicación web que se está desarrollando. Los archivos de mayor importancia para crear las vistas de una aplicación web son los archivos del directorio "modelo", donde se localizan las clases php para definir las clases que describen las características de los tipos de objetos del área de la aplicación que se mostrarán en la página.

5. CLASES DEL MODELO EN EL PISIS FRAMEWORK

Todas las clases del modelo en este *Framework* deben heredar de forma directa o indirecta de la clase objeto estándar "Obj_Std", esta clase se halla

localizada en el directorio "model" de la estructura de directorios del *Framework* y contiene toda la infraestructura que comparten las clases del modelo en la aplicación, Fig. 14.

```

class Obj_Std
{
// -----
// Atributos basicos
// Actualizacion: "D"= borrado, "E"= eliminado, "C"= cambiado, "I"= insertado
// Clave: array(<name1>=><valor1>,<name2>=><valor2>,...)
// -----
protected $Actualizacion=""; // registra si el objeto ha sido modificado
protected $Clave = array(); //todo objeto real debe tener un clave

// -----
// Componentes externas de la clase: Atributos derivados de las relaciones con otras clases
// -----

// -----
// Interfaz para consulta de los atributos de la clase
// -----
protected static $Clase_Clave = array();
protected static $Clase_Attr = array(/*'objeto' => '*' */);
protected static $Clase_Faceta = array();

// -----
// constructor de un objeto de la clase
// -----
function Obj_Std($Dt=NULL, //vector de datos del objeto
                $Dispositivo = NULL, //Gestor del dispositivo de almacenamiento
                $Componente = array() //arbol de componentes externas a ser incluidas
  )
  {
  }
}
  
```

Fig. 14. Clase Objeto Estándar "Obj_Std". Fuente: PISIS

Las clases de este *Framework* carecen de un constructor propio para instanciar los objetos, el constructor debe ser heredado de la clase objeto estándar ?Obj_Std? ya sea por herencia directa o por herencia de la clase padre a la que extienda cada clase. De la clase objeto estándar se heredan los atributos ?actualizacion? y ?clave?, que apoyan el proceso de actualización de la base de datos. Una clase tiene definido un conjunto de atributos básicos protegidos de tres tipos, Fig. 15. En primer lugar, están los atributos básicos que corresponden a los atributos miembros de la clase del diagrama de clases.

En segundo lugar, hay un conjunto de atributos denominados "facetas de la clase". Estos atributos se corresponden, en general, con los atributos derivados a través de las relaciones con otras clases. Estos atributos se consideran como conjuntos de objetos de las clases relacionadas y se guardan en arreglos para ser accedidos y manipulados desde de la clase.

```

<?php
class Documento extends Obj_Std
{
// -----
// Atributos basicos
// -----
protected $identificacion; //Una sigla para el autor principal seguida del año de publicacion todo entre [
protected $Titulo; //Titulo de la referencia.
protected $Fecha_publicacion; //Fecha de la publicacion de la referencia.
protected $url;
protected $editorial; //Nombre de la revista ó la Editorial.
protected $num_volumen;
protected $num_ejemplar;
protected $issn; //Para una revista.
protected $isbn; //Para un libro.
protected $tipo_publicacion; //libro, Revista, Tesis, Artículo.
protected $pagina_inicial;
protected $pagina_final;
protected $palabras_claves;
protected $resumen;
protected $prologo;
protected $imagen;
protected $descripcion;
protected $contenido;

// -----
// Facetas de la clase: Atributos derivados de las relaciones con otras clases
// -----
protected $autores = array(); //Apellidos y nombre de cada autor
protected $publicacion; //arreglo de instancias de Lectura
}
  
```

Fig. 15. Atributos Protegidos de una clase del *Framework* de PISIS. Fuente: PISIS

En tercer lugar, están los atributos derivados de la clase que son atributos cuyo valor se obtiene de diferentes operaciones entre los atributos de la clase. Como elemento sustancial para el *Framework*, es necesario que todas las clases tengan definido un conjunto de arreglos estáticos protegidos que le describen al *Framework* las características mismas de la cada clase. A estos atributos se les denomina metadatos de la clase y se presentan en las Fig. 16, Fig. 17, Fig. 18, Fig. 19 y Fig. 20. En los metadatos se define la clase padre a la que hereda la clase actual, los atributos de la clase padre se incorporan a esta clase directamente por herencia. A continuación se explican los cuatro atributos que conforman los metadatos de la clase:

5.1 Atributo clave del objeto

El primero de estos atributos es "Clase_Clave", que sirve para definir la clave del objeto y se corresponde con el atributo "Clave" de la clase objeto estándar. Para definirlo se emplea un array asociativo de la forma:

```
array(<id_valor1>=><tipo_valor1,  
id_valor2>=><tipo_valor2>...),
```

Donde id_valor1 es el nombre que identifica la componente i-esíma de la clave y tipo valor es el

tipo de dicha componente. Para el ejemplo de la Fig. 16, la clave está conformada por un solo valor de nombre "curso" y de tipo "integer".

```
protected static $Clase_Clave = array('curso'=>'*integer');
```

Fig. 16. Atributo clave de la clase. Fuente: PISIS

5.2 Array de atributos

El segundo de estos atributos de los metadatos, es un array asociativo de nombre "\$Clase_Attr" que contiene como clave el nombre del atributo y como valores los datos separados por coma; su tipo de dato, el tamaño en caracteres, el rotulo de presentación en los listados y su descripción, si no se definen en este array todos los atributos declarados se generara un mensaje de error. Estos atributos son tanto los atributos base de la clase que almacenan un dato, como también los atributos derivados de operaciones entre atributos y los atributos obtenidos por la clave. Para todos estos atributos se debe definir un método "get_<nombre_de_atributo>" que permite accederlos desde los métodos de la clase objeto estándar, Fig. 17.

```
protected static $Clase_Attr =  
array( 'datos_basicos'=>'string',  
       'identificacion'=>'string',  
       'nombre'=>      'string,40,NOM,Nombre de la Asignatura Correspondiente al Curso',  
       'horario'=>     'string,10,Hor,Horario en que se imparte el curso',  
       'lugar'=>      'string,10,SAL,Código del Salon Donde se Imparte el Curso',  
       'matr'=>       'integer,5,MATR,Número de estudiantes Matriculados',  
       'docente'=>    'string,20,DOC,Código del Docente a cargo del Grupo',  
       'grupo'=>     'integer,4,GRU,Código de grupo del curso',  
       'asignatura'=> 'integer,10,COD,Código SIA de la Asignatura',  
       'anio'=>      'integer,4,AÑO,Año del período académico',  
       'semestre'=>  'integer,2,SEM,Semestre del Período Académico' );
```

Fig. 17. Array de atributos de la clase. Fuente: PISIS

```
protected static $Clase_Faceta =  
array( 'curso'=>      'Curso,CURSO,Datos Propios del Curso',  
       'bibliografia'=>'Referencia,REFERENCIAS,Referencias Bibliográficas del Curso',  
       'documentos'=> 'Documento,DOCUMENTOS,Material de Apoyo al Curso',  
       'secciones'=>  'Seccion,SECCIONES,Secciones y Actividades del Curso',  
       'estudiantes'=>'Estudiante,ESTUDIANTES,Estudiantes Matriculados en el Curso',  
       'pruebas'=>   'Prueba,PRUEBAS,Pruebas de verificacion de aprendizaje',  
       'grupos'=>    'Grupo,GRUPOS,Grupos de trabajo');
```

Fig. 18. Array de atributos de las facetas de la clase. Fuente: PISIS

5.3 Atributo de la faceta de la clase

El atributo "clase_faceta" posee una faceta del mismo nombre de la clase, esta faceta permite obtener todos sus atributos para que sean editados y mostrados como datos de la clase en una vista del *Framework*. Adicionalmente, también tendrá definidas las otras facetas que hacen parte de la lógica de la clase, Fig. 18.

5.4 Atributos de visualización de la interfaz de usuario

Finalmente, el último atributo de los metadatos es "Clase_Opciones_Menu", que define las distintas pestañas de la vista de la clase base. Cada pestaña corresponde a una vista parcial de los datos de la clase, que puede contener atributos propios de la clase o atributos de los objetos contenidos en sus facetas.

Este metadato también emplea un array asociativo que usa como clave el nombre de la pestaña y como valor, otro array asociativo con los elementos que se describen a continuación; el elemento 'título' encabeza el contenido de la pestaña, el elemento 'plantilla' es el nombre de la plantilla que despliega los datos de la pestaña, los 'permisos' definen las opciones de acceso para los diferentes roles de usuario, 'componentes' describen los elementos de la información que deben ser leídos de la base de

datos para que sean desplegados en las pestañas. Existen dos plantillas predefinidas, la plantilla 'listado_complejo' que está definida para ser cargada por defecto en la clase, esta plantilla ayuda a desplegar las vistas de las clases que tengan datos ordenados en tablas anidadas, la otra plantilla es 'lista_de_atributos' que lista los datos en forma de filas y columnas.

El atributo "Clase_Opciones_Menu" posee un descriptor para los elementos del listado que tendrán las facetas a ser visualizadas y se encuentra en la clase "Curso"; este descriptor lo conforman el tipo de dato del atributo, el tamaño aproximado del atributo en número de caracteres, el rotulo con el que se identificará el atributo en las vistas, una descripción que será desplegada al pasar el cursor sobre el rotulo del atributo y el nombre de la plantilla que será utilizada para listar las secciones del curso y que será una de las plantillas arriba mencionadas.

El contenido del listado de estas facetas tiene como punto de partida a la clase "Curso", está clase muestra un conjunto de filas que son las secciones que tendrá cada curso. Cada una de estas filas está conformada por varias columnas, estas columnas pueden ser un simple listado de datos en forma de varias columnas o pueden ser columnas con tablas anidadas que también pueden tener columnas de datos, Fig. 19.

```
protected static $Clase_Opciones_Menu = //maximo 12 caracteres en la clave
array(
  "curso" =>
  array('título'=>"Datos Propios del Curso",
    'atributos' =>array("nombre, horario, lugar, matr, docente, asignatura, grupo, anio, semestre"),
    'plantilla' => 'lista_de_atributos',
    'permisos'=>array("estudiante"=>1000, "docente"=>1111),
    'componentes'=>array('curso', array())
  ),
),
```

Fig. 19. Asignando la plantilla 'lista_de_atributos'. Fuente: PISIS

Cada fila o sección posee en su primera entrada la posición que es una tabla por defecto, en la segunda entrada posee una tabla anidada de actividades conformada por tres columnas; la primera contiene la fecha de la actividad, la segunda contiene el lugar de la actividad y la tercera contiene el tipo de actividad.

En la tercera entrada muestra un listado de tópicos del curso y una sangría con la lista de sub tópicos

que corresponden a este tópico. En la cuarta entrada muestra una tabla con tres columnas; la primera el nombre de la referencia, en la segunda la url de dicha referencia y en la tercera los numerales de la referencia que deben ser leídos, especificando de la forma "referencia>url". Finalmente, en la quinta entrada de la fila de la sección se muestra una columna con el icono del documento que corresponde a un entregable, puede ser un ícono de documento word o un ícono de pdf, Fig 20.

```
"bibliografia"=>
array('titulo'=>"Referencias Bibliográficas del Curso",
      'atributos' =>"bibliografia>(referencia,titulo,editorial,
      isbn,pagina_inicial,pagina_final,num_volumen,num_ejemplar,fecha_publicacion)",
      'permisos'=>array("estudiante"=>1000,"docente"=>1111),
      'componentes'=>array("curso",array(array("bibliografia",array()))
    ),
"secciones"=>
array('titulo'=>"Secciones y Actividades del Curso",
      'atributos'=>"secciones(posicion,actividades(fecha,lugar,tipo),
      [L]temas(descripcion,[L]subtemas(descripcion)),
      lecturas(referencia>url,numerales),entregables(>url)",
      'permisos'=>array("estudiante"=>1000,"docente"=>1111),
      'componentes'=>array("curso",array(array("secciones",array("actividades","temas",
      "lecturas","entregables")), "documentos","bibliografia"))
    ),
```

Fig. 20. Asignando la plantilla 'listado_complejo' por defecto y definiendo la lista de atributos en tablas anidadas. Fuente: PISIS

6. ROLES DE USUARIOS

Los roles de los usuarios poseen diferentes opciones de visualización para acceder a los datos de los objetos. Para acceder a ellos, se define un conjunto de métodos protegidos que le sirven de interfaz sobre los atributos a la clase objeto estándar. Estos métodos son para acceder a los atributos de la clase, y para acceder a los atributos que son objetos de otras clases "facetas de la clase" y para acceder a los atributos derivados donde se debe incluir la operación de cálculo para obtener el valor del atributo, Fig. 21.

```
protected static $Clase_rol = array( "estudiante"=>"estudiantes",
                                     "docente"=>"docente"
                                   );
```

Fig. 21. Atributo para controlar el acceso de los usuarios. Fuente: PISIS

La clase objeto estándar, a su vez, es la responsable de definir la interfaz pública sobre los atributos de todas las clases del modelo. Estos métodos para la clase del ejemplo se muestran en la Fig. 22.

Si un atributo no cuenta con un método de acceso 'get?', el objeto de la clase Objeto Estándar muestra un mensaje de error. La función "leer_codigo" contiene un conjunto de arreglos con valores, estos valores son leídos directamente y sirven para cargar

localmente con datos todos los atributos de los objetos de la clase, esto permite crear un prototipo de la clase que facilita las labores de diseño y prueba antes de pensar en los aspectos de acceso a la base de datos. Estos valores para un objeto particular se pueden acceder por medio de su clave, Fig. 23.

```
//
// Interfaz publica para consulta de los atributos
//
public function get_identificacion() {return $this->identificacion;}
public function get_titulo() {return $this->titulo;}
public function get_nombre() {return $this->nombre;}
public function get_fecha_publicacion() {return $this->fecha_publicacion;}
public function get_url() {return $this->url;}
public function get_editorial() {return $this->editorial;}
public function get_num_volumen() {return $this->num_volumen;}
public function get_num_ejemplar() {return $this->num_ejemplar;}
public function get_issn() {return $this->issn;}
public function get_isbn() {return $this->isbn;}
public function get_tipo_publicacion() {return $this->tipo_publicacion;}
public function get_pagina_inicial() {return $this->pagina_inicial;}
public function get_pagina_final() {return $this->pagina_final;}
public function get_palabras_claves() {return $this->palabras_claves;}
public function get_resumen() {return $this->resumen;}
public function get_prologo() {return $this->prologo;}
public function get_documento() {return $this->Clave["documento"];}
public function get_autores() {return $this->autores;}
public function get_imagen() {return $this->imagen;}
public function get_descripcion() {return $this->descripcion;}
public function get_contenido() {return $this->contenido;}
public function get_datos_basicos() {return $this->get_editorial() . " / " .
                                     $this->get_fecha_publicacion();}
}
```

Fig. 22. Conjunto de métodos que sirven como interfaz pública para los atributos de una clase del Framework de PISIS. Fuente: PISIS

```
// -----
// Dispositivo "codigo", que almacena en codigo un conjunto de objetos de ejemplo
// -----

static function leer_codigo($Co,$Ds)
{
    switch ($Co['referencia'])
    {
        case "Abelson1985":
            $Dt = array(
                'fecha_publicacion'=>'1985',
                'editorial'=>' The MIT Press', 'isbn'=>'28676',
                'prologo'=>'',
                'imagen'=>'cover_abelson.jpg', 'descripcion'=>'',
                'autores' => array(new Persona(array("persona" => 7), $Ds),
                                new Persona(array("persona" => 8), $Ds),
                                new Persona(array("persona" => 9), $Ds),
                                )
            );
            break;
        case "Arango2008":
            .
            .
            .
    }
}
```

Fig. 23. Arreglo de datos local para los atributos de una Clase del *Framework* de PISIS. Fuente: PISIS

CONCLUSIONES

En este trabajo se presenta el desarrollo de un prototipo web con *Framework* PISIS, el cual maneja el dominio de los cursos y docentes del Departamento de Ciencias de la Computación y la Decisión de la Facultad de Minas - Universidad Nacional de Colombia Sede Medellín. El uso e implementación de *Frameworks* para la creación de aplicaciones y sitios Web facilita el trabajo de desarrolladores de software, ya que contribuye al incremento de su productividad y a la disminución de errores invalidez.

REFERENCIAS

- [1] M. R. J. Qureshi, "Agile software development methodology for medium and large projects," *IET Softw.*, vol. 6, no. 4, p. 358, 2012.
- [2] R. A. Santelices and M. Nussbaum, "A framework for the development of videogames," *Softw. Pract. Exp.*, vol. 31, no. 11, pp. 1091–1107, 2001.
- [3] M. Björemo and P. Trninic, *Evaluation of web application frameworks*, 1st ed., no. June. Göteborg, 2010.
- [4] M. Schwarz, "Design and Analysis of Web Application Frameworks." Arhus, pp. 7–8, 2013.
- [5] M. Salas, G. Hernández, R. Valencia, L. Rodríguez, A. Rodríguez, and J. López, "Analyzing best practices on Web development frameworks: The lift approach," *Sci. Comput. Program.*, vol. 102, pp. 1–19, 2015.
- [6] Oracle, "JavaServes Pages technology," JSP, 2016. [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>. [Accessed: 07-Jul-2016].
- [7] Struts, "Struts," Apache struts, 2016. [Online]. Available: <https://struts.apache.org/>. [Accessed: 07-Jul-2016].
- [8] M. Emoto, M. Yoshida, C. Iwata, S. Inagaki, and Y. Nagayama, "Efficient development of web applications for remote participation using Ruby on Rails," *Fusion Eng. Des.*, vol. 85, no. 3–4, pp. 622–624, 2010.

- [9] Grails, “Grails,” A powerful Groovy-based web application framework for the JVM, 2016. [Online]. Available: <https://grails.org/>. [Accessed: 07-Jul-2016].
- [10] M. Biermann, “A simple versatile solution for collecting multidimensional clinical data based on the CakePHP web application framework,” *Comput. Methods Programs Biomed.*, vol. 114, no. 1, pp. 70–79, 2014.
- [11] Django, “Django,” The web framework for perfectionists with deadlines., 2016. [Online]. Available: <https://www.djangoproject.com/>. [Accessed: 07-Jul-2016].
- [12] Catalyst, “Catalyst,” Perl MVC framework, 2016. [Online]. Available: <http://www.catalystframework.org/>. [Accessed: 07-Jul-2016].
- [13] Symfony, “Symfony,” Symfony is a set of reusable PHP components, 2016. [Online]. Available: <https://symfony.com/>. [Accessed: 07-Jul-2016].