



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD
DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Cristian Pavel Acu Jocol
Ervin Gerardo Ramírez Tacejo
Asesorado por el Ing. Edgar Danilo Salazar Solares

Guatemala, septiembre de 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD
DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

CRISTIAN PAVEL ACU JOCOL

ERVIN GERARDO RAMÍREZ TACEJO

ASESORADO POR EL ING. EDGAR DANILO SALAZAR SOLARES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, SEPTIEMBRE DE 2013

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Walter Rafael Véliz Muñoz
VOCAL V	Br. Sergio Alejandro Donis Soto
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Türk
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
EXAMINADORA	Inga. Sonia Yolanda Castañeda Ramírez
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD
DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha abril de 2013.



Cristian Pavel Acu Jocol



Ervin Gerardo Ramirez Tacejo



Guatemala, 06 de Abril de 2013

Inga. Sigrid Alitza Calderón de León
Directora EPS
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Un cordial Saludo:

Por medio de la presente le informo que los estudiantes **Cristian Pavel Acu Jocol** con número de carné **200412794** y **Ervin Gerardo Ramírez Tacejo** con número de carné **200412905**, concluyeron satisfactoriamente bajo mi asesoría y revisión, la elaboración del Reporte Final de Trabajo de Graduación de su EPS titulado: "GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA", el cual fue realizado en el período comprendido del **1 de septiembre de 2010 al 28 de Febrero de 2011**, en las instalaciones del Centro de Investigaciones de la Facultad de Arquitectura ubicado en el 2do. Nivel del edificio T-2 de la Facultad de Arquitectura de la Universidad de San Carlos de Guatemala.

Agradeciendo su atención a la presente,

Atentamente,


EDGAR DANILLO SALAZAR
INGENIERO EN CIENCIAS Y SISTEMAS
COLEGIADO NO. 9584

Ing. Edgar Danilo Salazar Solares
No. Colegiado 9584
Asesor de Proyecto de EPS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA
UNIDAD DE EPS

Guatemala, 20 de mayo de 2013.
REF.EPS.DOC.581.05.2013.

Inga. Sigríd Alitza Calderón de León
Directora Unidad de EPS
Facultad de Ingeniería
Presente

Estimada Ingeniera Calderón de León.

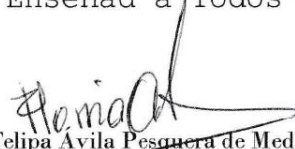
Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) de los estudiantes universitarios de la Carrera de Ingeniería en Ciencias y Sistemas, **Cristian Pavel Acu Jocol Carné 200412794 Ervin Gerardo Ramírez Tacejo** carné No. **200412905** procedí a revisar el informe final, cuyo título es **"GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"


Inga. Floriza Felipa Avila Pesquera de Medinilla
Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA



Universidad de San Carlos
de Guatemala



Facultad de Ingeniería
Decanato

Ref.DTG.621.2013

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ciencias y Sistemas, al trabajo de graduación titulado: **“GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, presentado por los estudiantes universitarios: **Cristian Pavel Acu Jocol y Ervin Gerardo Ramírez Tacejo**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.


Ing. Murphy Olympo Paiz Recinos
Decano



Guatemala, septiembre de 2013

/cc

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERIA
UNIDAD DE EPS

Guatemala, 20 de mayo de 2013.
REF.EPS.D.384.05.2013.

Ing. Marlon Antonio Pérez Turk
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Perez Turk.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **“GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, que fue desarrollado por los estudiantes universitarios **Cristian Pavel Acu Jocol Carné 200412794 Ervin Gerardo Ramírez Tacejo carné No. 200412905** quienes fueron debidamente asesorados por el Ing. Edgar Danilo Salazar Solares y supervisados por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y la Supervisora de EPS, en mi calidad de Directora apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
“Id y Enseñad a Todos”

Inga. Sigrid Alitz Calderón de León
Directora Unidad de EPS
Unidad de Prácticas de Ingeniería y EPS
Facultad de Ingeniería

SACdL/ra

E
S
C
U
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“GESTOR DE CONTENIDO DEL CENTRO DE INVESTIGACIONES DE LA FACULTAD DE ARQUITECTURA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**, realizado por los estudiantes CRISTIAN PAVEL ACU JOCOL y ERVIN GERARDO RAMÍREZ TACEJO , aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”



Ing. Marlon Antonio Pérez Furk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 05 de septiembre 2013

ACTO QUE DEDICO A:

- Dios** Por haberme dado fuerzas, las capacidades y la ayuda a través de mi familia y compañeros para llegar hasta el final y terminar lo que empecé.
- Mis padres** Camilo Acu por su apoyo incondicional en toda mi vida, su ejemplo, sabios consejos y preocuparse de cada uno de mis pasos. A mi madre Angélica Jocol, por ser un ángel que me ha cuidado durante cada etapa de mi vida, por confiar en mí y por ser una luz en momentos de oscuridad. Han sido los pilares de mi vida.
- Mis hermanos** Lesly y Lester Acu, por paciencia, comprensión y por hacerme la vida más feliz con su compañía.
- Mis abuelos** Marquina Quiñonez por su amor y apoyo. A Clemente Acu (q.e.p.d.) por su buen ejemplo, su recuerdo en cada momento me acompaña y espero nos encontremos en la vida eterna.
- Mis tíos y primos** Por ser parte de este camino, sus consejos, ideas y ese apoyo y confianza hacia mí.

Mis amigos	Carlos Ruiz, Ervin Ramírez, Alex Campos y Josué Obregón, por tantos buenos momentos vividos, por las metas que juntos alcanzamos y principalmente por su valiosa amistad.
Nuestro asesor	Edgar Salazar, por su efectivo apoyo a lo largo de la realización del trabajo de graduación.
Los ingenieros	Marlon Pérez Türk, Floriza Ávila y Mayra Corado, por su apoyo dentro de la Facultad de Ingeniería.
Colegio Salesiano Don Bosco	Por haberme formado durante mi niñez y adolescencia. Y darme las bases para ser una persona íntegra.
Facultad de Ingeniería	Por brindarme conocimientos que me llevaron a una superación personal y haber hecho de mi un profesional.

Cristian Pavel Acu Jocol

ACTO QUE DEDICO A:

Dios	Por prestarme la vida y proveerme las fuerzas para alcanzar un logro más en mi vida.
Mis padres	Víctor Ramírez y Ester Tacejo, por su amor, sus consejos, mi formación espiritual, su apoyo moral, económico y por ayudarme en todo momento.
Mi esposa	Julia Dardón, por su amor incondicional, apoyo y compartir sus conocimientos de forma positiva y ser mi ayuda idónea.
Mis hermanos	Víctor, Linely y Eduardo Ramírez, por su amor y sus consejos.
Mis abuelos	Por sus consejos sabios y su amor incondicional.
Mis amigos	Por su lealtad y comprensión.
Nuestro asesor	Edgar Salazar, por su efectivo apoyo a lo largo de la realización del trabajo de graduación.
Facultad de Ingeniería	Por brindarme conocimientos que me llevaron a una superación personal y haber hecho de mi un profesional.

Ervin Gerardo Ramírez Tacejo

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
GLOSARIO	VII
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. ORIGEN DE LOS GESTORES DE CONTENIDO.....	1
1.1. Web 1.0.....	1
1.1.1. Características	1
1.1.2. ¿Quiénes la usan?	3
1.1.3. Tecnología utilizada.....	4
1.2. Web 2.0.....	5
1.2.1. Características	5
1.2.2. ¿Quiénes la usan?	8
1.2.3. Tecnología utilizada.....	10
1.3. Gestores de contenidos	11
2. ANÁLISIS DE LA PROPUESTA DE SOLUCIÓN.....	15
2.1. Identificación del problema.....	16
2.2. Solución al problema.....	18
2.3. Tecnologías a utilizar.....	20
2.3.1. Generalidades	21
2.3.2. Descripción de arquitectura de software	22
2.3.3. Servidor Web.....	24
2.3.3.1. Apache Tomcat	24

2.3.4.	Base de datos	26
2.3.4.1.	MySQL Server.....	26
2.3.4.2.	Mapeo objeto relacional	27
2.3.4.3.	Persistencia.....	29
2.3.4.4.	Java Persistence API	30
2.3.4.5.	Transacciones.....	31
2.3.5.	Tecnologías para desarrollo de software.....	32
2.3.5.1.	Spring framework	32
2.3.5.2.	RichFaces	37
2.3.5.3.	PHP.....	40
3.	IMPLEMENTACIÓN Y RESULTADOS	43
3.1.	Aplicación del cliente	44
3.1.1.	Diseño de página principal.....	44
3.1.2.	Diseño de página secundaria.....	46
3.1.2.	Diseño de página de galería	47
3.2.	Aplicación del administrador	48
3.2.1.	Análisis y diseño	48
3.2.1.1.	Autenticación.....	50
3.2.1.2.	Agregar ubicación	51
3.2.1.3.	Agregar proyecto.....	52
3.2.1.4.	Agregar contenido	53
3.2.1.5.	Agregar personal.....	54
3.2.1.6.	Asignar personal	55
3.2.1.7.	Agregar descargas	56
3.2.1.8.	Agregar usuario.....	57
3.2.1.10.	Agregar ciudad maya	59
3.2.1.11.	Agregar publicación.....	60
3.2.1.12.	Agregar material multimedia.....	61

3.2.1.13.	Agregar colaborador	62
3.2.1.14.	Asignar colaborador.....	63
3.3.	Diagrama de clases y entidad relación.....	64
3.4.	Diagrama entidad relación	70
3.5.	Resultados	71
CONCLUSIONES		75
RECOMENDACIONES		77
BIBLIOGRAFÍA		79

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Web 1.0	4
2.	Proceso de solución	19
3.	Ambiente propuesto para la solución	20
4.	Modelo vista controlador	22
5.	Distribución de tecnologías por capa	23
6.	Arquitectura de Hibernate.....	29
7.	Arquitectura de Java Persistence API	31
8.	Arquitectura de Spring framework	33
9.	Archivo de configuración	36
10.	Tipos de componentes de RichFaces	39
11.	Estadísticas de RichFaces	40
12.	División del sistema.....	43
13.	Diseño de página principal	45
14.	Diseño de página secundaria	46
15.	Diseño de página de galería.....	47
16.	Módulo de administración.....	49
17.	Caso de uso del módulo de edición.....	58
18.	Clase persistente.....	65
19.	Diagrama de clases de persistencia.....	66
20.	Inyección de servicios	67
21.	Diagrama de clases de la capa del controlador 1.....	68
22.	Diagrama de clases de la capa del controlador 2.....	69
23.	Diagrama entidad relación.....	70
24.	Gráfica de tiempos empleados en los tipos de publicaciones	73

TABLAS

I.	Componentes que conforman Spring framework	34
II.	Descripción del caso de uso autenticación.....	50
III.	Descripción del caso de uso agregar ubicación	51
IV.	Descripción del caso de uso agregar proyecto.....	52
V.	Descripción del caso de uso agregar contenido	53
VI.	Descripción del caso de uso agregar personal.....	54
VII.	Descripción del caso de uso asignar personal	55
VIII.	Descripción del caso de uso agregar descargas	56
IX.	Descripción del caso de uso agregar usuario.....	57
X.	Descripción del caso de uso agregar ciudad maya	59
XI.	Descripción del caso de uso agregar publicación.....	60
XII.	Descripción del caso de uso agregar material multimedia	61
XIII.	Descripción del caso de uso agregar colaborador.....	62
XIV.	Descripción del caso de uso asignar colaborador	63
XV.	Especificaciones de los tipos de publicaciones	71
XVI.	Medición de tiempos de publicación del proceso anterior	72
XVII.	Medición de tiempos de publicación del proceso actual	73

GLOSARIO

AJAX	<i>AJAX</i> , acrónimo de <i>Asynchronous JavaScript And XML</i> (<i>JavaScript</i> asíncrono y <i>XML</i>), es una técnica de desarrollo web para crear aplicaciones interactivas.
API	Una interfaz de programación de aplicaciones o API (del inglés <i>Application Programming Interface</i>) es el conjunto de funciones y procesamientos que ofrece cierta biblioteca para ser utilizado por otro <i>software</i> como una capa de abstracción.
Arqueología de la Arquitectura	Es una rama de la arqueología que se dedica al estudio de los materiales arquitectónicos.
Arquitectura	La arquitectura es la técnica o el arte de quien realiza el proyecto y dirige la construcción de los edificios y estructuras, ya que, para los antiguos griegos, la palabra «Τεχνη (techne)» significa saber hacer alguna cosa.
AWT	La <i>Abstract Window Toolkit</i> (<i>AWT</i> , en español Kit de Herramientas de Ventana Abstracta) es un kit de herramientas de gráficos, interfaz de usuario, y sistema de ventanas independiente de la plataforma original de Java.

Backup o respaldo	Es la acción de crear soportes o copias de la información de tal forma que al perder la fuente o respaldo principal, sea posible su restauración.
CIFA	Centro de Investigaciones de la Facultad de Arquitectura.
Clase	Define como será el comportamiento del objeto y como almacenará su información. Es responsabilidad de cada objeto ir recordando el valor de sus atributos.
Cluster	Se aplica a los conjuntos de computadoras construidos mediante la utilización de componentes de hardware comunes y que se comportan como si fuesen una única computadora.
Contenido	Aplicado al proyecto se le denominara a cualquier elemento que forma de parte de un proyecto específico, que es creado y modificado por los usuarios de la plataforma, es decir documentos, comentarios, material audio visual, etc.

DOM	El <i>Document Object Model</i> (Modelo en Objetos para la representación de Documentos), abreviado <i>DOM</i> , es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos <i>HTML</i> y <i>XML</i> , un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.
EJB	Los <i>Enterprise JavaBeans</i> (también conocidos por sus siglas <i>EJB</i>) son una de las <i>API</i> que forman parte del estándar de construcción de aplicaciones empresariales <i>J2EE</i> de <i>Sun Microsystems</i> (ahora <i>JEE 5.0</i>). El objetivo de los <i>EJBs</i> es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia) para centrarse en el desarrollo de la lógica de negocio en sí.
Escalabilidad	Es la capacidad del sistema informático de cambiar su tamaño o configuración para adaptarse a las circunstancias cambiantes.
Framework	Es una estructura de soporte definida en la cual otro proyecto de <i>software</i> puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros <i>software</i> para desarrollo.

Gestor de Contenido	Un sistema de gestión de contenidos (en inglés <i>Content Management System</i> , abreviado <i>CMS</i>) es un programa que permite crear una estructura de soporte (<i>framework</i>) para la creación y administración de contenidos, principalmente en páginas <i>web</i> , por parte de los participantes.
Herencia	Especifica que una clase usa la implementación de otra clase, con la posible sobre escritura de la implementación de las interfaces.
HQL	<i>HQL (Hibernate Query Language)</i> es un lenguaje de consultas de objetos, similar a <i>SQL</i> . En <i>HQL</i> las consultas se realizan sobre Objetos (no tablas), y se retornan objetos.
IDE	Un entorno de desarrollo integrado o, en inglés, <i>Integrated Development Environment (IDE)</i> , es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.
Interface	Operación mediante la cual el cliente accede al objeto.
JAR	Un archivo <i>JAR</i> (por sus siglas en inglés, <i>Java Archive</i>) es un tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java.

JEE	<i>Java Platform, Enterprise Edition</i> o <i>Java EE</i> (anteriormente conocido como <i>Java 2 Platform, Enterprise Edition</i> o <i>J2EE</i> hasta la versión 1.4), es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de <i>software</i> modulares ejecutándose sobre un servidor de aplicaciones.
JVM	Una máquina virtual Java (en inglés <i>Java Virtual Machine, JVM</i>) es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (<i>Java byte code</i>), el cual es generado por el compilador del lenguaje Java.
Maquetación	Maquetar un diseño consiste en dar un formato a los documentos, a todo el conjunto de elementos que lo componen, las imágenes, los textos, etc.
Material audiovisual	Se considera a todo aquella información que es transmitida en formato visual y audio.
Modelo relacional	El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Multiplataforma	Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.
OID	Identificador de objeto, representación de cadena de un identificador de objeto que se compone de una lista de números decimales separados por puntos.
Plug-in	Un complemento (o <i>plug-in</i> en inglés) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la <i>API</i> .
Proyecto arquitectónico	Referente al ambiente de CIFA, se le denomina a todo proyecto que aprobado por el consejo y junta directiva de la facultad de arquitectura, el cual tiene asignados recursos y tiene como objetivo cumplir las metas establecidas para el mismo.
Serialización	Consiste en un proceso de codificación de un Objeto (programación orientada a objetos) en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria).

SQL	El Lenguaje de consulta estructurado (<i>Structured Query Language</i>) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Es un lenguaje de cuarta generación (4GL).
UML	Lenguaje Unificado de Modelado (<i>UML</i> , por sus siglas en inglés, <i>Unified Modeling Language</i>) es el lenguaje de modelado de sistemas de <i>software</i> más conocido y utilizado en la actualidad.
URL	<i>URL</i> significa <i>Uniform Resource Locator</i> (localizador uniforme de recurso). Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
Vista de casos de uso	Es la descripción del comportamiento que el sistema debería realizar según la percepción de los usuarios.
Vista lógica	Es el diagrama de clases que se obtiene como resultado del análisis de los casos de uso.
XML	Sigla en inglés de <i>Extensible MarkupLanguage</i> (lenguaje de marcas ampliable), es un metalenguaje extensible de etiquetas desarrollado por el <i>World Wide Web Consortium (W3C)</i> .

RESUMEN

El Centro de Investigaciones de la Facultad de Arquitectura (CIFA), es el ente responsable del diseño y coordinación de la investigación de corto y largo plazo en los diferentes campos prioritarios del conocimiento de la arquitectura, relacionados a: dinámicas y desequilibrios territoriales; sostenibilidad del medio ambiente; producción de objetos arquitectónicos; teorías e historias de diseño de la arquitectura; tecnología constructiva y vivienda; diseño gráfico; manejo, conservación patrimonial y de centros históricos; estudios de coyuntura.

Sin embargo, es necesario dar a conocer el trabajo que dicha entidad realiza, asimismo, el trabajo de los involucrados en el proyecto. Tanto para los proyectos que se encuentran en ejecución, como los proyectos que han sido ejecutados.

Por ello, el Gestor de Contenidos del Centro de Investigaciones de la Facultad de Arquitectura (GCCIFA), es un proyecto de software que tiene como objetivo ser una herramienta de apoyo para la gestión de contenidos de los proyectos del CIFA.

La plataforma de Gestor de Contenidos CIFA, es una aplicación en línea capaz de ser accesible desde cualquier ubicación y disponible 24 horas al día y 365 días del año.

El gestor de contenidos facilita el acceso a la publicación de contenidos a un rango mayor de usuarios. Permite que sin conocimientos de programación ni maquetación cualquier usuario pueda indexar contenido en el portal.

Además permite la gestión dinámica de usuarios y permisos, la colaboración de varios usuarios en el mismo trabajo, la interacción mediante herramientas de comunicación. Los costes de gestión de la información son muchos menores ya que se elimina un eslabón de la cadena de publicación. La maquetación es hecha al inicio del proceso de implantación del gestor de contenidos.

La actualización, *backup* y reestructuración del portal son mucho más sencillas al tener todos los datos vitales del portal, los contenidos, en una base de datos estructurada en el servidor.

El GCCIFA, brindará escalabilidad, alto rendimiento, alta usabilidad, rapidez y facilidad para la gestión del contenido de cada proyecto.

La creación del proyecto anexo, Atlas de las Ciudades Mayas en Guatemala, el cual estará integrado al Gestor de Contenido CIFA, para aprovechar las facilidades de la estructura general de la plataforma.

OBJETIVOS

General

Desarrollar un software para facilitar la gestión de la información del CIFA basado en una arquitectura fiable y escalable. Utilizando tecnologías web como parte de la arquitectura de la aplicación.

Específicos

1. Entregar un sistema funcional, confiable y que provea facilidad para la gestión de contenidos del CIFA.
2. Utilizar tecnologías de desarrollo estándar para garantizar la facilidad de mantenimiento incremental del sistema.
3. Utilizar tecnologías sin costo de licenciamiento.
4. Implementar una arquitectura de software fiable y escalable.
5. Instalar la aplicación en los servidores que están a disposición del CIFA.

INTRODUCCIÓN

La investigación científica se considera una actividad consubstancial de la universidad. Los académicos nacionales la definieron como un sistema. Con ello se ha buscado subrayar la integralidad de la misión académica, basada en la coordinación universitaria, la operación rigurosa y profunda de la labor científica, la difusión benefactora para el país, la vinculación con la comunidad científica nacional y otras funciones universitarias. Estas definiciones fueron consideradas en el reglamento del sistema de investigación, aprobado por el Consejo Superior Universitario en sesión del 24 de noviembre de 1993.

La investigación científica universitaria es una parte importante del proceso de modernización del país. El sistema de investigación universitario se describe como el funcionamiento de tres subsistemas: dirección, coordinación y operación.

El Centro de Investigaciones de la Facultad de Arquitectura, es el ente de dicha facultad encargado de realizar estudios e investigaciones de dicho campo. Obteniendo como resultado informes, planos, simulaciones de infraestructura, documentos, etc. los cuales se albergan en el CIFA, en la biblioteca de la facultad y en la biblioteca virtual de DIGI, encontrándose el material de forma digital.

Por ello el CIFA, a través de la implementación de CGCIFA busca la difusión y compartimiento de investigaciones y demás contenido resultante de los programas y proyectos ejecutados.

Con ello brindando a la sociedad guatemalteca un espacio en el cual se pueda informar y aportar conocimiento, en un ambiente amigable de fácil uso. Además de darse a conocer con otros centros de investigaciones acorde a su segmento.

Además CGCIFA, es un proyecto que está pensado para soportar el crecimiento del proyecto, proveer un alto rendimiento bajo la demanda del servicio, seguridad de la información que se contiene y que su funcionamiento se basa en la parametrización de su comportamiento para brindar una mayor maniobrabilidad a los administradores del sitio para actualizar fácil y rápidamente el contenido del sitio.

1. ORIGEN DE LOS GESTORES DE CONTENIDO

1.1. Web 1.0

El término Web 1.0 se refiere a la primera versión de la web, a veces, también conocida como la web de información, que se desarrollo a partir de 1991, era esencialmente una fuente de información creado por un pequeño número de autores para un número muy grande de usuarios.

La Web 1.0 consistía en una gran parte de páginas web estáticas con un poco de espacio para interactuar. Se podía visualizar como una gran biblioteca de libros para consultar.

El objetivo tras la creación de la web es el de tener un espacio de información en común por el cual las personas se comunicaran compartiendo información.

1.1.1. Características

Entre las características más importantes se pueden mencionar las siguientes:

- Información bastante estática
- Se actualiza con poca frecuencia
- Se caracteriza como brochureware que consiste en presentar información plana e unilateral en la red.
- Los elementos contenidos en las páginas comúnmente son: imágenes, iconos para navegar, texto y menú.

Las primeras opciones que permitió la web 1.0 fue la búsqueda de la información y lectura. Era muy poca la forma de interacción con el usuario o la contribución de contenido.

Los hipervínculos era otra manera de interactuar. Una persona podía crear una página web, y crear vínculos; tenía que dar un clic a la lectura y absorber más información. Todo transcurría a un ritmo mucho más lento ya que el recurso utilizado inicialmente eran módems de 56 k/s. Las Web 1.0 se basó en gran medida en creación de páginas web personales. Las personas configuraban una página sobre ellos o sobre un tema que les interesaba y tenían que ir a través de la codificación en muchos casos con poca ayuda.

La web 1.0, tiene un sinnúmero de útiles referencias materiales disponibles. En muchos casos, son referencias a libros impresos tradicionales, como diccionarios, enciclopedias, o atlas, o buscar material como folletos de información del gobierno, o folletos emitidos por una serie de organizaciones e instituciones. Así también, una gran colección de videos e imágenes históricas y una extensa colección de documentales.

Las primeras opciones que permitió la web 1.0 fue la búsqueda de la información y lectura. Era muy poca la forma de interacción con el usuario o la contribución de contenido.

1.1.2. ¿Quiénes la usan?

En el período de auge era usada por las personas que tenían los recursos para disponer de un ordenador y aún más para hacer uso de la red, en su mayoría eran personas que tenían conocimiento de informática; comúnmente era utilizada por estudiantes que pasaban el tiempo en la web, en busca de respuestas a una o más preguntas preestablecidas y que optaban en dar seguimiento a resolución de problemas o descubrimiento guiado haciendo uso de la información publicada; actualmente existe este tipo de usuario que busca información, pero en una Web 1.0 incorporada.

Asimismo, estaba la mayoría de los propietarios de los sitios web, que querían establecer una presencia en línea y hacer su información disponible a cualquier persona en cualquier momento. De esta manera, daban a conocer los productos a los clientes potenciales, tanto por medio de un catálogo o un folleto en una página web que los usuarios debían de descargar, usado como método para proporcionar productos a cualquier persona en el mundo, eliminando restricciones geográficas.

Así también estaban las personas que encontraban entretenimiento en los cuestionarios, rompecabezas y juegos, tal como los crucigramas que solían ser lo más atractivo que simples ejercicios. En resumen, las razones generales por la cual era utilizada la web son las siguientes:

- Presentación de información.
- La Intranet, que se utilizaba para cuestiones internas de la empresa, como por ejemplo: que los vendedores pudieran cargar o descargar remotamente lista de precios, catálogos y pedidos.

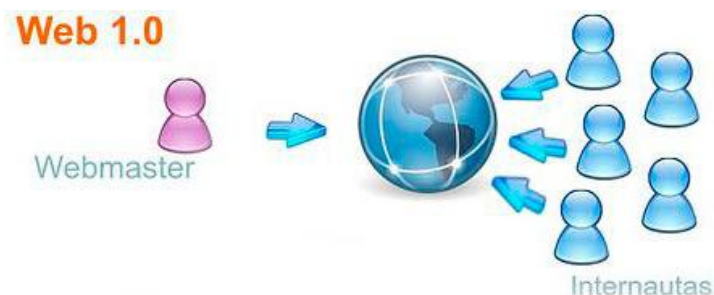
- Extranet, que busca comunicarse con clientes y/o proveedores.
- Comercio, que tiene la intención de incrementar ingresos en las compañías.

1.1.3. Tecnología utilizada

La forma más básica de la Web 1.0 era con navegadores de sólo texto, luego surgió el HTML que hizo las páginas web más agradables a la vista, pero seguían de forma estática, así como los primeros navegadores visuales tales como IE, Netscape. Por lo que, el usuario no podía interactuar con el contenido de la página (nada de comentarios, respuestas, citas, etc.), estando limitado a solo ver la información subida y comunicarse con el propietario de la página por correo electrónico únicamente.

Además, si se utilizaba algún software requería de ser instalado para su aplicación.

Figura 1. **Web 1.0**



Fuente: http://www.koala-soft.com/sites/default/files/images/web1_0.jpg. Consulta: 20 de junio de 2013.

Es importante darse cuenta que la web 1.0 no ha desaparecido se sigue incorporando en la web 2.0.

1.2. Web 2.0

La Web 2.0 es la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través de las web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio.

La web 2.0 es considerada como una etapa donde constantemente estaban surgiendo nuevas aplicaciones y sitios con sorprendentes funcionalidades, y desde donde han surgido nuevos proyectos y se preocupa por brindar mejores soluciones al usuario final. Donde sin duda la evolución del medio ha propuesto cosas más interesantes las cuales se ven en la actualidad.

Este concepto de Web 2.0 surge de una conferencia donde Dale Dougherty de O'Reilly Media compartió una lluvia de ideas junto a Craig Cline de MediaLive, en un evento donde se analizaban las empresas que sobrevivieron y otras que aparecieron con nuevas ideas y nuevos modelos de negocio en la web, para identificar las características claves de éxito y los principios de la web 2.0.

1.2.1. Características

En la actualidad estamos asistiendo a la web 2.0 o también llamada lectura-escritura, la cual tiene la capacidad de aportar contenidos e interactuar con otros usuarios de Internet, lo cual ha cambiado dramáticamente el panorama de la web en poco tiempo.

Asimismo, existe un número sorprendente de definiciones de lo que se constituye la Web 2.0, hay personas que piensan que si un sitio se construye utilizando una determinada tecnología tal como *Ruby onRails* o se emplea *Ajax* es una aplicación web 2.0; pero la definición se limita a exigir que los usuarios sean capaces de interactuar unos con otros o aportar contenido.

En la conferencia de O'Reilly da las primeras características del concepto y por consiguiente los siete principios de la Web 2.0:

- La web es la plataforma: las herramientas en la web 2.0 están basadas en que tanto el software como la información están alojados en internet, las aplicaciones son modulares y pueden ser combinadas haciendo uso de Apis, estándares y protocolos.
- La gestión de la base de datos: el valor de la web 2.0 está determinado por los datos que guarda y se hacen disponibles. La mayoría de personas actualizan sus perfiles, crean amistades, realizan búsquedas, en fin todas estas modalidades añaden valor a la aplicación. Asimismo, hay grupos que tienen otras aplicaciones con acceso a resultados de los procesos de negocio con los cuales pueden construir nuevas aplicaciones.
- Fin del ciclo de las actualizaciones: las aplicaciones web 2.0 se ejecutan a través de la propia web y operan dentro de un navegador, además son incrementales debido a que con mayor rapidez se realizan actualizaciones en un servidor web, ya que antes habían software empaquetados, distribuidos y comprados en CD.

- Aprovechar la inteligencia colectiva: una característica que sobresale de las aplicaciones web 2.0 es que mientras más personas las usen, más gente las usará. Y es debido a que realizan acciones como crear rankings, etiquetar, comentar; esta es la manera general como un usuario interactúa y contribuye con contenido de manera que los usuarios se han convertido en productores y consumidores del contenido web. Como ejemplos se encuentran:
 - Wikipedia: es una enciclopedia on-line, un sitio donde el valor aumenta cada vez que un usuario aporta o actualiza uno de sus artículos.
 - Amazon: en este sitio los clientes sugieren libros, realizan votaciones sobre estos, de manera que se puedan recomendar a los clientes proporcionándoles material de utilidad e interés.

- Modelos de programación ligeros: el secreto de las aplicaciones web 2.0 es que existen herramientas que facilitan la creación, y los datos que contienen interactúan de manera simple con distintas aplicaciones.
 - Una de las premisas es que los datos se guardan separados de cómo se presentan, facilita que funcionen en dispositivos o soportes distintos.
 - Crean una tendencia donde se dan soluciones código abierto que entreguen herramientas rápidas para prototipos de aplicaciones.

- El software no limitado a un solo dispositivo: cada vez más existen dispositivos móviles –teléfonos móviles, PDAs, consolas de juego, reproductores que soportan la navegación en la web, y por lo tanto, los sitios se deben crear para ser vistos correctamente en todos los dispositivos. La regla es separar el contenido y la presentación de forma que, por un lado está la lógica comercial de una aplicación la cual pueda mantenerse constante y por otro la presentación de la información la cual puede tener propiedades específicas para cierto dispositivo en el cual va a ser visualizado el contenido.
- Una experiencia de usuario enriquecida: sobresale la tecnología Ajax, ya que permite actualizar ciertas partes de una página, por lo tanto crea presentaciones más limpias y permite cambiar la apariencia y funciones de la página.

1.2.2. ¿Quiénes la usan?

En cuanto a los usuarios se ha pasado de consumidores de información a tratarlos como protagonistas activos, ya que participan en la elaboración y gestión de contenidos. Este cambio es debido a la confianza que el usuario tiene al realizar sus acciones, y en el uso que hará de los servicios, etc. Asimismo del aprovechamiento de la inteligencia colectiva que nace de la utilización del software social integrando las actividades y recogiendo a su vez conocimiento de los usuarios.

De manera que, los usuarios de la web 2.0 son todos los que participan y colaboran haciendo uso de los servicios y aplicaciones. Por ejemplo:

- Los usuarios crean contenidos como en el caso de los blogs y las wikis, registrando su conocimiento.
- Comparten objetos digitales, como videos, fotografías, documentos y enlaces favoritos.
- Aportan comentarios sobre el contenido en la web.
- Realizan valoraciones con puntuaciones respecto a algo.

Un buen ejemplo en la utilización de la web como plataforma es Google Apps, donde se editan, guarda y comparten documentos de texto, plantillas, presentaciones etc., de modo que provee un sistema de ofimática en línea.

Sin olvidar a nivel de negocio, una acción que se realiza de forma general y sencilla es la compra de productos, la cual causa inquietud, pero se animan al ver las opiniones de los usuarios que ya han comprado y lo describen de mejor forma que el vendedor.

Hay que mencionar, que el ordenador ya no es el único dispositivo para acceder a los servicios y aplicaciones de internet, ahora existen distintos aparatos que poseen conexión a internet, lo cual conlleva el aumento de usuarios de distintas edades en la red que se conectan con personas de otros lugares del mundo.

A su vez, la web 2.0 proporciona aplicaciones más pequeñas y más especializadas que disminuyen la curva de aprendizaje en los usuarios.

1.2.3. Tecnología utilizada

Existen tecnologías consideradas 2.0 las cuales constituyen las aplicaciones en la web. Estas aplicaciones son las que el permiten a los individuos comunicarse unos con otros y mantener una relación en la web, así también se suele incluir aquellas herramientas y servicios que permiten compartir información y objetos digitales.

La mayor parte de estas aplicaciones hacen uso de las tecnologías siguientes:

- XML: es el lenguaje para representar y compartir datos en la web. Tiene la capacidad de ser transferidos a través de RSS o haciendo uso del protocolo HTTP (hypertext transfer protocol).
- JavaScript: es un lenguaje de programación compatible con todos los navegadores. Generalmente se usan en formularios donde se deban de validar los datos solicitados al usuario o en imágenes desplegadas cuando se desplaza el ratón por alguna parte y otras funcionalidades.
- Ruby on Rails: es una estructura para construir aplicaciones web que acceden a base de datos. Permite escribir código evitando que se repita y favorece los aspectos únicos de la aplicación, sin preocuparse de la configuración.
- PHP: es un lenguaje de código abierto muy popular especialmente adecuado web y puede ser incrustado en HTML. Lo que distingue a PHP es que el código es ejecutado desde el servidor generando HTML y enviándolo al cliente, por lo que el cliente recibirá resultados, sin saber que código ha producido el resultado recibido y a la vez se obtiene un contenido más dinámico.

- Ajax: es una combinación de xml y javascript que posibilita crear aplicaciones web ejecutables en el cliente, reduciendo el tráfico de datos y la carga de trabajo del servidor, brindando mayor interactividad con el usuario. Por ejemplo Google Maps, GMail o Flickr hacen uso de ella.
- La plataforma ASP.NET de Microsoft y las familias de lenguajes como C# Y VB.NET proporcionan también aplicaciones web en el servidor.
- DHTML: utiliza HTML, JavaScript, y hojas de estilo en cascada para cambiar la apariencia o función de las páginas estáticas.
- RSS (Really Simple Syndication): es una de las tecnologías que facilita la publicación de las aplicaciones Web 2.0. Un RSS es una transmisión de datos formateado en XML.

1.3. Gestores de contenidos

Derivado de las necesidades emergentes de la web 1.0 y las características impulsadas en la transición hacia la web 2.0, surgen los gestores de contenidos.

Este concepto usualmente es asignado a los portales web, sin embargo, J. J. Merelo. Mereloen en un curso en el 2005, expresa que un gestor de contenido es: un conjunto de programas instalados en un servidor web y que comúnmente se apoya en una base de datos, además, de incluir programas cliente que brindan facilidad en el acceso a dichos programas.

Y desde la perspectiva del usuario final, se trata de administrar de forma estándar, fácil, un sitio web dinámico, con actualizaciones de forma periódica. Además, de permitir que una o más personas pueden realizar actividades según roles previamente determinados.

Los gestores de contenidos de la web 1.0, exigían que los administradores tuviesen en lenguajes de programación, hojas de estilos y de base de datos, en contraste con los gestores de contenidos de la web 2.0, que proveen una gran facilidad en la utilización de los mismos.

Entre las características generales más importante que distinguen a este tipo de software son:

- Facilidad en la creación de contenidos online.
- No requiere que el administrador tenga conocimientos en lenguajes de programación.
- Brinda control y distribución de las publicaciones, permitiendo administrar el período de vigencia de la publicación.
- Independencia entre el contenido y la presentación o estilo del sitio.
- Control de usuario, restringiendo las operaciones a realizar basados en catálogos de perfiles de usuarios.

También es indispensable conocer que existen diferentes tipos de gestores de contenidos, los cuales se catalogan según su uso, entre los cuales se pueden mencionar: blogs, wikis, educativos, foros y publicaciones digitales.

Blogs, son aquellos gestores de contenidos que permiten crear diversas entradas con contenido fotográfico o de video, pero que esencialmente están destinados a ser de uso personal.

Wiki, en contraste con los blogs, este tipo de gestor tiene la particularidad de permitir la participación de usuarios de la red. De modo que el conocimiento se va incrementando a medida que más usuarios participan activamente en el sitio, en otras palabras son publicaciones colaborativas.

Educativos, estos gestores de contenidos, proveen diversos módulos de apoyo para realizar cursos, tutorías, salas de trabajo, exámenes, videoconferencias, etc.

En la actualidad este gestor de contenidos está desplazando el paradigma de educación que hasta hoy día se considera efectivo. Está siendo utilizado en universidades y academias educativas de idiomas entre las más populares.

Foro, es un gestor que brinda una forma abierta de participar, tiene como objetivo el intercambio de conocimiento, información, etc.

De los tipos de gestores, es considerado uno de los más simples ya que busca ser fácil en la interacción o participación de los usuarios.

Aunque una de las características principales es la posibilidad que el usuario tiene de comentar u opinar sobre un tema de interés. Sin embargo una de las debilidades de este gestor es el tipo de indexado de contenido, ya que a la hora de realizar una búsqueda en internet, busca por el título del tema o por el nombre del autor.

Por último, se tiene el gestor de contenidos de publicaciones, este gestor está orientado a ser un facilitador en las publicaciones de organizaciones que generan gran cantidad de publicaciones y que no necesariamente requieren de la participación de terceros en la construcción de la información o que su objetivo se centra en dar a conocer información únicamente.

2. ANÁLISIS DE LA PROPUESTA DE SOLUCIÓN

El Centro de Investigaciones de la Facultad de Arquitectura (CIFA) es la entidad encargada de coordinar la investigación a corto, mediano y largo plazo en los diversos campos de la arquitectura. También es la encargada de publicar los resultados de las investigaciones.

Dentro del proceso de investigación como al final del mismo se producen diversos resultados, como pueden ser documentos, videos, fotografías, planos, etc. Estos documentos pueden presentarse de forma independiente o pueden ser organizados para realizar una publicación en una revista o artículo.

Sin embargo, el costo de realizar publicaciones por medios escritos es muy costoso. A pesar que actualmente se cuenta con el apoyo de de la Dirección General de Investigación de la Universidad de San Carlos de Guatemala (DIGI), para realizar publicaciones digitales dentro de un sitio web, el proceso de publicación no es instantáneo, debido a que está ligado a autorizaciones previas a publicar el contenido.

Además, el sitio WEB de CIFA que actualmente alberga la DIGI es un sitio WEB que está desarrollado completamente con tecnología HTML, siendo un sitio que depende de la intervención del administrador de servidores de dicha institución para realizar la actualización del sitio.

Sin embargo, el sitio actualmente cumple con el objetivo que es ser una tecnología de apoyo para dar a conocer el trabajo realizado por la institución.

No obstante, se desea mejorar el proceso de publicación actual, con el objetivo de obtener una mayor independencia y flexibilidad.

Además, de existir el apoyo de DIGI hacia CIFA para permitir hospedar aplicaciones, brindando los recursos y tecnologías que se encuentran instalados dentro de sus servidores.

2.1. Identificación del problema

El proceso actual para realizar las publicaciones, inicia con la revisión y validación de los resultados de la investigación. Posteriormente, se procede a seleccionar el contenido de texto, imágenes y videos que se van a publicar.

Esta información es trasladada a un colaborador para que agregue esta información en la plantilla que se tiene del sitio y se distribuya de forma apropiada de modo que la publicación este bien estructurada, a este proceso se le conoce como maquetado.

La información maquetada es supervisada una vez más por el colaborador responsable de publicaciones y de estar correcta autoriza guardar la información en un CD, este a su vez es enviado a la DIGI.

La publicación espera la autorización de la DIGI y posteriormente es trasladado al encargado de la administración de servidores de la unidad de informática.

El encargado ingresa a repositorio en donde se encuentra alojada la aplicación del CIFA y procede a realizar la actualización de páginas, archivos de video o imágenes y todo lo relacionado a la publicación.

Al conocer el proceso completo de publicación actual provee información importante para identificar actividades que pueden ser optimizadas. Estas optimizaciones se identifican en las actividades de reproceso, las cuales a su vez generan un costo escondido para el dueño del proceso, en este caso el costo es el tiempo.

Los reproceso identificados son: el maquetado de la información, la intervención del administrador de servidores.

Ya que estas actividades son las que no permiten la generación constante y rápida de publicaciones y la independencia para realizarlas en cualquier momento.

El maquetado de la información proviene de la limitante de la tecnología utilizada dentro del sitio actual. Pero a su vez el impacto lo produce en el tiempo que conlleva realizar esta actividad aun cuando la publicación a realizar no es grande.

Para optimizar la actividad de maquetado es necesario independizar el estilo del sitio de la información a publicar, con el objetivo que solo sea necesario ingresar la información y esta se acople al formato definido, o viceversa, que un cambio de estilo no afecte la información, esto evitará invertir tiempo en la estructuración del contenido.

Sin embargo, esto será posible únicamente si se utiliza tecnologías web que brinden la manipulación de información en tiempo de ejecución, en otras palabras que soporten contenido dinámico.

La intervención del administrador de servidores, actualmente es indispensable esto debido a que el contenido del sitio es estático. Sin embargo, para independizar las publicaciones del CIFA la alternativa será contar con una herramienta de software que brinde o traslade las operaciones y responsabilidad del administrador de servidores a un usuario de CIFA, el cual sea el encargado de realizar la publicación.

Sin embargo, al realizar este traslado es fundamental considerar que la herramienta debe funcionar en un ambiente WEB, debe brindar seguridad de acceso, amigable, fácil de utilizar e intuitiva de tal manera que no requiera conocimientos avanzados de informática para realizar dicha actividad. Ya que la persona que estará utilizando dicha herramienta posiblemente no sea una persona con perfil informático.

2.2. Solución al problema

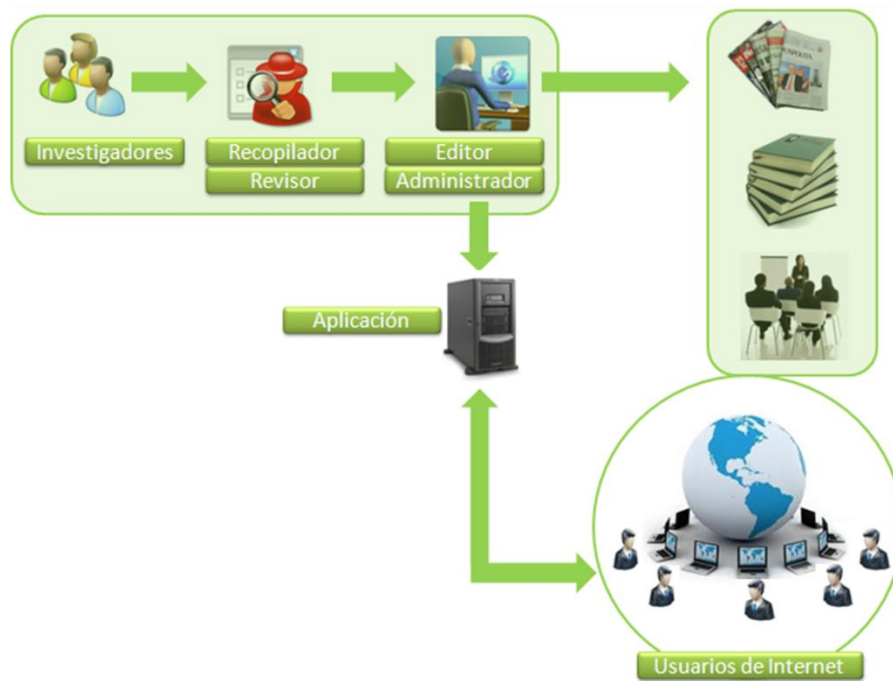
La solución tiene como principal objetivo agilizar el proceso de publicación en un entorno de flexibilidad de horario. Y basados en el proceso actual se ha identificado el nuevo proceso.

Por ello, el proceso óptimo debe iniciar con la revisión y validación de los resultados de la investigación. Posteriormente, se procede a seleccionar el contenido de texto, imágenes y videos que se van a publicar.

La información es trasladada al colaborador que está autorizado para realizar la publicación al quien se llamará editor. El editor ingresa al sistema por medio de un modulo de seguridad, el cual valida el acceso.

El editor publica el contenido de texto, imágenes y video y al finalizar dicha actividad el sitio se encuentra actualizado.

Figura 2. **Proceso de solución**



Fuente: elaboración propia.

A partir de las características del problema y de la propuesta de solución se ha identificado que la herramienta que se ajusta a estas necesidades es un gestor de contenidos. Un gestor de contenido para publicaciones digitales y de contenido multimedia.

Una característica importante del proyecto, es que la estructura del contenido de publicaciones de proyectos diversos es diferente a proyectos de ciudades mayas. Por esa razón se identificó qué es necesario personalizar una herramienta que brinde las características deseadas y que a su vez se fácil de utilizar.

Figura 3. **Ambiente propuesto para la solución**



Fuente: elaboración propia.

2.3. **Tecnologías a utilizar**

En informática existe una gran variedad de herramientas disponibles, pero estas deben analizarse para seleccionar las que se ajustan a las necesidades del proyecto. Es por ello que se recopilan los detalles que determinaron la elección de cada herramienta.

2.3.1. Generalidades

Para iniciar el tema se deben de identificar conceptos en los que se centra el desarrollo del proyecto, y los beneficios que proveen, como resultados al proyecto.

Todo proyecto surge como respuesta a una necesidad, acorde con la visión de la organización. Tomando en cuenta lo anterior denominamos el proyecto como un gestor de contenido, definimos el concepto de sistema de gestión de contenidos (del inglés *Content Management System, CMS*) como una aplicación que permite crear una estructura de soporte para la elaboración y administración de contenidos (Información, investigación, material audiovisual, etc.), enfocado en páginas web, por parte de los participantes.

Definidos estos conceptos podemos definir el proyecto como un gestor de contenido de proyecto del Centro de Investigaciones de la Facultad de Arquitectura. Cada proyecto cuenta con participantes quienes ingresan el progreso del proyecto, adjunto materiales audiovisuales, planos, apuntes, etc. De este modo se genera conocimiento y se puede transmitir de una manera sencilla y accesible.

Tras este breve resumen de lo que es en si el proyecto se decidió el uso de diferentes tecnologías que combinadas y estructuradas bajo una arquitectura sostenible hacen del desarrollo del proyecto la mejor elección en escalabilidad, robustez, presentación y rapidez de desarrollo.

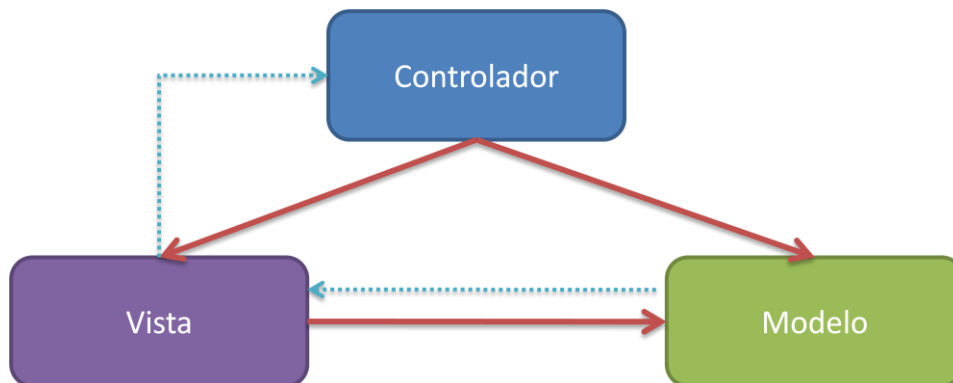
2.3.2. Descripción de arquitectura de software

Se necesita desarrollar la aplicación web que provea un desarrollo:

- Escalable
- Provea un almacenamiento seguro
- Brinde facilidad de uso

Considerando los aspectos anteriores, la arquitectura de aplicación con mejor perfil es el Modelo de Vista Controlador el cual se define así: “es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista”.

Figura 4. **Modelo vista controlador**

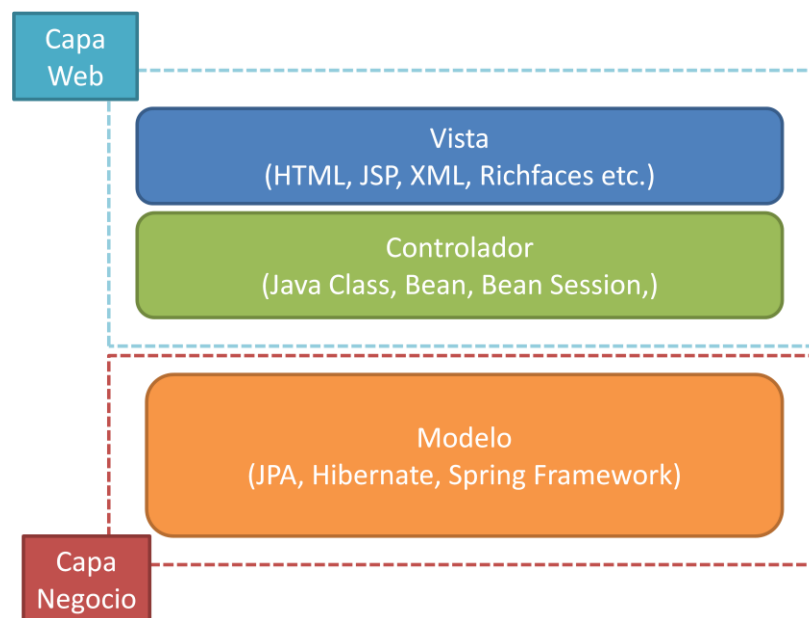


Fuente: elaboración propia.

En resumen el objetivo de la arquitectura es poder separar el desarrollo por capas, desde un punto de vista práctico (implementación) se define en:

- Capa de negocio: que contendrá las interfaces y la implementación para el acceso a la base de datos, manipulación de datos y encargada de conservar la integridad de la información.
- Capa web: contiene la implementación las clases (en este caso Bean) que genera la respuesta ya sea al mismo nivel o al nivel de capa de negocio, a las peticiones de la que surgen desde la interacción con el usuario (Páginas Web, o Java Server Pages).

Figura 5. **Distribución de tecnologías por capa**



Fuente: elaboración propia.

2.3.3. Servidor Web

Un servidor web es una herramienta que permite el almacenamiento y ejecución de aplicación web, a continuación el servidor web seleccionado.

2.3.3.1. Apache Tomcat

Es un contenedor web *HTTP* de código abierto para plataformas Unix, GNU/Linux, Microsoft Windows, Macintosh y otros sistemas operativos.

Se inicio en 1995 se tomo como código base el popular NCSA HTTPD 1.3, pero fue reescrito por completo posteriormente.

Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU.

El servidor Apache se desarrolla dentro del proyecto *HTTP Server (httpd)* de la *Apache Software Foundation*.

Desde 1996, Apache, es el servidor *HTTP* más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70 % de los sitios web en el mundo.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la *World Wide Web*. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

Apache es el componente de servidor web en la popular plataforma de aplicaciones *LAMP*, junto a *MySQL* y los lenguajes de programación *PHP/Perl/Python* (y ahora también *Ruby*).

Además de las características anteriores, Tomcat implementa Java *EE 5*. La plataforma Java *EE 5* incluye las versiones de tecnologías más recientes, como por ejemplo *JavaServerPages(JSP) 2.1*, *JavaServer Faces(JSF) 1.2*, *Servlet 2.5*, *Enterprise JavaBeans 3.0*, *Java API for Web Services(JAX-WS) 2.0*, *Java Architecturefor XML Binding(JAXB) 2.0*, *Web ServicesMetadata* para la plataforma Java 1.0.

Es importante el poder aclarar que cuando hablamos de Apache tomcat, hablamos de un contenedor web y no un servidor web. Se hace la relevancia ya que el contenedor tiene ciertas limitaciones con respecto a servidores web.

Nótese que un contenedor web no puede ejecutar todas las operaciones de la plataforma JAVA, tales como Enterprise Java Beans.

Servidores como Glassfish contienen todas las características para la ejecución de la plataforma JAVA con todo su potencial. En el caso del proyecto solo se necesita la ejecución de componentes web, por lo que en beneficio de la minimización de costos de recursos del sistema es superior que si se utilizase un servidor.

Después de haber dado una breve descripción del software, se muestran las características específicas que llevaron a la elección de la herramienta, tales razones son:

- Contenedor web que implementa tecnología JAVA.

- Licencia de Apache, que permite su utilización de forma gratuita.
- Los recursos del sistema son bajos comparados con otros contenedores.
- Independiente del Sistema Operativo.

2.3.4. Base de datos

En la actualidad es imprescindible el almacenamiento de datos, es decir que implícitamente se debe de incluir un sistema de base datos en todo sistema informático que se necesite desarrollar.

Un gestor de base de datos, proporciona un componente capaz de almacenar de forma duradera y segura la información. Se encuentran también diferentes paradigmas de bases de datos incluyendo sus respectivas implementaciones.

2.3.4.1. MySQL Server

Actualmente existe la tendencia de base de datos NoSQL, que permiten el almacenar grandes cantidades de información sin depender de un paradigma relacional, pero para el presente caso la cantidad de información a almacenar no es crítica, sino la organización de dicha de información. Por lo que se selecciona una RDBMS del paradigma de tradicional relacional.

Hoy en día existen infinidad de opciones disponibles en el mercado. Desde el mejor calificado Oracle, hasta implementaciones de pequeñas bases de datos como Firebird.

La elección en este caso ha sido la base de datos *MySQL Server*, es un sistema de gestión de bases de datos relacional, se encuentra bajo licencia *GNU-GPL* para aplicaciones no privativas. Fue adquirida por Oracle *Corporation* desde abril de 2009.

Los factores de claves para su selección son:

- Capacidad de procesos multihilo y multiusuario.
- Uso gratuito para aplicaciones software libre y sin ánimo de lucro.
- Compatibilidad con la mayoría de lenguajes.
- Implementación nativa del driver para Java.
- Menor uso de los recursos del servidor.
- Una gran cantidad de documentación disponible en internet.

2.3.4.2. Mapeo objeto relacional

Para un desarrollo estándar de aplicaciones se necesitan asegurar la integridad de los datos tanto en su creación como manipulación desde la base de datos a la interfaz de usuario y viceversa. Con dicho objetivo se han desarrollado *frameworks* y API's que implementan las características de transacciones de base de datos.

En el mercado existen muchas implementaciones de forma libre o de pago. Cada ORM implementa su propia lógica para el manejo de las diferentes características de cada RDBMS. Por lo que es necesario el análisis de cada opción para identificar los beneficios y desventajas, esto relacionado con el RDBMS a utilizar. La seleccionada para este caso es Hibernate.

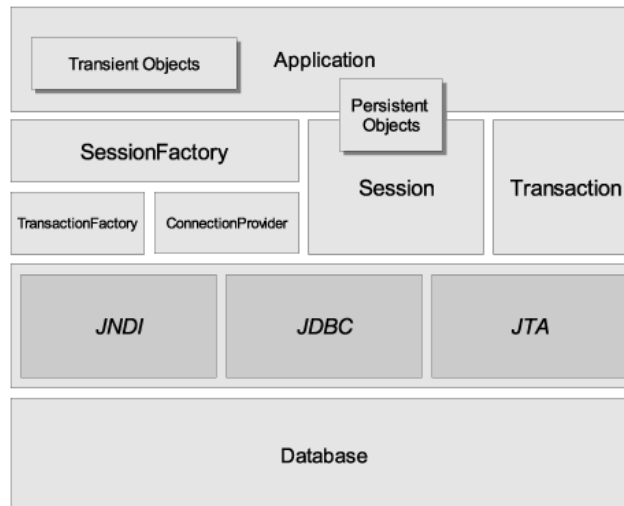
Hibernate, es un *framework* que ejecuta las funciones de mapeo Objeto Relacional (*ORM*), y es de código abierto. La característica principal de Hibernate es que permite mantener la persistencia con los datos, facilitando el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación. Esto por medio de archivos que definen las estructuras de las tablas de una base datos (*.HBN.XML) y que permiten establecer relaciones entre tablas. Además de soportar polimorfismo, colecciones, y un gran número de tipos de datos.

La selección de Hibernate se deriva de sus siguientes bondades:

- Soporta la mayoría de sistemas gestores de bases de datos SQL en el mercado.
- Se integra de buena manera y sin restricciones con los más populares servidores de aplicaciones J2EE, contenedores web, y aplicaciones cliente.
- Cumple con todas las propiedades de un ORM.
- Posee un lenguaje de consultas HQL (Hibernate Query Language).
- Se utiliza SQL nativo o consultas basadas en criterios.
- Compatible con Java.
- Amplio listado amplio de aplicaciones que respaldan su experiencia.

La arquitectura "completa" de Hibernate abstrae la aplicación de las *APIs* de *JDBC/JTA* y permite que Hibernate se encargue de los detalles.

Figura 6. **Arquitectura de Hibernate**



Fuente: <http://docs.jboss.org/hibernate/core/3.5/reference/es-ES/html/architecture.html>.

Consulta: 20 de abril de 2013.

2.3.4.3. **Persistencia**

Aplicado a los datos, se centra en conservar la integridad de los datos. Por ejemplo, un tipo de dato número en base de datos con una exactitud de 20 decimales puede perder su exactitud en la conversión a un tipo de dato numérico de un lenguaje de programación que soporte solo 8 decimales, en este caso no existe la persistencia de datos, en un ámbito científico o económico representan un gran problema. Derivado de esto la importancia de la persistencia, el poder mantener la integridad del dato tal y como se obtuvo. La Herramienta elegida fue *Java Persistence API*.

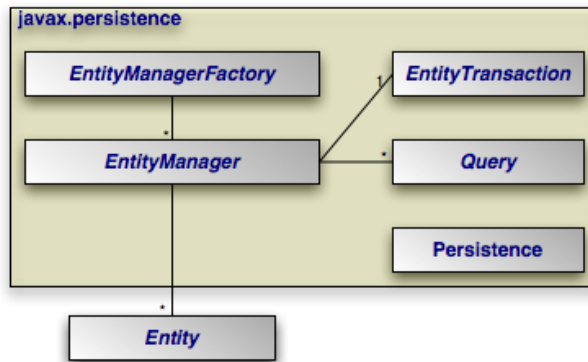
2.3.4.4. Java Persistence API

Es una interfaz de programación que implementa la comunicación entre componentes de software desarrollado bajo el lenguaje de programación Java y un gestor de base datos, y viceversa.

Su desarrollo inicio en *JSR 220 Expert Group* y posteriormente incluida la versión estándar *EJB3*. Cuenta con *un Persistence Query Language (JPQL)* para metadatos objeto/relacional y permite usar objetos regulares (conocidos como *POJOs*).

La principal razón para su uso en este proyecto, fue su total naturaleza java que permite mantener la integridad de manera segura entre la base de datos y el lenguaje de programación. Haciendo uso de anotaciones que permiten mantener la características entre un tipo de dato propio de base datos y su equivalente en el lenguaje de programación, incluyendo llaves compuestas, relaciones y tipos de datos especiales de cada gestor de base datos (*BLOB, File, Timestamp, etc.*).

Figura 7. **Arquitectura de Java Persistence API**



Fuente: elaboración propia.

2.3.4.5. Transacciones

Comúnmente en el área informática, existe el dicho “Para que inventar el agua azucarada”, esto se explica cómo no hay que programar lo que ya existe. En el ámbito informático se cuentan con variedad de framework que ejecutan procesos monótonos y repetitivos, lo cual evita se tenga que programar indefinidas veces. Un ejemplo claro es Spring framework que contiene las interfaces, implementaciones y librerías que permiten la interacción entre base de datos, *API's (JPA)* y otros framework que cooperan entre sí para la manipulación adecuada de datos.

En el caso de este proyecto el acceso a base datos se maneja con una combinación de Java Persistence Api (JPA) e Hibernate. Debido a que Hibernate facilita la configuración para el acceso a la base de datos y puede configurarse de forma fácil e inmediata en un ambiente de producción. Por su lado JPA proporciona una abstracción muy similar a la base de datos, ahorra tiempo de desarrollo y mantiene una integridad en los datos.

2.3.5. Tecnologías para desarrollo de software

La tecnología ha avanzado en el ámbito del desarrollo de software, creando una competencia de las diversas opciones de desarrollo, entre software libre y privativo.

Entre todas estas opciones se realizó una investigación y estudio de cada tecnología, basado en rendimiento, documentación, facilidad de uso, madurez de la herramienta y las restricciones de uso, en el sentido de uso de licencias.

2.3.5.1. Spring framework

El Spring framework (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java. Incluido en la publicación del libro *Expert One-on-One Java EE Design and Development* (Wrox Press, octubre 2002), el autor Rod Johnson desarrolló la primera versión bajo licencia Apache 2.0 License en junio de 2003.

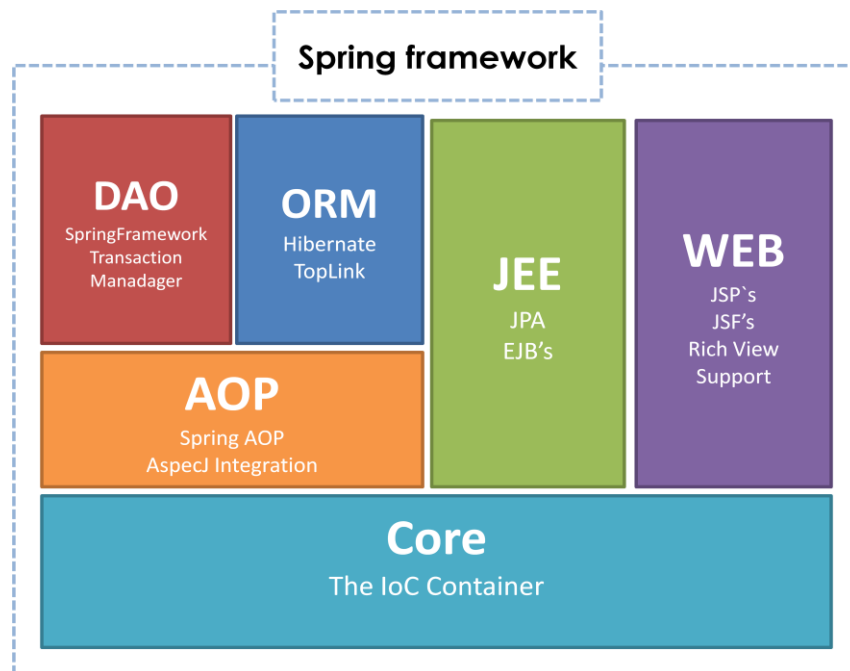
La versión 1.0, apareció en marzo de 2004. No obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al ser considerado como una alternativa y sustituto del modelo de *Enterprise JavaBean*. Por la libertad que ofrece a los desarrolladores en Java, Buena y abundante documentación y facilidad de uso.

Los pilares claves de este framework son:

- POJO: (Plain Old Java Object) enfatiza la simplicidad de las clases Java aportando manejo de transacciones de forma no intrusiva.

- XML: configuración basada en archivos XML, hace posible ajuste y cambios en producción.
- Seguridad: implementado como un aspecto (AOP) a través del framework Acegi.
- Remoting: ejecutando Java Remote Method Invocation (RMI) simplificado con acceso y publicación de web services.
- Testing (Pruebas): provee un package de testing específico para componentes del framework e integrado con JUnit.

Figura 8. **Arquitectura de Spring framework**



Fuente: elaboración propia.

Spring framework aportó la parte de implementación para la capa de control, ya que se integra de buena manera con Hibernate y JPA.

Permite una funcionalidad fluida entre la capa de control y la de base de datos, esto debido a su capacidad de adaptación a diferentes ORM.

La elección de la misma, es clave en el ahorro de tiempo de desarrollo de implementaciones, ya que la mayoría de operaciones mecánicas y repetitivas están contenidas en el framework, por lo que el desarrollador solo se preocupa por adaptar esa gama de funciones a la lógica de negocio del sistema. A continuación se presenta la descripción de los componentes que conforman Spring framework.

Tabla I. **Componentes que conforman Spring framework**

Componente	Descripción
DAO	El módulo OP, se encarga de la lógica del manejo de transacciones hacia el gestor de base de datos.
ORM	La funcionalidad <i>h</i> a este nivel está orientada a la compatibilidad e interacción con herramientas como <i>Hibernate, JDO, JPA, etc.</i>
JEE	Esta capa se encarga de la integración con la variedad de servicios <i>Enterprise</i> de la plataforma java
WEB	Este nivel proporciona componentes que proveen la versatilidad de implementaciones que interactúen con otros <i>framework</i> , aunque su facilidad a dirigido su mayor uso a plataformas web.
AOP	Una implementación de desarrollo según el paradigma de Orientación a Aspectos (<i>AOP</i>) que se presenta como una estructura simplificada para el desarrollo y utilización de aspectos (módulos de <i>MOC- Multiple Object Crosscutting</i>)
Core	Implementado en base a la Técnica Inversión de Control (<i>IoC</i>), que promueve el bajo acoplamiento a partir de la inyección de dependencias entre los objetos (relaciones).

Fuente: elaboración propia.

Spring framework, fue utilizado en la capa de negocio, para aprovechar las ventajas que provee en la administración del manejo transaccional y conectividad hacia la base de datos, además de brindar una implementación de del patrón de diseño para la inyección de objetos de la capa de negocio en los *Backing Bean* de la capa de presentación.

Para implementar el Spring framework en el proyecto fue necesario realizar lo siguiente:

- Modificar el archivo de configuración `applicationContext.xml` del proyecto de negocio.
- Modificar el archivo de configuración `faces-config.xml` del proyecto de web.
- Modificación de los *Backing Bean* que implementaran los servicios del negocio.

Archivo de configuración `faces-config.xml` del proyecto de negocio. Dentro de este archivo se especifica la conexión y tipo de conexión que utilizará la aplicación para poder acceder a los recursos de base de datos.

Lo recomendable es utilizar un *DataSource*, lo cual permite hacer referencia únicamente al nombre del objeto.

Además, los objetos *DataSource* utilizan el servicio de nombres de java *JNDI (Java Naming and Directory Interface)*. Una vez que un objeto *DataSource* se encuentre registrado dentro de dicho directorio, las aplicaciones puedan usar el API JNDI para acceder al objeto *DataSource*.

Por su parte el *DataSource* contiene propiedades que identifican y describen el origen de datos. Especificando las direcciones del servidor de base de datos, puerto de conexión, protocolo de red para la comunicación, etc. Esto aporta transparencia en la conectividad de las aplicaciones que usan el *DataSource* ya que si en un momento la conexión al origen de datos sufre variaciones únicamente se realiza el cambio en la configuración del *DataSource* y no en todas las aplicaciones que se conectan a dicho recurso.

Dentro del *applicationContext* la especificación del *DataSource* se realiza de la siguiente manera.

Figura 9. Archivo de configuración



```
*applicationContext.xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:aop="http://www.sp.
  xmlns:tx="http://www.springframework.org/schema/tx" xmlns:util="http://www.spr.
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util.xsd">
  <bean id="dataSource" class="org.springframework.jndi.JndiObjectFactoryBean">
    <property name="lookupOnStartup" value="true" />
    <property name="jndiTemplate" ref="jndiTemplate" />
    <property name="jndiName" value="java:comp/env/jdbc/CIFA_DS" />
    <property name="cache" value="false" />
    <property name="proxyInterface" value="javax.sql.DataSource" />
  </bean>
```

Fuente: elaboración propia.

2.3.5.2. RichFaces

Richfaces es un conjunto de librerías de código abierto, desarrollada en Java que permite crear aplicaciones web con AJAX (Asynchronous JavaScript And XML).

Se ejecuta sobre el framework de Java Server Faces (Version 1.6). Se implementan filtros que permiten realizar peticiones *AJAX* en la página. La ventaja principal del planteamiento, es que la petición *AJAX* provoca una ejecución en el servidor y finalmente una renderización parcial o total de la página del navegador realizando un cambio del lado del cliente de manera instantánea. Así, el control de lo que sucede está en el servidor. Richfaces está desarrollado y mantenido por JBoss. Actualmente forma parte del proyecto Exadel.

Se integra con los componentes para JSF y un avanzado framework para la integración de *AJAX*, que dan una facilidad en la capacidad de desarrollo de aplicaciones de negocio. Los componentes vienen listos para su uso *out-of-the-box*, para los desarrolladores se ahorra tiempo de implementación. Una ventaja de los componentes en gran medida mejora la experiencia del usuario más fiable y más rápidamente.

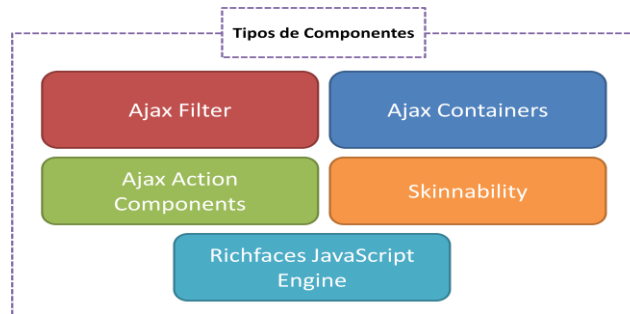
RichFaces incluye un fuerte apoyo para la *skinnability* (*temas personalizables*) y aprovecha al máximo los beneficios de JSF framework incluyendo, la validación y conversión de instalaciones, junto con la gestión de estática y dinámica los recursos.

El funcionamiento del framework consiste en que sus propias etiquetas generan eventos que envían peticiones al contenedor Ajax. Con esto se evita crear el código Javascript y el objeto XMLHttpRequest para que envíe la petición al servidor. Toda esta funcionalidad se activa únicamente con insertar la etiqueta.

Los componentes de Richfaces son:

- Ajax Filter: reconoce los mismos tipos de Request, además registran los recursos estaticos y dinamicos en el Resource Builder y el Resource Cache.
- Ajax Action Componentes: estos se encargan de capturar las solicitudes de acción.
- Ajax Containers: es una interface que define un área en el lienzo JSF, lo que permite el renderizado por sectores.
- Java Script Engine: corre simultáneamente con las áreas de JSF y se ejecutan procesos basados en la información del Ajax response.
- Skinnability: es un complemento que permite aplicar estilos a todos los componentes de forma sencilla utilizando temas predeterminados o el desarrollo de temas personalizados.

Figura 10. **Tipos de componentes de RichFaces**



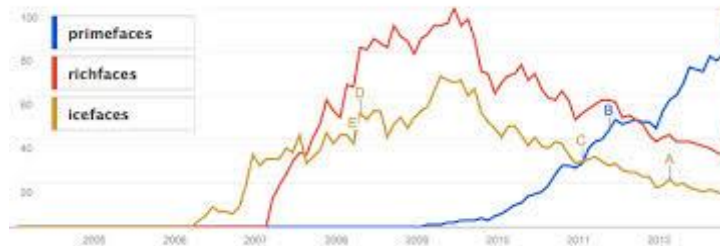
Fuente: elaboración propia.

Richfaces para ser el framework que da la cara al cliente, es un detalle delicado del diseño de sistemas ya que este debe ser capaz de contar con componentes que hagan la experiencia del usuario algo fácil y agradable. En este caso cuenta con más 400 componentes que se ajustan dependiendo de la necesidad de la función.

Un aspecto importante de resaltar es que se realizó una investigación con respecto al desempeño en producción (memoria, tiempo de respuesta, peso, etc.) de diferentes opciones, enfocado en la optimización de recursos. Se comparó con similares como ICEfaces y primefaces.

Donde en un estudio de rendimientos de cada framework, devolvió una buena calificación a richfaces.

Figura 11. **Estadística de RichFaces**



Fuente: <http://ingmmurillo.blogspot.com/2012/07/popularidad-de-primefaces.html>. Consulta: 20 de abril de 2013

La decisión se tomo en base al rendimiento y tiempo en el mercado, ya que es un factor importante para identificar el acceso a soporte disponible.

2.3.5.3. **PHP**

PHP significa Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Rasmus Lerdorf en 1994, inicio el desarrollo el kit de herramientas; sin embargo la implementación principal de PHP es producida por The PHP Group de donde se define el estándar. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

PHP posee una programación estructurada, al estilo C y Perl, lo que permiten crear aplicaciones complejas con una curva de aprendizaje muy corta. Al igual involucrarse con aplicaciones de contenido dinámico.

Su funcionamiento, consiste en que el cliente hace una petición al servidor. El servidor por su parte ejecuta el intérprete de *PHP*. Éste procesa el script solicitado que generará el contenido de manera dinámica. El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. El código fuente escrito en *PHP* se ejecuta en el servidor y envía su resultado *HTML* al navegador.

Una característica importante, es que permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Realizando un análisis de la herramienta, se definieron:

Otras ventajas:

- Es un lenguaje multiplataforma.
- Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5).

Inconvenientes:

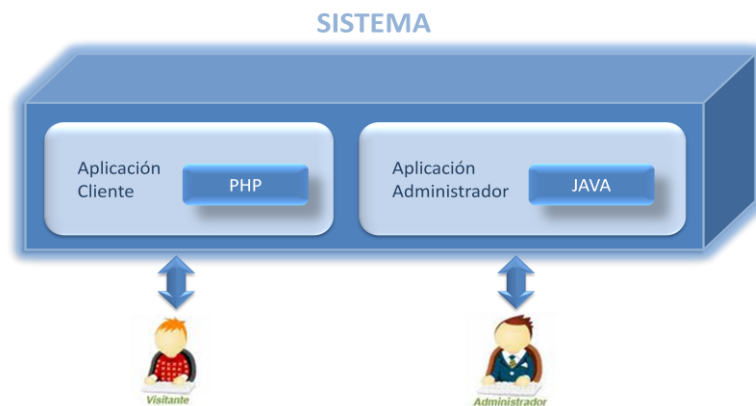
- Por ser un lenguaje interpretado, el código fuente no es totalmente oculto.

3. IMPLEMENTACIÓN Y RESULTADOS

Para implementar el sistema fue necesario dividirlo en dos proyectos, el primero denominado la aplicación del cliente y el otro denominado la aplicación del administrador.

Debido a que esta aplicación es la encargada de visualizar el contenido que ingresa el administrador, se ha identificado que la tecnología de desarrollo que mejor se ajusta a las necesidades de esta aplicación es PHP. Y para la aplicación del administrador se selecciono Java.

Figura 12. **División del sistema**



Fuente: elaboración propia.

3.1. Aplicación del cliente

En esta aplicación estará disponible a todo usuario que navegue en internet, por ello, el factor determinante de éxito de esta aplicación reside en gran parte de la usabilidad y amigabilidad de la misma. Por ello, fue necesario recopilar información y características visuales.

Dentro de las características visuales se identificaron tres diseños de pantallas y la distribución de cada una de ellas. Los diseños de páginas son: Página principal, secundaria y galería.

3.1.1. Diseño de página principal

La página principal, como su nombre lo indica es el diseño de página empleado en la página de ingreso de la aplicación el cliente, cuya finalidad es presentar la información relevante del sitio, evitando el exceso de información para no generar contaminación visual para los usuarios visitantes.

Este diseño cuenta con un apartado en la parte superior de la página que comúnmente se conoce como área de banner, esta sección contiene un conjunto de imágenes representativas de la institución o del contenido del sitio. Por ser la página principal este banner será interactivo, ya que permitirá que pueda mostrarse una a una las imágenes en una frecuencia de tiempo apropiado.

En la parte superior del área de banner, se ha posicionado el área de menú, un área importante para brindar facilidad en la navegación del usuario dentro de las opciones más importantes del sitio.

Además, se incorporo en la parte derecha de la página e inferior al área de banner, el área de enlaces que a su vez se divide en dos grupos: enlaces principales y enlaces relacionados.

Enlaces principales, esta sección tiene como propósito contener vínculos relacionados directamente con el contenido del sitio o con la institución, por otra parte, la sección de enlaces relacionados tiene la finalidad de incorporar vínculos a sitios y documentos de información relacionada al sitio.

Por último, está el área de contenido que para esta página es un área en donde puede mostrarse información breve e importante para el usuario o información importante para la institución.

Figura 13. **Diseño de página principal**



Fuente: elaboración propia.

3.1.2. Diseño de página secundaria

La página secundaria, es empleada para el contenido que no es parte de la página principal, pero tampoco es contenido de material de galería.

El diseño de esta página contiene un área de banner, que a diferencia del banner de la página principal este no será interactivo, únicamente mostrará una imagen. Esto debido a que el contenido de interés del usuario visitante se encontrará en este tipo de páginas, por ello, el tiempo visita de esta página será mayor y no debe desviarse la atención del usuario del contenido hacía otro elemento de la página.

También esta página incorpora un área de menú, en el cual se alojarán las opciones relevantes del sitio. Además, incluye un área de contenido que contiene más de un 70 % de la distribución de la página con el propósito que el usuario pueda enfocar su atención hacia el contenido visitado.

Figura 14. **Diseño de página secundaria**



Fuente: elaboración propia.

3.1.3. Diseño de página de galería

Este estilo de página ha sido diseñado para presentar imágenes y videos, además, se requiere que exista la menor cantidad de elementos visuales a los indicadores anteriormente, para evitar que la atención el usuario pueda afectarse por otro elemento del sitio.

Por ello, esta página cuenta con el área de menú en la parte superior de la página para dar la posibilidad al usuario visitante de navegar por las opciones importantes del sitio. Y el resto de la página es el correspondiente al área de contenido, en el cuál se agregaran las imágenes y videos de una forma interactiva, para brindarle al usuario una buena experiencia en el sitio.

Figura 15. **Diseño de página de galería**



Fuente: elaboración propia.

3.2. Aplicación del administrador

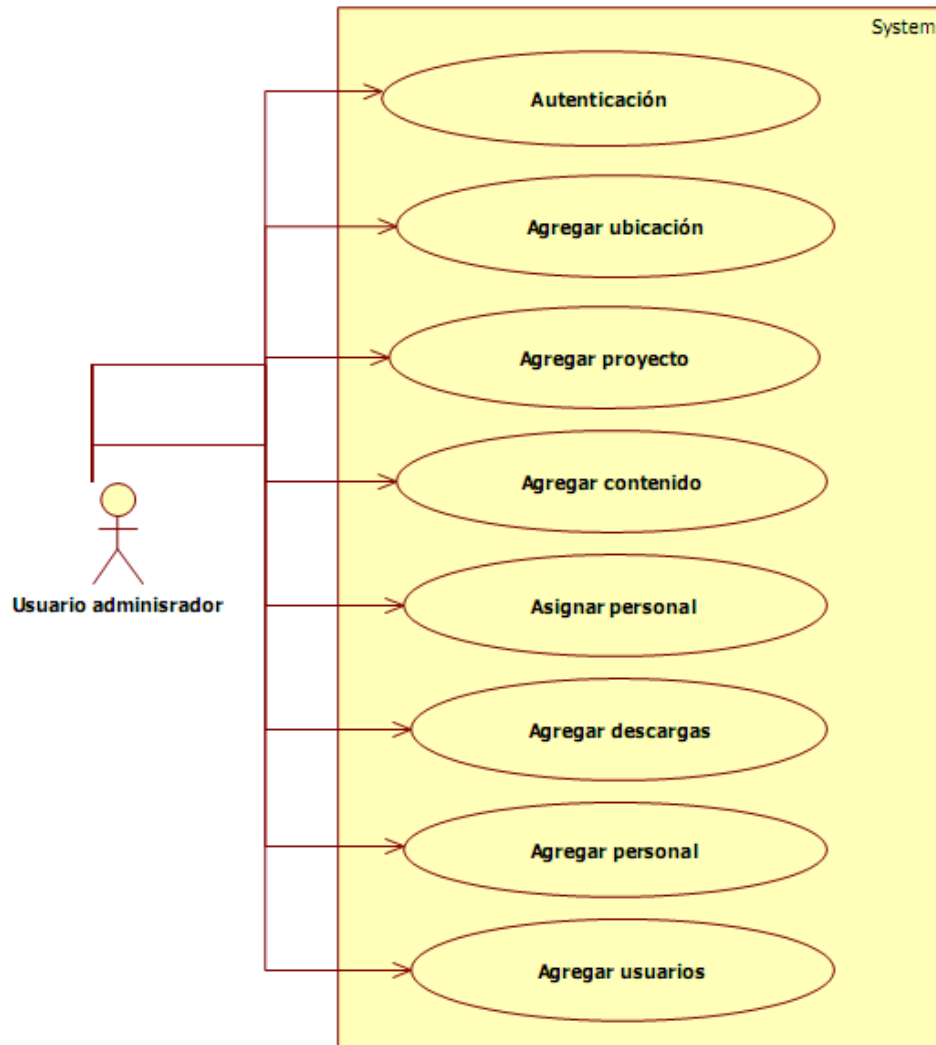
Este proyecto a diferencia del proyecto del cliente no debe ser accedido por cualquier usuario, sino que debe estar restringido a un grupo mínimo de usuario con previa autorización por la institución. Por ello, fue necesario desarrollar un componente de seguridad.

Para el ingreso de seguridad de diseño una página que permite el ingreso del usuario y contraseña, esta información es enviada a un componente validador de credenciales. Si este componente al validar determina que la información es correcta asigna un ticket de acceso por sesión y este ticket es validado en cada una de las páginas de la aplicación.

3.2.1. Análisis y diseño

Se procedió a trasladar los requerimientos del proyecto a diagramas de casos de uso con sus respectivas descripciones, ya que estos brindan una mejor abstracción de la solución y facilita la comunicación y la clarificación de la solución para los dueños de los productos de software, como para el equipo de desarrollo, brindando apoyo en la definición del alcance del proyecto.

Figura 16. **Módulo de administración**



Fuente: elaboración propia.

3.2.1.1. Autenticación

Debido a que esta aplicación no será accedida por todos los usuarios, es necesario establecer el conjunto de actividades a realizar para permitir o denegar el acceso a los usuarios.

Tabla II. Descripción del caso de uso de autenticación

Identificador	Autenticación										
Descripción	El sistema deberá validar la autenticación del usuario de internet según se describe en el siguiente caso de uso:										
Precondición	El usuario de internet deberá estar registrado en el sistema y contar con un usuario y contraseña.										
Secuencia Normal	<table border="1"><thead><tr><th>Paso</th><th>Acción</th></tr></thead><tbody><tr><td>1</td><td>El usuario ingresa el usuario y la contraseña respectiva.</td></tr><tr><td>2</td><td>El sistema, valida la información proporcionada.</td></tr><tr><td>2a</td><td>Si (usuario o contraseña errónea), el sistema muestra un mensaje de error.</td></tr><tr><td>3</td><td>El sistema muestra la página de inicio.</td></tr></tbody></table>	Paso	Acción	1	El usuario ingresa el usuario y la contraseña respectiva.	2	El sistema, valida la información proporcionada.	2a	Si (usuario o contraseña errónea), el sistema muestra un mensaje de error.	3	El sistema muestra la página de inicio.
Paso	Acción										
1	El usuario ingresa el usuario y la contraseña respectiva.										
2	El sistema, valida la información proporcionada.										
2a	Si (usuario o contraseña errónea), el sistema muestra un mensaje de error.										
3	El sistema muestra la página de inicio.										
Post condición	Ninguna										
Comentarios	Ninguno										

Fuente: elaboración propia.

3.2.1.2. Agregar ubicación

A pesar que el sistema incluye una carga inicial de información, es necesario habilitar un módulo para permitir el ingreso de ubicaciones si fuese necesario, entiéndase esto como departamentos y municipios por medio de una interface para el usuario.

Tabla III. Descripción del caso de uso de agregar ubicación

Identificador	Agregar ubicación														
Descripción	El sistema debe permitir el ingreso de nuevos departamentos según se describe en el siguiente caso de uso:														
Precondición	El usuario de internet se debe autenticar.														
Secuencia Normal	<table border="1"><thead><tr><th>Paso</th><th>Acción</th></tr></thead><tbody><tr><td>1</td><td>El usuario ingresa el nombre del departamento.</td></tr><tr><td>2a</td><td>Si el departamento existe, entonces, paso 3.</td></tr><tr><td>2</td><td>El usuario agregar el departamento.</td></tr><tr><td>3</td><td>El usuario ingresa el nombre del municipio.</td></tr><tr><td>3a</td><td>El sistema muestra un mensaje indicando que el municipio ya existe.</td></tr><tr><td>4</td><td>El usuario agrega el municipio.</td></tr></tbody></table>	Paso	Acción	1	El usuario ingresa el nombre del departamento.	2a	Si el departamento existe, entonces, paso 3.	2	El usuario agregar el departamento.	3	El usuario ingresa el nombre del municipio.	3a	El sistema muestra un mensaje indicando que el municipio ya existe.	4	El usuario agrega el municipio.
Paso	Acción														
1	El usuario ingresa el nombre del departamento.														
2a	Si el departamento existe, entonces, paso 3.														
2	El usuario agregar el departamento.														
3	El usuario ingresa el nombre del municipio.														
3a	El sistema muestra un mensaje indicando que el municipio ya existe.														
4	El usuario agrega el municipio.														

Fuente: elaboración propia.

3.2.1.3. Agregar proyecto

Este módulo permitirá al usuario administrador crear diversos proyectos asociados a una ubicación específica.

Tabla IV. Descripción del caso de uso de agregar proyecto

Identificador		Agregar Proyecto	
Descripción	El sistema deberá permitir al usuario ingresar un proyecto para un departamento y municipio específico.		
Precondición	El usuario de internet deberá estar registrado en el sistema y contar con un usuario y contraseña.		
Secuencia Normal	Paso	Acción	
	1	Seleccionar la ubicación del proyecto, que es el departamento y municipio.	
	2	El sistema muestra el listado de los proyectos existentes.	
	3	El usuario ingresa el nombre y la descripción del proyecto.	
	3a	Si el proyecto ya existe el sistema devuelve un mensaje de error.	
Post condición	Ninguna		
Comentarios	Ninguno		

Fuente: elaboración propia.

3.2.1.4. Agregar contenido

Este módulo es básico para el sistema ya que este es el responsable de permitir la gestión del contenido de cada uno de los proyectos

Tabla V. Descripción del caso de uso de agregar contenido

Identificador	Agregar contenido	
Descripción	El sistema debe permitir el ingreso de diferentes tipos de contenido nuevos para cada proyecto, según se describe en el siguiente caso de uso:	
Precondición	El usuario de internet se debe autenticar.	
Secuencia Normal	Paso	Acción
	1	El usuario selecciona el departamento a consultar.
	2	El sistema lista los municipios del departamento.
	3	El usuario selecciona el municipio a consultar.
	4	El sistema lista los proyectos del municipio.
	5	El usuario selecciona el proyecto al cual quiere cargar contenido.
	6	El sistema lista la información general del sistema.
	7	El usuario selecciona el tipo de contenido a cargar.
	8	El usuario selecciona el contenido a cargar.
	9	El usuario carga el contenido.
	9a	Si el proceso de carga finaliza sin ningún problema, entonces, muestra un mensaje de carga exitosa.
	9b	Si el proceso de carga tiene algún problema, entonces, muestra un mensaje de carga errónea.

Fuente: elaboración propia.

3.2.1.5. Agregar personal

Este módulo fue diseñado para dar crédito al trabajo que cada colaborador desempeña en cada proyecto, por ello, es necesario brindar la funcionalidad de incorporar colaboradores al sistema.

Tabla VI. Descripción del caso de uso de agregar personal

Identificador		Agregar Personal	
Descripción	El sistema deberá permitir ingresar nuevo personal, es decir personas que trabajan en los proyectos existentes en el sitio.		
Precondición	Deberá existir un usuario administrador quien deberá estar registrado en el sistema y contar con permisos para realizar la tarea.		
Secuencia Normal	Paso	Acción	
	1	Seleccionar la opción administrar personal de CIFA.	
	2	El sistema muestra el listado del personal o colaboradores de los proyectos.	
	3	El usuario ingresa la información del nuevo colaborador:	
		<ul style="list-style-type: none"> • Nombre • Apellido • Fecha de nacimiento • Fecha de defunción • Profesión 	
	4	El sistema, valida la información ingresando al nuevo personal.	
	4a.	Si la persona ya existe el sistema devuelve un mensaje de error.	
Post condición	Ninguna		
Comentarios	Ninguno		

Fuente: elaboración propia.

3.2.1.6. Asignar personal

Este módulo complementa el caso de uso de agregar personal, ya que este permite seleccionar un colaborador del listado de colaboradores de la institución para asignarlo en la participación de uno o más proyectos en lo que ha estado involucrado.

Tabla VII. Descripción del caso de uso de asignar personal

Identificador	Asignar personal	
Descripción	El sistema debe permitir la asignación de personal al proyecto, según se describe en el siguiente caso de uso:	
Precondición	El usuario de internet se debe autenticar.	
Secuencia Normal	Paso	Acción
	1	El usuario selecciona el departamento a consultar.
	2	El sistema lista los municipios del departamento.
	3	El usuario selecciona el municipio a consultar.
	4	El sistema lista los proyectos del municipio.
	5	El usuario selecciona el proyecto al cual quiere asignar personal.
	6	El sistema lista el personal que existe.
	7	El usuario selecciona la persona que desea asignar al proyecto.
	7a	Si la persona no está asignada al proyecto, entonces, muestra un mensaje de asignación exitosa.
	7b	Si la persona ya esta asignada al proyecto, entonces, muestra un mensaje informativo que la persona ya esta asignada a dicho proyecto.

Fuente: elaboración propia.

3.2.1.7. Agregar descargas

La finalidad de este módulo es permitir seleccionar que contenido podrá ser descargado y que contenido únicamente podrá ser visualizado.

Tabla VIII. Descripción del caso de uso de agregar descargas

Identificador		Agregar descargas	
Descripción	El sistema debe permitir agregar contenido para descarga de los diferentes proyectos, según se describe en el siguiente caso de uso:		
Precondición	El usuario de internet se debe autenticar.		
Secuencia Normal	Paso	Acción	
	1	El usuario selecciona el departamento a consultar.	
	2	El sistema lista los municipios del departamento.	
	3	El usuario selecciona el municipio a consultar.	
	4	El sistema lista los proyectos del municipio.	
	5	El usuario selecciona el proyecto al cual quiere asignar personal.	
	6	El sistema lista el contenido que existe del proyecto.	
	7	El usuario selecciona el contenido que desea habilitar para ser descargado.	
	7a	Si el contenido no está habilitado para ser descargado, entonces, el sistema muestra un mensaje de habilitación de descarga exitosa.	
	7b	Si el contenido ya está habilitado para descarga, entonces, el sistema muestra un mensaje informativo que el contenido seleccionar ya está habilitado para ser descargado.	

Fuente: elaboración propia.

3.2.1.8. Agregar usuario

Esta funcionalidad ha sido implementada para que si hubiese necesidad de permitir a más usuario tener acceso al sistema se puedan incorporar sin ningún inconveniente.

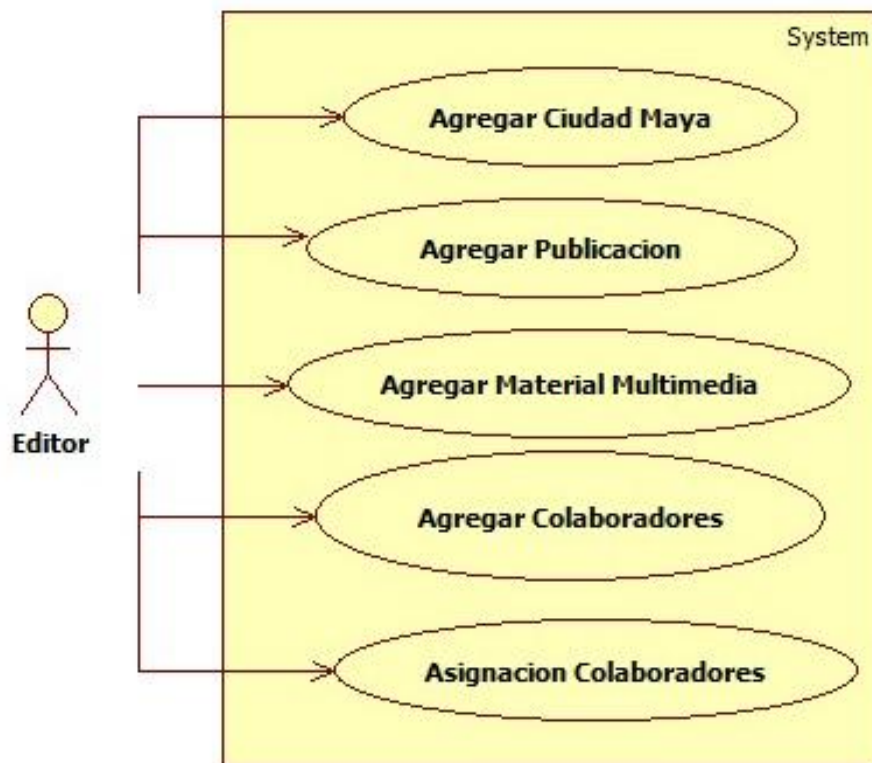
Tabla IX. Descripción del caso de uso de agregar usuario

Identificador	Agregar Usuario	
Descripción	El sistema deberá permitir ingresar nuevos usuarios.	
Precondición	Deberá existir un usuario administrador quien deberá estar registrado en el sistema y contar con permisos para realizar la tarea.	
Secuencia Normal	Paso	Acción
	1	Seleccionar la opción agregar usuario al sistema.
	2	El sistema muestra el listado de los usuarios.
	3	El usuario elige la opción crear usuario.
	4	El sistema le muestra una sección para ingresar los datos del usuario.
	5	El usuario administrador ingres el nuevo usuario y contraseña respectiva.
	5a.	Si el usuario ya existe el sistema devuelve un mensaje de error.
Post condición	Ninguna	
Comentarios	Ninguno	

Fuente: elaboración propia.

Este módulo ha sido considerado debido a características de personalización que se desea para los proyectos de ciudades de mayas.

Figura 17. **Caso de uso del módulo de edición**



Fuente: elaboración propia.

3.2.1.9. Agregar ciudad maya

Este módulo permite la incorporación de ciudades mayas según la necesidad, como característica principal resalta la incorporación de coordenadas geográficas para permitir la visualización de cada ciudad en un mapa.

Tabla X. Descripción del caso de uso de agregar ciudad maya

Identificador	Agregar ciudad maya	
Descripción	El sistema deberá permitir al usuario Agrega/Modificar Ciudades Mayas para el Atlas de Ciudades Mayas:	
Precondición	Ninguna	
Secuencia Normal	Paso	Acción
	1	El sistema visualiza los campos a llenar para agregar ciudades mayas.
	2	El Usuario ingresa el periodo al que pertenece la ciudad.
	3	El usuario selecciona el departamento de la ubicación.
	3	El sistema visualiza los municipios del departamento seleccionado.
	4	El usuario selecciona el municipio de la ubicación.
	5	El usuario ingresa las coordenadas geográficas de la ciudad. <ul style="list-style-type: none"> • Latitud • Longitud
	6	El usuario ingresa el id UNESCO (Opcional)
	7	El Usuario Presiona Guardar
	8	El sistema visualiza las ciudades existentes.

Fuente: elaboración propia.

3.2.1.10. Agregar publicación

Este módulo permite crear diversos temas relacionados a cada ciudad, asimismo, permite incluir el contenido.

Tabla XI. Descripción del caso de uso de agregar publicación

Identificador	Agregar Publicación	
Descripción	El sistema deberá permitir al usuario Agrega/Modificar Publicaciones para Ciudades Mayas para el Atlas de Ciudades Mayas:	
Precondición	La ciudad Maya a la cual se agregara la publicación debe haber sido creada anteriormente.	
Secuencia Normal	Paso	Acción
	1	El sistema visualiza las ciudades mayas disponibles.
	2	El Usuario selecciona la ciudad.
	3	El usuario ingresa el titulo de la publicación.
	3	El usuario ingresa el contenido de la ciudad (Se permite texto e imágenes).
	4	El usuario puede ingresar el orden en que se mostrara la publicación (Opcional).
	5	El Usuario Presiona Guardar
	6	El sistema visualiza las publicaciones de ciudad existentes.
Post condición	Ninguna	
Comentarios	No se necesita autenticación para poder ver el contenido de los proyectos.	

Fuente: elaboración propia.

3.2.1.11. Agregar material multimedia

Este módulo permite incluir material adicional a cada ciudad que podrá ser utilizado posteriormente en los temas publicados o de forma independiente.

Tabla XII. Descripción del caso de uso de agregar material multimedia

Identificador	Agregar Material Multimedia	
Descripción	El sistema deberá permitir al usuario Agregar material multimedia (imágenes, videos) respecto a Ciudades Mayas para la galería en el Atlas de Ciudades Mayas:	
Precondición	La ciudad Maya a la cual se agregaran los archivos debe haber sido creada anteriormente.	
Secuencia Normal	Paso	Acción
	1	El sistema visualiza las ciudades mayas disponibles.
	2	El Usuario selecciona la ciudad.
	3	El usuario selecciona la ruta del archivo(s)
	3	El sistema carga los archivos
	4	El Usuario Presiona Guardar
	5	El sistema visualiza el material multimedia existente.
Post condición	Ninguna	
Comentarios	No se necesita autenticación para poder ver el contenido de los proyectos.	

Fuente: elaboración propia.

3.2.1.12. Agregar colaborador

Inicialmente se ha realizado una carga inicial de colaboradores de la institución, sin embargo, este módulo da la posibilidad de incorporar más colaboradores.

Tabla XIII. Descripción del caso de uso de agregar colaborador

Identificador	Agregar Colaborador	
Descripción	El sistema deberá permitir ingresar nuevos colaboradores, es decir personas que trabajan en los proyectos de ciudades mayas existentes en el sitio.	
Precondición	Deberá existir un usuario administrador quien deberá estar registrado en el sistema y contar con permisos para realizar la tarea.	
Secuencia Normal	Paso	Acción
	1	Seleccionar la opción crear Investigadores.
	2	El sistema muestra el listado del personal o colaboradores de los proyectos. El usuario ingresa la información del nuevo colaborador:
	3	<ul style="list-style-type: none"> • Nombre • Apellido • Fecha de nacimiento • Fecha de defunción • Profesión (Investigador, Arquitecto, Ingeniero, etc.)
	4	El sistema, valida la información ingresando al nuevo personal.
	4a.	Si la persona ya existe el sistema devuelve un mensaje de error.

Fuente: elaboración propia.

3.2.1.13. Asignar colaborador

Este módulo permite gestionar la participación que los colaboradores tienen en cada proyecto, por ello, permite asignar y desasignar colaboradores.

Tabla XIV. Descripción del caso de uso de asignar colaborador

Identificador		Asignar Colaborador
Descripción	El sistema deberá permitir asignar/ eliminar asignación de colaboradores, es decir personas que trabajan en los proyectos de ciudades mayas existentes en el sitio.	
Precondición	Deberá existir un usuario administrador quien deberá estar registrado en el sistema y contar con permisos para realizar la tarea.	
Secuencia Normal	Paso	Acción
	1	Seleccionar la opción Asignar colaborador.
	2	El sistema muestra el listado del <ul style="list-style-type: none"> • personal disponible (izquierda) y • personal asignado colaboradores de los proyectos (derecha).
	3	El Usuario selecciona a <u>el</u> (los) colaboradores
	4	El usuario presiona el botón si asigna y este se mueve al listado de asignados
	5	El Usuario selecciona el botón desasignar y este se mueve al listado de desasignados
	6	El Usuario presiona guardar
	4a.	Si la persona ya existe el sistema devuelve un mensaje de error.

Fuente: elaboración propia.

3.3. Diagrama de clases y entidad relación

La elaboración de un diagrama de clases es una estrategia de desarrollo de software que brinda la facilidad de identificar entidades físicas del sistema a desarrollar y trasladar estas entidades a una entidad conceptual y con esta abstracción se puede logra identificar entidades de software.

Con la identificación de este tipo de entidades se facilita la actividad de reconocimiento de atributos y operaciones relacionadas a cada entidad.

Y a medida que más entidades sean reconocidas, también se puede ir identificando si existen jerarquías entre ellas, que en la POO se representa como herencia. Además, se pueden observar el tipo de relaciones e interacción entre estas.

Cabe mencionar que las aplicaciones de escritorio se facilitan el desarrollo de la POO debido a que no existen elementos externos a la aplicación que incrementen la complejidad de las interacciones de los objetos, como es el caso de las aplicaciones WEB.

Además, este proyecto se basa en el patrón de diseño MVC, que significa Modelo Vista Controlador. La estrategia de este patrón es separar los elementos de la vista de los elementos del controlador y estos del modelo, para que un cambio en la vista como pueda ser agregar o quitar un tag de html o de Richfaces, no implique hacer un cambio en el código del controlador.

Y tanto la capa del controlador como del modelo permite trabajar con clases. La capa de modelo utiliza las clases para establecer la persistencia.

La persistencia se ha realizado utilizando el API de persistencia de Java JPA. El cual permite crear una clase serializada y definir atributos, y cada atributo es vinculado a una columna de una tabla de la base de datos a partir de anotaciones de dicha API. Es importante que el tipo de dato utilizado para el atributo de la clase sea el equivalente al tipo de dato de la base de datos.

En el caso de las tablas que contiene detalles, este mapeo se realiza a través de listas de objetos de persistidos, estos objetos persistidos son a su vez instancias de la clase persistente de la tabla detalle y es en estas características en donde se aprovecha los beneficios de la POO para facilitar al desarrollador la manipulación de la información.

Figura 18. Clase persistente

```

    * Actividad generated by hbm2java
    */
    @Entity
    @Table(name = "actividad", catalog = "gestor_cifa")
    public class Actividad implements java.io.Serializable {

        private int idActividad;
        private Proyecto proyecto;
        private String descripcion;
        private Date fechaInicio;
        private Date fechaFin;
        private Integer avance;
        private Set<ActividadPersona> actividadPersonas = new HashSet<ActividadPersona>(
            0);
        private Set<ActividadRecurso> actividadRecursos = new HashSet<ActividadRecurso>(
            0);

        public Actividad() {

        }

        public Actividad(int idActividad, Proyecto proyecto, String descripcion,
            Date fechaInicio, Date fechaFin) {

        }

        public Actividad(int idActividad, Proyecto proyecto, String descripcion,
            Date fechaInicio, Date fechaFin, Integer avance,
            Set<ActividadPersona> actividadPersonas,
            Set<ActividadRecurso> actividadRecursos) {

        }

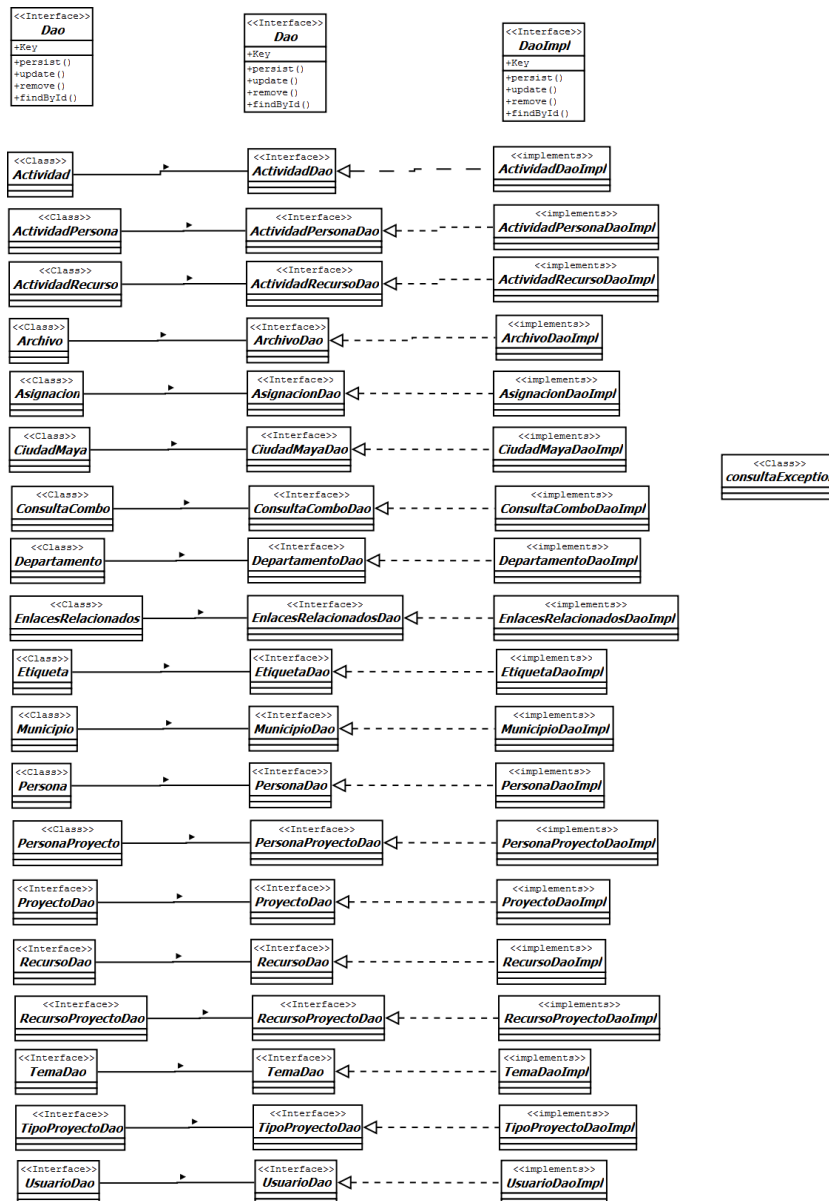
        @Id
        @Column(name = "id_actividad", unique = true, nullable = false)
        public int getIdActividad() {
            return this.idActividad;
        }
    }

```

Fuente: elaboración propia.

También el API, permite implementar los métodos de la clase DAO, que no son más que las operaciones básicas que se pueden realizar en la base de datos, tales como: select, insert, update y delete.

Figura 19. Diagrama de clases de persistencia



Fuente: elaboración propia.

La utilización de clases en la capa del controlador también es posible y esto se debe en que en diversas ocasiones es necesario, conocer el estado o valores de ciertos componentes de la capa de presentación para realizar una determinada operación. Es allí donde se hace uso de los conocidos Backing Bean.

Un Backing Bean, no es más que bean de java pero que codifican comportamientos asociados a cada Managed Bean. Otra de las características es que pueden ser compartidos por un mismo Managed Bean, permitiendo que diferentes páginas puedan hacer uso de comportamientos comunes.

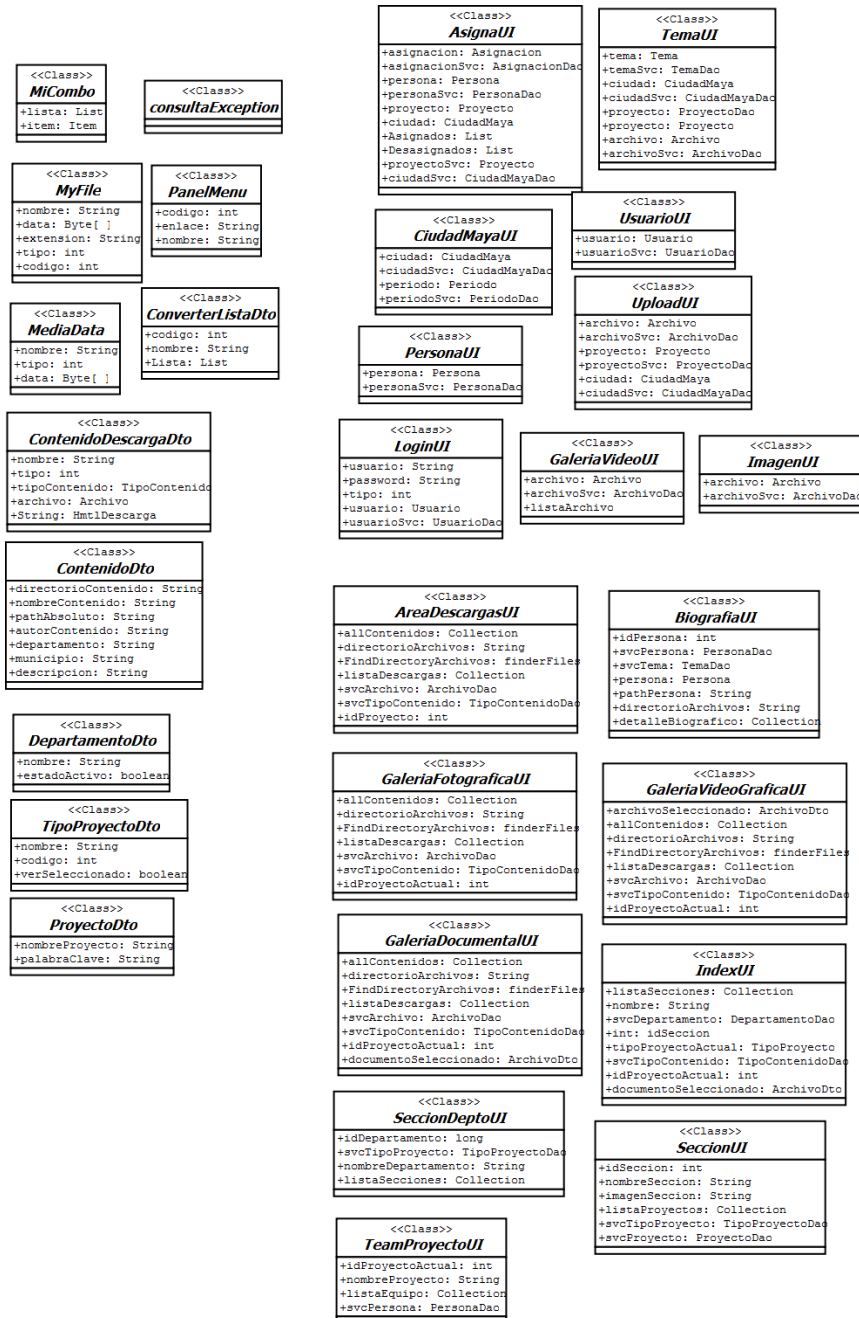
En el Backing Bean, también es posible definir servicios de la capa de negocio para poder ser usados dentro del mismo, permitiendo con ello la comunicación de las capas. Este servicio se define con un atributo normal dentro de la clase, sin embargo este servicio debe estar previamente definido en la capa del modelo.

Figura 20. **Inyección de servicios**

```
<?xml version="1.0"?>
<!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD JavaServer Faces Config 1.1//EN"
"http://java.sun.com/dtd/web-facesconfig_1_1.dtd">
<faces-config>
  <managed-bean>
    <managed-bean-name>bkn_cifa_seccion</managed-bean-name>
    <managed-bean-class>gt.edu.cifa.web.SeccionUI</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
    <managed-property>
      <property-name>svcTipoProyecto</property-name>
      <value>#{TipoProyectoCrud}</value>
    </managed-property>
    <managed-property>
      <property-name>svcProyecto</property-name>
      <value>#{ProyectoCrud}</value>
    </managed-property>
  </managed-bean>
</faces-config>
```

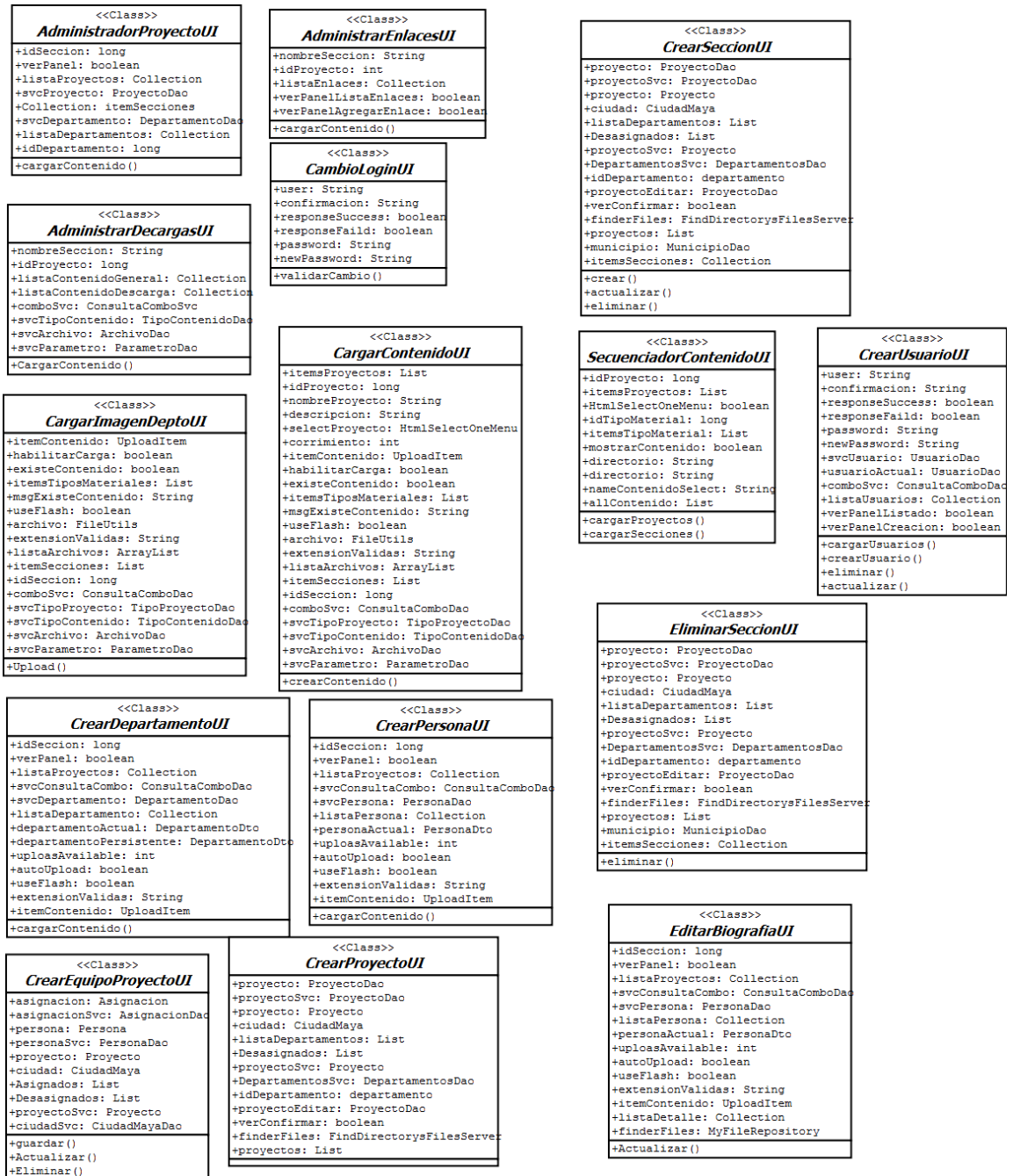
Fuente: elaboración propia.

Figura 21. Diagrama de clases de la capa del controlador 1



Fuente: elaboración propia.

Figura 22. Diagrama de clases de la capa del controlador 2

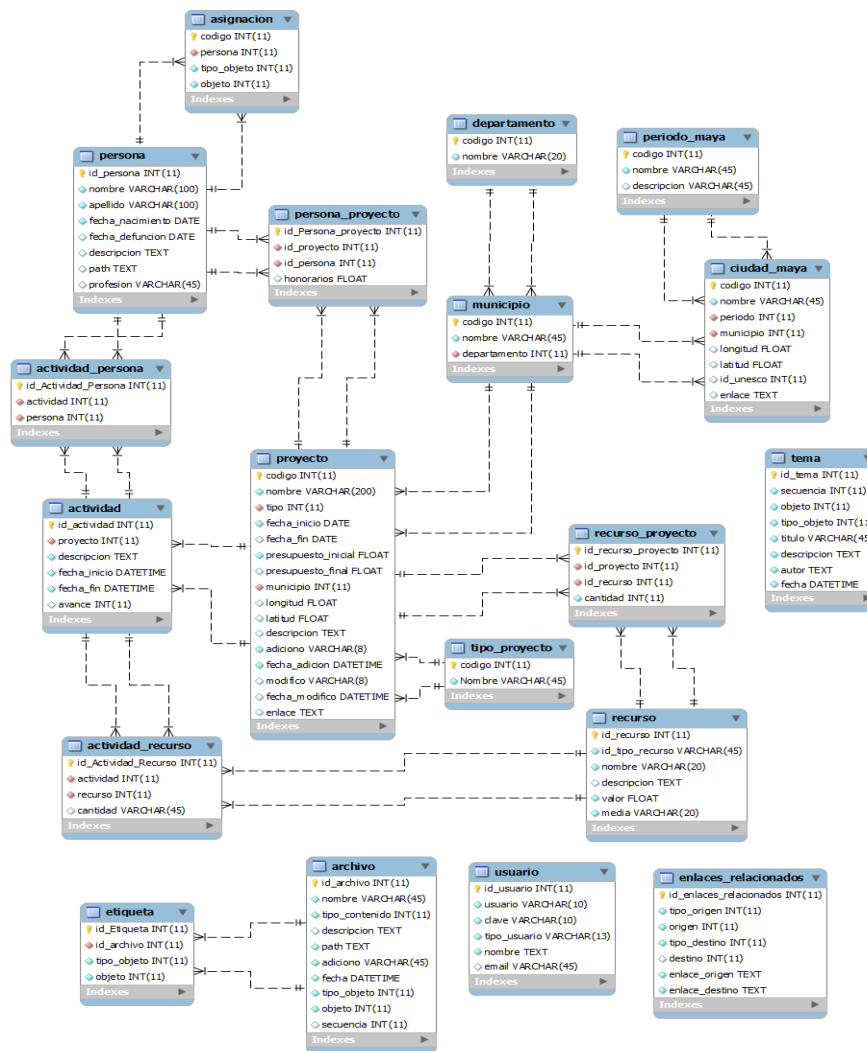


Fuente: elaboración propia.

3.4. Diagrama entidad relación

Posterior a recolectar información del sistema se procedió a identificar las estructuras de almacenamiento necesarias para satisfacer las necesidades del mismo. Como resultado de esta actividad se obtuvo el modelo de base de datos sobre el cual se realizó el desarrollo.

Figura 23. Diagrama entidad relación



Fuente: elaboración propia.

3.5. Resultados

Debido a que el objetivo de la implementación es optimizar el proceso de publicación de contenidos actual, es necesario realizar mediciones de los procesos para poder establecer si el objetivo ha sido alcanzado.

Para realizar la comparación de los procesos fue necesario identificar criterios de medición, a partir de las características de las publicaciones. También, fue necesario preparar y obtener material de publicación, y partir de esto realizar las mediciones de las actividades.

A partir de los criterios de medición se elaboró una clasificación de las publicaciones, con el objetivo de facilitar la medición y el análisis de los resultados. Las especificaciones de esta clasificación se detallan a continuación.

Tabla XV. **Especificaciones de los tipos de publicaciones**

Detalle de elemento del tipo de publicación		
Tipo de publicación	Criterios de medición	
	No. Páginas de contenido	Archivos
Pequeña	1-3	30
Mediana	4-8	70
Grande	>8	>70

Fuente: elaboración propia.

Al obtener el material a publicar se inician las actividades del proceso anterior y asimismo se mide el tiempo que conlleva la realización de cada actividad. Sin embargo, la misma actividad bajo las mismas condiciones a personas con diferentes niveles de experiencia conlleva tiempo diferente, por lo cual se obtuvo el tiempo promedio de estas mediciones, obteniendo los resultados que se detallan en la siguiente tabla.

Tabla XVI. **Medición de tiempos de publicación del proceso anterior**

Mediciones del proceso anterior			
Nombre de la actividad	Publicación		
	Pequeña (Hrs.)	Mediana (Hrs.)	Grande (Hrs.)
Maquetado de la información.	3	11	No se tiene medición
Guardar información en medio portátil.	0,08	0,08	No se tiene medición
Aprobación y publicación por DIGI.	8	8	No se tiene medición
Total	11,08	19,08	

Fuente: elaboración propia.

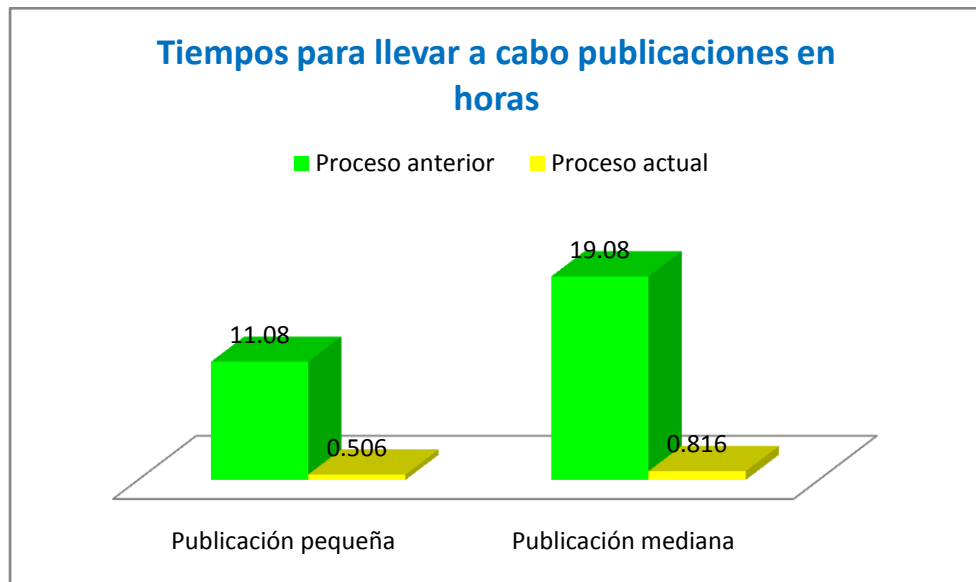
Además, el mismo material de publicación fue empleado para llevar a cabo el nuevo proceso. También en este proceso se consideró a usuarios con diferentes niveles de experiencia y los resultados que se presentan a continuación estaban basados en los tiempos promedios de dichas experiencias por cada actividad.

Tabla XVII. **Medición de tiempos de publicación del proceso actual**

Mediciones del proceso actual			
Nombre de la actividad	Publicación		
	Pequeña (Hrs.)	Mediana (Hrs.)	Grande (Hrs.)
Ingreso a la aplicación.	0,016	0,016	No se tiene medición
Carga de archivos.	0,16	0,05	No se tiene medición
Edición de contenido.	0,33	0,75	No se tiene medición
Total	0,506	0,816	

Fuente: elaboración propia.

Figura 24. **Gráfica de tiempos empleados en los tipos de publicaciones**



Fuente: elaboración propia.

Se procedió a analizar la información para determinar el porcentaje de optimización que provee el proceso actual respecto del proceso anterior.

El proceso anterior para realizar una publicación pequeña requiere aproximadamente 11,08 horas lo que representa el 100 %. Sin embargo, aplicando el proceso actual, este requiere 0,506 horas que es un 4,60 % del tiempo empleado en el proceso anterior. Reduciendo el tiempo invertido para realizar la misma tarea en un 95,40 %.

Para la publicación mediana el proceso anterior requiere 19,08 horas que es el 100 %. Sin embargo, el proceso actual requiere de 0,816 horas para llevar a cabo la misma tarea, lo que representa un 4,28% del tiempo invertido por el proceso anterior y una reducción del tiempo en un 95,72%.

Por tanto, se puede deducir que el proceso actual permitió optimizar el proceso anterior un 95,56 % del tiempo invertido en la realización de publicaciones.

CONCLUSIONES

1. Con base en la experiencia obtenida del actual proyecto, el éxito y tiempo óptimo de desarrollo de una aplicación independiente de la plataforma, la tecnología y nivel de dificultad, el factor determinante una buena toma de requerimientos.
2. La toma de requerimientos, es un proceso crítico. Pero después de este proceso es necesario elaborar una línea base referente a los requerimientos para que durante el desarrollo no se desvíe del objetivo inicial.
3. En el caso de sistemas web, el tema de la usabilidad es importante. Realizar esquemas gráficos son un apoyo para determinar como el usuario interactúa con el sistema. En el proyecto actual se trabajo con arquitectos, por lo que poder mostrar un modelo grafico de la aplicación fue una herramienta para acelerar la toma de requerimientos.

4. Se determinaron perfiles de usuario, dependiendo del módulo, las actividades que realiza y la cantidad de usuarios que utilizará dicho módulo. Este factor determina que parte del sistema requerirá ajustes de rendimiento, desde la base de desarrollo o ya sea desde la base de la infraestructura lógica y física de los servidores que proporcionan el alojamiento web. El proyecto actual se determinó que el número de usuarios que darán mantenimiento del sistema se encuentra en un promedio de 25 personas (administradores, editores). En contra parte la parte de consulta se calcularon 100 usuarios o visitantes, por lo que el enfoque de optimización va del lado de mayor cantidad de usuarios.

RECOMENDACIONES

1. A los gerentes informáticos se recomienda estudiar a fondo las necesidades del negocio, la capacidad de la infraestructura y el tiempo disponible para el desarrollo de las aplicaciones.
2. La variable de rendimiento es vital, sobre todo en sistemas donde los recursos son limitados, en este caso la parte del cliente que es únicamente de consulta se tomo la decisión de un lenguaje sobre todo liviano y rápido como lo es php.
3. Cuando se trata de velocidad de desarrollo lenguajes como java son la opción recomendada, por contar con miles de implementaciones en la cual existen variedad de opciones que disminuyen el tiempo de programación de funciones que son repetitivas.
4. Actualmente existen aplicaciones llamadas Mockup, este software permite mostrar una versión grafica del producto. Lo que apoya al momento de tomar decisiones de funcionalidad.
5. Se le insta a las diferentes escuelas a impulsar una automatización de sistemas en plataforma web, ya que proporciona una mayor accesibilidad, y la plataforma siempre está disponible, que se resumen en proporcionar un mejor servicio.

6. Se recomienda que no existan más de 3 administradores del sistema, para evitar que se pierda el control de seguridad y depuración de la información a publicar.

7. También se recomienda que antes que las personas que están destinadas a interactuar con el sistema pueda capacitarse o se les brinde la guía de uso de la aplicación para obtener el máximo beneficio de la misma.

BIBLIOGRAFÍA

1. *Documentación de PHP*. [en línea]
<<http://www.php.net/>> [Consulta: 20 enero de 2013]
2. FOWLER, Adam. *El libro de Tomcat*. [en línea]
<<http://tomcatbook.sourceforge.net>> [Consulta: 20 enero de 2013]
3. KING, Gavin; BAUER, Christian; BERNARD, Emmanuel; EBERSOLE, Steve. *Hibernate*. [en línea]
<http://docs.jboss.org/hibernate/core/3.6/quickstart/en-US/html_single/> [Consulta: 18 enero de 2012]
4. *RichFaces Developer Guide*. [en línea]
<http://docs.jboss.org/richfaces/latest_3_3_X/en/devguide/html_single> [Consulta: 12 marzo de 2012]
5. *Sistema de gestión de contenidos*. [en línea]
<http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos> [Consulta: 24 de abril de 2012]
6. *Spring framework*. [en línea]
<<http://www.springsource.org/spring-framework#documentation>>
[Consulta: 24 mayo de 2012]