PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS

TRABAJO DE DISERTACIÓN PREVIO AL TÍTULO DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN Y COMPUTACIÓN

NOMBRE:
CRISTOPHER PÉREZ

TÍTULO:

DESARROLLO DE UNA APLICACIÓN EN AMBIENTE WEB CON DISTRIBUCIÓN GNU ORIENTADO AL MANEJO DE METADATOS DESTINADO A LA BIBLIOTECA DE LA COMUNIDAD DOMINICOS DE QUITO

QUITO, 2015

Índice

Agradecimientos 5
Introducción 6
Capítulo 1: Características del sistema de metadatos para catalogar información 7
1.1 Características del sistema de metadatos Dublin Core 7
1.2 Descripción general del esquema de metadatos Dublin Core 8
1.3 Clasificación y elementos de Dublin Core 9
1.3.1Elementos relacionados principalmente con el contenido 9
Título 9
Tema 9
Descripción 10
Fuente 10
Idioma 10
Relación 10
Cobertura 10
1.3.2 Elementos relacionados principalmente con el recurso registrado como propiedad intelectual 11
Creador 11
Editor 11
Colaboradores 11
Derechos 11
1.3.3 Elementos relacionados principalmente con la instanciación del recurso. 12
Fecha 12
Tipo 12
Formato 12
Identificador 13
1.4 Vocabulario de metadatos 13
Capítulo 2: Requerimientos para la aplicación web orientada al manejo de metadatos 17
2.1 Necesidades de la aplicación 17
2.1.1 Requerimientos básicos 17
2.1.2 Requerimientos específicos 18
2.1.3 Acceso Público 18
2.1.4 Acceso restringido 18
2.2 Descripción de módulos necesarios 21
2.2.1 Modelo de base de datos 21

2.2.2 Distribución de módulos 24
2.3 Diseño de la aplicación 25
2.3.1 Diseño de secciones públicas de la página 25
2.3.2 Diseño de secciones privadas de la página 28
Capítulo 3: Seguridades necesarias para la aplicación en ambiente web 32
3.1 Acceso al portal web 32
3.2 Restricciones para el uso del portal web 32
3.2.1 Distribución del aplicativo web 32
3.2.2 Roles de usuarios 34
3.2.2.1 Administrador 34
3.2.2.2 Editor 34
3.2.2.3 Visitante 34
3.3 Implementación de protocolos de seguridad para el portal web. 34
3.3.1 Security 34
3.3.2 Security-domain 35
3.3.2.1 Module option 35
3.3.3 Security-constraint 36
3.3.4 Login-config 37
3.3.5 Security Role 38
Capítulo 4: Aplicación de características requeridas y desarrollo de la aplicación web 39
4.1 Aplicación de diseño establecido 39
4.1.1 Distribución de Página principal 41
4.1.2 Distribución de página de información adicional 41
4.1.3 Distribución de página de contacto 42
4.1.4 Distribución de página de catálogos 43
4.2 Desarrollo de la funcionalidad de módulos 48
4.2.1 Implementación de código 48
4.2.1.1 Usuario 49
4.2.1.2 Registro 50
4.2.1.3 Catálogo 50
4.2.1.4 Ubicación 52
4.2.1.5 Media 53
4.3 Integración de módulos 54
4.3.1 Relaciones entre entidades 54
4.3.1.1 Usuario y Registro 54

4.3.1.2 Catálogo y Ubicación 56
4.3.1.3 Catálogo y Media 56
4.3.2 Mapa del aplicativo web. 58
Capítulo 5: Pruebas de funcionamiento de la aplicación 61
5.1 Pruebas de compatibilidad con navegadores 61
5.1.1 Chrome 62
5.1.2 Firefox 65
5.1.3 Internet Explorer 66
5.1.4 Observaciones 69
5.2 Pruebas individuales de módulos 70
5.3 Pruebas exhaustivas con módulos integrados 71
Capítulo 6: Conclusiones y recomendaciones del proyecto 72
6.1 Conclusiones 72
6.2 Recomendaciones 72
Bibliografía 74

Agradecimientos

Por medio del presente trabajo quiero expresar mi agradecimiento a toda mi familia quienes me han apoyado en toda mi formación como persona y como profesional. Este trabajo va dedicado para todos ellos.

Introducción

La Biblioteca San Ignacio perteneciente a la Comunidad Dominicos de Quito contiene un aproximado de 30847 volúmenes los cuales requieren de un esquema que guíe en publicación de datos por medio un sistema informático amigable que facilite varias tareas como la clasificación, registro y publicación de datos.

El presente trabajo busca ayudar a los investigadores al ingreso de datos por medio de un sistema informático orientado a web, donde se aplica el esquema de metadatos Dublin Core en el que se publican los datos ya sea de forma privada o abierta al público.

Esencialmente se requiere desarrollar un sistema informático donde se almacenen los datos referentes a los catálogos de los libros de la Biblioteca San Ignacio y que permita su acceso desde la web donde los usuarios encuentren que la funcionalidad del sistema cumple con los requerimientos para ingresar los respectivos datos siguiendo un esquema de metadatos.

Dublin Core es un sistema de modelamiento de metadatos que posee un vocabulario de 15 propiedades para ser usadas en la descripción de recursos. Este vocabulario forma parte de las especificaciones mantenidas por la DCMI[1], donde también se especifica términos de metadatos, conjuntos de clases de recursos, esquemas de codificación de vocabulario, y esquemas de codificación de sintaxis. Los elementos que contiene Dublin Core son respaldados por los estándares ISO[2] 15836:2009, ANSI/NISO[3]Z39.85-2012 y IETF RFC [4]5013.

[1] **DCMI (Dublin Core Metadata Initiative):** Junta que regula los contenidos del esquema Dublin Core.
[2] **ISO (International Organization for Standardization):** Organizmo encargado de promover el desarrollo de normas internacionales para todas las ramas industriales menos la eléctrica y electrónica.
[3] **ANSI/NISO (American National Standards Institute):** Estándar regulado por el instituto americano de estándares, el cual para este caso regula el grupo de elementos que forman los metadatos de Dublin Core
[4] **IETF RFC (The Internet Engineering Task Force, Request for Comments):** Grupo que tiene como objetivo un mejor uso de internet por medio de una buena documentación de la información.

Capítulo 1: Características del sistema de metadatos para catalogar información

1.1 Características del sistema de metadatos Dublin Core

El objetivo principal de la implementación del sistema de metadatos Dublin Core es el de facilitar la obtención de información a través de internet aplicando referencias al contenido que se desea facilitar su búsqueda. “La web podría alcanzar su verdadero potencial si se convierte en un lugar donde los datos pueden ser compartidos y procesados tanto por herramientas automáticas como por personas”.

En este caso específico se busca facilitar la búsqueda de información a través de internet de la biblioteca de la comunidad de Dominicos de Quito, lo cual se lo piensa hacer por medio de una aplicación web basado en el sistema de metadatos Dublin Core para facilitar su búsqueda a personas que deseen obtener información relacionada con la que cuenta en la ya mencionada biblioteca y que sirva como referencia a la misma comunidad.

Para lograr lo ya mencionado, se requiere conocer las características, clasificaciones elementos y especificaciones de metadatos que se indican en Dublin Core para poder implementarlos posteriormente en la aplicación Web.

Las características principales que hacen que las personas escojan este sistema de metadatos son:

El sistema tiene un alcance internacional, debido a que su objetivo es el de facilitar la obtención de información a través de internet, desde cualquier parte del mundo.

Posee calificadores y elementos de matización que ayudan a mejorar la semántica del vocabulario implementado en el sistema de metadatos.

Es bastante flexible, debido a que sus normas no obligan al usuario a seguir un modelo rígido para referenciar su información.

Es extensible, ya que las normas del esquema permiten añadir elementos al vocabulario implementado.

```
<?xml version="1.0" encoding="UTF-8"?>

<metadata
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/">

<dc:title>G.R.A. military dog tags</dc:title>
<dc:creator>Gerald R. Atwater</dc:creator>
<dc:subject xsi:type="dcterms:TGM">Dog tags</dc:subject>
<dc:subject xsi:type="dcterms:LCSH">Military</dc:subject>
<dc:subject xsi:type="dcterms:LCSH">United States. Army</dc:subject>
<dc:subject xsi:type="dcterms:LCSH">Identification numbers, Personal</dc:subject>
<dc:subject>Gerald R. Atwater</dc:subject>
<dc:subject xsi:type="dcterms:AAT">metal</dc:subject>
<dc:subject xsi:type="dcterms:LCSH">Gifts</dc:subject>
<dc:description>These are the dog tags of Gerald R. Atwater, who served in the Army for three years after he graduated high school.  The
<dc:publisher>U.S. Army</dc:publisher>
<dc:contributor>Emily G. Atwater</dc:contributor>
<dcterms:created>1960</dcterms:created>
<dc:type xsi:type="dcterms:DCMIType">Image</dc:type>
<dc:type xsi:type="dcterms:DCMIType">PhysicalObject</dc:type>
<dc:format xsi:type="dcterms:IMT">jpeg</dc:format>
<dc:format>44 KB (Image)</dc:format>
<dc:format>2" x 1" (PhysicalObject)</dc:format>
<dc:format>Metal (PhysicalObject)</dc:format>
<dc:identifier>armyobjectsfront.jpg</dc:identifier>
<dc:source>PhysicalObject, given by creator, Gerald R. Atwater.</dc:source>
<dc:language xsi:type="dcterms:ISO639-2">eng</dc:language>
<dcterms:isPartOf>armyobjectsback.jpg</dcterms:isPartOf>
<dcterms:spatial xsi:type="dcterms:TGN">Tucson</dcterms:spatial>
<dcterms:spatial xsi:type="dcterms:TGN">Arizona</dcterms:spatial>
<dcterms:temporal xsi:type="dcterms:W3CTDF">start=1959; end=1962; scheme=W3C-DTF</dcterms:temporal>
<dcterms:audience>Emily G. Atwater</dcterms:audience>
<dcterms:provenance>This object was issued to Gerald R. Atwater by the U.S. Army during his service from 1959-1962.  Ownership transferr
<dcterms:rightsholder>Emily G. Atwater</dcterms:rightsholder>
<dcterms:accrualmethod>Gift</dcterms:accrualmethod>
```

Ilustración 1: Ejemplo de aplicación del sistema de metadatos Dublin Core. (Meta-data records in XML, Emily Atwater, https://emilyatwatermslis.wordpress.com/work-examples/meta-data-records/).

1.2 Descripción general del esquema de metadatos Dublin Core

El término metadatos se lo conoce generalmente como “datos sobre datos”, pero es formalmente definido como datos que facilitan la obtención y manejo de información por medio de elementos como descripciones, explicaciones, localizaciones, etc. Por lo tanto los metadatos son datos estructurados y legibles usados para organizar datos electrónicos. En el artículo “Metadatos y prácticas de categorización” se menciona a los metadatos como “estándares que asisten en identificar, describir y localizar recursos electrónicos” (1) refiriéndose a la descripción de recursos en línea.

Sus orígenes se remontan al año 1995 en la convención ubicada en Dublin, Ohio donde varios expertos en los campos de clasificación y lenguajes de mercado se reunieron con el propósito del desarrollo de estándares para la descripción de recursos informativos y que se facilite su correspondiente recuperación.

Como producto de esta reunión se obtuvieron descripciones pensadas para su inclusión en el momento en que se elabora un tipo de escrito, pero estas descripciones fueron adquiriendo mayor atención a nivel mundial, siendo utilizadas en diversos campos.

Como ejemplo del alcance del sistema de metadatos Dublin Core, en la actualidad es bastante utilizado para la obtención de información a través de la web, mediante

lenguajes como Html[5] y XML[6] para temas relacionados mayormente a bibliotecas y documentación.

El sistema de metadatos tiene como objetivo la ubicación de datos de un tema requerido por medio de internet para su descripción, identificación, procesamiento y recuperación de un documento subido a internet que cumpla con los patrones requeridos. En resumen, el sistema de metadatos busca facilitar la búsqueda de información a través de internet por medio de sus estándares.

1.3 Clasificación y elementos de Dublin Core

El sistema Dublin Core cuenta con un conjunto de metadatos diseñados para la descripción de documentos por medio de etiquetas que describen los elementos que componen los correspondientes documentos.

Dublin Core cuenta con 15 elementos principales para ayudar en la descripción de los metadatos, los cuales se clasifican en elementos relacionados principalmente con el contenido, relacionados principalmente con el recurso registrado como propiedad intelectual, y relacionado principalmente con la instanciación del recurso.

Como aclaraciones para la inclusión de los elementos que forman parte del sistema de metadatos, no es necesario que cada uno de los elementos sea incluido en el metadato del recurso a referenciar, y estos pueden repetirse las veces que se crea conveniente.

1.3.1Elementos relacionados principalmente con el contenido

Título

Es el nombre formal que tiene el recurso referenciado.

Usualmente por medio de este elemento es como se va a conocer o descubrir el contenido, se recomienda que este campo se escriba de la misma forma que el recurso en HTML de ser el caso.

Como opción adicional se puede incluir la opción “alternative” para incluir más de un título si así se requiere. Este recurso es muy útil para incluir títulos en diferentes idiomas.

Ejemplo:

<dc:title>Título del recurso</dc:title>

<dc:title.alternative>Title of the resource</dc:title.alternative>

Tema

El tema asociado al recurso referenciado, este puede ser referenciado de muchas formas, ya sea con palabras clave, códigos de clasificación, término científico, etc.

Ejemplo:

<dc:subject>Tema del recurso</dc:subject>

[5] **HTML (Hypertext Markup Language):** Lenguaje de marcado utilizado para la construcción de páginas web.

[6] **XML (Extensible markup language):** Lenguaje de marcas utilizado para almacenar datos en forma legible.

Descripción
Información breve sobre el contenido del recurso referenciado, puede contar con varios elementos desde un resumen hasta tablas de contenidos.

Se recomienda que este campo tenga información concreta, con frases completas y con datos que proporcionen mayor información al usuario que quiera acceder al recurso. Para ello se puede incluir notas sobre el contenido, información no especificada en los otros elementos, e inclusive múltiples descripciones ordenadas por importancia.

Ejemplo:

<dc:description>Descripción del contenido</dc:subject>

Fuente
Referencias sobre otros recursos de los que se obtuvieron datos para la elaboración del contenido intelectual que se busca referenciar.

Se debe aclarar que la referencia puede referirse a la totalidad del recurso o a una parte de este, por lo que se permite añadir frases a la etiqueta para especificar el tipo de fuente

Ejemplo:

<dc:source.localizacion>http://www.ejemplo.com</dc:source.localizacion>

Idioma
Lenguaje con el que se realizó el recurso referenciado.

Si el contenido referenciado se encuentra realizado en más de un idioma, se pueden incluir varias etiquetas de idioma.

Ejemplo:

<dc:language>español</dc:language>

Relación
Referencias a otros recursos que pueden aportar al contenido intelectual que se busca referenciar.

Las relaciones se pueden especificar de forma recíproca o en un solo sentido.

Este elemento tiene importancia dentro del esquema, debido a que puede ayudar al usuario a encontrar una mayor cantidad de la información, por lo que se sugiere que la relación contenga información suficiente para que los usuarios sean capaces de citar, identificar y localizar el recurso a relacionar.

Ejemplo:

<dc:relation>http://www.ejemplo.com(Información adicional) </dc:relation>

Cobertura
Descripción de la extensión del recurso referenciado, puede tener elementos correspondientes al tema del contenido referenciado como localizaciones espaciales en caso de referencias geográficas.

Para la descripción de la cobertura se sugiere utilizar un vocabulario seleccionado de acuerdo al tipo de contenido referenciado. Además de los ejemplos de contenidos mencionados anteriormente, se sugiere utilizar nombres de lugares y períodos de tiempo para referenciar la cobertura.

Ejemplo:

<dc:coverage>Siglo XVII</dc:coverage>

1.3.2 Elementos relacionados principalmente con el recurso registrado como propiedad intelectual.

Creador
Descripción de la entidad responsable del contenido referenciado, pueden ser los nombres y apellidos de un autor, el nombre de una organización o de un servicio.

Se pueden utilizar varias etiquetas para referenciar a diferentes creadores, los cuales pueden ser incluidos y se recomienda citar a los autores incluyendo primero el apellido seguido de su primer nombre. También se sugiere que si coinciden los nombres de creadores y editores, se omita utilizar la etiqueta de editor.

Ejemplo:

<dc:creator>Apellido, Nombre, 1890-1933</dc:creator>

Editor
Identificación de la entidad responsable de la distribución del contenido referenciado.

Ejemplo:

<dc:editor>Nombre de editora</dc:editor>

Colaboradores
Descripción de las entidades de una forma han colaborado con la elaboración del contenido referenciado.

Para este elemento se aplican las mismas indicaciones y recomendaciones para el elemento de creador.

Ejemplo:

<dc:contributor>Apellido, Nombre</dc:contributor>

Derechos
Información relevante sobre los derechos legales de propiedad intelectual registrados en el contenido referenciado. Usualmente se cita una declaración de gestión de derechos para el recurso.

La información de los derechos puede ser incluido como texto o referenciados por medio de una dirección URL o ambas.

Ejemplo:

<dc:rights>Contenido protegido por la ley Nro 15.215 sobre la Propiedad Intelectual</dc:rights>

1.3.3 Elementos relacionados principalmente con la instanciación del recurso.

Fecha
Indicador de fecha relacionada con un evento dentro del contenido referenciado, esta puede ser la fecha de publicación, fecha de edición, etc.

Para la publicación de la fecha se recomienda seguir el estándar ISO8601[7]: YYYY-MM-DD.

Ejemplo:

<dc:date>2014-10-19</dc:date>

Tipo
Categorización del contenido referenciado, se contemplan categorías, niveles de agregación, funciones, géneros dentro del tema del contenido intelectual referenciado.

Para especificar el tipo del que deriva el recurso, se recomienda utilizar los vocabularios propios del esquema de metadatos Dublin Core.

Es posible añadir varias etiquetas para el elemento Tipo en el caso de que el contenido referenciado derive de diferentes categorizaciones.

Ejemplo:

<dc:type>text.libro</dc:type>

Formato
Indicaciones de la forma en la que se encuentra el contenido referenciado, se especifican atributos en las que el contenido referenciado fue hecho como la descripción de dimensiones en caso de un documento físico o el tipo de media en caso de un recurso digital.

Al igual que en el elemento Tipo, se recomienda utilizar el vocabulario de metadatos Dublin Core o un esquema de codificación de elementos para referenciar el formato.

Ejemplo:

<dc:format>text/document</dc:format>

<dc:format>text/docx</dc:format>

[7] **ISO8601 (Representation of dates and time):** Norma perteneciente a los estándares ISO que especifica la fecha y hora usadas para el registro de eventos.

Identificador
Este elemento busca referenciar al contenido deseado de forma puntual y clara, dentro de un contexto establecido como por ejemplo un estándar de numeración o direcciones URL[8].

Ejemplo:

<dc:identifier>http://www.ejemplo.com</dc:identifier>

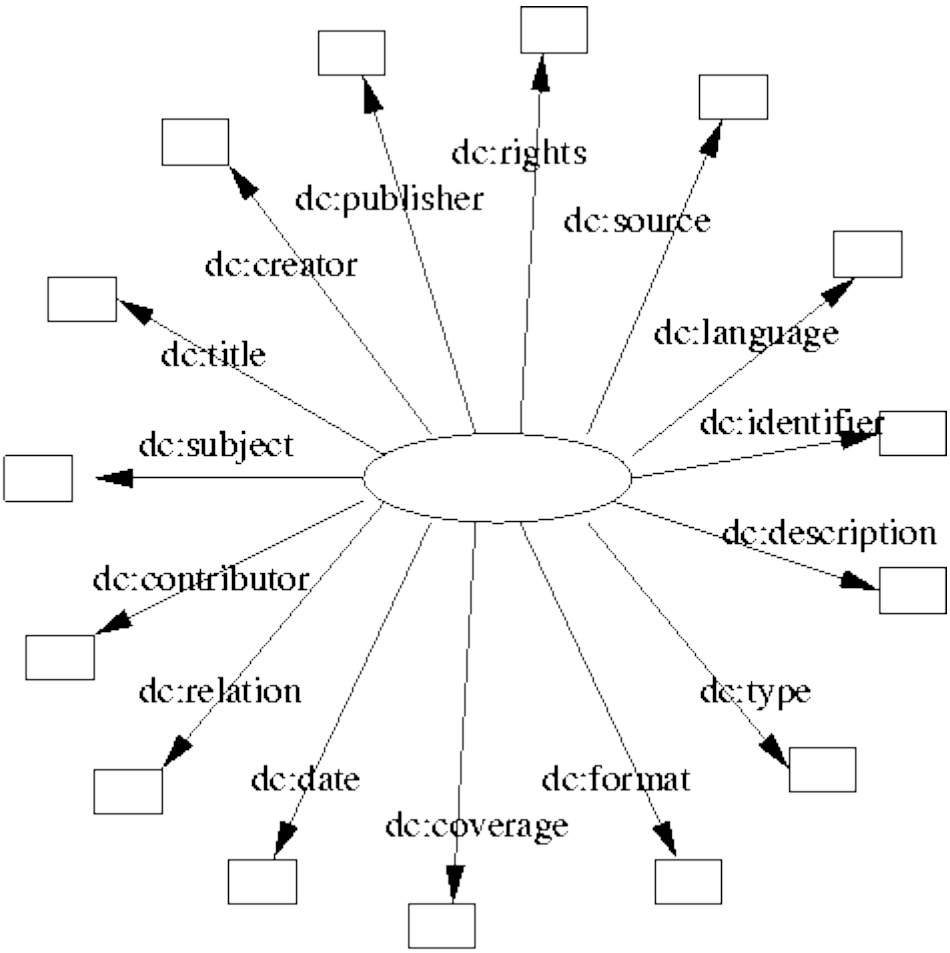


Ilustración 2: Elementos de Dublin Core (Dublin Core Metadata Initiative, http://dublincore.org/documents/dcq-rdf-xml/)

1.4 Vocabulario de metadatos

Los vocabularios en Dublin Core son los esquemas de metadatos sobre diferentes temas formados por un conjunto de elementos descriptivos. Estos se encargan de definir la información mediante una representación semántica con el objetivo de lograr una recuperación rápida de esta información.

Referente al vocabulario de metadatos que se requiere lograr, el esquema Dublin Core permite la incorporación de calificadores para los elementos ya vistos, estos permiten aclarar y especificar de forma precisa los metadatos utilizados en las referencias.

[88] **URL (uniform resource locator):** Indicador formado por un conjunto de caracteres que referencian a información ubicada en Internet.

Antes de optar por utilizar estos calificadores en los elementos utilizados en la referencia del contenido deseado, es necesario tomar en cuenta que la complejidad de los metadatos puede aumentar y complicar la referencia. Se recomienda utilizar calificadores especificados en los términos de metadatos de la DCMI.

Dentro de estos calificadores la DCMI se ha establecido varios tipos de vocabularios los cuales son descritos en la siguiente tabla:

Tabla No. 1 Vocabularios establecidos por la DCMI

Colection	Agregación de los recursos, se describe a un grupo
Dataset	Datos agrupados en una estructura definida como tablas, listas, etc.
Event	Una ocurrencia no persistente basada en el tiempo. Posee información descriptiva referente al propósito, localización, duración y agentes responsables de un evento
Image	Una representación visual sin contar con texto informativo
Interactive Resource	Recurso que requiere la interacción con un usuario para poder ser entendido
Moving Image	Un conjunto de representaciones visuales que dan el efecto de movimiento
Physical Object	Objeto inanimado y tangible de tres dimensiones
Service	Sistema que provee de una o más funciones.
Software	Referencia de un programa de computación ya sea en su código fuente o en su forma compilada
Sound	Recurso utilizado para ser escuchado, se puede incluir archivos de sonido
Still image	Representación visual estática
Text	Recurso compuesto por palabras. Hecho para ser leído

Fuente: Cristopher A. Pérez

Además de los tipos de vocabularios ya vistos, también especifica varios tipos de esquemas de codificación de vocabularios según diferentes esquemas además de la DCMI, los cuales son mencionados en la siguiente tabla:

Tabla No. 2 Esquemas de codificación establecidos por la DCMI

DCMI Type Vocabulary	Conjunto de clases especificadas por la DCMI para categorizar los recursos deseados.
DDC	Grupo de recursos conceptuales especificados por la Dewey Decimal Classification. Es usados para la clasificación de bibliotecas
IMT	Conjunto de recursos multimedia especificados por la Internet Assigned Numbers Authority. Se manejan protocolos de internet
LCC	Recursos conceptuales especificados por la Library of Congress Classification. Se usan clasificaciones destinadas a las bibliotecas de investigación y académicas
LCSH	Conceptos etiquetados por la Library of Congress Subject Headings. Se usa para el registro de bibliografías
MESH	Conceptos etiquetados por la Medical Subject Headings. Contempla un vocabulario terminológico destinado para publicaciones científicas

NLM	Conjunto de recursos conceptuales especificados por la National Library of Medicine Classification. Se cubren los campos relacionados con medicina y ciencias básicas preclínicas.
TGN	Información de lugares especificados por Getty Thesaurus of Geographic Names
UDC	Recursos conceptuales especificados por la Universal Decimal Classification. Provee de una disposición sistemática para la clasificación de diferentes ramas del conocimiento humano

Fuente: Cristopher A. Pérez

Dentro de los calificadores, se especifican diferentes clases con lo que se puede facilitar la identificación de los calificadores, las clases especificadas por la DCMI son las siguientes:

Tabla No. 3 Clases de calificadores establecidos por la DCMI

Agent	Recurso con la capacidad de actuar con el entorno, puede ser una organización, una persona, etc.
Agent Class	Se menciona a un grupo de agentes mencionados en el campo anterior.
Bibliographic Resource	Recurso bibliográfico en forma de documento. Puede ser una publicación, un libro, etc.
File Format	Especificación del formato de un recurso digital.
Frequency	Un rango en el que sucede un evento descrito en el recurso.
Jurisdiction	Especificación del rango jurídico del recurso.
License Document	Documento legal donde se especifican las restricciones del uso del recurso
Linguistic System	Descripción de un sistema lingüístico como la escritura, las señas o un lenguaje hablado.
Location	Descripción de una región espacial.
Location, Period, or Jurisdiction	Además de lo descrito en el campo anterior, se pueden añadir descripciones amplias de rangos o períodos de tiempo
Media Type	Formato de un recurso multimedia o medio físico
Media Type or Extent	Adición de una extensión al formato descrito en el campo anterior
Method of Accrual	Se menciona un método por el cual el recurso es añadido a una colección
Method of Instruction	Se menciona un método de enseñanza relacionado con el recurso
Period of Time	Descripción de un período de tiempo donde se menciona su fecha de inicio y fin
Physical Medium	Materiales que sirven como referencias físicas como libros, DVDs o documentos.
Physical Resource	Objeto material asociado al contenido referenciado
Policy	Un conjunto de acciones especificadas para establecer decisiones

Provenance Statement	Descripción de cambios en el recurso, se incluyen datos de creación, custodia, integridad, autenticidad e interpretación
Rights Statement	Declaración de los derecho de propiedad intelectual del recurso
Size or Duration	Se mencionan dimensiones relacionados al contenido del recurso
Standard	Conjunto de normas reguladoras utilizadas para comparar un contenido referente al contenido del recurso

Fuente: Cristopher A. Pérez

Como se puede apreciar en el cuadro anterior, existen clases que son iguales a otra pero permiten añadir información adicional. La decisión de utilizar una de estas clases depende del grado de precisión que se desee incluir en la referencia y según las clases que el usuario considera necesario incluir para el contenido a referenciar.

Cabe aclarar que una vez que se requiere referenciar la información deseada mediante el esquema de metadatos Dublin Core, se debe tomar en cuenta los siguientes aspectos:

La normativa de este contempla dos niveles: el simple, que posee los 15 elementos básicos de Dublin Core vistos anteriormente, y el cualificado, conformado por audiencia, origen y titular de derechos.

También se puede contar con un conjunto de elementos de matización los cuales están encargados de refinar la semántica de los elementos impuestos en el esquema para facilitar su recuperación de información a través de Internet.

Si se requiere revisar una lista extensa y completa de los calificadores y como estos pueden relacionarse con otros para los propósitos de referenciar contenido en internet se puede visitar la página oficial de Dublin Core Metadata Initiative.

Capítulo 2: Requerimientos para la aplicación web orientada al manejo de metadatos

2.1 Necesidades de la aplicación

2.1.1 Requerimientos básicos

Como se había planteado en el objetivo general de este trabajo, se busca facilitar el trabajo de personas encargadas de trabajos de investigación por medio de una herramienta de software.

Esta herramienta debe ser capaz de ayudar en la publicación de información, por lo cual se ha acordado en utilizar internet como medio de propagación de la información deseada.

Adicionalmente se necesita publicar esta información desde ubicaciones diferentes, con el fin de facilitar el trabajo de quienes quieran publicar información desde diferentes estaciones de trabajo.

Por este motivo la aplicación a construir será orientada a un ambiente web, por medio del cual un usuario puede acceder a través de un dispositivo con conexión a internet y que posea un navegador web actualizado. Para lograr lo anterior se tiene planeado construir el aplicativo por medio del lenguaje de programación Java[9] y complementando con el servidor de aplicaciones Jboss As[10] para poder orientar el aplicativo a la web.

Los medios por los que se pretende construir el aplicativo web son de código abierto y están en una versión estable lo cual garantiza su correcto funcionamiento en un ambiente de trabajo real. Estas son las razones por las que se escogieron estos medios de desarrollo de software para el proyecto a realizar.

Se necesita que esta herramienta sea fácil de usar y que posea una interface amigable para quienes deseen publicar información a través de ella.

Para poder cumplir este requerimiento se hará un diseño preliminar el cual va a ser descrito en la tercera parte de este capítulo.

Este diseño será visualizado por los usuarios que van a utilizar la herramienta para saber si lo consideran intuitivo y fácil de usar como interface de la aplicación.

Otro punto a considerar es el hecho de publicar información y diferenciar sobre el tipo de contenido que se busca referenciar a través de internet y el contenido que requiere un acceso privado. Para ello se debe implementar seguridades adecuadas a la aplicación web y conservar las referencias establecidas por medio del esquema de metadatos.

Para ello se pretende implementar el inicio de sesión o autenticación de usuario como requisito fundamental para poder manejar el contenido que se publica en el aplicativo web. Además se debe identificar la información más delicada e importante para poder

[9] **Java:** Lenguaje de programación de distribución libre y con paradigma orientado a objetos diseñado por Sun Microsistem y actualmente soportado por Oracle Coorporation.
[10] **Jboss AS:** servidor de aplicaciones de código abierto implementado en el lenguaje de programación Java.

almacenarla por medio de algortimos de codificación con el propósito de codificar la información antes de almacenarla en la correspondiente base de datos.

2.1.2 Requerimientos específicos

Para poder conocer los requerimientos específicos del portal web se han tenido conversaciones los delegados de la Comunidad de Dominicos de Quito para conocer a detalle el tipo de información que se desea referenciar y en base a estos se ha decidido elaborar el diseño preliminar del aplicativo y elaborar la estructura del mismo conformada por módulos.

Adicionalmente, se pretende revisar algunos de los libros y materiales que se busca referenciar para poder determinar cuáles serían los elementos necesarios del sistema de metadatos Dublin Core para que la información subida a la página sea referenciada de la mejor forma.

Por medio de estas conversaciones se ha concluido que es necesario identificar dos partes importantes del aplicativo web, donde una parte es de acceso público y otra parte es de acceso restringido, para lo cual se ha especificado las características que necesita cada parte del aplicativo web en particular:

2.1.3 Acceso Público

Dentro de esta sección se encuentra información relacionada con la biblioteca, sobre su historia, reseñas, imágenes y cualquier información adicional que el convento considere pertinente. Esta información será publicada con el fin de atraer la atención del público interesado en el tema.

Una vez incluida la información general sobre la biblioteca, también es necesario el acceso a una sección donde se listen los catálogos de libros identificados en la correspondiente biblioteca, para cumplir el objetivo de difundir la información relevante sobre la información ubicada en la biblioteca por medio del sistema de metadatos.

2.1.4 Acceso restringido

Al ingresar los datos de usuario requeridos por el aplicativo web para ingresar a esta sección, se debe proceder a administrar la información correspondiente a la información de la página y a los catálogos de libros que se busca referenciar ya sea revisar, ingresar, eliminar o actualizar esta información.

La biblioteca en la actualidad cuenta con un conjunto limitado de catálogos de varios libros que han logrado identificar como es el caso de los siguientes ejemplos:

Ejemplo1: Catálogo de la Biblioteca Fray Ignacio de Quezada

```
* FIQ 40.1.3 C-305

* JV

Toledo, Francisco de, S.I (1532-1593)

      Institución de sacerdotes y Suma de casos de conciencia /

compuesto  por el.... Cardenal Francisco Toledo, religioso de la

compañía de Jesus; con las adiciones y anotaciones de Andrés

Victorelo...; y añadido el tratado del Sacramento de la Orden;
```

```
traducida del latin en castellano por el Doctor Nenriquez
de Salas y nuevamente corregida y enmendadapor el Padre Juan de
Salas de la Compañía de Jesús en esta tercera impresión.... --
En Valladolid : por Francisco Fernandez de Córdova : a costa de
Antonio Vazquez...., 1619
      [6], 303, [20] h. ; Fol. (31cm)
      Colofón
      Sign.: [@]>6<, A-Z>8<, 2A-2P>8<, *-2*>8<, 3*>3<
      Port. a dos tintas con grab.  xil. de la compañía de Jesús
      Texto a dos col.

Sacerdotes-Formación
Pecados Mortales
Teología Pastoral
```

Fuente: Convento de Santo Domingo de Quito

Ejemplo 2: Catálogo de la Biblioteca Fray Ignacio de Quezada

```
* FIQ 39.2.5 L-961
* PC
Luis de Granada, O.P. (1504-1588)
      Guía de pecadores: en la que se contiene una larga, y
copiosa exhortación a la virtud y guarda de los mandamientos.
tomo primero / compuesto por..Fr. Luis de Granada del Sagrado
Orden de Predicadores... -- [Madrid]: en el convento de Jesús
María de Valverde : en la oficina de Manuel Fernandez.. en
Madrid...,1730
      [56], 578, [34] p., [1] h. de grab. ; 4`
      Sign.: [calderón]-3[calderón]\p4\s, *-3*\p4\s, []\p4\s,
A-4G\p4\s, 4H\p2\s
      Texto a dos col.
      Texto con apostillas marginales
      H. de grab. calc.: "J\ps\s á Palomo. del. et sculp."
```

```
      Grab. calc. en cabecera de p.1

      Letras capitales ornadas




Espiritualidad

Fernández, Manuel, imp.

Convento de Jesús María de Valverde (Madrid), imp

España-Madrid
```

Fuente: Convento de Santo Domingo de Quito

Al revisar los catálogos que la biblioteca posee con los encargados de la misma, se ha delimitado la necesidad básica de catalogar los libros en el aplicativo web utilizando los siguientes elementos:

- Nombre de biblioteca
- Estado
- Iniciales
- Autor
- Título
- Tipo
- Editorial
- Páginas
- Tamaño
- Temas
- Imagen

Como requisito adicional se ha planteado la opción de poder subir archivo multimedia, concretamente imágenes y documentos digitales relacionados con el catálogo respectivo. Para ello se añadirá una opción adicional de subir varias imágenes en diferentes formatos y documentos en formato PDF[11].

Una vez descrito los elementos necesarios para catalogar los libros de la biblioteca, se ha decidido implementar los siguientes metadatos correspondientes al sistema Dublin Core:

- dc:title
- dc:subject
- dc:type
- dc:description
- dc:source

[11] **PDF (Portable Document Format):** Formato de almacenamiento para documentos digitales de tipo compuesto y de plataforma independiente.

- dc:creator
- dc:editor
- dc:format (Estado)
- dc:coverage (Iniciales) / biblioteca

2.2 Descripción de módulos necesarios

Al establecer los requerimientos de la aplicación web y al relacionarlos con el sistema de metadatos que se busca implementar, se puede realizar la construcción inicial del aplicativo mediante la descripción de sus componentes y sus funciones.

Como punto inicial, se realizará un modelo de la base de datos del aplicativo, ya que en base a esta, es posible proceder con la mención y breve descripción de los módulos necesarios para lograr el funcionamiento adecuado del aplicativo web.

Como último punto se ha acordado la elaboración preliminar y aprobación de los diseños del aplicativo web, los cuales muestran diferentes secciones de la interface de la página web, contemplando las secciones con acceso restringido como las secciones de acceso público.

2.2.1 Modelo de base de datos

Para esto se ha realizado como primer paso el diseño del modelo relacional de la base de datos necesaria para el correcto funcionamiento del aplicativo web acorde al sistema de metadatos Dublin core para poder catalogar la información requerida.

En este modelo se puede visualizar las entidades, sus correspondientes relaciones y los campos requeridos para registrar los datos deseados y las acciones de los usuarios.

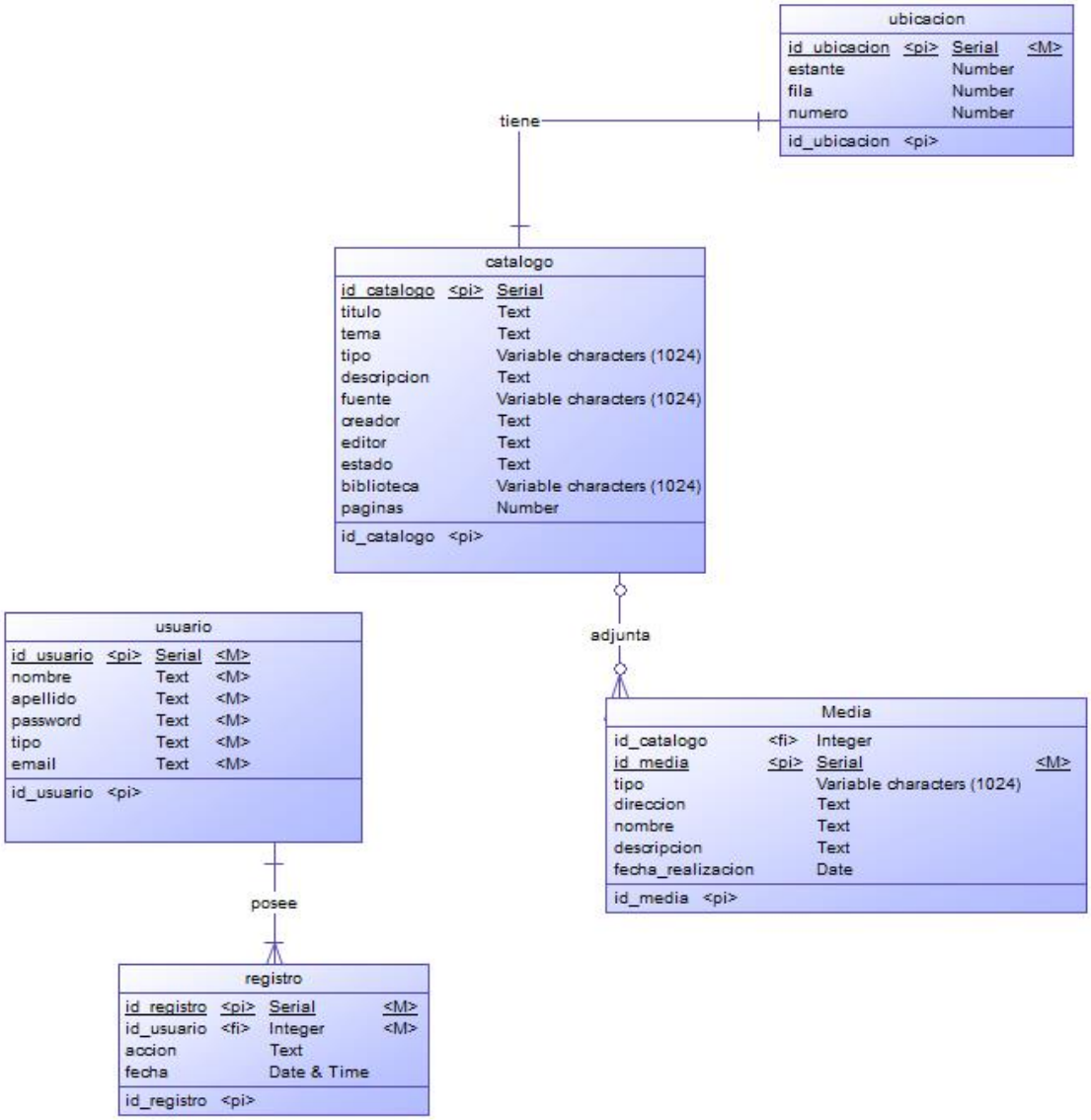


Ilustración 3: Modelo relacional de la base de datos del aplicativo web. (Cristopher A. Pérez)

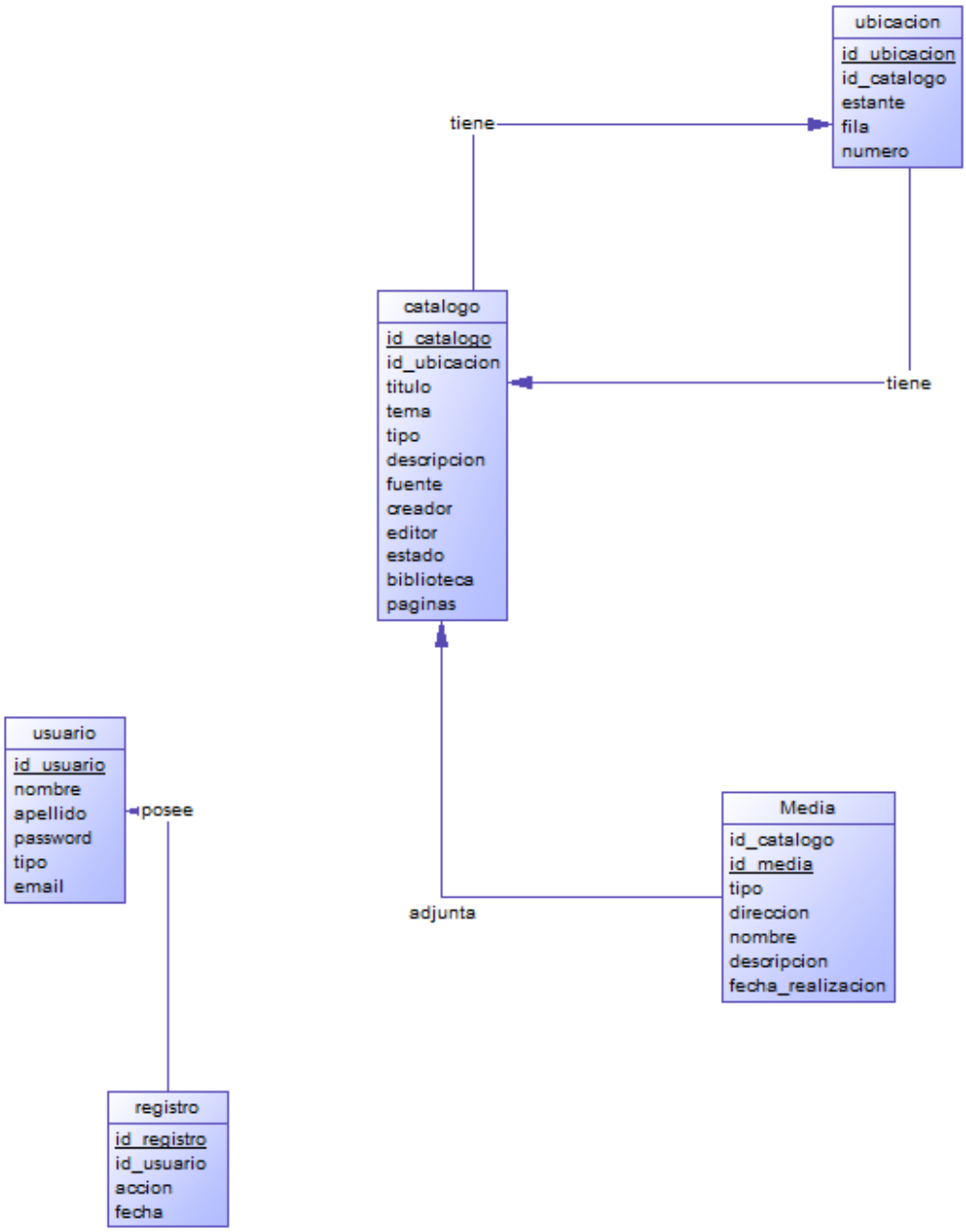


Ilustración 4: Modelo físico de la base de datos del aplicativo web. (Cristopher A. Pérez)

Dentro de esta base de datos podemos visualizar la entidad “Usuario” la cual se encarga de almacenar los datos de las personas a cargo de la administración de la información relevante al aplicativo web. Esta entidad puede contener varios “registros” donde se guarda un historial de las acciones de administración de los usuarios.

Para la entidad “Catálogo” se han tomado en cuenta los campos necesarios para el registro de información acorde a lo solicitado, además de asociarla a una entidad “Media” en el caso de que se requiera subir contenido multimedia relacionado con el catálogo, y agregando la entidad “Ubicación” para facilitar la ubicación física del correspondiente libro de acuerdo a como se maneja en la biblioteca.

2.2.2 Distribución de módulos

La distribución de los archivos que conforman el aplicativo será distribuida en diferentes capas en las cuales se concentrarán los módulos de acuerdo a la funcionalidad de estos conjuntos. Para ello se ha contemplado la siguiente distribución para el aplicativo web:

- **DataManager:** Se encarga de comunicar las funciones de inserción, consulta, actualización y eliminación realizadas desde la interfaz gráfica.
- **Controller:** Maneja las operaciones de inserción, consulta, actualización y eliminación de los datos de las entidades hacia la base de datos.
- **Service:** Permite llamar a los métodos que contienen las operaciones necesarias para el funcionamiento lógico de la aplicación.
- **Dao:** Realiza las consultas en lenguaje SQL[12] para la identificación de datos por medio de parámetros ingresados por el usuario.
- **Model:** Contiene la configuración de la base de datos de las entidades incluyendo sus respectivas entidades.

Dentro de esta distribución se manejará la información de las entidades ya identificadas en el diseño de la base de datos.

Es necesario destacar que además de la distribución de módulos ya planificada, se requiere la distribución de la capa de visualización, que viene a ser la interacción del usuario con el aplicativo web. Estas son organizadas en archivos xhtml[13] donde se especifica el contenido de las interfaces gráficas por medio de expresiones XML para poder aplicar el sistema de metadatos requerido.

Se utilizarán hojas de estilo CSS[14] para poder almacenar la estructura visual de los componentes de la página web los cuales incluyen estilos de letra, tamaño de elementos, distribución de colores, etc.

Esto se realiza con el propósito de lograr un control centralizado del estilo del sitio web y mejorar la accesibilidad del diseño de la página.

[12] **SQL (Structured Query Language):** Lenguaje declarativo usado para el acceso y manejo de base de datos relacionales.

[13] **XHTML (Extensible Hypertext Markup Language):** Lenguaje de marcado presente en páginas web expresado como sentencias XML válidas.

[14] **CSS (Cascading style sheed):** Hoja de estilo en cascada utilizada para la definición y presentación de un documento estructurado en lenguaje HTML o XML.

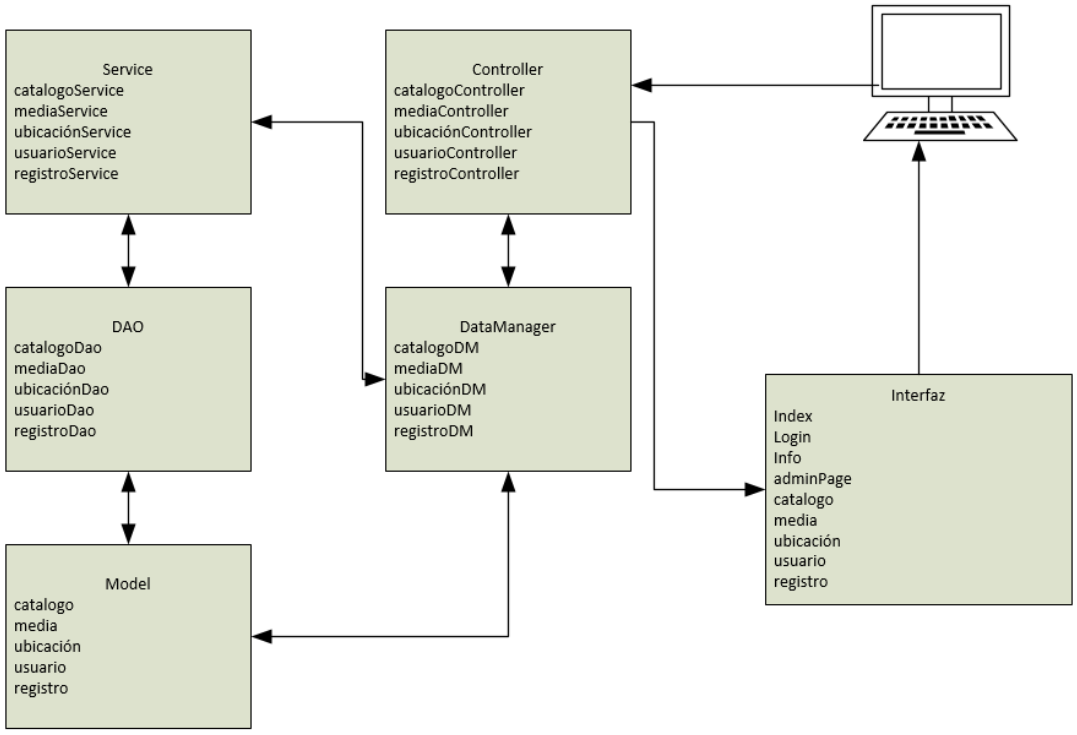


Ilustración 5: Modelo vista controlador del aplicativo web. (Cristopher A. Pérez)

2.3 Diseño de la aplicación

En esta sección se muestran los diseños preliminares de diferentes vistas que componen a esta aplicación incluyendo las partes accesibles por cualquier usuario y las partes requeridas para administrar la información y catálogos de la misma.

Estos diseños tienen el propósito de dar una impresión a los futuros usuarios de la página web del aspecto que esta tendrá para que esté acorde a sus necesidades y lo consideren una plataforma fácil de usar.

2.3.1 Diseño de secciones públicas de la página

En estas secciones hay información relevante a la biblioteca en sí, la cual se especificará en la página principal como en la sección de información adicional, y la información de los catálogos registrados en la página. En la página principal se ubicará la información general de la biblioteca en sí, así como los vínculos a las diferentes secciones de la página.



Ilustración 6: Diseño preliminar de la página principal. (Cristopher A. Pérez)

En la sección de información adicional se lista información específica sobre la biblioteca, su historia y otros datos relevantes para el interés de los visitantes.



Ilustración 7: Diseño preliminar de la sección de información adicional. (Cristopher A. Pérez)

Para la sección de catálogos se dispone de una tabla en la que se listan los correspondientes catálogos, donde se pueden realizar búsqueda por medio de diferentes parámetros y al seleccionar un elemento de la tabla, se puede proceder a una sección donde se despliega toda la información relevante al catálogo seleccionado.



Ilustración 8: Diseño preliminar de la tabla de catálogos disponibles (Cristopher A. Pérez)

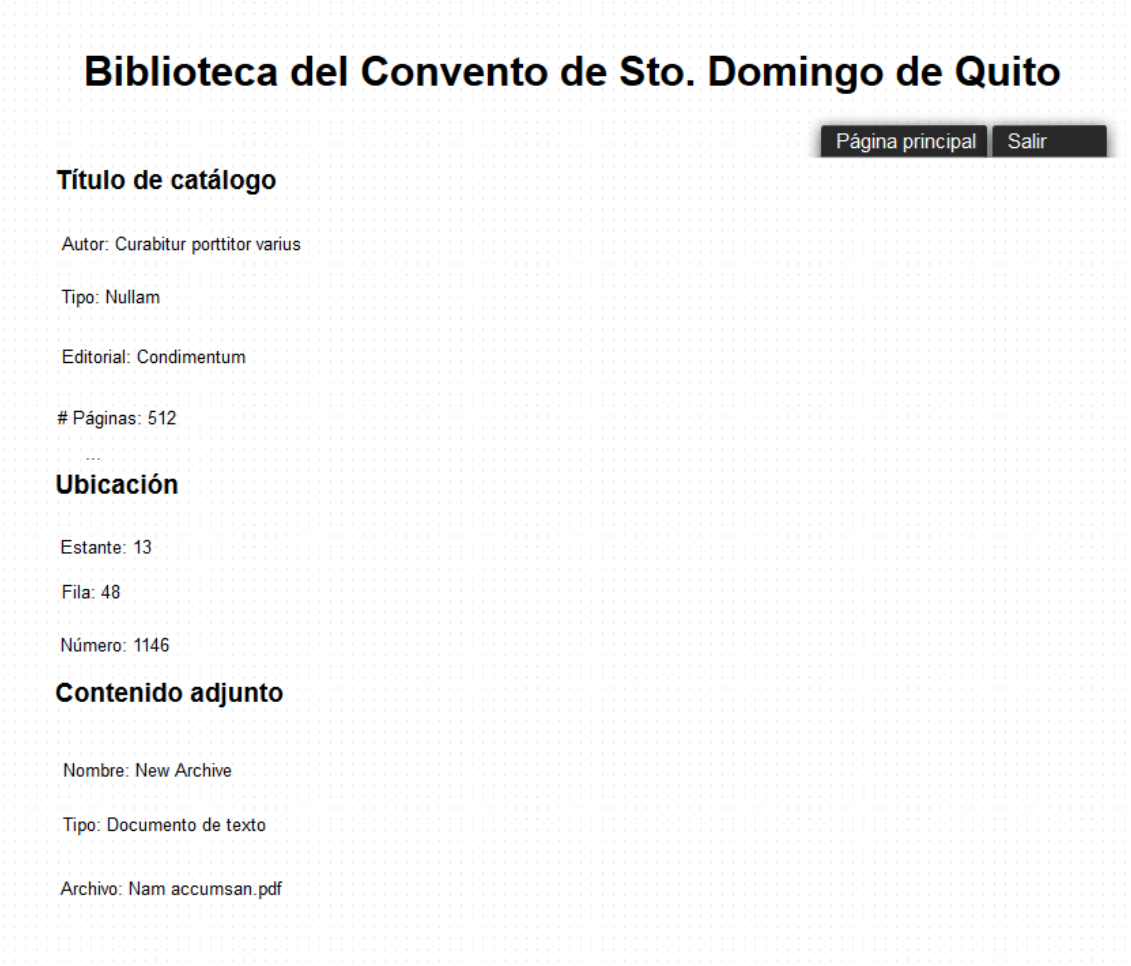


Ilustración 9: Diseño preliminar del despliegue de información de catálogos (Cristopher A. Pérez)

2.3.2 Diseño de secciones privadas de la página

Como se indicó anteriormente, para poder acceder a esta sección es necesario ingresar los datos de nombre de usuario y contraseña los cuales se solicitan al intentar ingresar a cualquier dirección de la sección privada de la página.

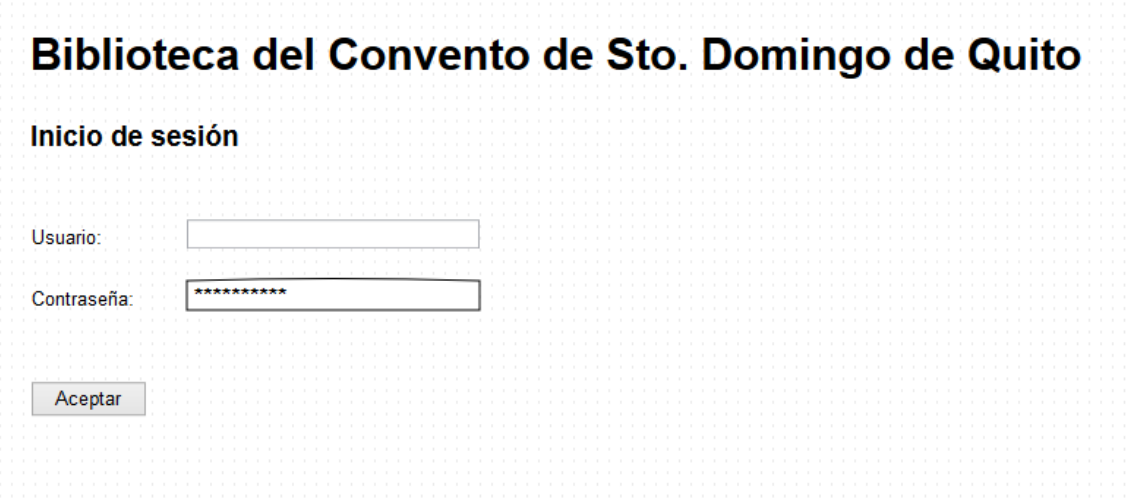


Ilustración 10: Diseño preliminar de la sección de inicio de sesión. (Cristopher A. Pérez)

Una vez ingresado correctamente los datos de nombre de usuario y contraseña, es posible acceder a la sección de administración de contenido, donde se puede visualizar y administrar el contenido de las entidades de la aplicación.



Ilustración 11: Diseño preliminar de la sección de administración de contenido. (Cristopher A. Pérez)

Para administrar el contenido de las entidades del aplicativo, se dispone de tablas en donde se listan los contenidos que al seleccionarlos se puede editar o eliminar la información correspondiente, además de agregar nuevo contenido.



Ilustración 12: Diseño preliminar de la tabla de contenido multimedia. (Cristopher A. Pérez)

Al editar o ingresar nuevo contenido, se despliega una nueva sección donde se despliega un formulario para el manejo de los datos de acuerdo a la correspondiente entidad.



Ilustración 13: Diseño preliminar del formulario de contenido multimedia. (Cristopher A. Pérez)

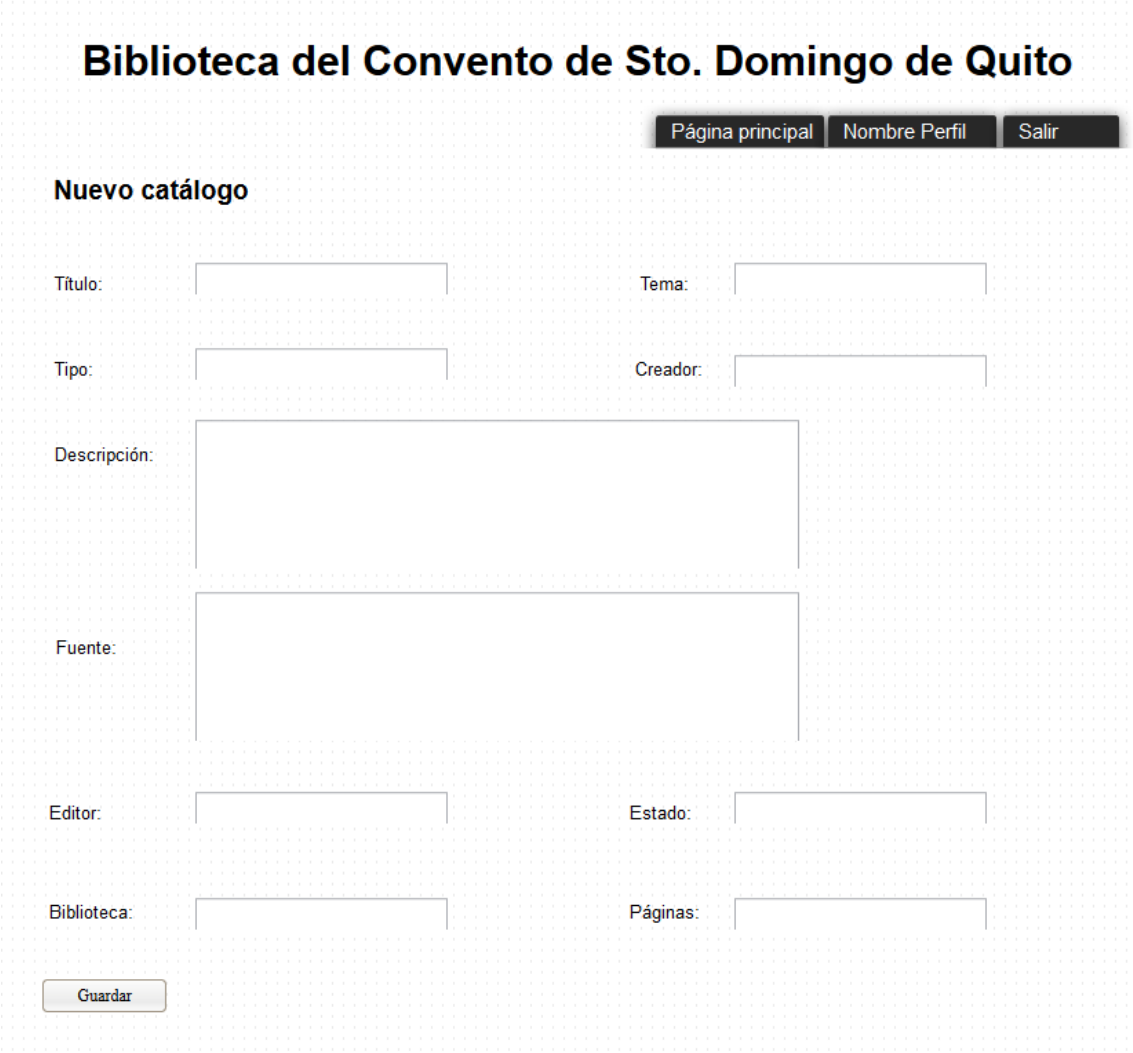


Ilustración 14: Diseño preliminar del formulario de catálogos. (Cristopher A. Pérez)

Capítulo 3: Seguridades necesarias para la aplicación en ambiente web

3.1 Acceso al portal web

Como parte de los requisitos mencionados en el capítulo anterior, es necesario establecer seguridades para que solo los usuarios autorizados del portal web puedan manejar la información que se requiere subir y que los usuarios comunes puedan tener acceso al contenido referenciado por medio del sistema de metadatos Dublin Core.

Para ello la propuesta a implementar es la autentificación de usuarios por medio del registro de datos específicos para las personas destinadas al manejo del portal web. Los datos personales que el usuario debe registrar en el portal web se listan a continuación:

Tabla 4: Datos personales de usuario

Nombre	Apellido	Correo Electrónico	Nombre de usuario	Contraseña

Fuente: Cristopher A. Pérez

Adicional a esto el usuario debe tener uno de los tres tipos de roles con el objetivo de diferenciar entre las tareas que pueden realizar los usuarios autorizados para el manejo de la página:

Tabla 5: Roles de usuario

Administrador	Editor	Visitante

Fuente: Cristopher A. Pérez

Para poder acceder al portal web y trabajar con los datos almacenados en la página es necesario ingresar a la misma identificándose por medio del nombre de usuario y contraseña.

El manejo de datos destinados al portal web dependerá del rol asignado al usuario registrado.

3.2 Restricciones para el uso del portal web

3.2.1 Distribución del aplicativo web

Es necesario diferenciar las secciones de la página web que serán abiertas para cualquier usuario y las secciones que permiten el manejo de la información.

Dentro de las páginas abiertas al público se encuentran las siguientes:

- **Home:** Página principal que contiene información básica del sitio web y la biblioteca en sí. Además esta sección posee enlaces hacia las otras secciones de la página.
- **Info:** Contiene información adicional relevante a la Biblioteca del Convento Santo Domingo.
- **Contactos:** Información de contacto sobre la biblioteca.
- **Catálogo:** Listado de catálogos de libros subidos a la página.

Las secciones de la página que requieren el inicio de sesión de usuarios registrados son las siguientes:

- **adminPage:** Sección donde se provee acceso a las partes de administración de contenido de la página.
- **Listas:** Secciones que despliegan el contenido de las entidades correspondientes a catálogos, usuarios, ubicaciones, registros y media.
- **Formularios:** Son secciones que permiten el manejo de información correspondiente a las entidades de catálogos, usuarios, ubicaciones, registros y media.

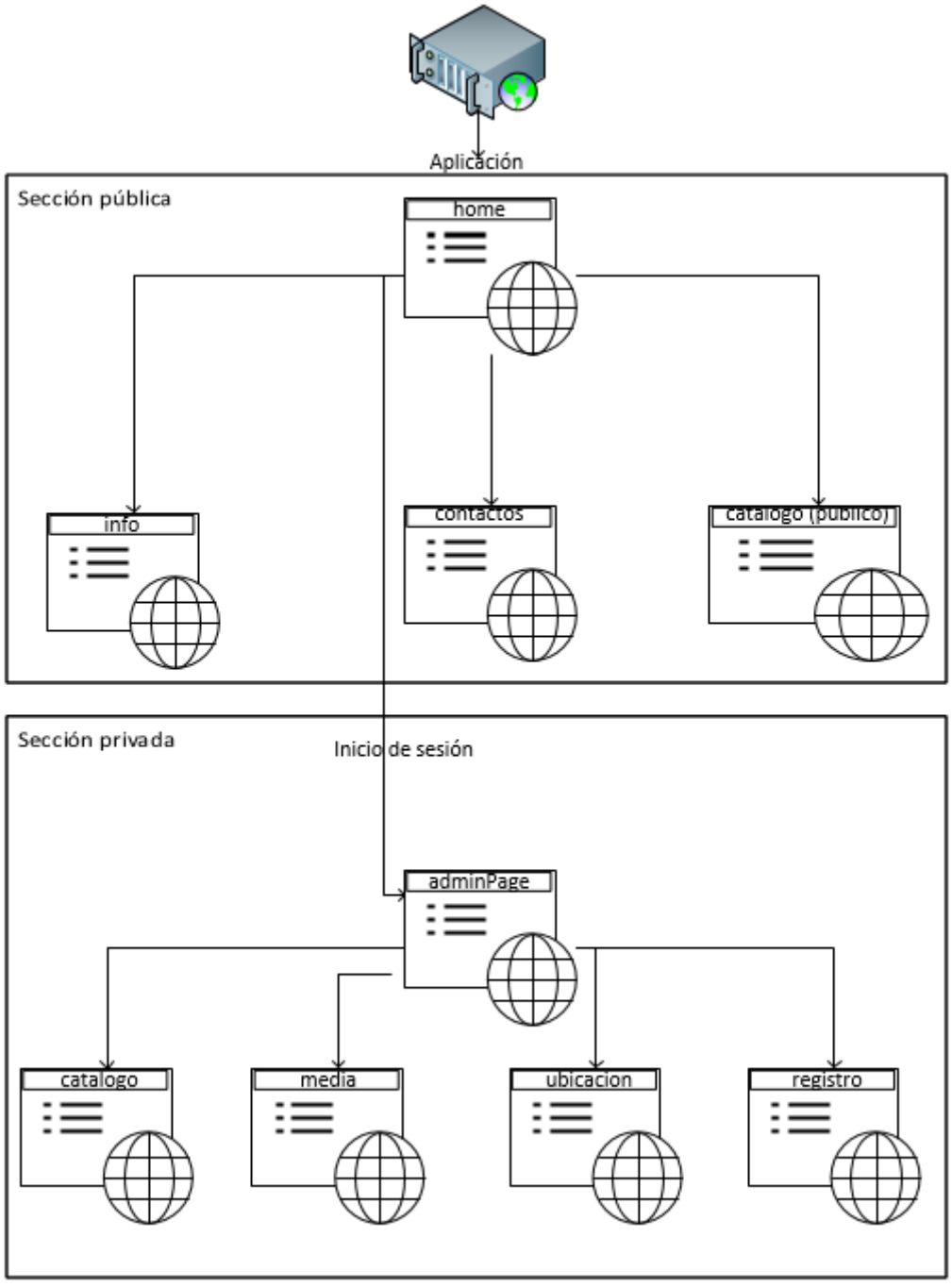


Ilustración 15: Distribución del aplicativo web (Cristopher Pérez)

3.2.2 Roles de usuarios

Para la administración del contenido del portal web se requiere identificar a los usuarios designados para esta tarea los cuales se describen a continuación:

3.2.2.1 Administrador

Este rol tendrá los privilegios de ingreso, edición y eliminación de datos dentro del portal web. Además de poder ingresar nuevos usuarios que trabajen en la página y poder modificar los datos de los usuarios ya registrados, de manera que datos como nombres, correos electrónicos, contraseñas e incluso los roles de usuarios pueden ser alterados.

3.2.2.2 Editor

El perfil de edición permite el ingreso y alteración de la información pero no se permite el registro de nuevos usuarios ni la alteración de sus datos sin contar con el cambio de rol. Este tipo de usuario tendrá acceso a las secciones de tipo formulario para acceder y manejar los correspondientes datos.

3.2.2.3 Visitante

Finalmente el rol de visita permite al usuario revisar el contenido ingresado en la página y le permite el acceso a las secciones de la página no autorizadas para usuarios no registrados. Este rol no puede ingresar ni alterar información dentro de la página y no puede manejar los datos de los usuarios registrados en la misma.

Es necesario mencionar que ningún tipo de usuario, incluyendo al tipo administrador, puede cambiar la información sobre los registros de acciones que han realizado los usuarios en la misma página

3.3 Implementación de protocolos de seguridad para el portal web.

Dentro del archivo de configuración standalone[15] se especifican las siguientes etiquetas que nos permiten la integración de la aplicación web con los elementos necesarios para cumplir con los requerimientos de seguridad de la misma.

3.3.1 Security

Para esta etiqueta debemos incluir el nombre de usuario y contraseña con el que la aplicación debe conectarse con el motor de base de datos, esto se especifica por medio de las etiquetas internas “user-name” y “password”.

Esquema 1: Especificación de contenido de la etiqueta security en el archivo stanalone.xml

```
<security>
        <user-name>postgres</user-name>
        <password>********</password>
</security>
```

[15] **Standalone:** archivo de configuración de jboss donde se listan los archivos de configuración, información de despliegue de contenido, configuraciones de perfil, información de archivos temporales e información a retener cuando el servidor es reiniciado.

Fuente: Cristopher A. Pérez

3.3.2 Security-domain

Esta etiqueta es vital para la identificación de usuarios en el momento en que estos deseen iniciar sesión en la página para trabajar con su información. Para ello contaremos con la etiqueta "authentication" en donde se especificará el módulo de sesión, así como la consulta hacia la base de datos en lenguaje SQL para verificar los datos ingresados en la página con los guardados en la base de datos.

Adicionalmente, se especifica el módulo encargado de la codificación de información correspondiente a la contraseña de los usuarios guardada en la base de datos.

Esquema 2: Especificación de contenido de la etiqueta security-domain en el archivo stanalone.xml

```
<security-domain name="bibliotecaSecurity">
      <authentication>
            <login-module code="Database" flag="required">
                  <module-option name="dsJndiName"
value="java:jboss/bibliotecaDS"/>
                  <module-option name="principalsQuery" value="select
password from usuario where lower(nombre)=lower(?)"/>
                  <module-option name="rolesQuery" value="select tipo,
'Roles' from usuario where lower(nombre)=lower(?)"/>
                   <module-option name="hashAlgorithm" value="SHA-256"/>
                   <module-option name="hashEncoding" value="base64"/>
             </login-module>
     </authentication>
 </security-domain>
```

Fuente: Cristopher A. Pérez

3.3.2.1 Module option

En esta etiqueta se especificarán los medios utilizados para la codificación de datos que no deben ser reconocidos por nadie aparte del creador de dichos datos como son las contraseñas de usuarios o ubicaciones de los archivos multimedia.

Se utiliza la función de codificación SHA[16]-256 en la etiqueta "module-option name="hashAlgorithm" para especificar la codificación de valores clave dentro del aplicativo web. Esta perteneciente al conjunto de funciones hash pertenecientes a SHA-2 diseñada por la Agencia de Seguridad Nacional y publicada por el Instituto Nacional de Estándares y tecnologías.

Esta función trabaja únicamente de un solo lado o camino para que no se pueda descifrar el código y se encarga de transformar un conjunto dado de datos en un valor

[16] **SHA (Secure hash algorithm):** Algoritmo de hash seguro que contiene funciones para la codificación de un dato recibido.

de longitud fija del tamaño de 256 bits o 32 bytes conocido como hash. Por medio de este valor se verifica la integridad de la copia del dato procesado como hash en comparación con el dato original.

Las funciones contempladas en SHA-2 corrigen los problemas de colisión de datos y fallos de seguridad detectados en la versión anterior de funciones conocidas como SHA-1. Estos fallos son producto de una falta de resistencia o vulnerabilidades en funciones matemáticas utilizadas para calcular los valores de longitud fija.

Complementado lo anterior, se utiliza el sistema de numeración posicional Base64 el cual contiene esquemas de codificación para la representación de datos binarios en una cadena de texto en código ASCII[17] de máximo 64 caracteres.

Por medio de estas funciones se pretende codificar los datos requeridos y guardarlos como una cadena de caracteres en la base de datos como se muestra a continuación:

Ejemplo 3: Codificación de datos utilizando SHA-256 y base64

```
TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCBieSB0aGlzI
HNpbmd1bGFyIHBhc3Npb24gZnJvbSBvdGhlciBhbmltYWxzLCB3aGljaCBpcyBhIGx1c3Qgb2YgdG
hlIG1pbmQsIHRoYXQgYnkgYSBwZXJzZXZlcmFuY2Ugb2YgZGVsaWdodCBpbiB0aGUgY29udGludWV
kIGFuZCBpbmRlZmF0aWdhYmxlIGdlbmVyYXRpb24gb2Yga25vd2xlZGdlLCBleGNlZWRzIHRoZSBz
aG9ydCB2ZWhlbWVuY2Ugb2YgYW55IGNhcm5hbCBwbGVhc3VyZS4=
```

Fuente: http://en.wikipedia.org/wiki/Base64

En el archivo de configuración web.xml[18] del aplicativo web se requiere la especificación de nuevas etiquetas donde se controlarán las seguridades internas de la aplicación web.

3.3.3 Security-constraint

Dentro de esta etiqueta se encuentran las especificaciones de acceso para las diferentes partes del aplicativo web. Para el presente caso se especifican las direcciones que necesitan un acceso libre hacia el público. Esto se especifica dentro de la etiqueta "web-resource-colection".

Se utilizará la etiqueta "web-resource-name" para identificar el tipo de acceso a las direcciones de la página web, para el presente caso se utilizará el nombre "Público" para identificar las direcciones de libre acceso.

A continuación se procede a especificar las direcciones URL de las partes del aplicativo web que requieren un acceso libre sin necesidad de que el usuario se registre e inicie sesión en la página. Las direcciones URL se incluyen dentro de las etiquetas "url-pattern".

Esquema 3: Especificación de contenido de la etiqueta security-constraint en el archivo web.xml

```
<security-constraint>
            <web-resource-collection>
                  <web-resource-name>Publico</web-resource-name>
                  <url-pattern>/index.html</url-pattern>
                  <url-pattern>/home.jsf</url-pattern>
                  <url-pattern>/pages/contactos.jsf</url-pattern>
```

[17] **ASCII (American Standard Code for Information Interchange):** Código de caracteres estándar estadounidense basado en el alfabeto latino y utilizado para el intercambio de información.

[18] **Web.xml:** archivo en formato xml que describe sentencias referentes al comportamiento específico de un aplicativo web.

```
                    <url-pattern>/pages/info.jsf</url-pattern>
                    <url-pattern>/pages/catalogos.jsf</url-pattern>
              </web-resource-collection>
</security-constraint>
```

Fuente: Cristopher A. Pérez

Una vez especificado la configuración necesaria para las direcciones de tipo público para la página web, es necesario realizar el mismo procedimiento para las direcciones de la página exclusivas para los usuarios con roles de administración, editor y visitante.

Para ello agregamos la etiqueta “auth-constraint” donde se especifica el rol de usuario asignado para las correspondientes restricciones de seguridad.

Esquema 4: Especificación de contenido de la etiqueta security-contraint en el archivo web.xml

```
<security-constraint>
              <web-resource-collection>
                    <web-resource-name>Admin</web-resource-name>
                    <url-pattern>/pages/admin/*</url-pattern>
              </web-resource-collection>
              <auth-constraint>
                    <role-name>ADMINISTRADOR</role-name>
              </auth-constraint>
      </security-constraint>
      <security-constraint>
              <web-resource-collection>
                    <web-resource-name>Editor</web-resource-name>
                    <url-pattern>/pages/admin*</url-pattern>
              </web-resource-collection>
              <auth-constraint>
                    <role-name>EDITOR</role-name>
                    <role-name>ADMINISTRADOR</role-name>
              </auth-constraint>
      </security-constraint>
      <security-constraint>
              <web-resource-collection>
                    <web-resource-name>Visitante</web-resource-name>
                    <url-pattern>/admin/adminPage</url-pattern>
              </web-resource-collection>
              <auth-constraint>
                    <role-name>VISITANTE</role-name>
              </auth-constraint>
</security-constraint>
```

Fuente: Cristopher A. Pérez

3.3.4 Login-config

En la presente etiqueta se debe especificar la dirección de la página web correspondiente al ingreso de datos para el inicio de sesión del usuario. Esto se lo realiza con el propósito de redirigir a esta dirección a los usuarios que no han iniciado sesión e intentan acceder a direcciones restringidas para el público abierto.

Es necesario identificar el método de autenticidad para la configuración de inicio de sesión por medio de la etiqueta “auth-method”, la cual es de tipo Form debido a que estamos utilizando un formulario simple para obtener los datos de ingreso del usuario, las configuraciones se listan dentro de la etiqueta “form-login-conf”.

Dentro de la etiqueta “form-login-page” se especifica la dirección en la que se encuentra ubicado la página de inicio de sesión de la página.

Como dato adicional se puede especificar una dirección para redirigir a una página adicional en caso de fallos en el proceso de inicio de sesión, eso se lo realiza mediante la etiqueta “form-error-page”.

Esquema 5: Especificación de contenido de la etiqueta login-config en el archivo web.xml

```
<login-config>
            <auth-method>FORM</auth-method>
            <form-login-config>
                  <form-login-page>/login.jsf</form-login-page>
                  <form-error-page>/error.html</form-error-page>
            </form-login-config>
</login-config>
```

Fuente: Cristopher A. Pérez

3.3.5 Security Role

Dentro de esta etiqueta se listan los roles de usuarios previamente mencionados para el manejo adecuado del aplicativo web. Para ello se introducen los roles respectivos dentro de las etiquetas “role-name”.

Esquema 6: Especificación de contenido de la etiqueta security-role en el archivo web.xml

```
<security-role>
            <role-name>ADMINISTRADOR</role-name>
      </security-role>
      <security-role>
            <role-name>EDITOR</role-name>
      </security-role>
      <security-role>
            <role-name>VISITANTE</role-name>
</security-role>
```

Fuente: Cristopher A. Pérez

Capítulo 4: Aplicación de características requeridas y desarrollo de la aplicación web

En el presente capítulo se procederá a explicar el desarrollo y funcionamiento del aplicativo web basándose en los diseños preliminares y los requerimientos funcionales explicados en capítulos anteriores. Para cumplir esta tarea se ha escogido trabajar con la metodología de programación extrema o también conocida como "Extreme Programming".

Se ha escogido esta metodología de programación debido a que permite "agilizar el proceso de desarrollo revisando el código a la par de la programación, aplicar diseños simples a la programación, simplicidad y claridad en el código fuente, acoger un desarrollo incremental y realizar pruebas en conjunto con las acciones correctivas al detectar errores en las pruebas de funcionamiento" (2).

4.1 Aplicación de diseño establecido

De acuerdo a los diseños preliminares para las partes públicas y privadas del aplicativo web, se ha optado por utilizar las librerías de "Primefaces"19 para el despliegue de información y navegación de las partes públicas, además de anexar la interfaz gráfica del aplicativo web con su esquema funcional escrito en java servlet faces.

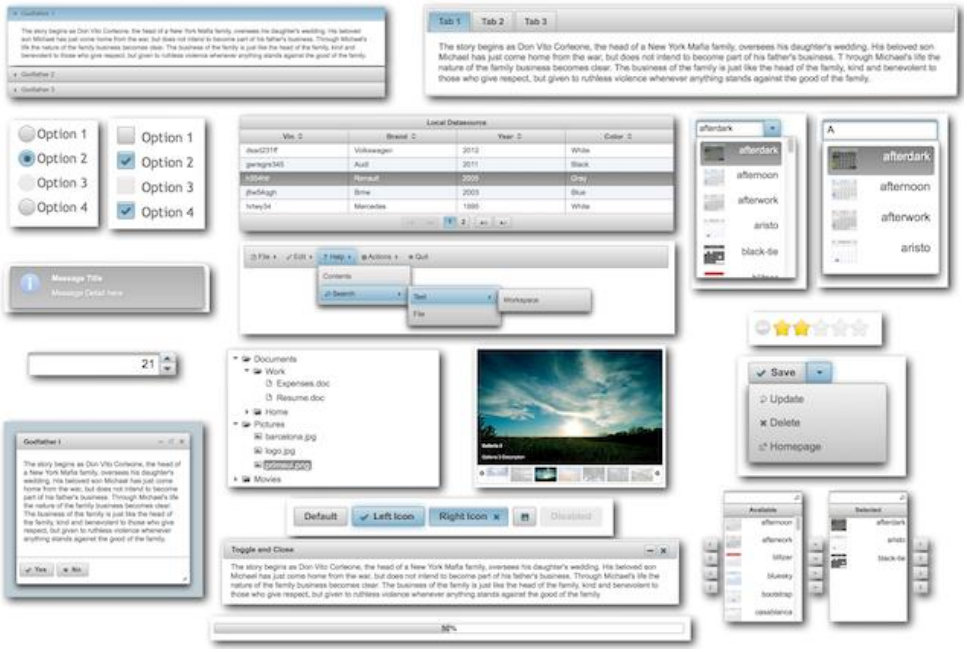


Ilustración #16: Despliegue de interfaces de Primefaces. (www.primefaces.org)

Las secciones públicas del aplicativo web contarán con un template[20] en el que se especificará la cabecera, menú y disposición del contenido individual para evitar repetir contenido general en las páginas individuales:

Esquema #7: Template para secciones públicas del aplicativo web.

[19] **PrimeFaces:** Librerías escritas en CSS y JS para facilitar la creación de aplicaciones web por medio de componentes enfocados en Java Servlet Faces.

[20] **Template:** Plantilla de diseño web utilizada de forma uniforme en una página web.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">
<h:head>
      <title>
            Biblioteca del Convento de Sto. Domingo de Quito
      </title>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
      <h:outputStylesheet library="css" name="estilos.css" />
      <!--   <h:outputStylesheet name="css/screen.css" />-->
</h:head>
<h:body>
      <div id="header" class="ui-widget-header">
            <h1>Biblioteca del Convento de Sto. Domingo de Quito</h1>
      </div>
      <div>
      <div class="menuPublico">
      <p:breadCrumb>
      <p:menuitem value="Home"
url="http://localhost:8080/biblioteca/home.jsf" icon="fa fa-home"
style="display:inline-flex; background-image:none; text-indent: 0; width:
67px; line-height: 14px; margin-top: 4px;"/>
      <p:menuitem value="Información adicional"
url="http://localhost:8080/biblioteca/pages/info.jsf" icon="fa fa-info"
style="display:inline-flex;"/>
      <p:menuitem value="Contáctos"
url="http://localhost:8080/biblioteca/pages/contacto.jsf" icon="fa fa-map-
marker" style="display:inline-flex;"/>
      <p:menuitem value="Catálogos" url="home.jsf" icon="fa fa-book"
style="display:inline-flex;"/>
      <p:menuitem value="Iniciar sesión"
url="http://localhost:8080/biblioteca/pages/admin/adminPage.jsf" icon="fa fa-
user" style="display:inline-flex;"/>
   </p:breadCrumb>
   </div>
            <ui:insert name="content">
                  [Template content will be inserted here]
    </ui:insert>
      </div>

</h:body>
</html>
```

Fuente: Cristopher A. Pérez

De igual forma que en las partes públicas, las partes privadas del aplicativo web tendrán su respectivo template donde se especificará la cabecera, información del usuario en sesión y disposición del contenido individual.

A continuación se detallará la composición del diseño aplicado a las diferentes secciones de la parte pública de esta aplicación, las cuales a excepción de la sección de catálogos se dispone de un espacio central de 900 pixeles en donde se despliega su respectivo contenido.

4.1.1 Distribución de Página principal

Esta sección está compuesta por una imagen principal desplegada en la parte superior de la página, seguida por texto informativo sobre la biblioteca, este texto se despliega en dos columnas alineadas al largo de la imagen.



Ilustración #17: Pantalla principal del aplicativo web. (Cristopher Pérez)

4.1.2 Distribución de página de información adicional

Se compone de una serie de imágenes desplegadas en la parte izquierda de la sección de contenido de la página y a lado derecho se muestra el respecto texto acerca de la información adicional de la biblioteca.



Ilustración #18: Pantalla de información adicional del aplicativo web. (Cristopher Pérez)

4.1.3 Distribución de página de contacto

Esta sección es similar a la de información adicional debido a que su estructura es similar al desplegar el texto correspondiente en el lado derecho del contenido de la página pero en lugar de las imágenes del lado izquierdo, se muestra un mapa con la ubicación del Convento de Santo Domingo.

Para el despliegue del mapa en la sección de la página, se utiliza la propiedad gmap[21] de Primefaces el cual se conecta a la página de Google maps22 y al bean MapPoint para agregar las coordenadas del punto a desplegar.



Ilustración #19: Pantalla de información de contactos del aplicativo web. (Cristopher Pérez)

4.1.4 Distribución de página de catálogos

La presente sección es más similar a las vistas en el tipo de contenido privado que en el del tipo público, debido a que en esta sección se despliega la información contenida de los catálogos en una tabla.

Al elegir un elemento de la tabla se despliega su información en la parte inferior de la pantalla ubicándose debajo de la tabla.

[21] **GMAP:** Conjunto de librerías para programación geo referencial por medio de Google Maps.
[22] **Google Maps:** Servicio de cartografía otorgado por Google a través de internet con variedad de opciones para el despliegue de mapas.



Ilustración #20: Pantalla de catálogos del aplicativo web. (Cristopher Pérez)

Los elementos desplegados correspondientes a los catálogos se encuentran referenciados por medios de las etiquetas utilizadas para el registro de metadatos Dublin Core descritos en el capítulo 1.

Esquema #8: Ejemplo de aplicación de etiquetas para metadatos Dublin Core.

```
<head profile="http://dublincore.org/documents/dcq-html/">
<h2>Título:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.titulo}" />
<meta name="DC.title" content="#{catalogoDM.catalogo.titulo}" />
<h2>Tema:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.tema}" />
<meta name="DC.subject" content="#{catalogoDM.catalogo.tema}" />
<h2>Tipo:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.tipo}" />
<meta name="DC.type" content="#{catalogoDM.catalogo.tipo}" />
<h2>Creador:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.creador}" />
<meta name="DC.creator" content="#{catalogoDM.catalogo.creador}" />
<h2>Descripción:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.descripcion}" />
<meta name="DC.description"
content="#{catalogoDM.catalogo.descripcion}" />
<h2>Fuente:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.fuente}" />
<meta name="DC.source" content="#{catalogoDM.catalogo.fuente}" />
<h2>Editor:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.editor}" />
<meta name="DC.editor" content="#{catalogoDM.catalogo.titulo}" />
<h2>Estado:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.estado}" />
<h2>Biblioteca:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.biblioteca}" />
<meta name="DC.relation" content="#{catalogoDM.catalogo.biblioteca}"
/>
<h2>Páginas:</h2>
<h:outputLabel value="#{catalogoDM.catalogo.paginas}" />
<meta name="DC.identifier" content="#{catalogoDM.catalogo.paginas}" />
```

```
</head>
```

Fuente: Cristopher A. Pérez

La identificación, ubicación y contextura de los elementos que componen el aplicativo web se encuentran registrados en una hoja de estilo en formato CSS.

Esquema #9: Hoja de estilo del aplicativo web

```
@CHARSET "UTF-8";
#header{
	height:120px;
	text-align:center;
}
#header h1{
	margin-top:40px;
}
#contenedorPaneles{
	width:940px;
	margin-left:auto;
	margin-right:auto;
}

#imagenPrincipal{
	width:900px;
	height:450px;
}
.imagenPrincipal{
	width:900px;
	margin-left:auto;
	margin-right:auto;
}

.ui-panel{
	width:250px;
}

.login{
	margin-left:auto;
	margin-right:auto;
	margin-top:10px;
	width:300px;
}

.menuPublico .ui-breadcrumb.ui-module.ui-widget.ui-widget-header.ui-helper-
clearfix.ui-corner-all{
	margin-left:auto;
	margin-right:auto;
	margin-top:10px;
	width: 889px;
}

.menuPublico .ui-breadcrumb-chevron{
	margin-right:55px;
}

.menuPublico .ui-widget-header .ui-icon {
	background-image:none;
```

```
}
/*
.ui-icon.fa {
     text-indent:0px;
     background-image:none;
}*/
.infoPrincipal {
    -webkit-column-count: 2; /* Chrome, Safari, Opera */
    -moz-column-count: 2; /* Firefox */
    column-count: 2;
    -webkit-column-gap: 40px; /* Chrome, Safari, Opera */
    -moz-column-gap: 40px; /* Firefox */
    column-gap: 40px;
    -webkit-column-rule-style: solid; /* Chrome, Safari, Opera */
    -moz-column-rule-style: solid; /* Firefox */
    column-rule-style: solid;
    width:900px;
    margin-left:auto;
    margin-right:auto;
}

.infoAdicional{
       width:900px;
       margin-left:auto;
       margin-right:auto;
}

.infoImagenes{
       width:400px;
}

.infoTexto{
       width:400px;
       margin-left: 475px;
    margin-top: -818px;
}

.infoContacto{
       width:900px;
       margin-left:auto;
       margin-right:auto;
}

.mapa{
       width:400px;
}

.textoContacto{
       width:400px;
       margin-left: 498px;
    margin-top: -706px;
}


#navBar{
       min-width:960px;
       width:100%;
```

```
        height:33px;
        margin-top:10px;
        margin-bottom:10px;
}
#navegacion{
        float:left;
        width:80%;
}
#navegacionFunciones{
        float: left;
    font-size: 12px;
    height: 23px;
    margin-left: 2px;
    padding: 6px 10px 0 0;
    text-align: right;
    width: 18%;
}
#perfil{
        margin-right:10px;
}

#menuBar{
        margin-top:10px;
        width:100%;
        height:60px;
        min-width:960px;
}
#menuBarIzquierda{
        float:left;
        width:70%;
}
#menuBarDerecha{
        float:left;
        width:30%;
        min-width:300px;
        text-align:right;
}

#footer{
        text-align:center;
        font-size:12px;
        clear:both;
}

.izquierda{
        text-align:center;
        margin-top:auto;
        margin-left:20px;
        margin-right:20px;
        margin-bottom:auto;
}

.derecha{
        text-align:center;
        margin-top:auto;
        margin-left:20px;
        margin-right:20px;
```

```
        margin-bottom:auto;
}
/* Tamaño menor de la letra*/
.ui-widget, .ui-widget .ui-widget {
    font-size: 92%;
}
.fileupload-content{
        min-height:40px;
}
/*Para el formulario de Taxonomia/Ficha*/


div.tablaFichaEspecie table{
   width: auto;
}
.fichaReferenciaColumna{
        width: 300px;
}
```

Fuente: Cristopher A. Pérez

4.2 Desarrollo de la funcionalidad de módulos

Como se mencionó en el capítulo 2, los módulos principales que contemplan la funcionalidad principal del aplicativo web son los siguientes:

Tabla #6: Módulos del aplicativo web.

Catálogo	Media	Ubicación	Usuario	Registro

Fuente: Cristopher A. Pérez

Su composición permite que cada una de las entidades pueda funcionar de acuerdo a lo descrito en la implementación de código.

4.2.1 Implementación de código

Al realizar la descripción del diseño implementado y mencionado las entidades que conforman el aplicativo web, procedemos a describir cómo funcionan estas entidades para poder almacenar y presentar los datos a tratar.

En cada sección correspondiente a una entidad contará con una tabla donde se podrá desplegar sus datos. Estas tablas contarán con un área de texto de búsqueda general y otras en cada una de las columnas que representan la información desplegada con el objetivo de facilitar las consultas específicas de información.

Para acceder a estas funciones se debe elegir la opción “Iniciar sesión” del menú principal el cual llevará a una nueva página donde se debe ingresa un nombre de usuario y contraseña básica.



Ilustración #21: Pantalla de inicio de sesión. (Cristopher Pérez)

Si los datos ingresados son correctos, se abrirá la ventana de administración de contenido, donde se puede elegir entre las entidades ya descritas.



Ilustración #22: Página principal de administración del sitio web. (Cristopher Pérez)

4.2.1.1 Usuario

En esta entidad se registran los datos de los usuarios encargados de utilizar el aplicativo web para el registro de los catálogos correspondientes. Los datos de usuario se despliegan en su correspondiente tabla.



Ilustración #23: Página de administración de datos de usuarios. (Cristopher Pérez)

Para agregar un nuevo usuario, se debe presionar el botón “Nuevo” donde aparece una nueva ventana para ingresar los correspondientes datos. Si estos datos fueron ingresados correctamente, al pulsar el botón “Guardar” se podrá visualizar el nuevo contenido en la tabla.

El procedimiento para editar un registro de usuario es similar al de ingreso de un nuevo registro, con la excepción que primero se debe elegir el elemento a editar de la tabla y posteriormente presionar el botón “Editar”.

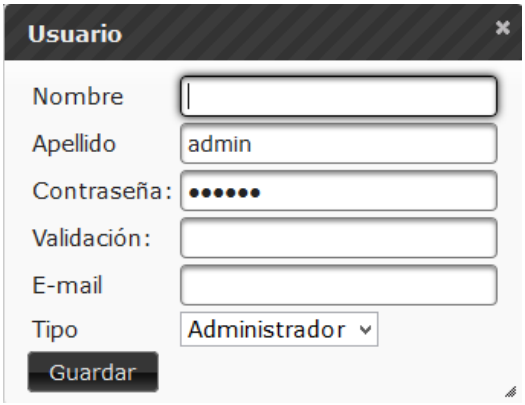


Ilustración #24: Ingreso de datos de usuario. (Cristopher Pérez)

4.2.1.2 Registro

Esta entidad sirve para poder registrar y tener constancia de las acciones de los usuarios. Por esta razón esta entidad no requiere de modificación y eliminación de datos. La entidad en sí está enfocada únicamente en la consulta de datos.

Para esto los datos de registros podrán ser revisados únicamente a través de su tabla.



Ilustración #25: Pantalla de registros del aplicativo web. (Cristopher Pérez)

4.2.1.3 Catálogo

Para este esquema en particular existen dos secciones o partes enfocadas a los catálogos registrados referentes a la parte de visita pública y privada.

En la visita pública un usuario cualquiera podrá acceder a este sección por medio del enlace con el nombre de “catálogos” ubicada en el menú principal.

En ella se puede seleccionar un elemento y al presionar “Mostrar catálogo” se despliegan los datos en la parte inferior dentro de las etiquetas utilizadas por el sistema de metadatos Dublin Core para referenciar dicho contenido en Internet.



Ilustración #26: Pantalla de catálogos del aplicativo web. (Cristopher Pérez)

La vista privada es aquella que permite manejar el contenido de los catálogos y solo puede ser accedida por medio de un inicio de sesión y por medio del enlace concedido en la página principal de administración previa al inicio de sesión.

De igual forma que en la vista pública, en la vista privada se visualiza una tabla con los contenidos de catálogos ya registrados, con la adición de poder ingresar, editar y eliminar nuevos contenidos.



Ilustración #27: Pantalla de administración de catálogos. (Cristopher Pérez)

Al presionar el botón “Nuevo” se despliega una pantalla con los datos a ingresar referentes al catálogo ubicado en la etiqueta “Catálogo”.

En la etiqueta “Ubicación” se puede elegir una ubicación disponible para el presente catálogo a ingresar.

Dentro de la etiqueta “Contenido multimedia” se puede anexar el presente catálogo a los recursos multimedia disponibles eligiendo los registros de la tabla de contenido y presionando el botón y confirmando la selección con el botón o el botón para cancelar un anexo del recurso. Los elementos anexados al catálogo actual se desplegarán en la lista ubicada en la parte inferior de la tabla de contenido multimedia.

Al terminar este proceso y presionar el botón “Guardar” se cerrará la ventana y se volverá a visualizar la tabla anterior, con el nuevo registro lista en la presente tabla.

El proceso citado anteriormente sirve para editar catálogos ingresados anteriormente, a lo que previamente se debe elegir un elemento de la tabla y presionar el botón “Editar”. Después de esto, el proceso es igual al de ingreso de un nuevo registro.

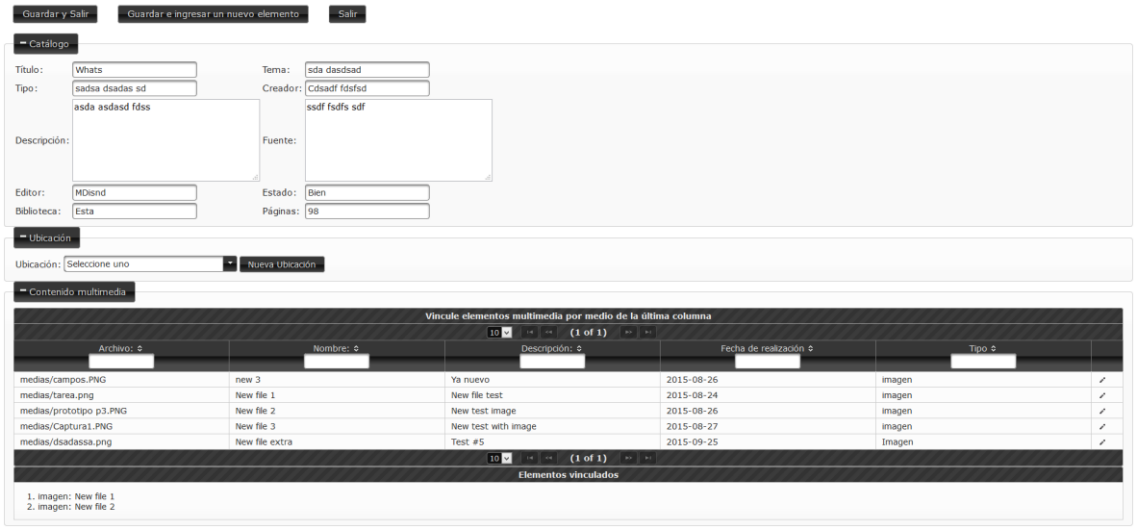


Ilustración #28: Ingreso de datos de catálogo. (Cristopher Pérez)

Para eliminar un catálogo registrado se debe elegir el elemento deseado de la tabla y presionar el botón “Borrar” y confirmar el mensaje que se muestra. Con este procedimiento el registro no aparecerá en la tabla.

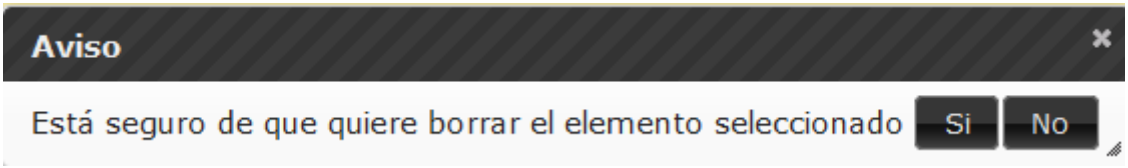


Ilustración #29: Diálogo de confirmación para eliminar datos. (Cristopher Pérez)

4.2.1.4 Ubicación

Permite el registro de la ubicación física en la biblioteca referente a la información referenciada en los catálogos. Estos se despliegan en su correspondiente tabla.



Ilustración #30: Página de administración de ubicaciones. (Cristopher Pérez)

Para ingresar una nueva ubicación es necesario presionar el botón “Nuevo” donde aparecerá una nueva ventana donde se deben ingresar los correspondientes datos de ubicación.

Al presionar el botón de “Guardar”, el nuevo registro debe estar listado en la tabla. Se debe seguir los mismos pasos para poder editar un registro después de seleccionar el registro de la tabla y presionar el botón editar.



Ilustración #31: Ingreso de datos de ubicación. (Cristopher Pérez)

Para poder eliminar un registro se lo debe elegir de la tabla y presionar el botón “Eliminar”, al hacer esto aparecerá una pantalla que confirma al usuario de su decisión. Al confirmar la acción y presionar “Si”, el registro habrá desaparecido de la tabla indicando su correcta eliminación.

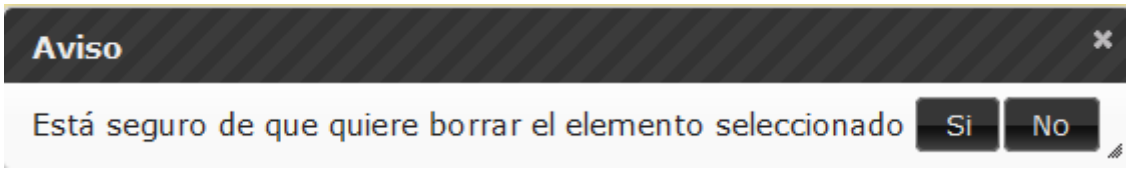


Ilustración #32: Diálogo de confirmación para eliminar datos. (Cristopher Pérez)

4.2.1.5 Media

Es referente al contenido multimedia que puede estar asociado a los catálogos registrados, los registros de contenidos multimedia en el sistema se pueden revisar en la siguiente tabla.



Ilustración #33: Pantalla de administración de contenido multimedia. (Cristopher Pérez)

Para agregar nuevo contenido, se debe presionar el botón nuevo, el cual despliega una nueva ventana con los datos a ingresar. Aparte de esto es necesario agregar un archivo con el botón “Choose” y seleccionar el archivo en el explorador que aparece a continuación.

Una vez seleccionado el archivo se debe presionar el botón “Upload” y esperar a que un mensaje confirme que el proceso de subida de archivo ha finalizado exitosamente.

Como último paso se debe presionar el botón “Guardar” y el nuevo registro aparecerá en la tabla.

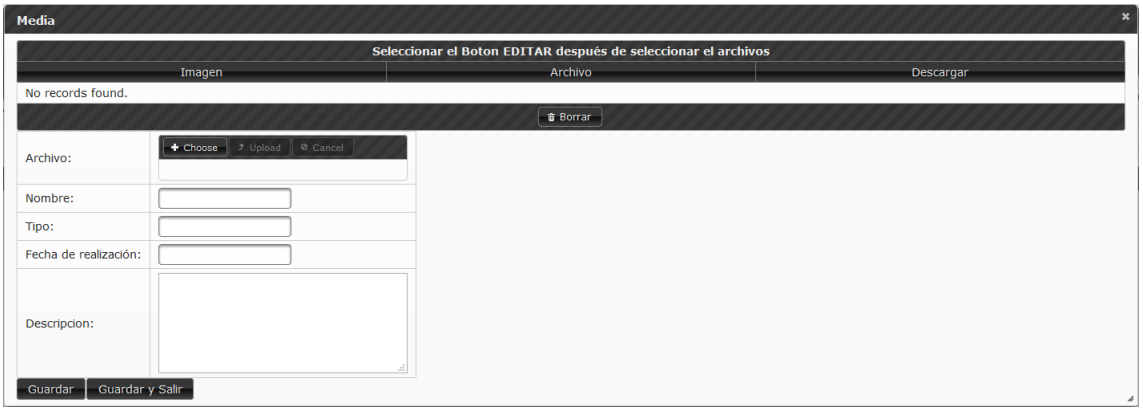


Ilustración #34: Ingreso de contenido multimedia. (Cristopher Pérez)

El proceso de edición de datos es similar al de agregar un nuevo registro pero primero se debe elegir un elemento de la tabla y presionar el botón “Editar”.

Para eliminar un registro se debe elegir el elemento deseado en la tabla y presionar el botón “Eliminar”, donde aparecerá un diálogo que nos pide confirmar el dato a eliminar. Al presionar sí en el diálogo, el elemento seleccionado será eliminado del registro.

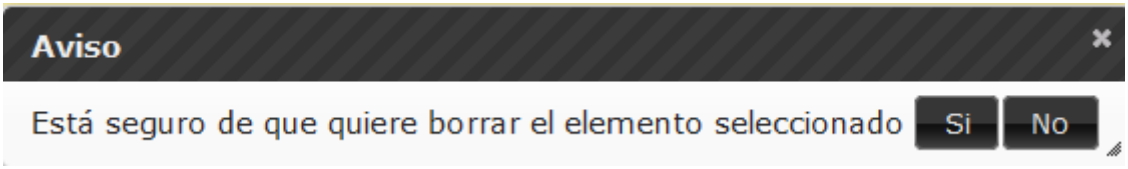


Ilustración #35: Diálogo de confirmación para eliminar datos. (Cristopher Pérez)

Para revisar el código fuente del aplicativo web bajo la licencia GNU[23] revise los anexos de este trabajo.

4.3 Integración de módulos

Los componentes de cada entidad estarán divididos en paquetes para diferenciar entre las funcionalidades principales de cada componente. Estas entidades tienen que formar relaciones para cumplir sus funciones y consultas en el aplicativo web. Su composición puede ser apreciada en el mapa del aplicativo web que se muestra más adelante.

4.3.1 Relaciones entre entidades

Estas son las relaciones que se han forjado entre las entidades para el funcionamiento del aplicativo web de acuerdo a las necesidades obtenidas de los futuros usuarios de la misma.

4.3.1.1 Usuario y Registro

Para poder obtener constancia de las acciones realizadas por los usuarios encargados del manejo del aplicativo, se ha implementado la relación donde un usuario puede poseer más de un registro.

[23] **GNU (General Public Licence):** Licencia utlizada en productos de software el cual garantiza a los usuarios la libertad de uso, estudio, modificación y distribución del producto de software.

La entidad de usuario requiere de un generador Hash de caracteres para ocultar las contraseñas de los usuarios al ingresarlas en el registro de usuarios previamente explicado. Este generador es llamado como un servicio al ingresar una contraseña.

Esquema #10: Generador hasher de caracteres.

```
package biblioteca.service;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MD5Hasher implements Hasher {

	private static final char[] HEXADECIMAL = { '0', '1', '2', '3', '4',
'5',
			'6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };

	@Override
	public String generateHash(final String stringToHash) {
		try {
			MessageDigest md = MessageDigest.getInstance("MD5");
			byte[] bytes = md.digest(stringToHash.getBytes());
			StringBuilder sb = new StringBuilder(2 * bytes.length);
			for (int i = 0; i < bytes.length; i++) {
				int low = (bytes[i] & 0x0f);
				int high = ((bytes[i] & 0xf0) >> 4);
				sb.append(HEXADECIMAL[high]);
				sb.append(HEXADECIMAL[low]);
			}
			return sb.toString();
		} catch (NoSuchAlgorithmException e) {
			return null;
		}
	}

}
```

Fuente: Cristopher A. Pérez

Para poder llevar a cabo los registros dentro del aplicativo web, estas dos instancias deben estar reflejadas en el resto de instancias y obtener la información de usuario autentificado al momento de manejar los recursos del sistema.

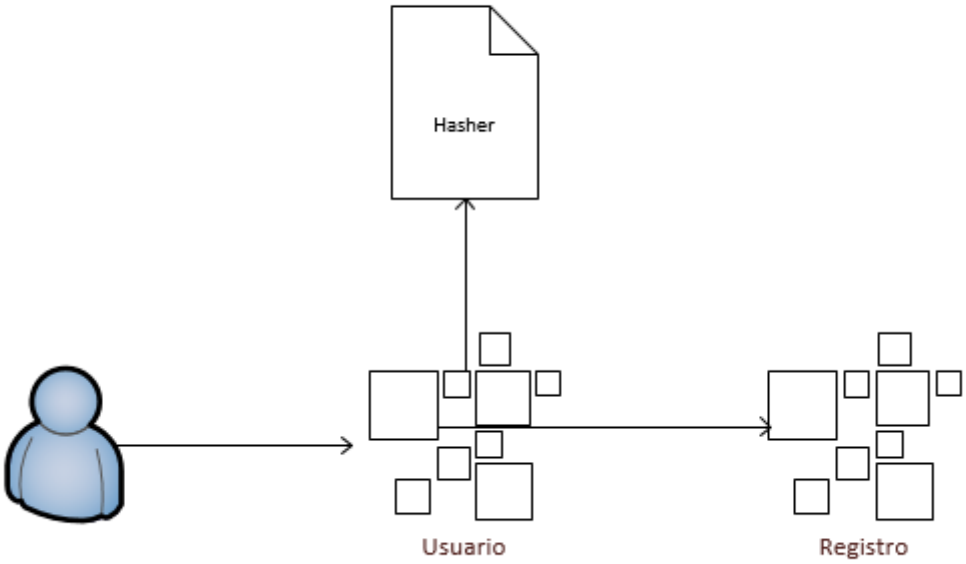


Ilustración #36: Gráfico de relación entre las entidades de usuario y registro. (Cristopher Pérez)

4.3.1.2 Catálogo y Ubicación

El registro de un catálogo en el aplicativo web puede ser asociado a una ubicación en específico dentro de la biblioteca, por lo que su relación corresponde a que un catálogo puede tener una ubicación de forma no obligatoria.

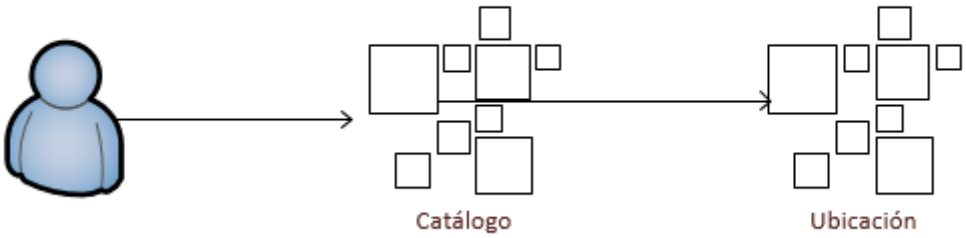


Ilustración #37: Gráfico de relación entre las entidades de catálogo y ubicación. (Cristopher Pérez)

4.3.1.3 Catálogo y Media

Cada catálogo registrado puede contener documentos, imágenes u otros elementos multimedia que haga referencia al mismo catálogo, por lo que la correspondiente asociación da lugar a que un catálogo puede tener uno o más contenidos multimedia de forma no obligatoria.

Adicionalmente, la entidad de media requiere de un manejador de archivos donde se den instrucciones sobre el nombre y almacenamiento físico de los archivos llevado a la par al registro del contenido multimedia en el sistema.

Esquema #11: Manejo de archivos multimedia

```
package biblioteca.aws;

import java.util.Date;

public class Archivo {
	public String nombre;

	public String url;

	public Archivo() {
	}

	public Archivo(String url) {

		this.url = url;
		for(int i=0;i<url.length();i++ ){
				if (url.charAt(i)=='/')
				{
					nombre=url.substring(i+1, url.length());
					break;
					}
			}
	}

	public Archivo(String nombre, Date fechaInicial, Date
fechaFinal) {
		this.nombre = nombre;
		this.nombre = nombre;
	}
	public String getNombre() {
		return nombre;
	}

	public void setNombre(String nombre) {
		this.nombre = nombre;
	}

	public String getUrl() {
		return url;
	}

	public void setUrl(String url) {
		this.url = url;
	}

}
```

Fuente: Cristopher A. Pérez



Ilustración #38: Gráfico de relación entre las entidades de catálogo y media. (Cristopher Pérez)

4.3.2 Mapa del aplicativo web.

Por medio del siguiente gráfico se puede visualizar la composición total del aplicativo web.

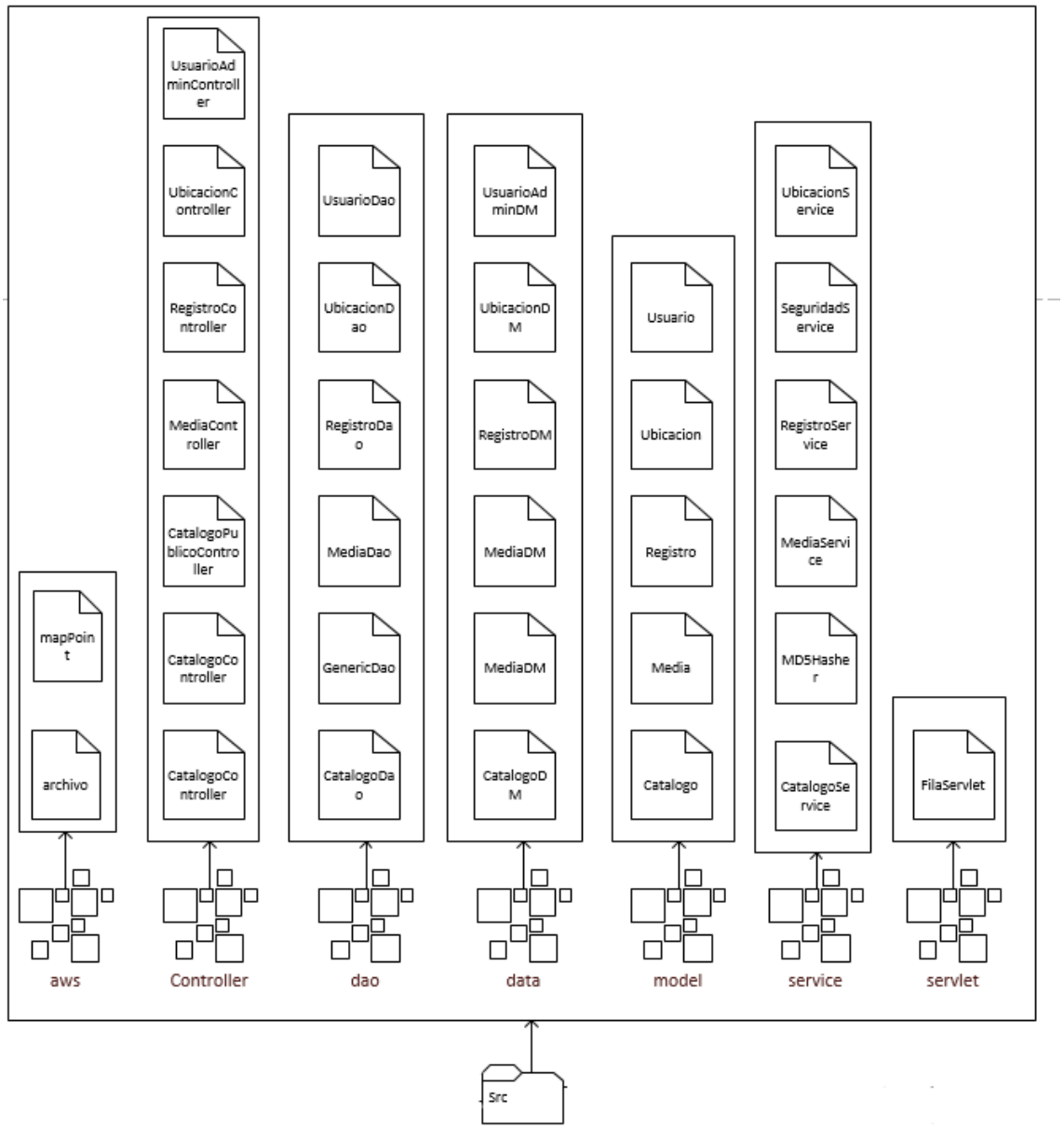


Ilustración #39: Composición de elementos de procesamiento de datos del aplicativo web. (Cristopher Pérez)

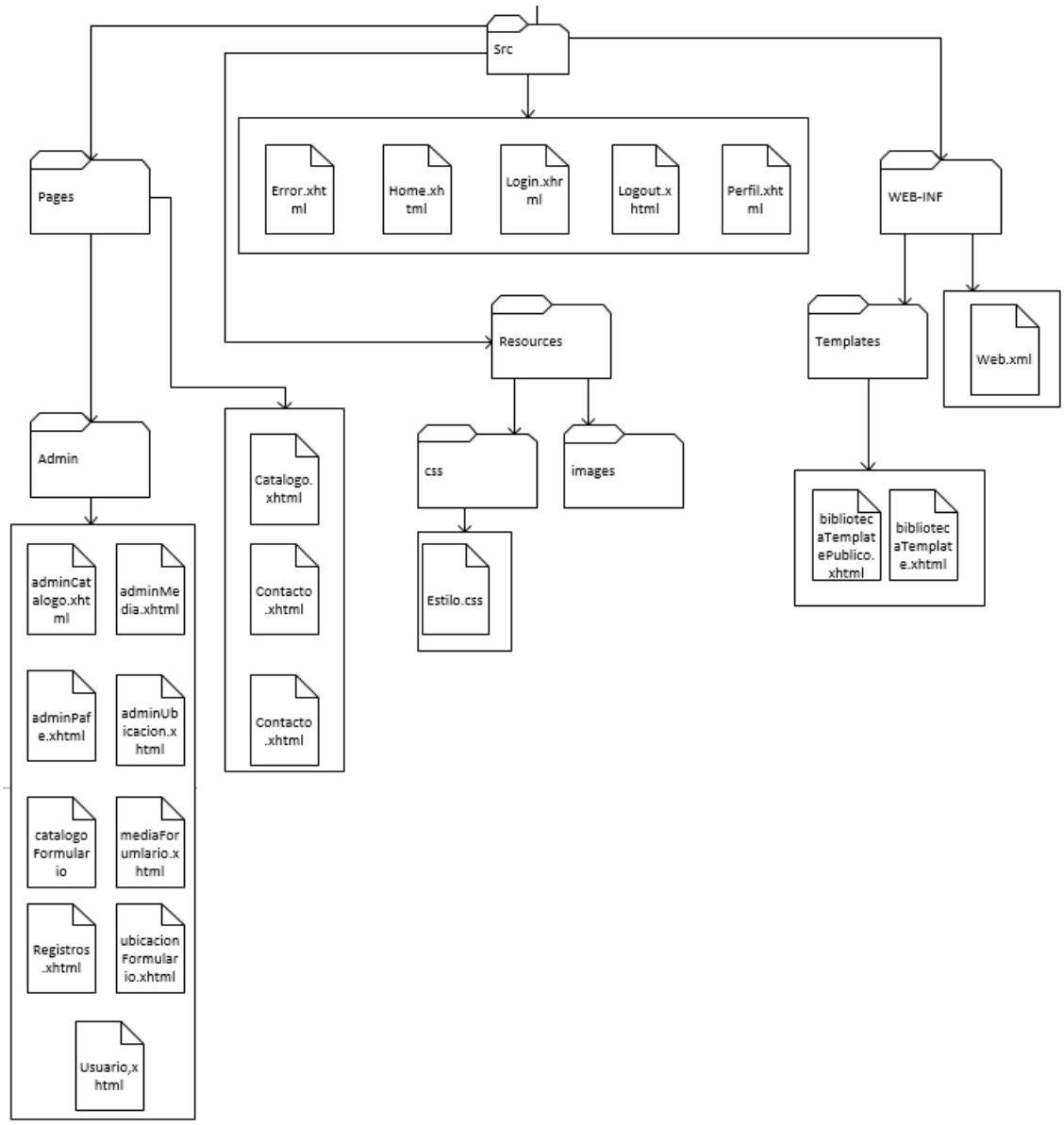


Ilustración #40: Composición de la interfaz del aplicativo web. (Cristopher Pérez)

Capítulo 5: Pruebas de funcionamiento de la aplicación

5.1 Pruebas de compatibilidad con navegadores

Al completar el desarrollo del aplicativo web, se realizarán pruebas de compatibilidad con los navegadores web con mayor uso de acuerdo a la siguiente fuente:

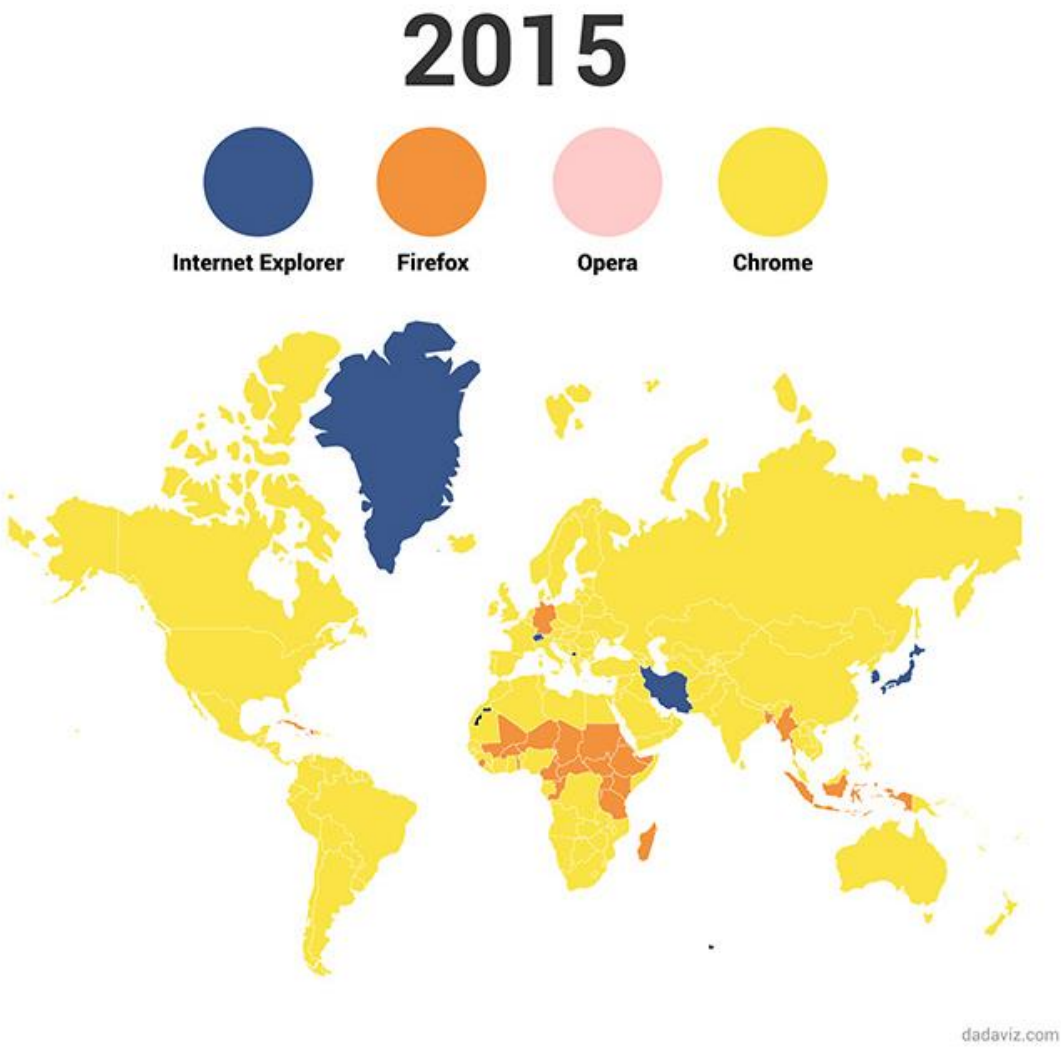


Ilustración #41: Evaluación de navegadores web más usados (3).

Dada esta información, se ha decidido realizar las pruebas de compatibilidad con los tres navegadores Chrome, Firefox e Internet Explorer.

5.1.1 Chrome



Ilustración #42: Ejemplo de despliegue del aplicativo web en Chrome v45.0.2454.85 (Cristopher Pérez)



Ilustración #43: Ejemplo de despliegue del aplicativo web en Chrome v45.0.2454.85 (Cristopher Pérez)



Ilustración #44: Ejemplo de despliegue del aplicativo web en Chrome v45.0.2454.85 (Cristopher Pérez)



Ilustración #45: Ejemplo de despliegue del aplicativo web en Chrome v45.0.2454.85 (Cristopher Pérez)



Ilustración #46: Ejemplo de despliegue del aplicativo web en Chrome v45.0.2454.85 (Cristopher Pérez)

5.1.2 Firefox



Ilustración #47: Ejemplo de despliegue del aplicativo web en Firefox v40.0.3 (Cristopher Pérez)



Ilustración #48: Ejemplo de despliegue del aplicativo web en Firefox v40.0.3 (Cristopher Pérez)



Ilustración #49: Ejemplo de despliegue del aplicativo web en Firefox v40.0.3 (Cristopher Pérez)

5.1.3 Internet Explorer

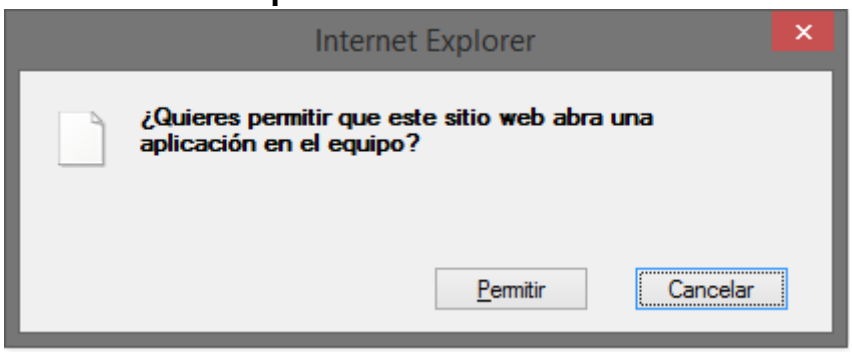


Ilustración #50: Ejemplo de despliegue del aplicativo web en Internet Explorer v11.0.9600.16384 (Cristopher Pérez)



Ilustración #51: Ejemplo de despliegue del aplicativo web en Internet Explorer v11.0.9600.16384 (Cristopher Pérez)



Ilustración #52: Ejemplo de despliegue del aplicativo web en Internet Explorer v11.0.9600.16384 (Cristopher Pérez)



Ilustración #53: Ejemplo de despliegue del aplicativo web en Internet Explorer v11.0.9600.16384 (Cristopher Pérez)



Ilustración #54: Ejemplo de despliegue del aplicativo web en Internet Explorer v11.0.9600.16384 (Cristopher Pérez)

5.1.4 Observaciones

Al realizar las pruebas de visualización y funcionamiento básico del aplicativo web en los diferentes navegadores, se han observado ligeros cambios en el despliegue de los elementos como la diferencia en el tamaño de letra y variaciones en la disposición de cuadros de texto en la página de inicio de sesión como se puede apreciar en las imágenes #5 y #13.

Para la corrección de despliegues de elementos en diferentes navegadores web, es necesario aplicar instrucciones en la hoja de estilo CSS dirigidas a los navegadores web en específico.

Esquema #12: Instrucciones de hoja de estilo dirigida a Chrome

```
@media screen and (-webkit-min-device-pixel-ratio:0) {
      .login_user input:-webkit-autofill, textarea:-webkit-autofill,
select:-webkit-autofill {
      -webkit-box-shadow: 0 0 0px 1000px white inset;
      }

      .login_pass input:-webkit-autofill, textarea:-webkit-autofill,
select:-webkit-autofill{
      -webkit-box-shadow: 0 0 0px 1000px white inset;
      }

}
```

Fuente: Cristopher A. Pérez

Esquema #13: Instrucciones de hoja de estilo dirigida a Internet Explorer

```
@media all and (-ms-high-contrast: none), (-ms-high-contrast: active)
{
     .login_pass input {
     width:151px;}
}
```

Fuente: Cristopher A. Pérez

A parte de estas observaciones no se ha detectado cambios o complicaciones en la funcionalidad del aplicativo web además del mensaje de confirmación presentado en Internet Explorer al momento de abrir el aplicativo web por primera vez.

Esto ha demostrado que la herramienta Primefaces tiene muy buena compatibilidad con los navegadores web más usados en la actualidad.

5.2 Pruebas individuales de módulos

Se han realizado pruebas de funcionamiento básicas con las entidades que conforman el aplicativo web y comprobar su funcionamiento individual. Al hacer esto se han encontrado observaciones y errores listados a continuación.

- Excepción NullPointerException

  Este error se presentó al tratar de realizar consultas en las distintas entidades y fue necesario una revisión al código de las instancias de controlador y manejo de datos. Al realizar esto se descubrió una mal instancia del manejador de datos en las tablas y en los controladores de las correspondientes entidades.

Esquema #14: Ejemplo de conexión de un manejador de datos de una entidad en una tabla de datos

```
<p:dataTable var="catalogos" rowKey="#{catalogos.idCatalogo}"
                  rows="10" value="#{catalogoDM.catalogos}"
id="tablaCatalogo"
                  selection="#{catalogoDM.catalogo}" paginator="true"
                  paginatorTemplate="{RowsPerPageDropdown}
{FirstPageLink} {PreviousPageLink} {CurrentPageReport} {NextPageLink}
{LastPageLink}"
                  rowsPerPageTemplate="5,10,15" selectionMode="single"
lazy="true">

</p:dataTable>
```

Fuente: Cristopher A. Pérez

Esquema #15: Ejemplo de conexión de un manejador de datos de una entidad desde el controlador de la entidad.

```
@ManagedBean(name = "catalogoController")
@ViewScoped
public class CatalogoController {

      @ManagedProperty(value = "#{catalogoDM}")
      private CatalogoDM catalogoDM;

}
```

Fuente: Cristopher A. Pérez

- Error de despliegue de paneles

  Se observó que al intentar desplegar nuevos paneles o ventanas, estos no aparecían o no se producía la conexión entre los elementos desplegados en la pantalla. Esto se debe a que se utilizaban instrucciones desactualizadas para la versión utilizada de Primefaces.

Esquema #16: Ejemplo de instrucción para desplegar un panel en Primefaces v5.2

```
<p:commandButton id="nuevobtn" value="Nuevo" process="@this"

      update=":ubicacionAdminForm:ubicacionModal"
```

```
      action="#{ubicacionController.nuevo()}"

      oncomplete="PF('ubicacionPanel').show()" />
```

Fuente: Cristopher A. Pérez

5.3 Pruebas exhaustivas con módulos integrados

Al comprobar la correcta funcionalidad de las entidades de forma individual, se procede a realizar pruebas exhaustivas relacionando las entidades y comprobando la funcionalidad total del sistema. De esto se tienen las siguientes observaciones:

- Sentencias SQL no reconocidas

  Al intentar realizar consultas o instrucciones específicas en lenguaje SQL como encontrar las ubicaciones físicas que aún no han sido vinculadas a ningún catálogo, se producía este tipo de error.

  Para ello se requiere revisar las sentencias SQL y verificar que los nombres de los campos escritos esta coincidan con las variables en el modelo de la entidad escritos en Java.

Esquema #17: Ejemplo de esquema escrito en lenguaje SQL desde Java

```
String sql = new String(
                        "select u from Ubicacion u where not exists
(select c.ubicacion from Catalogo c where c.ubicacion =
u.idUbicacion)");
```

Fuente: Cristopher A. Pérez

- Instancia no inicializada

  Al intentar obtener registros a través de las relaciones entre entidades se despliega un mensaje indicando que una instancia no ha sido inicializada a tiempo. Para esto es necesario cambiar el tipo de asociación especificada en las relaciones entre entidades a “Eager[24]”.

Esquema #18: Ejemplo de asociación entre elementos de entidades

```
@OneToMany(fetch = FetchType.EAGER, mappedBy = "catalogo")
      private Set<Media> medias = new HashSet<Media>(0);
```

Fuente: Cristopher A. Pérez

[24] **Eager:** Instrucción de conexión en Java perteneciente a “Java Persistence API” para especificar recuperación temprana de datos.

Capítulo 6: Conclusiones y recomendaciones del proyecto

Tras concluir el presente trabajo se ha llegado a las siguientes conclusiones y recomendaciones:

6.1 Conclusiones

- La herramienta Primefaces es útil para el manejo de datos en un aplicativo web y no tiene problemas de funcionamiento ni mayores problemas de compatibilidad con los navegadores web utilizados en la actualidad.
- Mediante el sistema de metadatos Dublin Core es posible referenciar diferentes tipos de contenido al internet para el conocimiento del público en general. Además, el sistema de metadatos resulta bastante útil para referenciar catálogos de bibliotecas por medio de sus etiquetas debido a su flexibilidad en la aplicación de sus normas.
- El software libre utilizado como herramientas para la elaboración del aplicativo web ha sido lo suficientemente estable como para no causar problemas en las fases de desarrollo y pruebas del aplicativo web y no fue necesario cambiar de herramientas durante el desarrollo del mismo.
- El uso de la metodología de programación extrema o “Extreme Programing” permitió un avance rápido en la fase de desarrollo de la aplicación web al realizar pruebas de funcionamiento y revisión de código a la par del proceso de programación. A pesar de que todo el proceso fue realizado por una sola persona.
- Siendo una herramienta de software libre, la base de datos PostgresSQL ha cumplido con los requerimientos de las entidades presentadas para el aplicativo web y su conexión con Hibernate, y al igual que el resto de herramientas de software libre escogidas, PostgresSQL puede trabajar bajo diferentes plataformas.
- La encriptación de caracteres es un factor importante para la confidencialidad de datos de los usuarios y para la seguridad de un sitio web en sí.

6.2 Recomendaciones

- Es necesario conocer los esquemas de metadatos que ofrece Dublin Core antes de utilizar sus etiquetas en una página web con el motivo de conocer la orientación y significado de las etiquetas a implementar.
- Al desarrollar aplicaciones orientadas a internet, es necesario verificar la disponibilidad de los puertos que va a utilizar el servidor de aplicaciones y otras herramientas orientadas a internet. Esto se hace para evitar que este produzca conflictos de funcionamiento con otras aplicaciones que utilicen los mismos puertos en el equipo.
- Al trabajar con archivos multimedia dentro de una plataforma, se debe considerar varios aspectos como los tipos de archivos y tamaño permitidos de acuerdo a las necesidades presentadas y a los recursos disponibles.
- Al construir relaciones entre los modelos que representan las entidades en la base de datos por medio de Hibernate, hay que asegurarse de las conexiones

entre entidades se especifiquen con la relación “Eager” en lugar de “Lazy” [25] el cual se encuentra por defecto. Esto evita que existan problemas de conexión entre entidades.

- Además de instruir a los usuarios sobre la funcionalidad del aplicativo web, es válido aclara a los mismos la importancia de la privacidad de los datos requeridos para el ingreso al portal web.
- Para verificar el correcto funcionamiento del servidor de aplicaciones, se debe revisar la correcta configuración almacenada en el archivo XML de configuración del mismo servidor de aplicaciones y verificar la debida conexión con la base de datos y el aplicativo web en sí.
- Dentro de la facultad, la expansión en las áreas de trabajo como la tratada en este proyecto puede ayudar a los estudiantes interesados en prácticas pre-profesionales.

[25] **Lazy:** Instrucción de conexión en Java perteneciente a “Java Persistence API” para especificar recuperación tardía de datos en la primera consulta de datos.

Bibliografía

1. *Metadatos y prácticas de categorización.* Chile : Biblioteca Nacional de Chile, 2009.

2. Extreme Programming. *USFCA.* [En línea] 15 de Julio de 2015. http://www.cs.usfca.edu/~parrt/course/601/lectures/xp.html.

3. Evaluación de navegadores web más usados. *pixfans.* [En línea] 18 de Agosto de 2015. http://www.pixfans.com/evolucion-de-los-navegadores-mas-usados-desde-2008/.

4. Stinson, Barry. *PostgreSQL Essential Reference.* s.l. : Sams Publishing.

5. Gary Shelly, Thomas J. Cashman, Joy Starks, Michael Mick. *Java Programming: Comprehensive Concepts and Techniques.* s.l. : Cengage Learning.

6. Jboss. Jboss Aplication server. *Jboss Aplication server documentation.* [En línea] https://docs.jboss.org/author/display/AS71/Documentation.

7. jtech. *Introducción a los servidores de aplicaciones.* [En línea] http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/sesion1-apuntes.htm.

8. Coyle, Karen. *Linked Data Tools: Connecting on the Web.* s.l. : American Library Association, 2012.

9. Qian, Kai. *Java Web Development Illuminated.* s.l. : Jones & Bartlett Learning.

10. Sikos, Leslie. *Web Standards: Mastering HTML5, CSS3, and XML.* s.l. : Apress.

11. Metadata Dublin Core innovation. *Dublin Core.* [En línea] http://dublincore.org.

12. Caplan, Priscilla. *Metadata Fundamentals for all Librarians.* s.l. : American Library Association.

13. Downey, Tim. *Web Development with Java: Using Hibernate, JSPs and Servlets.* s.l. : Springer Science & Business Media.

14. Curtis Poole, Janette Bradley. *Developer's Digital Media Reference: New Tools, New Methods.* s.l. : Taylor & Francis, 2014.

15. Lapuente, María Jesús Lamarca. Hipertexto. *Metadatos Dublin Core.* [En línea] http://www.hipertexto.info/documentos/dublin_core.htm.