

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

**SISTEMA DE GESTIÓN Y DIFUSIÓN DE LA REVISTA
POLITÉCNICA NACIONAL**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y COMPUTACIÓN**

CHRISTIAN FERNANDO CHÁVEZ SALAS

chavez_christian1409@yahoo.es

DIRECTORA: MSC. MARIA HALLO

maria.hallo.epn.edu.ec

QUITO, ENERO 2016

DECLARACIÓN

Yo, CHRISTIAN FERNANDO CHÁVEZ SALAS, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional, y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo los derechos de propiedad intelectual correspondientes a este trabajo a la Escuela Politécnica Nacional según lo establecido por la ley de propiedad intelectual, por su reglamento, y por la normativa institucional vigente.

Christian Fernando Chávez Salas

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Christian Fernando Chávez Salas, bajo mi supervisión.

Msc. María Hallo

DIRECTORA DE PROYECTO

CONTENIDO

CONTENIDO	4
RESUMEN.....	10
INTRODUCCIÓN.....	11
CAPITULO 1 : MARCO TEÓRICO.....	13
1.1 DESCRIPCIÓN DEL PROBLEMA	13
1.2 REVISTAS CIENTÍFICAS DIGITALES.....	14
1.2.1 <i>REVISTAS CIENTÍFICAS Y ACCESO ABIERTO</i>	14
1.2.2 <i>VENTAJAS</i>	15
1.2.3 <i>RIESGOS</i>	15
1.2.4 <i>DOI</i>	16
1.2.5 <i>DUBLIN CORE</i>	16
1.3 ETAPAS DEL PROCESO DE PUBLICACIÓN	17
1.3.1 <i>INVESTIGACIÓN Y ENTREGA</i>	17
1.3.2 <i>REVISIÓN Y APROBACIÓN</i>	18
1.3.3 <i>PROCESOS DE EDICIÓN</i>	19
1.3.3.1 <i>Corrección de estilo</i>	20
1.3.3.2 <i>Edición de maquetación</i>	21
1.3.4 <i>PUBLICACIÓN</i>	22
1.4 PUBLICACIÓN DE DATOS ABIERTOS.....	22
1.4.1 <i>ACCESO ABIERTO (OPEN ACCESS)</i>	22
1.4.2 <i>INICIATIVAS</i>	23
1.4.2.1 <i>Budapest</i>	23
1.4.2.2 <i>Betsheda</i>	24
1.4.2.3 <i>Berlín</i>	24
1.4.3 <i>OAI PMH</i>	25
1.4.4 <i>TIPOS DE OPEN ACCESS</i>	26
1.4.4.1 <i>Acceso abierto verde (Green OA)</i>	27
1.4.4.2 <i>Acceso abierto dorado (Gold OA)</i>	27
1.4.4.3 <i>Acceso abierto gratis</i>	27
1.4.4.4 <i>Acceso abierto libre</i>	28
1.4.4.5 <i>Relación entre los tipos de acceso abierto</i>	28
1.4.5 <i>BENEFICIOS DEL OPEN ACCESS SOBRE LAS BIBLIOTECAS</i>	28

1.4.6	WEB SEMÁNTICA	29
1.4.6.1	<i>Datos enlazados (Linked Data)</i>	30
1.4.6.2	<i>URIs.....</i>	31
1.4.6.3	<i>RDF (Resource description framework).....</i>	31
1.4.6.4	<i>Enlaces entre sujeto y objeto.....</i>	33
1.4.6.5	<i>SPARQL</i>	33
1.4.6.6	<i>Punto SPARQL.....</i>	34
1.4.7	PROCESO DE PUBLICACIÓN DE LINKED DATA	34
1.5	METODOLOGÍA DE DESARROLLO	34
1.5.1	DEFINICIÓN DE LA METODOLOGÍA.....	34
1.5.1.1	<i>Definición de RUP.....</i>	34
1.5.1.2	<i>Principios De RUP</i>	35
1.5.1.3	<i>Fases.....</i>	36
1.5.1.4	<i>Justificación</i>	37
1.5.2	HERRAMIENTAS A UTILIZAR.....	37
1.5.2.1	<i>Open Journal Systems</i>	37
1.5.2.2	<i>Base de datos: Mysql.....</i>	38
1.5.2.3	<i>Lenguaje de programación: PHP.....</i>	39
1.5.2.4	<i>Servidor web: Apache</i>	39
1.5.2.5	<i>Software de extracción de datos: Triplify.....</i>	40
1.5.2.6	<i>Publicación de punto SPARQL: Virtuoso Opensource</i>	40
1.5.2.7	<i>Front end para el punto SPARQL: Pubby.....</i>	41
1.5.2.8	<i>Contenedor de servlets para Pubby: Tomcat</i>	41
1.5.2.9	<i>JDK (Java Development Kit).....</i>	42
CAPITULO 2	INICIO Y ELABORACIÓN.....	43
2.1	MODELAMIENTO DEL NEGOCIO	43
2.1.1	ACTORES DEL NEGOCIO	43
2.1.2	CASOS DE USO DEL NEGOCIO	44
2.1.2.1	<i>Administrar revista.....</i>	44
2.1.2.2	<i>Revisar artículo</i>	45
2.1.2.3	<i>Corregir estilo</i>	47
2.1.2.4	<i>Publicar Edición.....</i>	49
2.2	CAPTURA DE REQUERIMIENTOS.....	50
2.2.1	CARACTERÍSTICAS DE USUARIOS.....	50
2.2.2	REQUERIMIENTOS NO FUNCIONALES	51
2.2.3	CASOS DE USO DEL SISTEMA (REQUISITOS FUNCIONALES)	51

2.2.3.1	Acceder al sistema.....	52
2.2.3.2	Configurar revista	53
2.2.3.3	Administrar usuarios.....	54
2.2.3.4	Notificar usuarios.....	56
2.2.3.5	Entregar artículo.....	57
2.2.3.6	Revisar artículo	58
2.2.3.7	Corregir estilo	61
2.2.3.8	Editar maquetación	63
2.2.3.9	Publicar edición	64
2.2.3.10	Publicar Datos semánticos.....	66
2.2.4	RESTRICCIONES.....	67
2.3	ANÁLISIS Y DISEÑO	69
2.3.1	DIAGRAMAS DE CLASES	69
2.3.1.1	Clases Handlers	69
2.3.1.2	Clases DAO.....	72
2.3.1.3	Clases de formulario	73
2.3.1.4	Clases administrativas	75
2.3.2	MODELO DE BASE DE DATOS.....	78
2.3.3	DISEÑO DE INTERFACES.....	78
2.3.3.1	Interfaz principal.....	78
2.3.3.2	Bloque contenido.....	79
CAPITULO 3 CONSTRUCCIÓN Y TRANSICIÓN		85
3.1	DESARROLLO	85
3.1.1	ESTÁNDARES DE CODIFICACIÓN.....	85
3.1.2	ARQUITECTURA	86
3.1.3	DIAGRAMA DE DESPLIEGUE	88
3.1.4	DIAGRAMA DE COMPONENTES [16].....	90
3.1.5	COMPONENTES	91
3.1.5.1	Publicación y administración de la revista	91
3.1.5.2	Publicación SPARQL.....	94
3.2	PRUEBAS DEL SISTEMA	97
3.2.1.1	Ingreso Al Sistema.....	97
3.2.1.2	Configuración de la revista.....	98
3.2.1.3	Crear usuario	100
3.2.1.4	Modificar usuario.....	101
3.2.1.5	Deshabilitar usuario.....	103

3.3	IMPLANTACIÓN	105
3.3.1	REQUISITOS DEL SISTEMA	105
3.3.1.1	Revista	105
3.3.1.2	Base de datos.....	105
3.3.1.3	Triplify.....	105
3.3.1.4	Punto SPARQL Virtuoso	105
3.3.1.5	Front end SPARQL.....	106
3.3.2	PLAN DE INSTALACIÓN.....	106
3.3.2.1	Preparación De Software Requerido	106
3.3.2.2	Instalación Y Configuración De Software Requerido	107
CAPITULO 4 ANÁLISIS DE RESULTADOS.....		108
4.1.1	OBTENCIÓN DE INFORMACIÓN.....	108
4.1.2	EVALUACIÓN DE INFORMACIÓN OBTENIDA.....	109
4.1.3	INTERPRETACIÓN DE LOS RESULTADOS.....	110
CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES.....		112
5.1	CONCLUSIONES	112
5.2	RECOMENDACIONES.....	113
BIBLIOGRAFIA		114
GLOSARIO.....		116

FIGURAS

ILUSTRACIÓN 1 - RUP.....	36
TABLA 1 - ACTORES DE NEGOCIO	43
ILUSTRACIÓN 2 – CUN01 (CASO DE USO DEL NEGOCIO) ADMINISTRAR REVISTA.....	45
TABLA 2 – CUN01 ADMINISTRAR REVISTA.....	45
ILUSTRACIÓN 3 – CUN02 REVISAR ARTÍCULO.....	46
TABLA 3 – CUN02 REVISAR ARTÍCULO	46
ILUSTRACIÓN 4 – CUN03 CORREGIR ESTILO	48
TABLA 4 – CUN03 CORREGIR DE ESTILO	48
ILUSTRACIÓN 5 – CUN04 PUBLICAR EDICIÓN	49
TABLA 5 – CUN04 PUBLICAR EDICIÓN	50
TABLA 6 - CASOS DE USO DEL SISTEMA.....	51
ILUSTRACIÓN 6 - CUS01 ACCEDER AL SISTEMA	52
TABLA 7 - CUS01 ACCEDER AL SISTEMA	52
ILUSTRACIÓN 7 - CUS02 CONFIGURAR REVISTA	53
TABLA 8 - CUS02 CONFIGURAR REVISTA	54
ILUSTRACIÓN 8 – CUS03 ADMINISTRAR USUARIOS.....	55
TABLA 9 - CUS03 ADMINISTRAR USUARIOS.....	55
ILUSTRACIÓN 9 - CUS04 NOTIFICAR USUARIOS.....	56
TABLA 10 - CUS04 NOTIFICAR USUARIOS	56
ILUSTRACIÓN 10 – CUS05 ENTREGAR ARTÍCULO.....	57
TABLA 11 - CUS05 ENTREGAR ARTÍCULO.....	58
ILUSTRACIÓN 11 - CUS06 REVISAR ARTÍCULO	59
TABLA 12 - CUS06 REVISAR ARTÍCULO.....	59
ILUSTRACIÓN 12 – CUS07 CORREGIR ESTILO.....	61
TABLA 13 - CUS07 CORREGIR ESTILO.....	61
ILUSTRACIÓN 13 - CUS08 EDITAR MAQUETACIÓN	63
TABLA 14 - CUS08 EDITAR MAQUETACIÓN	63
ILUSTRACIÓN 14 - CUS09 PUBLICAR EDICIÓN	64
TABLA 15 - CUS09 PUBLICAR EDICIÓN.....	65
ILUSTRACIÓN 15 – CUS10 PUBLICAR DATOS SEMÁNTICOS.....	66
TABLA 16 - CUS10 PUBLICAR DATOS SEMÁNTICOS	66
ILUSTRACIÓN 16 - CLASES CORE	70
ILUSTRACIÓN 17 - CLASES HANDLER	71
ILUSTRACION 18 - CLASES DAO.....	72

ILUSTRACIÓN 19 - CLASES DE FORMULARIO	74
ILUSTRACIÓN 20 - CLASES ADMINISTRATIVAS.....	76
ILUSTRACIÓN 21 - MODELO DE BASE DE DATOS	77
ILUSTRACIÓN 22 - INTERFAZ PRINCIPAL	78
ILUSTRACIÓN 23 – INTERFAZ MENÚ ADMINISTRADOR.....	79
ILUSTRACIÓN 24 – INTERFAZ MENÚ DE CONFIGURACIÓN.....	80
ILUSTRACIÓN 25 - INTERFAZ DE DETALLE DE CONFIGURACIÓN.....	81
ILUSTRACIÓN 26 – INTERFAZ PRINCIPAL DE CONFIGURACIÓN DE USUARIOS	82
ILUSTRACIÓN 27 – INTERFAZ DE EDICIÓN DE USUARIO	83
ILUSTRACIÓN 28 - INTERFAZ DE CREACIÓN DE USUARIO.....	84
ILUSTRACIÓN 29 - MODELO VISTA CONTROLADOR	87
ILUSTRACIÓN 30 - DIAGRAMA DE DESPLIEGUE	89
ILUSTRACIÓN 31 - DIAGRAMA DE DESPLIEGUE PUNTO SPARQL.....	89
ILUSTRACIÓN 32 - DIAGRAMA DE COMPONENTES OPEN JOURNAL SYSTEMS.....	90
ILUSTRACIÓN 33- DIAGRAMA DE COMPONENTES SPARQL.....	90
TABLA 17 - COMPONENTE NAVEGADOR WEB.....	91
TABLA 18 - COMPONENTE SERVIDOR WEB	92
TABLA 19 - COMPONENTE OPEN JOURNAL SYSTEMS	92
TABLA 20 - COMPONENTE DE BASE DE DATOS	93
TABLA 21 - COMPONENTE SISTEMA DE ARCHIVOS DE ALMACENAMIENTO	93
TABLA 22 - COMPONENTE PLUGIN TRIPLIFY	94
TABLA 23 - COMPONENTE SERVIDOR SPARQL	95
TABLA 24 - SERVIDOR WEB TOMCAT.....	95
TABLA 25 - FRONT END PUBBY	96
TABLA 26 - CASO DE PRUEBA INGRESO AL SISTEMA	97
TABLA 27 - CASO DE PRUEBA CONFIGURACIÓN DE LA REVISTA	98
TABLA 28 - CASO DE PRUEBA CREAR USUARIO	100
TABLA 29 - CASO DE PRUEBA MODIFICAR USUARIO	101
TABLA 30 - CASO DE PRUEBA DESHABILITAR USUARIO.....	103
TABLA 31 – MUESTRA DE ENCUESTA.....	108
TABLA 32 - RESULTADOS ENCUESTA DE EVALUACIÓN	109

RESUMEN

El presente proyecto permite hacer la implantación de un sistema que administre la Revista Politécnica, además de la publicación de los datos semánticos correspondientes en un punto SPARQL para consultas de este tipo.

En el primer capítulo se expone la organización de la revista, los tipos de acceso a datos de revistas científicas, los procesos que se manejan en la publicación de artículos por parte de editores, autores y revisores, finalmente se realiza una introducción a la web semántica y sus usos en la publicación de datos.

En el segundo capítulo se analiza y justifica la metodología a utilizar, se enlistan las herramientas que se usan en el proyecto, se definen los requerimientos, casos de uso, y diagramas de diseño del software.

Finalmente, en el capítulo tres, se define la arquitectura, se analizan los componentes del sistema, así también se define las pruebas de cada una de las actividades principales.

Adicionalmente se hace un análisis de resultados mediante una encuesta planteada a los usuarios de la revista.

INTRODUCCIÓN

En la época actual es de suma importancia que el conocimiento sea universal y sea accesible mediante varias fuentes de comunicación, por ejemplo, desde hace mucho tiempo existen ya las bibliotecas en donde la gente podía obtener mediante escritos de otras personas los conocimientos necesarios o información que estuviere buscando.

Sin embargo, la divulgación de conocimientos en un inicio se veía fuertemente mermada a aspectos de orden geográfico, por ejemplo, la información para desarrollo de embarcaciones o información de producción de cultivo. En el inicio de los tiempos tecnológicos tan solo podía hacerse empíricamente a menos que se tuviere acceso a las fuentes tecnológicas en cada uno de los países donde estos se desarrollasen.

Con el paso del tiempo el conocimiento ha ido evolucionando, se empezó a divulgar libros y con el crecimiento de las bibliotecas este conocimiento se vio rápidamente propagado.

A pesar de esto, es con el uso de internet como herramienta donde la gente en la época actual puede tener el conocimiento al alcance de su mano tan solo con una búsqueda en internet, el usuario obtiene el conocimiento rápidamente, mismo que en otras épocas hubiese sido muy difícil de encontrar.

Esta información, divulgada en internet, a pesar de proporcionar una ayuda inmensa al fortalecimiento del conocimiento, se encuentra generalmente de una manera muy desorganizada, es decir, dispersa, suelta, sin seguir una temática, una estructura, en varios idiomas, etc. y además esta información no siempre se la puede encontrar libre, muchas veces está restringida a ciertos usuarios o a algún tipo de pago.

Frente a este problema han surgido alternativas para tratar que la información llegue todo el que la busque y de una manera más organizada.

En el presente proyecto analiza la Implementación de una Revista Científica como alternativa de difusión de información mediante *Open Access* o *Acceso Abierto* que es la práctica que promueve el libre acceso a revistas científicas.

Así también se desarrolla la implementación de esta misma información en la web semántica, en donde esta es desplegada y de libre acceso de una manera más organizada e inteligente, de tal modo que no solo sea entendible para el ser humano sino también para las máquinas, por medio de las cuales se pueden hacer búsquedas más específicas y mejor detalladas de la información que se necesite.

CAPITULO 1 : MARCO TEÓRICO

1.1 DESCRIPCIÓN DEL PROBLEMA

Actualmente existe una revista científica en la Escuela Politécnica Nacional, esta revista ha permanecido vigente durante algún tiempo, ha tenido su estructura y funcionamiento de una manera adecuada. Sin embargo, no ha tenido oportunidad de que sus procesos se lleven a cabo mediante un software de control.

Es por esta razón que el presente proyecto pretende que la Revista pueda existir en un formato digital, dando lugar a que el proceso de presentación, revisión y publicación sea automatizado.

A su vez lo que se busca es que esta información pueda ser consultada e indexada de manera global, dándose así a conocer de manera universal y promoviendo la divulgación del conocimiento científico de las distintas investigaciones realizadas hasta la fecha y posteriores.

Por tal motivo es necesario la implementación de un sistema que pueda abarcar todo el proceso y facilitar el acceso a los diferentes actores del mismo, para que así se culmine con ésta de una manera más efectiva, y rápida.

Además de la implementación del sistema en cuestión, es necesario que estos artículos puedan ser accedidos por aplicaciones de web semántica, que cada vez se encuentra con mayor uso en el campo de bibliotecas digitales y Revistas científicas, para esto también es necesario la integración de datos, herramientas y servicios que permitirán que la información publicada por la Revista Politécnica sea accedida a través de dicha tecnología.

1.2 REVISTAS CIENTÍFICAS DIGITALES

La extensión del conocimiento es fundamental para el desarrollo científico universal, durante años el despliegue del mismo ha fortalecido la tecnología y el crecimiento de la ciencia. [4]

Un ámbito importante es la investigación, a través de los años investigaciones científicas han servido para que futuras generaciones o generaciones contemporáneas puedan hacer aportes significativos a la ciencia por medio de ese conocimiento.

Un aspecto básico para la publicación de la investigación han sido las revistas científicas, dichas revistas empezaron a surgir alrededor del siglo XVIII como publicaciones breves de resultados investigativos, cuyo número fue incrementándose con el tiempo.

Hoy en día se sabe que el impacto que logra internet con la investigación científica es fundamental, puesto que las revistas científicas se valen de este medio de divulgación; actualmente existen varias de ellas situadas en la red en donde cualquier persona puede acceder a gran número de artículos en constante publicación y con un sinnúmero de investigaciones.

Las revistas científicas digitales consisten generalmente en una organización digital, mediante un software o aplicación en donde los artículos son sometidos a *revisión por pares*. Sus gastos consisten en este tipo de revisión, la preparación del artículo y el alojamiento web, a veces las revistas científicas tienen un subsidio de la universidad o la sociedad profesional, eso quiere decir que existe la posibilidad de que estas revistas acepten una paga por derecho de artículos.

1.2.1 REVISTAS CIENTÍFICAS Y ACCESO ABIERTO

Las revistas científicas digitales se fundamentan generalmente sobre políticas de Acceso abierto u *Open Access* (ver sección 1.4.1), esto quiere decir que sus contenidos son abiertos, de libre difusión, es decir, la información que las revistas

científicas proveen está en la red disponible sin importar el tiempo o lugar de donde se la busque siempre y cuando se tenga acceso a internet [4].

1.2.2 VENTAJAS

La publicación de Revistas científicas electrónicas ha proporcionado a las instituciones y personas ávidas de conocimiento algunas ventajas, entre las cuales se pueden destacar las siguientes [9]:

- Aumento en la velocidad de comunicación
- Diseminación y absorción del conocimiento
- Nuevas formas de encontrar y analizar información.
- Reduce el espacio físico usado en estanterías de bibliotecas
- Disminución del proceso de publicación y lo hace más barato
- Mejora la calidad de publicación a través de la retroalimentación de los lectores por medio de comentarios.
- Reduce costos (impresión, entrega y recepción)

1.2.3 RIESGOS

Así como la publicación electrónica es una gran ventaja, tiene algunos riesgos, entre los cuales se puede enumerar los siguientes: [9]

- Debido a la alta velocidad con que son desarrollados los artículos en otras revistas, la presión por la "competencia electrónica" y su rapidez en el proceso de publicación, se puede dar lugar a que su control de calidad sea tomado a la ligera.
- Al no tener un control de calidad adecuado la información puede quedar distorsionada.
- Se puede dar lugar al plagio o robo de ideas de información alojada en lugares poco conocidos en un medio local.

1.2.4 DOI

El sistema DOI (Digital Object Identifier – Identificador de Objeto Digital) es un sistema administrado para mantener la persistencia de la identificación de un contenido, puede ser usado para identificar entidades físicas, digitales o abstractas. [13]

El DOI de un documento permanece constante durante el tiempo de vida de este mientras que su localización y otro tipo de metadato pueda cambiar, por eso es aconsejable que al referirse a un documento en línea el administrador de la publicación cambie solo la URL de conexión actualizando tan solo sus metadatos.

1.2.5 DUBLIN CORE

Dublin Core es una iniciativa para crear un “catálogo de tarjetas de biblioteca” digital. Este estándar está constituido de metadatos que ofrecen una información catalogada extensa y mejorada para documentos digitales. [6]

Los elementos de metadatos más comúnmente utilizados en Dublin Core son:

- *Title* (Título)
- *Creator* (Autor)
- *Subject* (Tema o Título referente del documento)
- *Description* (Descripción del contenido del documento)
- *Publisher* (El responsable de publicar el recurso)
- *Date* (La fecha que el recurso estuvo disponible por primera vez)
- *Type* (La categoría del contenido)
- *Format* (Formato del presentación del recurso)
- *Identifier* (Identificador numérico del recurso)
- *Source* (Origen de donde proviene el documento)
- *Language* (Idioma en el que está el contenido)
- *Relation* (Como está relacionado con otros recursos)
- *Coverage* (Donde está el recurso ubicado físicamente)
- *Rights* (declaración de copyright)

1.3 ETAPAS DEL PROCESO DE PUBLICACIÓN

La publicación de revistas tiene un proceso de publicación, basado en mejores prácticas, este proceso consiste en una serie de etapas en donde el artículo es creado, editado, revisado y controlado para el consumo del lector. A continuación se enlistan dichas etapas con su respectiva descripción general: [19] [20]

- **Investigación y Entrega:** El autor escribe un artículo en donde describe los resultados de su investigación
- **Revisión y Aprobación:** El autor entrega el artículo a la revista científica, en este medio el artículo es sometido a revisión por pares, es aceptado, rechazado o retornado para revisión.
- **Procesos de la Edición:** La revista lleva los procesos de corrección y maquetación.
- **Publicación:** Una vez se han completado los procesos previos, se lleva a cabo la difusión del artículo a través del sitio web correspondiente. La publicación también se da a nivel de un punto SPARQL, mediante herramientas especializadas en extracción de datos RDF y Publicación.

1.3.1 INVESTIGACIÓN Y ENTREGA

Una revista científica se basa en la investigación de sus autores, es por eso que esta debe tener varios de autores que en base a su experiencia e investigación puedan aportar los artículos necesarios para la divulgación del conocimiento.

El software que use la revista debe aportar con acceso y facilidades para que los autores puedan subir dichos artículos a su servidor. Es recomendable que la aplicación que maneje la revista tenga una opción para suscribirse o solicite que haya un método de registro para los autores.

El ingreso de autores puede hacerse ya sea desde la página web o mediante algún método operativo, como el envío de correo para solicitar un usuario o clave, lo recomendable es que este método sea lo más automatizado posible.

Una vez el autor tiene su usuario es necesario que este haga el documento en donde esté contenido el artículo, este documento debe estar sujeto a las normas de la revista, por ejemplo:

- El archivo donde se ha realizado el artículo debe ser en formato docx (Microsoft Word 2007+) o también podría ser formato *ODT (Open Document)*.
- El artículo no debe ser publicado previamente en otra revista.
- Normas respecto al interlineado y tipo de letra.
- Reglas para indicar registros bibliográficos
- Normas adicionales respecto al formato del documento apta para la revisión por pares

Cuando el autor se ha asegurado que el formato del documento es el adecuado, este debe subirlo a través del sistema.

La aplicación a su vez debe avisar al encargado que se ha ingresado un nuevo artículo.

1.3.2 REVISIÓN Y APROBACIÓN

Cuando se ha entregado un documento es necesario que los encargados de la edición y corrección sepan que hay un nuevo artículo, para que estos coordinen como se va a realizar la revisión.

El editor encargado debe asignar 3 revisores para que estos realicen el proceso de revisión por pares respectivo.

Estos revisores pueden ser ingresados previamente al sistema, donde los autores pueden ser habilitados para acceder también al mismo como revisores

Esta etapa consiste en la aprobación o desaprobación del artículo según los siguientes criterios de los 3 revisores:

- Si dos de los tres revisores mínimo aprueban el artículo, este pasa a la siguiente etapa.
- Existe la posibilidad de que no se apruebe el artículo en la primera revisión, sin embargo, este artículo se califica con distintos criterios, se puede mandar a hacer las correcciones respectivas y el autor debe volver a entregar el mismo para su posterior revisión.
- La última posibilidad es que se desestime el mismo, el artículo se puede enviar otra vez.
- Si el artículo ha sido mandado a corregir por tercera vez y no se aprueba, este se desecha y no será apto para más envíos.

En cada caso el sistema debe contar con las opciones que permitan este flujo de datos. La comunicación se la realiza a través de correo electrónico (administrado por el propio sistema de la revista), y el sistema almacena cada artículo con cada corrección realizada.

El editor puede ver cada uno de los artículos enviados luego de cada corrección, a su vez también se encarga de, en base a las respuestas de los revisores, tomar una decisión final, donde el artículo tiene las siguientes opciones:

- pasa a la siguiente etapa.
- se envía para una nueva revisión
- se desecha.

Una vez tomada esta decisión el artículo pasa a los procesos de edición: corrección, y maquetación.

1.3.3 PROCESOS DE EDICIÓN

Cuando un artículo es admitido, se lo debe pasar por procesos de edición, el primer paso es hacer la corrección de estilo.

1.3.3.1 Corrección de estilo

El autor se ha encargado de investigar y plasmar sus ideas en un documento, a criterio de él es un documento válido, puesto que contiene el resultado de su investigación o ideas en el mismo, sin embargo, un autor no es necesariamente un experto en la materia gramatical, su redacción puede estar llena de errores gramaticales o semánticos, aun así el artículo podría contener términos científicos que tal vez los usuarios no podrían entender. [20][8]

Es por este problema que es necesaria la intervención de personas que sean en realidad expertas en escritura y redacción. La buena redacción del artículo es importante en todo el proceso, asegura que el consumo del artículo por parte de los lectores va a ser óptimo y es a la vez parte de la imagen que la revista da al medio.

Las personas encargadas de la corrección de estilos deberán ser avisadas por el sistema que existe un nuevo archivo, el mismo que ha sido aprobado por la decisión editorial para corregirlo.

Se consideran 3 pasos básicos para la corrección de estilos:

Corrección Inicial:

Como su nombre lo indica, esta corrección se la hace preliminarmente al archivo aprobado por el editor. El corrector de estilo deberá ver todas las fallas gramaticales y semánticas que se pudieren presentar.

Es aconsejable que se maneje un procesador de texto con control de cambios para poder diferenciar las correcciones, sin embargo, esto debe ser aceptado como normal general por todas las personas que intervengan en estos procesos.

El editor puede efectuar este proceso si así lo considere conveniente o designar a un corrector de estilo que lo haga.

Corrección del Autor:

Puesto que el corrector de estilo no es el creador del artículo, este podría cometer un error de cambiar la idea que puede tener el autor o en general podría hacer algún cambio que no es del agrado del escritor, el artículo debe pasar por las manos del autor nuevamente para que este se encargue de, en base a los avisos o mensajes del corrector de estilo, hacer los cambios respectivos al documento.

Corrección final

En esta fase solo se verifica que el autor haya hecho los cambios oportunos al artículo en cuestión, una vez verificado esto se puede seguir a la siguiente fase.

Como se puede apreciar, el proceso de corrección de estilos es minucioso, el sistema debe tener la opción de poder subir los cambios tanto para las 3 etapas del mismo.

1.3.3.2 Edición de maquetación

El editor de maquetación transforma las versiones de la corrección de estilos en galerías de *HTML* y *PDF*. [20]

En vista que hasta este punto se ha trabajado solamente con procesadores de texto, es necesario que el editor de maquetación diseñe el contenido *HTML* a publicar así como también el formato final *PDF*. Por lo general tan solo es necesario subir un archivo final *PDF*, pero en ocasiones se suele subir también una versión *HTML*. Y archivos adicionales.

Se deberán ingresar las últimas versiones en el sistema, mismas que serán las versiones finales.

1.3.4 PUBLICACIÓN

La publicación de una revista es la fase final para su consumo, para llegar a esta etapa es importante haber pasado por los respectivos procesos de entrega, revisión, edición y diseño, la omisión de alguno de los tres últimos estos procesos puede dar lugar a una falla, ya sea de estilo o propiamente de fondo. [20]

El control de calidad del artículo por tanto es algo que en toda revista se debe tener mucho cuidado y contar con el personal adecuado que tenga los conocimientos, experiencia y tiempo necesarios para cada una de las tareas en este proceso

Finalmente, contando ya con los archivos de maquetación se debe indicar en que número se debe publicar el artículo, aun así el editor puede editar las últimas versiones incluso luego a su publicación.

Las publicaciones deben tener un tiempo establecido, es decir por ejemplo cada 6 meses, cada 3 meses, o cada año y se debe hacer el respectivo llamado a la recepción de nuevos artículos, ya sea por mail o en publicación en el sitio oficial de la revista.

1.4 PUBLICACIÓN DE DATOS ABIERTOS.

1.4.1 ACCESO ABIERTO (*OPEN ACCESS*)

El Acceso abierto se lo podría especificar como *“La información que está disponible en línea para ser leída de manera libre por cualquier persona en cualquier momento y desde donde sea, mientras se tenga acceso a internet”*. [4]

Las Revistas Científicas entregan los resultados de su investigación en la forma de artículos revisados por pares para que los demás puedan sacar provecho de este conocimiento. Es así como la sociedad se beneficia de este principio, donde se intercambia información de manera libre.

El *Acceso Abierto* es un principio que se lo aprovecha de gran manera a través de investigadores y bibliotecas, cuya misión es diseminar el conocimiento no solo de la manera clásica sino a través del internet. Se define el *Acceso Abierto* como una fuente completa de conocimiento humano y herencia cultural que ha sido aprobado por la comunidad científica.

A continuación, se define las versiones de Acceso abierto de las 3 iniciativas destacadas

1.4.2 INICIATIVAS

Las causas para el origen del movimiento de *Acceso Abierto* no solo fueron aspectos de costos debido a la subida de precios de revistas científicas en la década de los 80, también fue el control de los derechos de copyright sobre trabajos que han sido publicados por parte de las editoriales y constaban con políticas de acceso y distribución. [4]

El avance tecnológico y la resolución de la comunidad científica ante estos problemas y valiéndose de una vía de comunicación como internet han originado el movimiento *Acceso Abierto* a través de 3 declaraciones:

- Budapest
- Bushida
- Berlín

1.4.2.1 Budapest

Viene de un encuentro que sostuvo la *Open Society Institute* en diciembre de 2001, donde su resultado se lo obtuvo en febrero de 2002, se define al Acceso abierto de la siguiente manera:

“Se entiende por acceso abierto a la literatura que se encuentra en libremente disponible en el internet permitiendo a los usuarios leer, descargar, copiar, distribuir, imprimir, buscar o enlazar textos completos de estos artículos sin pedir

permiso previo del editor o el autor. Para poder usarlo con cualquier propósito legal sin ninguna barrera financiera, legal o técnica más que las que sean necesarias para ganar acceso al internet, la única imposición en cuanto a la reproducción y distribución y el único papel del copyright en este ámbito debe ser el de dar a control a los autores sobre la integridad de su trabajo y el derecho a ser reconocidos y citados de manera adecuada”.

1.4.2.2 Betsheda

Esta declaración fue lanzada en junio de 2003, en un encuentro de un día con dos docenas de participantes.

Según esta declaración se define una publicación Acceso abierto de la siguiente manera:

“El autor otorga a todos los usuarios un derecho perpetuo, mundial, irrevocable y libre de acceso y una licencia para copiar, usar, distribuir, transmitir y mostrar el trabajo públicamente y para hacer trabajos de distribución derivativa en cualquier medio digital para cualquier propósito responsable sujeto a la propia atribución de propiedad”.

También se definió en este encuentro que una versión completa del trabajo y todos los materiales suplementarios incluida una copia del permiso establecido anteriormente en un formato electrónico debe ser ingresado inmediatamente en al menos un repositorio online.

1.4.2.3 Berlín

La definición es prácticamente idéntica a la de Budapest a excepción de la necesidad de un repositorio, es decir toma los términos de las dos declaraciones anteriores, establece dos condiciones para acceso abierto:

1) El (los) autor(es) y depositario(s) de la propiedad intelectual de tales contribuciones deben garantizar a todos los usuarios por igual, el derecho gratuito, irrevocable y mundial de acceder a un trabajo erudito; lo mismo que licencia para

copiarlo, usarlo, distribuirlo, transmitirlo y exhibirlo públicamente; y para hacer y distribuir trabajos derivados en cualquier medio digital, para cualquier propósito responsable. Todo ello está sujeto al reconocimiento apropiado de autoría (los estándares de la comunidad continuarán proveyendo los mecanismos para hacer cumplir el reconocimiento apropiado y uso responsable de las obras publicadas, como ahora se hace), lo mismo que al derecho de efectuar copias impresas en pequeño número para su uso personal.

2) Una versión completa del trabajo y todos sus materiales complementarios, que incluya una copia del permiso del que se habla arriba, en un conveniente formato electrónico estándar, se deposita (y así es publicado) en por lo menos un repositorio electrónico que utilice estándares técnicos aceptables (tales como las definiciones del Acceso Abierto), que sea apoyado y mantenido por una institución académica, sociedad erudita, agencia gubernamental, o bien una organización establecida que busque implementar el acceso abierto, distribución irrestricta, interoperabilidad y capacidad archivística a largo plazo.

1.4.3 OAI PMH

La idea de tener archivos abiertos es permitir el acceso a los recursos en la web por medio de repositorios que, intercomunicándose unos con otros, permitan el intercambio de metadatos, su almacenamiento y publicación. Surgió la necesidad de una solución de interoperabilidad de bajo nivel para el acceso entre repositorios que da lugar a la creación de la *Open Archives Initiative* (OAI). La misma que se puede definir de la siguiente manera: [3] [12]

“La Iniciativa de Archivos Abiertos desarrolla y promueve estándares de interoperabilidad que apuntan a facilitar la diseminación de contenido eficiente. la OAI tiene sus raíces sobre el acceso abierto y los movimientos de repositorios institucionales. El apoyo continuo de este trabajo sigue siendo una piedra angular del programa de Archivos Abiertos. Con el tiempo, sin embargo, el trabajo de la OAI se ha expandido para promover un amplio acceso a los recursos digitales para eScholarship, eLearning y eScience” [12]

La sociedad se beneficia de la estrategia de archivos abiertos. Internet y la gran cantidad de documentos en formato digital han hecho crecer el número de usuarios potenciales de muchos repositorios de información. Los documentos pueden ser accesibles y además utilizados con distintos objetivos de los que motivaron inicialmente la creación de los repositorios.

En otro aspecto, el poder acceder a varios repositorios permite la construcción de nuevos servicios que pueden dar soporte a las necesidades de los usuarios. También existe la posibilidad de reducción de costos gracias a los nuevos modelos de procesos de comunicación académica que apoyan a esta estrategia.

El Protocolo OAI para la Recolección de Metadata (*OAI-PMH*) define un proceso de recolección de datos que contiene información de los metadatos de los repositorios. Este protocolo ofrece a las fuentes de datos una opción para publicar a sus metadatos que se basan en estándares como HTTP o XML.

El uso del protocolo es importante en la estrategia de *Acceso Abierto*, siendo el beneficio principal el de servir al nuevo modelo de comunicación académica. A pesar de que el protocolo OAI-PMH es sencillo, la construcción del mismo enfocado a la satisfacción del usuario final sigue siendo compleja. Sin embargo, este protocolo se proyecta para ser una estructura importante en la web tanto como lo es hoy el protocolo HTTP.

1.4.4 TIPOS DE OPEN ACCESS

Entre los tipos principales de Acceso abierto se tiene [4]:

- Green OA (Acceso abierto verde)
- Gold OA (Acceso abierto dorado)
- Gratis OA (Acceso abierto gratis)
- Libre OA (Acceso abierto libre)

1.4.4.1 Acceso abierto verde (Green OA)

Consiste en artículos revisados por pares depositados libremente en repositorios institucionales donde la mayoría de estos utiliza protocolo OAI-PMH, Los autores pueden depositar sus artículos sin necesidad de ninguna autorización previa. Estos artículos denominados '*preprint*' (pueden haber sido revisados por pares, pero no están publicados aún) son depositados en repositorios institucionales. Si un autor transfiere el copyright del artículo a un publicador, este solo afecta a la versión publicada (no a la depositada en el repositorio)

Cabe destacar que este tipo de acceso abierto no requiere de consentimientos ni cambios de política de publicadores, además no posee versión final de los artículos publicados.

1.4.4.2 Acceso abierto dorado (Gold OA)

El *Acceso abierto dorado* consiste en la publicación de artículos previamente revisados por pares, de manera libre y con acceso ilimitado, sin ningún tipo de paga o registro, sin embargo los publicadores pueden cobrar tarifas de procesamiento al autor, este tipo de acceso abierto permite que también se pueda hacer versiones impresas para su venta, sin que esto signifique que se viole el carácter de acceso abierto mientras haya acceso libre en línea, a diferencia del acceso abierto verde este provee un acceso inmediato a artículos finales.

1.4.4.3 Acceso abierto gratis

El acceso abierto gratis considera que cualquiera puede leer los artículos sin costo. La existencia de este tipo de acceso abierto y la existencia de las declaraciones principales requieren que exista un conjunto de condiciones, estas condiciones son difíciles de cumplir para algunas revistas científicas digitales, ya que reducen la disponibilidad de los artículos para lectura libre, que es un objetivo clave para la mayoría de investigadores y usuarios. Todos estos problemas son cubiertos por el Acceso abierto gratis.

1.4.4.4 Acceso abierto libre

A diferencia del acceso abierto gratis, el acceso abierto libre declina por lo menos algunos permisos para aquellos que deseen utilizar los artículos para tareas más allá de leer y copiarlos, A cambio la única restricción para usarlos es la citación de la fuente y la atribución de la misma.

1.4.4.5 Relación entre los tipos de acceso abierto

Actualmente la Mayoría de Información de acceso abierto, sin importar que sea *verde, dorada o gratis* se encuentra entre *Gratis y Libre*.

Por ejemplo, si se prepara un libro o cualquier producto comercial, el autor de este texto debería poder utilizar el acceso abierto libre, pero probablemente este autor deba pedir autorización para usar artículos de acceso abierto gratis.

1.4.5 BENEFICIOS DEL OPEN ACCESS SOBRE LAS BIBLIOTECAS

Las bibliotecas han sido siempre fuente de conocimiento y han ayudado a la comunidad investigativa a lo largo del tiempo, sin embargo, su uso, se ha visto limitado a factores como su stock, su actualización a nuevas ediciones, falta de ejemplares, etc. Por otro lado, del acceso abierto en cambio ayuda a la comunidad a superar estos problemas comunes en las bibliotecas, a continuación, se detalla cada uno de manera específica:

- La comunidad puede tomar ventaja del trabajo más actual en cualquier campo sin necesidad de que este conocimiento necesariamente sea añadido a una biblioteca en particular.
- Las bibliotecas no siempre pueden proveer de artículos científicos que pueden ser útiles a la comunidad. El acceso abierto por supuesto no tiene esa limitación.
- El acceso abierto ayuda a que los profesionales accedan a información necesaria para su uso con un mínimo esfuerzo y con mayor velocidad,

permitiendo a estos estar al día en su campo, lo que permite no solo que se beneficien estos, sino la comunidad entera.

- El acceso abierto permite que haya libertad en cuanto a investigación de contenido y en cuanto a profundidad del mismo, dando a cada persona la oportunidad de poder investigar tanto cuanto y como quiera.
- Permite que sus contenidos sean accedidos no solo por gente de instituciones que puedan tener a su disposición una biblioteca, centros de investigación, laboratorios. Permite el acceso no solo a personas pertenecientes a instituciones académicas o de investigación, posibilita su uso a cualquier persona que tenga acceso a internet.
- El presupuesto bibliotecario está siendo cada vez más limitado debido a los costos de publicación e impresión de revistas científicas, particularmente en los campos de ciencias como ingeniería, tecnología, medicina, que crecen a pasos agigantados y en donde comprar ejemplares, que tienen una caducidad rápida no es una buena inversión. El acceso abierto permite estar al día en publicaciones tecnológicas que pueden quedar obsoletas en años o incluso meses.

1.4.6 WEB SEMÁNTICA

La web semántica o web 3.0 es una tecnología web, una forma de enlazar información entre sistemas o entidades que permite relaciones que son expresadas de una manera auto descriptiva y está disponible en el universo web. [7]

Esta información marca un cambio en el pensamiento de publicación de datos, de manera que la información web ya no es tan solo publicable en formato de lectura humana (*HTML*), sino también esta información está enlazada mediante un formato tal que las máquinas puedan entender. Esto Significa que los dispositivos en cierto punto utilizando este formato pueden obtener búsquedas avanzadas entendiéndolas en todo su contexto.

Se puede apreciar lo descrito en el siguiente ejemplo: Suponiendo que un usuario desee buscar un artículo de *petróleos* que le pertenezca al autor *Juan Pérez*, la manera convencional de búsqueda es a través de un motor de búsqueda, por ejemplo, *Google*, lo que el usuario ingresará en el buscador serán los términos “Juan Pérez petróleo”, el buscador encontrará resultados que contengan esas palabras, ya sea en texto o en etiquetas de enlace. Sin embargo, el buscador no “sabe” que Juan Pérez es el autor del artículo, este tan solo busca las palabras mencionadas y muestra los sitios que contengan tal información.

Este tipo de búsqueda de información hasta ahora ha ayudado mucho a los usuarios, sin embargo, se aleja de ser efectiva, debido a que no siempre se enlaza de manera adecuada, puede mostrar páginas que contengan *autor(es)* de la ciudad de San *Juan*, de la universidad Julio *Pérez* Maldonado, por poner un ejemplo.

Lo que se busca en la web semántica es dar a entender a los buscadores (máquinas), que el usuario requiere el artículo cuyo título sea referente a *petróleos* y que su autor sea *Juan Pérez*.

Esto hasta hace unos años era imposible, sin embargo, la tecnología ha ido avanzando y se han podido implementar protocolos de búsqueda que ayuden a que las máquinas tengan mejor comunicación con las personas.

1.4.6.1 Datos enlazados (Linked Data)

Se refiere a un conjunto de mejores prácticas para publicar y relacionar la información estructurada en la web, estas prácticas tienen los siguientes principios:
[7] [17]

- Usar URIs como nombres para objetos
- Usar las URIs HTTP para que las personas puedan buscar esos nombres
- Cuando alguien busque una URI, proveer información útil usando estándares (RDF, SPARQL).
- Incluir los links a otras URIs para que se pueda descubrir más información.

Una URI no es más que un conjunto de caracteres utilizados para identificar el nombre de un recurso. Esa identificación permite la interacción con representaciones del recurso sobre la red. El URI más común es la URL, que normalmente se refiere a una dirección web.

La idea de la Linked data es aplicar la estructura general de la web con la idea de compartir información estructurada a una escala global.

El documento web tendrá un identificador único que será su URI; HTTP, como mecanismo de acceso universal, HTML como un formato de contenido.

1.4.6.2 URIs

Para publicar información en la web, los objetos deben ser primeramente identificados, hay cosas cuyas propiedades y relaciones pueden ser descritas sobre datos, que puede incluir documentos web, así como entidades o conceptos abstractos. [7]

Las URIs proveen una manera simple de crear nombres únicos en la web, estas URIs no solo sirven como nombres sino como medio de acceso a información relacionada con el objeto.

Una URI tiene una estructura similar a un URL, pero a diferencia tiene una descripción, es decir identifican un objeto, mientras que una URL es un localizador.

En pocas palabras se debe pensar en las URIs como nombres para las cosas en lugar de direcciones de documentos web.

1.4.6.3 RDF (Resource description framework)

El modelo RDF está diseñado para integrar la representación de información que se origina en múltiples fuentes, el objetivo de este modelo es que sea empleado como una lengua común utilizada para transportar información entre diversos orígenes. [7]

En el modelo RDF la descripción de un recurso está representada por un número de *Triples*. Este Triple está dividido por 3 partes, sujeto, predicado y objeto. Un triple representa una estructura de una oración como puede verse en el siguiente ejemplo:

Juan Pérez tiene una Casa.
 sujeto predicado objeto

El sujeto del triple es una URI identificando el recurso descrito, el objeto puede ser un valor *literal*, que puede ser una cadena de texto, un número o una fecha, en la mitad el predicado, indica que tipo de relación existe, en este caso el predicado es *tiene*, que es como están relacionados, el tipo de dato del predicado es otro recurso, que tiene un URI, estos URI vienen de *vocabularios*, que son colecciones de URIs que se usan para la representación de relaciones entre objetos.

Se pueden distinguir dos tipos de RDF, *literal* y *enlaces RDF*.

Literal:

Son los triples cuyo objeto es un literal, este puede ser un número, una cadena o una fecha, son utilizados para describir las propiedades de los recursos.

Un ejemplo de triple literal es:

<http://example.org/spiderman> <http://xmlns.com/foaf/0.1/name> "Peter Parker"

En donde se puede ver que el objeto es una cadena texto.

Enlaces RDF:

Describe la relación entre 2 recursos, consiste en 3 referencias URI, el URI en el sujeto, el URI del predicado y el URI del objeto, en este caso se establecerá una relación entre un RDF con otro RDF. Por ejemplo:

<http://example.org/#spiderman>

<http://www.perceive.net/schemas/relationship/enemyOf> <http://example.org/#green-goblin>

1.4.6.4 Enlaces entre sujeto y objeto

El cuarto principio de los datos enlazados es configurar enlaces RDF que apunten a otros orígenes RDF. Técnicamente un RDF externo es un triple RDF en el cual el sujeto del triple es una referencia URI a un conjunto de datos, mientras que el objeto y el predicado son cada una referencias a conjuntos de datos externos. Y así sucesivamente, de esta manera es como se relacionan los objetos en la web semántica.

Existe 3 tipos de enlaces RDF.

- **Enlaces de Relación:** apuntan a los objetos relacionados en otras fuentes de datos, por ejemplo, otras personas, lugares, etc. Este tipo de relaciones permiten apuntar a información acerca del lugar de nacimiento o información bibliográfica acerca de las publicaciones que han escrito.
- **Enlaces de Identidad:** Apuntan a otras URIs consideradas alias, en otros orígenes de datos, pero que representan al mismo objeto en la vida real, permiten encontrar mayor información del mismo objeto en otra fuente de datos.
- **Enlaces Tipo Vocabulario:** Apuntan a las definiciones de vocabulario que son usadas para representar información, este tipo de enlaces hacen que la información sea auto descriptiva y permiten a las aplicaciones para datos enlazados entender e integrar estos datos.

1.4.6.5 SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) es un lenguaje de consulta orientado a datos en el que consulta información contenida en grafos RDF, un grafo RDF no es más que un conjunto de triples. Mediante este lenguaje se puede retornar y manipular información almacenada en un RDF.

1.4.6.6 Punto SPARQL

El punto SPARQL es un servicio que permite a usuarios, pueden ser humanos o software hacer consultas a un conjunto de datos RDF mediante lenguaje SPARQL cuyos resultados son retornados en formato procesable.

1.4.7 PROCESO DE PUBLICACIÓN DE LINKED DATA

El proceso de publicación de datos puede variar de acuerdo a las necesidades y criterios del administrador o de la institución a publicarlos.

Para la Revista científica se utiliza el siguiente proceso:

- Tomar en cuenta como origen la base de datos de la revista.
- Extraer los datos necesarios de la base utilizando una herramienta de tal manera que se pueda crear un documento en formato RDF.
- Configurar el Servidor SPARQL
- Subir el archivo RDF generado mediante la interfaz web del servidor SPARQL a un nuevo grafo RDF.
- Configurar un Front end HTML para el punto SPARQL.

1.5 METODOLOGÍA DE DESARROLLO

1.5.1 DEFINICIÓN DE LA METODOLOGÍA

1.5.1.1 Definición de RUP

La metodología RUP (Rational Unified Process) Es un proceso de desarrollo de software que es iterativo, centrado en arquitectura, y dirigido por casos de uso, este proceso de desarrollo es definido y estructurado, define quien, como y cuando realizar las tareas, también muestra una estructura del ciclo de vida del proyecto, mostrando de una manera clara objetivos y puntos de decisión. [10]

La base de RUP se basa en algunos principios fundamentales que soportan el desarrollo iterativo, los mismos que representan el espíritu de RUP; estos principios son los siguientes:

- Atacar los mayores riesgos de manera temprana y continua, o ellos atacarán.
- Asegurarse de dar valor al cliente.
- Permanecer enfocado sobre software ejecutable.
- Acomodar los cambios de manera temprana en el proyecto.
- Tener como línea de base una arquitectura ejecutable desde el inicio.
- Construir el sistema con componentes.
- Trabajar juntos como equipo
- Hacer de la calidad una forma de vida

1.5.1.2 Principios De RUP

El Proceso RUP fue diseñado con técnicas similares a aquellas utilizadas en el diseño de software, este proceso tiene dos estructuras

- **Estructura Dinámica**

La dimensión horizontal representa la estructura dinámica o la dimensión del tiempo en el proceso, muestra el proceso expresado en términos de ciclo, fases e iteraciones que se desarrollan durante el ciclo de vida del proyecto.

- **Estructura Estática**

La dimensión vertical representa la estructura estática del proceso, describe como los elementos del proceso: actividades, disciplinas, artefactos y roles, están lógicamente agrupados en procesos básicos.

A continuación, en la ilustración 1 se pueden apreciar las dos estructuras

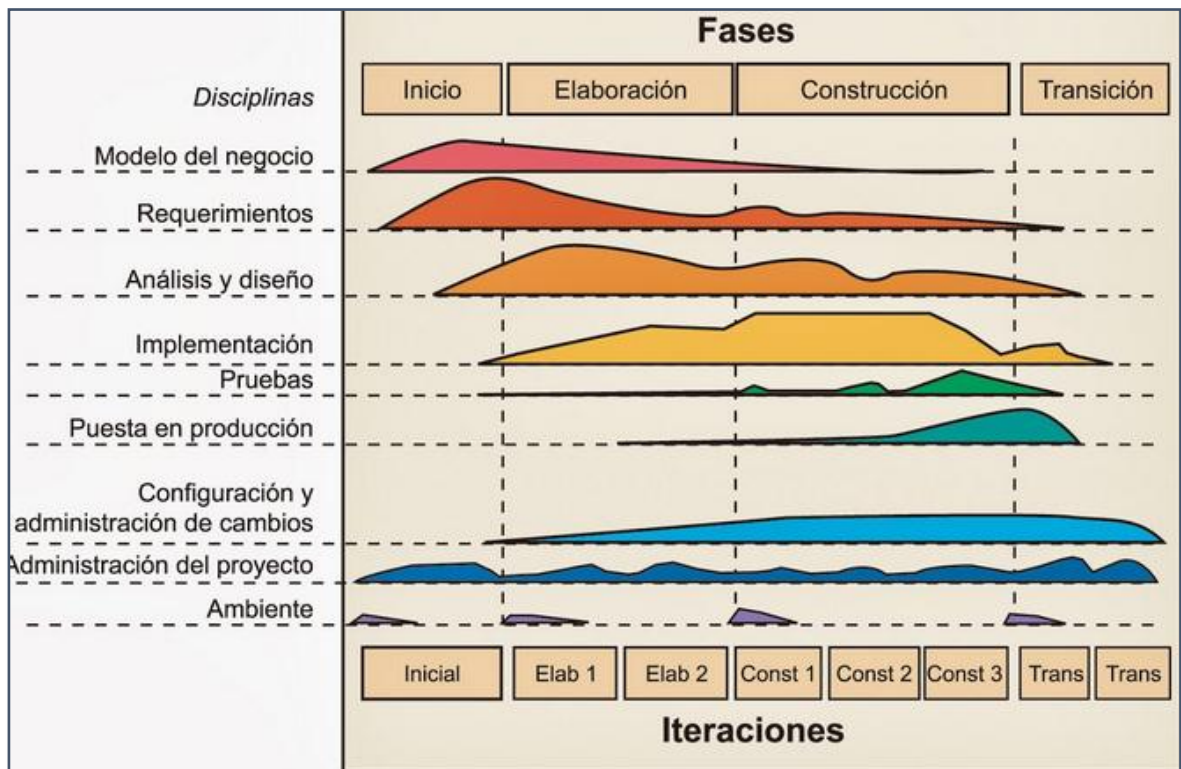


Ilustración 1 - RUP

1.5.1.3 Fases

El propósito de las fases de RUP no es el de partir las actividades por su tipo, sino cumplir objetivos, las fases definen los estados del proyecto, mientras que los estados definen que riesgo se está mitigando. [10]

La fase de Inicio se enfoca en manejar los riesgos del proyecto, se debe hacer la pregunta si es factible o si es conveniente.

La elaboración se la realiza en forma iterativa e incremental, se revisa el alcance a medida que los requerimientos están entendidos de mejor manera, tomando en cuenta los riesgos técnicos.

En la fase de Construcción se desarrollan los diferentes módulos identificando los riesgos lógicos para sacar mejor provecho al trabajo realizado

En la fase de Transición se presta importancia a los riesgos asociados con los riesgos logísticos de la instalación del producto.

Estos máximos hitos son puntos de decisión claves para el proyecto donde las decisiones deben hacerse acerca de la continuación del proyecto, su alcance, estrategia de financiación, entrega o planificación de tiempo.

1.5.1.4 Justificación

La metodología cumple con un proceso conveniente para el presente proyecto, dado que requiere de una gestión capaz de administrar un proceso complejo en varias etapas. Es necesario que el presente proyecto se vaya trabajando en las diferentes fases para tener una respuesta de los usuarios de la revista. Y así poder hacer una implementación de la indexación semántica de datos.

1.5.2 HERRAMIENTAS A UTILIZAR

En el presente proyecto se tiene contemplado la implementación de una revista científica digital para la administración de la Revista Politécnica, con base en los procesos de la revista científica antes mencionados se debe elegir una herramienta que pueda soportar las características de dicho proceso.

Una herramienta que presenta una estructura acorde a lo estudiado es el CMS (Content Management Systems), *Open Journal Systems*. Este aplicativo permite la configuración de un sitio web que soporta la recepción, edición y publicación de artículos.

1.5.2.1 Open Journal Systems

Este CMS es un sistema de publicación y administración para revistas científicas, su desarrollo ha sido concretado por el PKP (*Public Knowledge Project*), entre sus características están: [20]

- Su instalación se realiza en un servidor local, lo que permite el acceso directo a sus recursos facilitando su configuración.

- Los usuarios asignados pueden configurar a su disposición los requerimientos, las secciones y sobre todo los procesos.
- El aplicativo cuenta con la opción de envío de correo electrónico, donde los usuarios de distinto tipo pueden comunicarse con los demás, esta herramienta sirve para la agilización del flujo de los distintos procesos de la revista.
- El manejo de contenidos y la recepción de artículos se la realiza en línea
- El *Open Journal Systems* es un software que está disponible de manera libre para revistas a nivel mundial con el objetivo de hacer uso del acceso abierto para la divulgación de los artículos.
- El aplicativo soporta los procesos básicos de recepción, revisión, edición y publicación de artículos.

1.5.2.2 Base de datos: Mysql

MySQL es un sistema de administración de base de datos que brinda mucha facilidad al usuario que lo utilice además de ser veloz y versátil, usa el *Structured Query Language* (Lenguaje de consulta estructurado), que es el lenguaje más ampliamente utilizado para el acceso a bases de datos.[11]

Sus principales características son las siguientes:

- MySQL es una base de datos relacional, sus datos están guardados en tablas mismas que establecen relaciones entre sí que permiten un manejo de datos eficiente.
- Es de tipo abierto, su código fuente está disponible de manera gratuita para su uso no comercial.
- MySQL es una base de datos ágil y de fácil uso.

En el presente proyecto se usará MySQL para almacenar la información del sistema Open Journal Systems, ya que su instalación depende de un sistema gestor de base de datos que guarde su configuración.

1.5.2.3 Lenguaje de programación: PHP

PHP es un lenguaje de programación orientado a desarrollo web, por esta razón, PHP interactúa con lenguaje HTML, PHP genera código HTML y lo envía al cliente, el mismo que recibe código generado HTML sin que sea necesario que este conozca que fue generado a partir de un script. [14]

Sus principales características son las siguientes:

- PHP orientado a desarrollo web, aunque esto no quiere decir que no tenga otro modo de ejecución.
- es multiplataforma
- Es ampliamente utilizado, tiene facilidad de conexión con bases de datos de libre acceso como MySql y Postgres.
- El código PHP es invisible al cliente, siendo el servidor el encargado de generar código HTML y pasarlo al cliente, lo que permite tener mayor seguridad.

PHP es un lenguaje sencillo y de fácil uso, sin que esto signifique que no tenga características avanzadas, incluso características que permiten su manejo sin necesidad de un navegador, y aún más utilizándolo desde la consola de comandos.

Open Journal Systems está desarrollado en PHP y utiliza de este lenguaje de programación para poder ejecutarse, además PHP se utiliza en otra herramienta, *Triplify*, cuyo uso permite la extracción de datos de la base para la generación de datos RDF.

1.5.2.4 Servidor web: Apache

Apache es un servidor web de código abierto, se puede instalar en varias plataformas (Windows, BSD, Linux, etc), ya que su uso va de la mano con el lenguaje de programación PHP, y es de acceso libre, es ampliamente usado a nivel mundial para un sinnúmero de aplicaciones. [18]

Apache posee un buen desempeño además de ser seguro gracias a sus módulos de autenticación y autorización, y es precisamente el uso de sus módulos lo que permite su flexibilidad, fácil uso y configuración.

Para este proyecto se lo utiliza para alojar la aplicación de la revista.

1.5.2.5 Software de extracción de datos: Triplify

Triplify es un software escrito en PHP que proporciona una funcionalidad semántica a las aplicaciones web, funciona a manera de plug-in y permite obtener estructuras semánticas disponibles en forma de linked data. [5]

Triplify contiene un conjunto de archivos PHP en los que el usuario debe configurar principalmente la conexión con la base de datos de la aplicación de donde desea obtener la información extraída.

Para obtener la información Triplify necesita de la configuración de *consultas* SQL, estas consultas deben asociar las tablas necesarias para obtener registros que puedan convertirse en triples, Triplify en base a estas consultas se encargará de generar la información necesaria disponible de dos formas: Para acceso web, vía navegador y también como conjunto de datos.

Triplify se usará para extraer un conjunto de datos desde la base de la Revista Politécnica.

1.5.2.6 Publicación de punto SPARQL: Virtuoso Opensource

Como ya se señaló anteriormente SPARQL es un lenguaje utilizado para realizar consultas de tipo semántico, muy parecido a SQL debido a su utilización sobre base de datos.

Por tanto SPARQL no es un lenguaje diseñado para consultas de datos relacionales, sino más bien para consultas de información de acuerdo a un modelo RDF, la información que se encuentra en formato RDF permite disponer de conjuntos de datos, incluso los que se encuentran en las bases de datos

relacionales (a través de herramientas de extracción y transformación) con una interfaz de datos estandarizados.

Actualmente el software que necesita comunicación a través de datos enlazados suele tener soporte para lenguaje de consulta SPARQL, sin que haya necesidad de implementar un API de comunicación con estas fuentes de datos. Esta disponibilidad de información ha permitido a los usuarios acceder a una gran variedad de datos públicos. Esta información está disponible a través de un punto SPARQL, el punto SPARQL no es más que un URI en donde se envían consultas SPARQL y a su vez este punto SPARQL responde a esta petición de manera adecuada, ya sea entregando información o dando una respuesta de actualización.

1.5.2.7 Front end para el punto SPARQL: Pubby

La mayoría de los datos de web semántica están ubicados en *Triplestores*, Un Triplestore es un repositorio de datos RDF, estos suelen tener un punto SPARQL para sus consultas, siendo muy difícil conectarse a estos repositorios con otras fuentes externas. [2]

Pubby hace posible que un punto SPARQL pueda ser un servidor de *datos enlazados*, dando una interfaz para navegadores web aportando así a una óptima lectura de datos semánticos por parte de los usuarios y también brinda una interfaz para acceso de datos RDF ya sea para datos remotos o locales.

Por estas razones Pubby se lo utilizará en el sistema de la Revista Politécnica para dar una interface web para usuarios y también para brindar una interface linked data para datos RDF de la revista.

1.5.2.8 Contenedor de servlets para Pubby: Tomcat

Es necesario que exista una instalación de Tomcat (también puede ser Jetty) para la instalación de Pubby, ya que se necesita un contenedor de servlets que pueda interactuar con los servlet u otros archivos java que tenga Pubby.

1.5.2.9 JDK (Java Development Kit)

El Java Development Kit es una implementación de una de las plataformas Java, tiene opción de ser instalada en Solaris, Linux, Mac OS X o Windows.

Este Development Kit necesario para la instalación de Pubby y para el correcto funcionamiento de Tomcat.

CAPITULO 2 INICIO Y ELABORACIÓN

2.1 MODELAMIENTO DEL NEGOCIO

2.1.1 ACTORES DEL NEGOCIO

En la revista politécnica como tal intervienen algunas personas, cada uno tiene una tarea, a continuación, se presenta una lista de los actores a tomar en cuenta:

Tabla 1 - Actores de Negocio

ACTOR	DESCRIPCIÓN
Lector	Persona que utiliza los contenidos finales de la revista científica para su beneficio
Autor	Persona que investiga, y trabaja en el desarrollo de artículos, encargada de entregar los artículos para su posterior edición
Revisor	Es quien ha sido designado para inspeccionar y verificar la correcta estructura de un artículo, también analiza si este está en lo correcto con respecto a su información
Editor	El editor de la revista se encarga del manejo general de la misma, el será quien asigne, a revisión y verifique si se cumple las expectativas de aprobación
Editor de sección	El editor de sección se encargará de las mismas labores que el editor general, pero solo con artículos que hayan sido entregados en su sección.

Corrector de estilo	El corrector de estilo verifica que el artículo no tenga faltas ortográficas, que las frases tengan un contenido entendible y que el artículo esté bien escrito
Editor de maquetación	Es la persona que, una vez terminada la edición del artículo hace los archivos finales, ya sea HTML o PDF y los sube a la revista publicada (o a posterior publicación)
Administrador de revista	Es la persona encargada de administrar todas las configuraciones, personalizaciones y demás tareas generales de la revista científica.
Administrador linked data	Es la persona encargada de administrar el sitio donde se aloja el contenido RDF de la revista
Usuarios de RDF	Todos los lectores (por medio de interfaz web), aplicaciones linked data o demás entidades que utilicen el servicio del punto SPARQL.

2.1.2 CASOS DE USO DEL NEGOCIO

2.1.2.1 Administrar revista

A continuación, se presenta el diagrama de caso de uso de negocio:

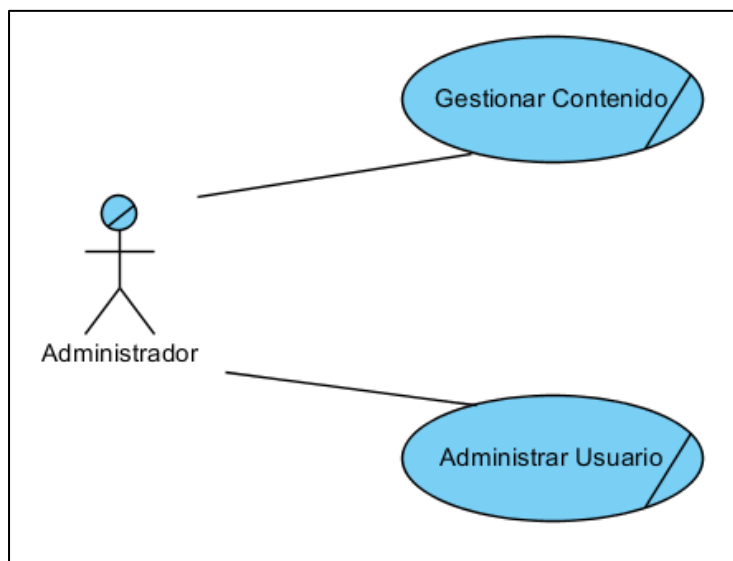


Ilustración 2 – CUN01 (Caso de uso del negocio) Administrar revista

A continuación, se muestra la descripción del caso de uso

Tabla 2 – CUN01 Administrar revista

IDENTIFICADOR	DESCRIPCION
Gestionar contenido	El administrador diseña el contenido de la revista
Administrar usuarios	El administrador maneja los datos de las personas que tienen relación con la revista; editores, autores y revisores.

2.1.2.2 Revisar artículo

A continuación, se muestra el diagrama de caso de uso de negocio correspondiente a la revisión de artículo.

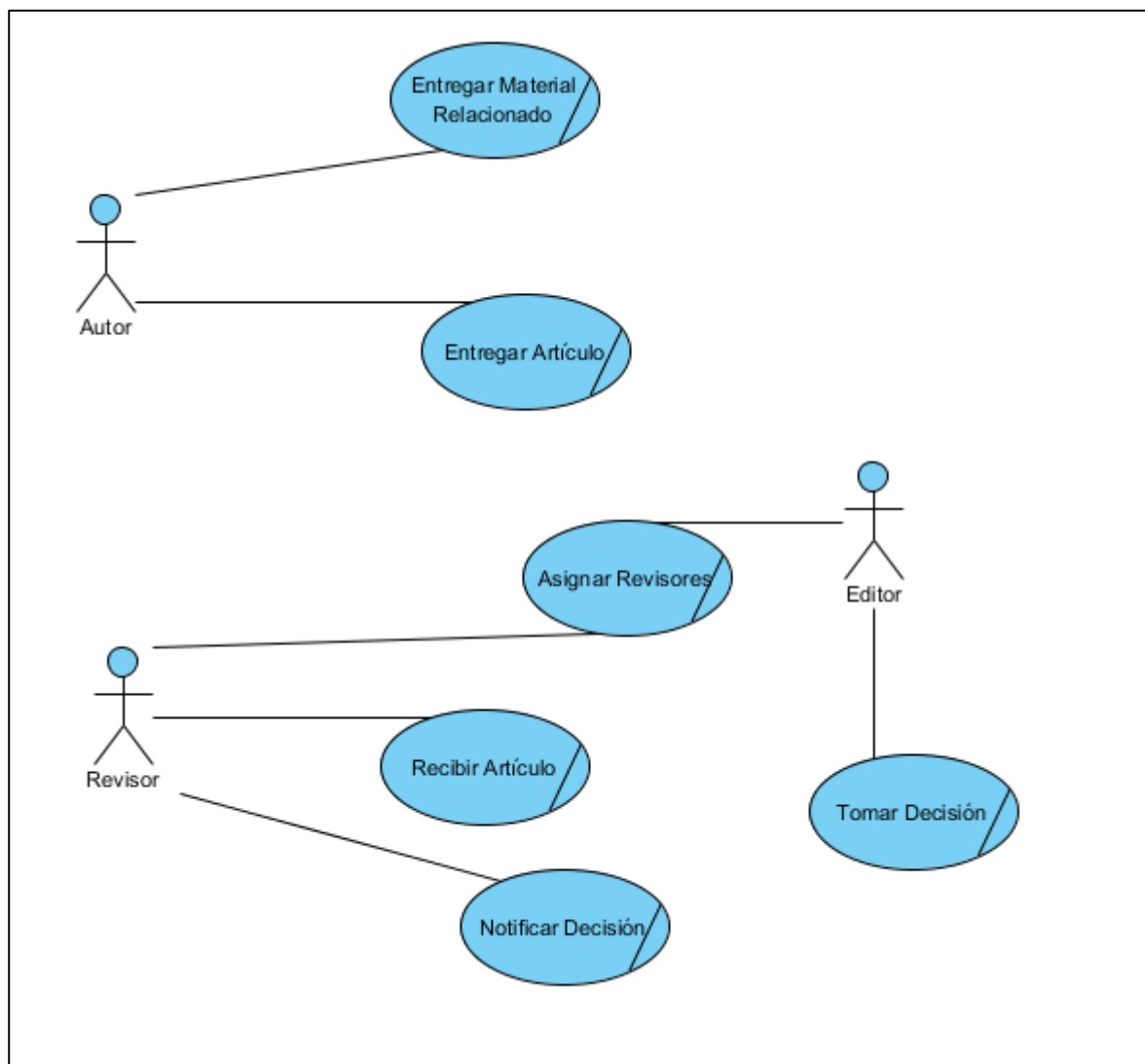


Ilustración 3 – CUN02 Revisar artículo

A continuación, se muestra la descripción del caso de uso:

Tabla 3 – CUN02 Revisar artículo

IDENTIFICADOR	DESCRIPCION
Entregar artículo	El autor entrega el artículo.
Entregar material relacionado	Opcionalmente el autor puede entregar material relacionado a su

	artículo
Asignar revisores	El editor asigna los revisores para el artículo proporcionado
Recibir artículo	El revisor obtiene una copia del artículo.
Notificar decisión	El revisor toma una decisión con respecto al artículo, si lo aprueba o no, notifica dicha decisión a la edición de la revista
Tomar decisión final	El editor en base a las decisiones de los revisores toma una decisión final

2.1.2.3 Corregir estilo

El diagrama de este caso de uso se lo puede apreciar a continuación:

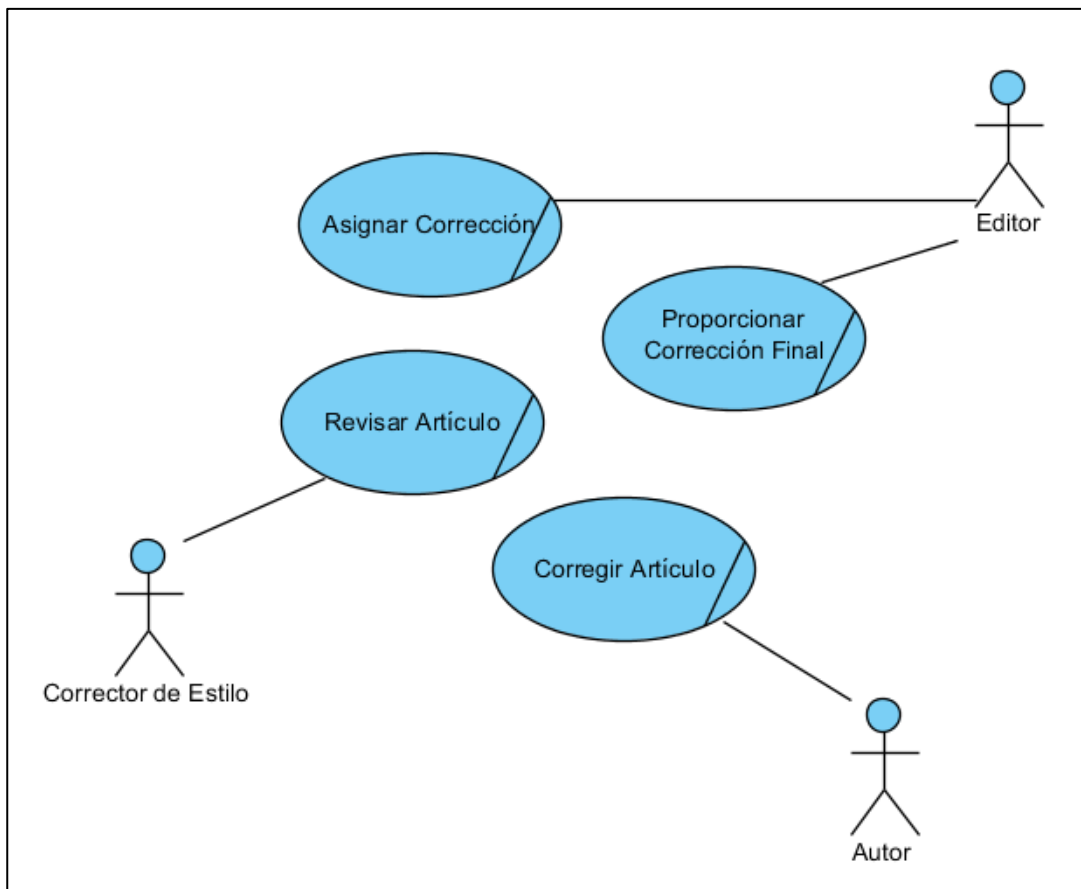


Ilustración 4 – CUN03 Corregir Estilo

La descripción correspondiente se muestra a continuación:

Tabla 4 – CUN03 Corregir de Estilo

IDENTIFICADOR	DESCRIPCION
Revisar artículo	El corrector de estilo hace una revisión de artículo
Asignación de corrección	Cuando se ha aprobado un artículo, el editor debe notificar al corrector de estilo que tiene un nuevo documento asignado para su gestión.

Revisar artículo	El corrector de estilo verificará la correcta expresión gramatical de la información en el artículo.
Corregir artículo	El autor deberá basarse en la revisión proporcionada por el corrector de estilo para hacer las correcciones necesarias
Proporcionar corrección final	El editor verificará la validez de las correcciones hechas por el autor, valiéndose en la revisión proporcionada por el corrector de estilo.

2.1.2.4 Publicar Edición

El diagrama de casos de uso de publicar edición es el siguiente:

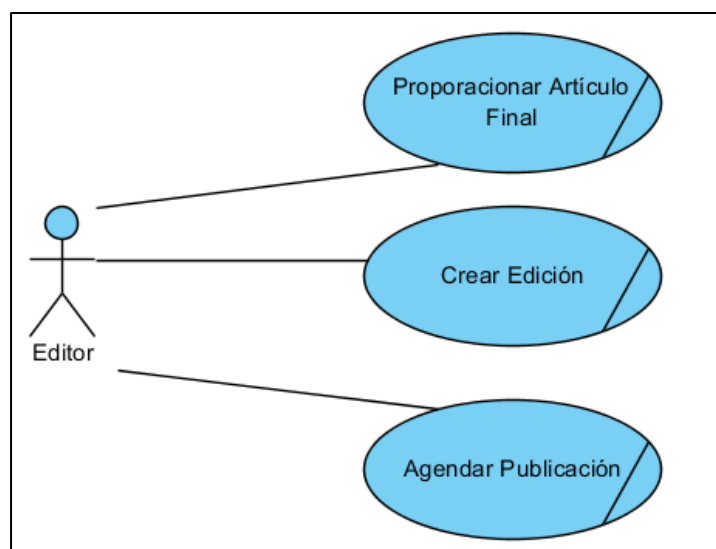


Ilustración 5 – CUN04 Publicar Edición

Tabla 5 – CUN04 Publicar Edición

IDENTIFICADOR	DESCRIPCION
Proporcionar artículo final	Una vez se hayan hecho las correcciones finales por el corrector de estilo, el editor obtiene una versión final del artículo.
Agendar publicación	Se define en el tiempo la salida de la revista
Crear edición	Una nueva edición es creada misma que contendrá los artículos.

2.2 CAPTURA DE REQUERIMIENTOS

Para la implementación del presente proyecto es necesario que se extraiga los distintos requisitos y en base a estos poder determinar su análisis y diseño.

2.2.1 CARACTERÍSTICAS DE USUARIOS

- Los usuarios que utilicen el sistema tienen que estar al tanto del proceso de publicación de artículos en revistas científicas.
- El Corrector de estilos debe tener buenos conocimientos de gramática y ortografía.
- El Editor debe saber todo el Procedimiento de Edición de Revistas Científicas.
- La persona encargada de generar los datos RDF debe tener los conocimientos necesarios para poder correr un “shell script” en linux.

2.2.2 REQUERIMIENTOS NO FUNCIONALES

- El sistema debe ser alojado en un servidor Linux, Debian.
- La base de datos para la funcionalidad del mismo debe ser MySql
- El sistema deberá tener un acceso a un directorio con suficiente espacio para guardar los archivos de los artículos que vayan a subirse.
- El módulo de indexación necesita acceso a la base de datos de la revista politécnica.
- El servidor de Front End debe tener un servlet container.

2.2.3 CASOS DE USO DEL SISTEMA (REQUISITOS FUNCIONALES)

A continuación, se listan los Casos de Uso del Sistema

Tabla 6 - Casos de uso del sistema

NOMENCLATURA	CASO DE USO
CUS01	Acceder al sistema
CUS02	Configurar revista
CUS03	Administrar usuarios
CUS04	Notificar usuarios
CUS05	Entregar artículo
CUS06	Revisar artículo
CUS07	Corregir estilo
CUS08	Editar maquetación
CUS09	Publicar edición
CUS10	Publicar datos semánticos

2.2.3.1 Acceder al sistema

El diagrama del caso de uso de acceder al sistema es el siguiente:

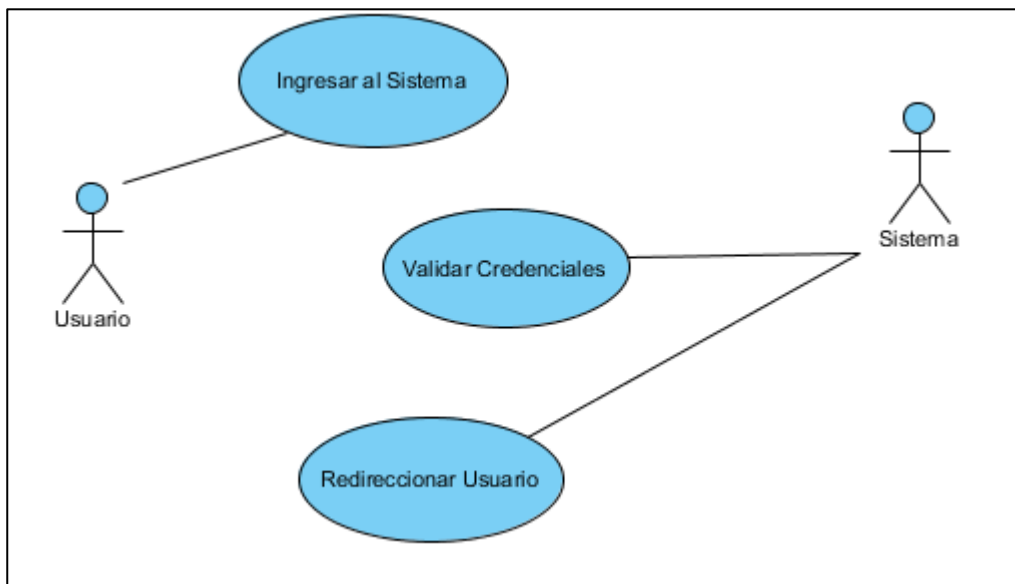


Ilustración 6 - CUS01 Acceder al sistema

A continuación, se presenta el detalle del caso de uso:

Tabla 7 - CUS01 Acceder al sistema

ITEM	DETALLE
Caso de Uso:	Acceder al sistema
ID:	CUS01
Descripción:	Permite el ingreso por parte del usuario al sistema
Actores:	Usuario, Sistema
Precondiciones:	El usuario debe haber sido ya creado en el sistema.

Flujo principal:	<ul style="list-style-type: none"> • El usuario abre la pantalla de logueo • El usuario ingresa su nombre de usuario y contraseña • El sistema verifica la validez del mismo • El sistema permite el acceso al usuario
Flujo alternativo	<ul style="list-style-type: none"> • Si el nombre de usuario no es correcto el sistema muestra un mensaje de error en el logueo
Post-condiciones:	El usuario tiene las funcionalidades del sistema conforme a su rol

2.2.3.2 Configurar revista

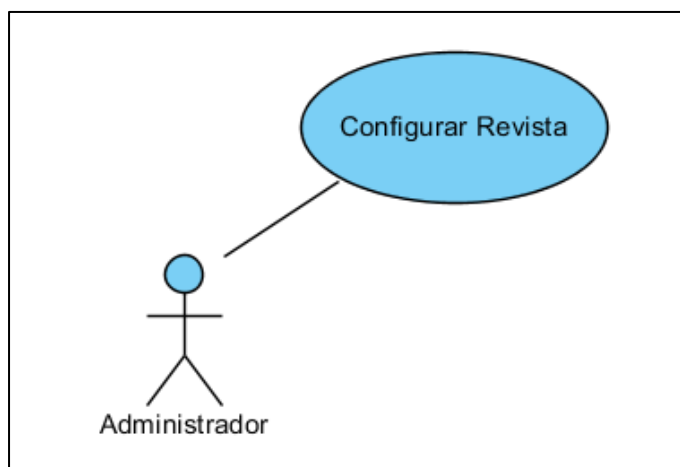


Ilustración 7 - CUS02 Configurar revista

A continuación, se presenta el detalle del caso de uso:

Tabla 8 - CUS02 Configurar revista

ITEM	DETALLE
Caso de uso:	Configurar revista
ID:	CUS02
Descripción:	Permite gestionar las opciones de configuración de la revista
Actores:	Administrador del sistema
Precondiciones:	El Sistema debe estar correctamente instalado
Flujo principal:	<ul style="list-style-type: none"> • El administrador ingresa al sistema • El administrador elige la opción de configuración • El sistema muestra las distintas opciones de configuración que permite el aplicativo. • El administrador elige una opción de configuración. • El administrador guarda los datos de configuración
Post-condiciones:	El sistema presentará la configuración seleccionada por el administrador.

2.2.3.3 Administrar usuarios

El diagrama de caso de uso se muestra a continuación:

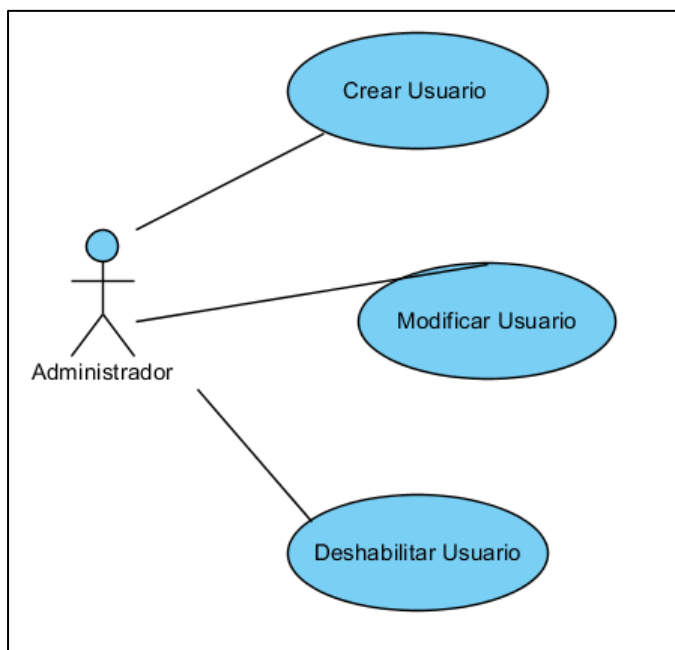


Ilustración 8 – CUS03 Administrar usuarios

La siguiente tabla muestra la descripción del caso de uso

Tabla 9 - CUS03 Administrar usuarios

ITEM	DETALLE
Caso de uso:	Administrar usuarios
ID:	CU03
Descripción:	Permite gestionar los usuarios de la revista
Actores:	Administrador del sistema
Precondiciones:	El sistema debe estar correctamente instalado
Flujo principal:	<ul style="list-style-type: none"> • El administrador ingresa al sistema

	<ul style="list-style-type: none"> • El administrador elige la opción de configuración de usuarios • El administrador crea/modifica/deshabilita un usuario. • El sistema actualiza la información requerida
Post- condiciones:	El sistema presenta los cambios que hayan sido realizados por el administrador en cuanto a los usuarios.

2.2.3.4 Notificar usuarios

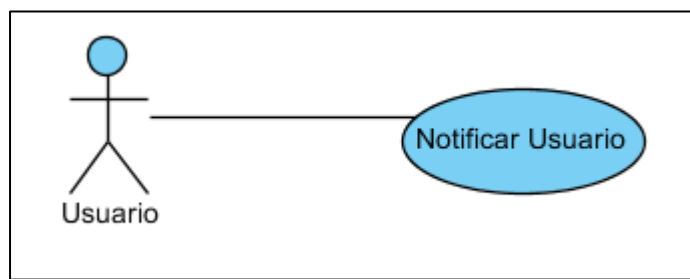


Ilustración 9 - CUS04 Notificar usuarios

A continuación, se muestra el detalle del caso de uso:

Tabla 10 - CUS04 Notificar Usuarios

ITEM	DETALLE
Caso de uso:	Notificar usuarios
ID:	CU04
Descripción:	Permite avisar a los usuarios distintos aspectos de la revista
Actores:	Administrador del sistema, editor, corrector de estilo, editor de maquetación, revisor

Precondiciones:	<ul style="list-style-type: none"> • El usuario notificador debe estar previamente creado • Debe haberse configurado una cuenta de correo electrónico para el sistema.
Flujo Principal:	<ul style="list-style-type: none"> • El usuario realiza la tarea correspondiente dentro del sistema • El sistema muestra la opción de notificación • El usuario ingresa a la pantalla de notificación • El sistema muestra los campos de notificación. • El usuario envía la notificación
Flujo Alternativo	<ul style="list-style-type: none"> • El usuario cancela el envío de notificación.
Post-Condicion:	El sistema envía un correo electrónico a la dirección correspondiente (cualquier dirección de correo, no necesariamente de un usuario en el sistema)

2.2.3.5 Entregar artículo

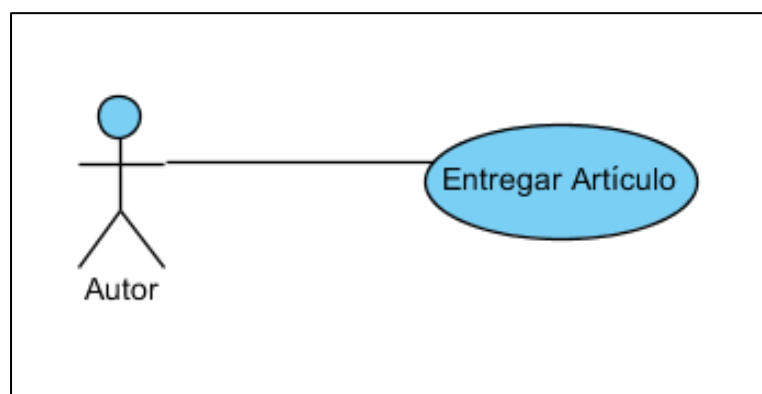


Ilustración 10 – CUS05 Entregar artículo

A continuación, se muestra el detalle del caso de uso:

Tabla 11 - CUS05 Entregar artículo

ITEM	DETALLE
Caso de Uso:	Entregar Artículo
ID:	CUS05
Descripción:	Permite subir el archivo del artículo por parte del autor
Actores:	Autor
Precondiciones:	El sistema debe estar correctamente instalado
Flujo principal:	<ul style="list-style-type: none"> • El autor ingresa al sistema • El autor elige la opción de nuevo envío • El autor ingresa los datos de nuevo envío • El sistema muestra las pantallas de nuevo envío • El autor sube el archivo que contiene el artículo • El sistema guarda el archivo en el sistema
Flujo alternativo	<ul style="list-style-type: none"> • Si el autor no chequea todas las opciones de envío el sistema no permite subir el documento.
Post- Condiciones:	El sistema presenta los cambios que hayan sido realizados por el administrador.

2.2.3.6 Revisar artículo

En el siguiente gráfico se muestra el diagrama de caso de uso de Revisar Artículo

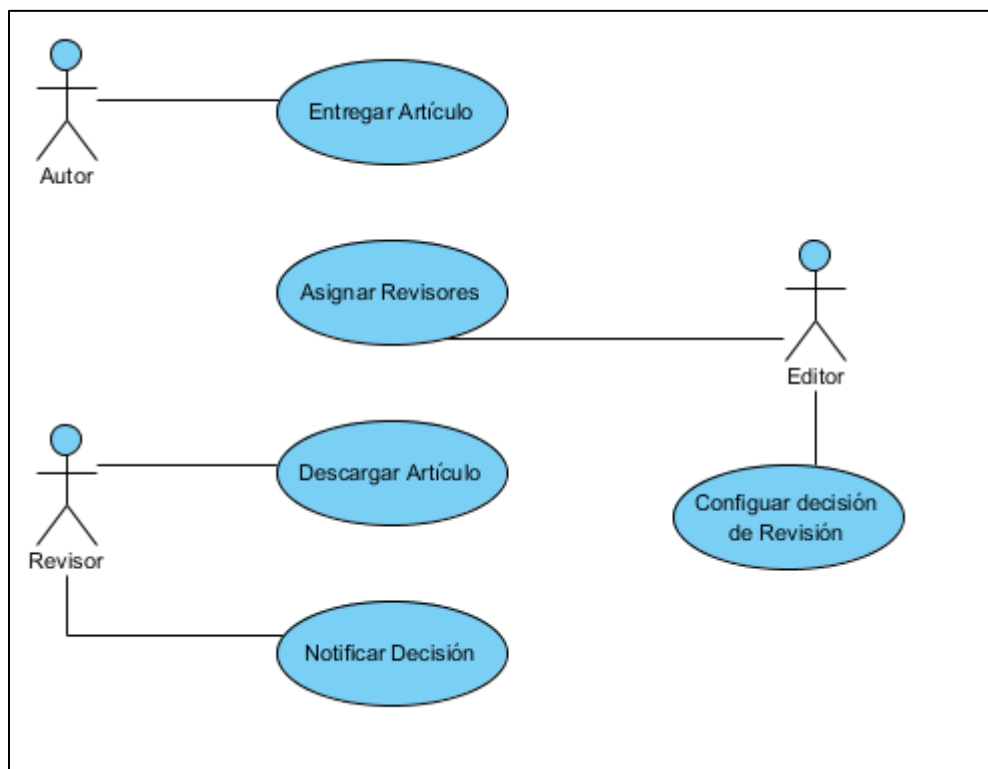


Ilustración 11 - CUS06 Revisar artículo

A continuación, se muestra el detalle del caso de uso:

Tabla 12 - CUS06 Revisar artículo

ITEM	DETALLE
Caso de Uso:	Revisar artículo
ID:	CUS06
Descripción:	Permite la aprobación o no del artículo suministrado por el autor
Actores:	Revisor, editor

Precondiciones:	<ul style="list-style-type: none"> • El artículo a revisar debe haber sido enviado por el autor. • Deben existir revisores disponibles.
Flujo principal:	<ul style="list-style-type: none"> • Se notifica al editor que existe un nuevo archivo listo para revisión. • El editor debe asignar a los revisores correspondientes para el artículo proporcionado. • El editor notificará a los 3 revisores de que han sido asignados para la revisión del artículo • El revisor confirma que ha recibido la asignación, si acepta el sistema lo incluirá en la lista de revisores del artículo • Si el revisor no acepta el sistema notifica al editor para que elija otro revisor. • Una vez asignado, el revisor debe verificar la validez del artículo, enviará los resultados de su revisión al sistema. • El revisor enviará también la información de comentarios de revisión y la recomendación de decisión. • El sistema mostrará al editor, los datos de cada revisión. • El editor, en base a su criterio tomará la decisión final que es lo que se debe hacer con el artículo, podrá mandarlo a nueva revisión, aceptarlo o rechazarlo
Flujo alternativo	<ul style="list-style-type: none"> • Si la decisión final es rechazar el autor podrá volver a subir una nueva versión del artículo.

Post- condiciones:	Si el artículo ha sido aceptado el mismo pasará a la fase de edición.
-------------------------------	---

2.2.3.7 Corregir estilo

A continuación se muestra el diagrama de caso de uso:

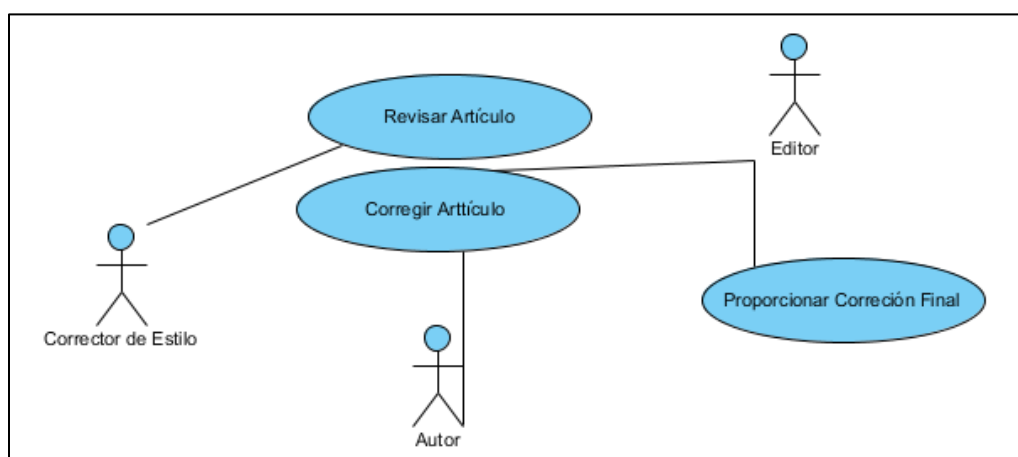


Ilustración 12 – CUS07 Corregir estilo

El detalle de caso de uso se muestra en la siguiente tabla:

Tabla 13 - CUS07 Corregir estilo

ITEM	DETALLE
Caso de uso:	Corregir estilo
ID:	CUS07
Descripción:	Permite corregir errores gramaticales o semánticos en el artículo
Actores:	Autor, corrector de estilo, editor

Precondiciones:	El artículo debe haber pasado la fase de revisión y debe haber sido aprobado.
Flujo principal:	<ul style="list-style-type: none"> • El editor ingresa al sistema • El sistema le notificará que hay un nuevo artículo en revisión • El editor sube el archivo del artículo al sistema • El editor inicia la fase de corrección de estilo • El corrector de estilo corrige el artículo utilizando un estándar de corrección, y sube un nuevo archivo. • El autor puede ser notificado que su artículo se encuentra en revisión y/o que ya está revisado. • El autor sube un nuevo archivo con las correcciones realizadas. • El sistema muestra al editor si hay nuevos archivos subidos al sistema en este proceso. • El editor completa el ciclo de corrección
	<ul style="list-style-type: none"> • El editor puede dar por completado el proceso antes de la corrección del autor.
Post-condiciones:	El sistema guarda la última versión revisada del archivo donde se encuentra el artículo

2.2.3.8 Editar maquetación

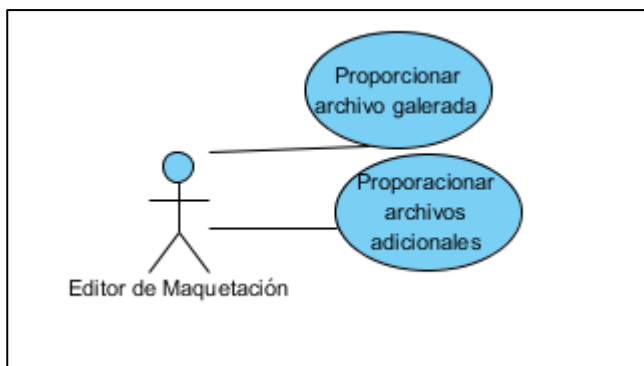


Ilustración 13 - CUS08 Editar Maquetación

A continuación, se muestra el detalle del caso de uso

Tabla 14 - CUS08 Editar maquetación

ITEM	DETALLE
Caso de Uso:	Editar maquetación
ID:	CUS08
Descripción:	Permite configurar el diseño del artículo y su posterior publicación
Actores:	Editor, editor en maquetación
Precondiciones:	El artículo debe haber pasado la fase de revisión y debe haber sido aprobado. Es recomendable que el artículo haya pasado también la fase de Corrección de estilo, aunque esto no es mandatorio.

Flujo principal:	<ul style="list-style-type: none"> • El editor sube un archivo con la última versión del artículo para la fase de maquetación. • El editor de maquetación descarga el archivo. • El editor de maquetación opcionalmente sube archivos adicionales • El sistema muestra que archivos se han subido para los formatos galerada y archivos adicionales
Flujo alternativo	<ul style="list-style-type: none"> • El editor de maquetación opcionalmente sube un archivo PDF para el formato de galerada • El editor de maquetación opcionalmente sube un archivo HTML para el formato de galerada
Post- Condiciones:	El sistema guarda la configuración del artículo para su posterior publicación.

2.2.3.9 Publicar edición

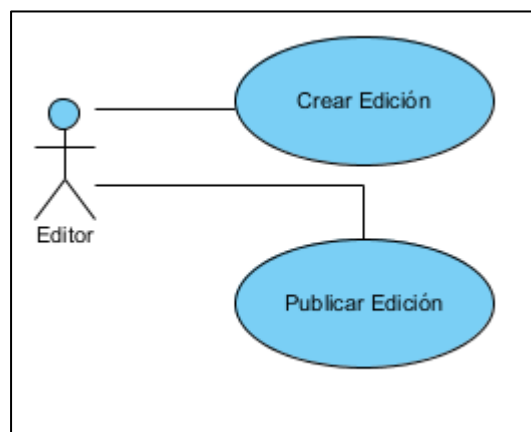


Ilustración 14 - CUS09 Publicar edición

A continuación, se presenta el detalle del caso de uso:

Tabla 15 - CUS09 Publicar Edición

ITEM	DETALLE
Caso de Uso:	Publicar edición
ID:	CUS09
Descripción:	Permite la difusión y divulgación de la edición de la revista en la página principal del sistema
Actores:	Editor
Precondiciones:	<ul style="list-style-type: none"> • El sistema debe estar correctamente instalado
Flujo principal:	<ul style="list-style-type: none"> • El editor ingresa al sistema • El editor elige la opción de Crear Número. • El sistema muestra la pantalla de ingreso de datos del número. • El editor ingresa los datos correspondientes. • Una vez ingresada la edición de la revista el sistema muestra cuales números/ediciones el sistema tiene disponible. • El editor elige una de estas ediciones. • El sistema muestra los artículos que se encuentren en cada edición y la opción de publicar el respectivo número • El editor publica el número.
Post-	El sistema muestra la edición publicada en su página

condiciones:	principal
---------------------	-----------

2.2.3.10 Publicar Datos semánticos

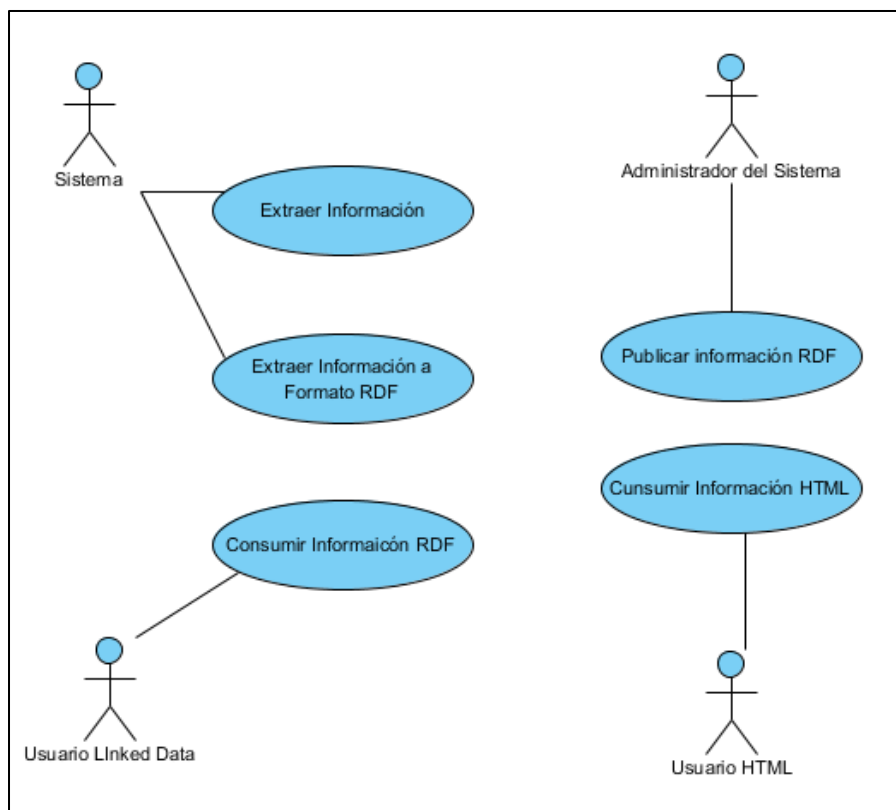


Ilustración 15 – CUS10 Publicar datos semánticos

A continuación, se presenta el detalle del caso de uso:

Tabla 16 - CUS10 Publicar datos semánticos

ITEM	DETALLE
Caso de uso:	Publicar datos semánticos
ID:	CUS10
Descripción:	Permite la generación del archivo RDF con la información

	extraída de la revista y la publicación de esta información en un punto SPARQL
Actores:	Administrador del sistema
Precondiciones:	El servidor de la base de datos de la revista debe tener configurado el acceso remoto
Flujo principal:	<ul style="list-style-type: none"> • El administrador ingresa al servidor de SPARQL • Navegar hasta el script ejecutable del sistema • Correr el script de generación del archivo • Ingresar al sistema web de Openlink Virtuoso • Crear/Actualizar el grafo del punto SPARQL
Post-condiciones:	El sistema web front end muestra los datos nuevos o actualizados extraídos de la revista.

2.2.4 RESTRICCIONES

- El Sistema no realiza una validación de documentos, los mismos deben ser validados por cada uno de los usuarios que tengan esta tarea.
- El envío de correos electrónicos estará a cargo del servidor SMTP, el sistema debe tener uno accesible, así como una cuenta para el envío desde el aplicativo.
- El sistema no será configurado para realizar ningún tipo de respaldo de datos automático. Corre a cargo de la persona/entidad que lo utilice el hacer los correspondientes backups.
- Los datos a implementar en la base de datos de la revista científica digital correspondientes a la desambiguación de autores solo servirán para la publicación de información de datos enlazados. No tienen ningún tipo de interacción de entrada con la revista científica digital.

2.3 ANÁLISIS Y DISEÑO

2.3.1 DIAGRAMAS DE CLASES

El *Open Journal Systems* consta de algunos módulos, su gran variedad de clases permite una funcionalidad separada por distintos bloques, Este sistema utiliza el modelo MVC para su funcionamiento en donde en base al *request* del usuario se direcciona la petición mediante un *handler*, este *handler* direcciona a la clase adecuada que a su vez se encarga de la lógica del negocio, para luego poder mostrar los resultados al usuario mediante las clases *Form*. A continuación se describe las clases más importantes del sistema. [15]

2.3.1.1 Clases Handlers

Los Handlers permiten el manejo de las peticiones HTTP, permiten la redirección de la petición, esta redirección debe ser manejada por una clase concreta, encargada de procesar la información de manera apropiada. (Public Knowledge Project)

Adicionalmente estas clases son responsables por la autenticación y controlan que los *request* proporcionados por los usuarios sean válidos.

Los Handlers son manejados por dos clases principales, encargadas de asociar cada *Handler* con la petición correspondiente. Estas son *PKPPageRouter*, *PKPRouter* y *PKPHandler*.

A continuación, el diagrama de Clases correspondiente al *core* y *handlers*

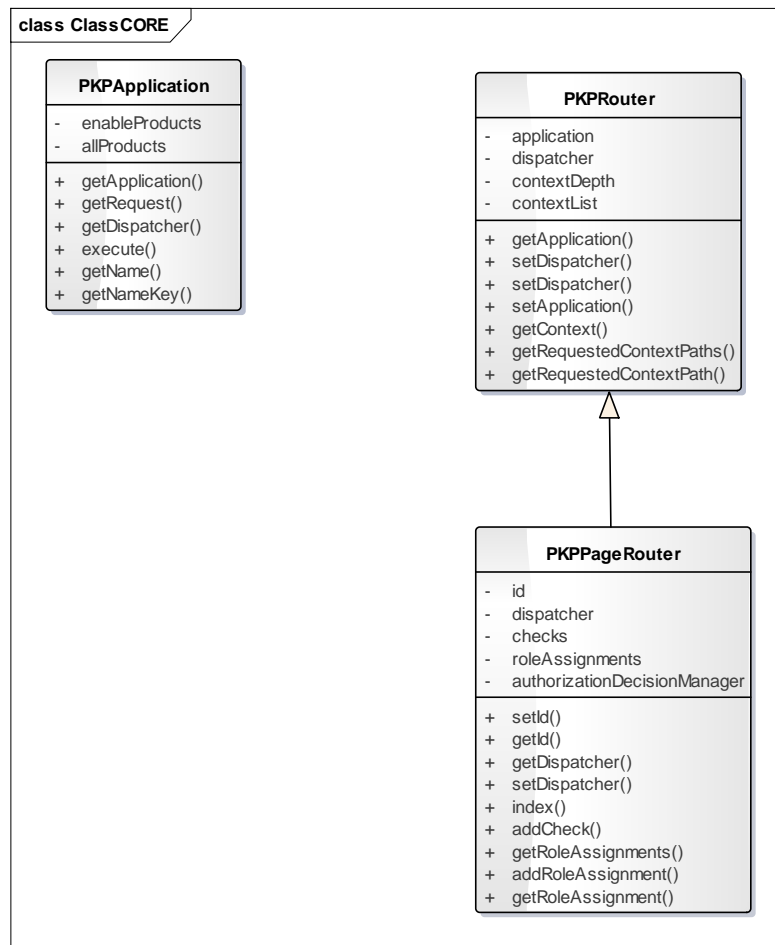
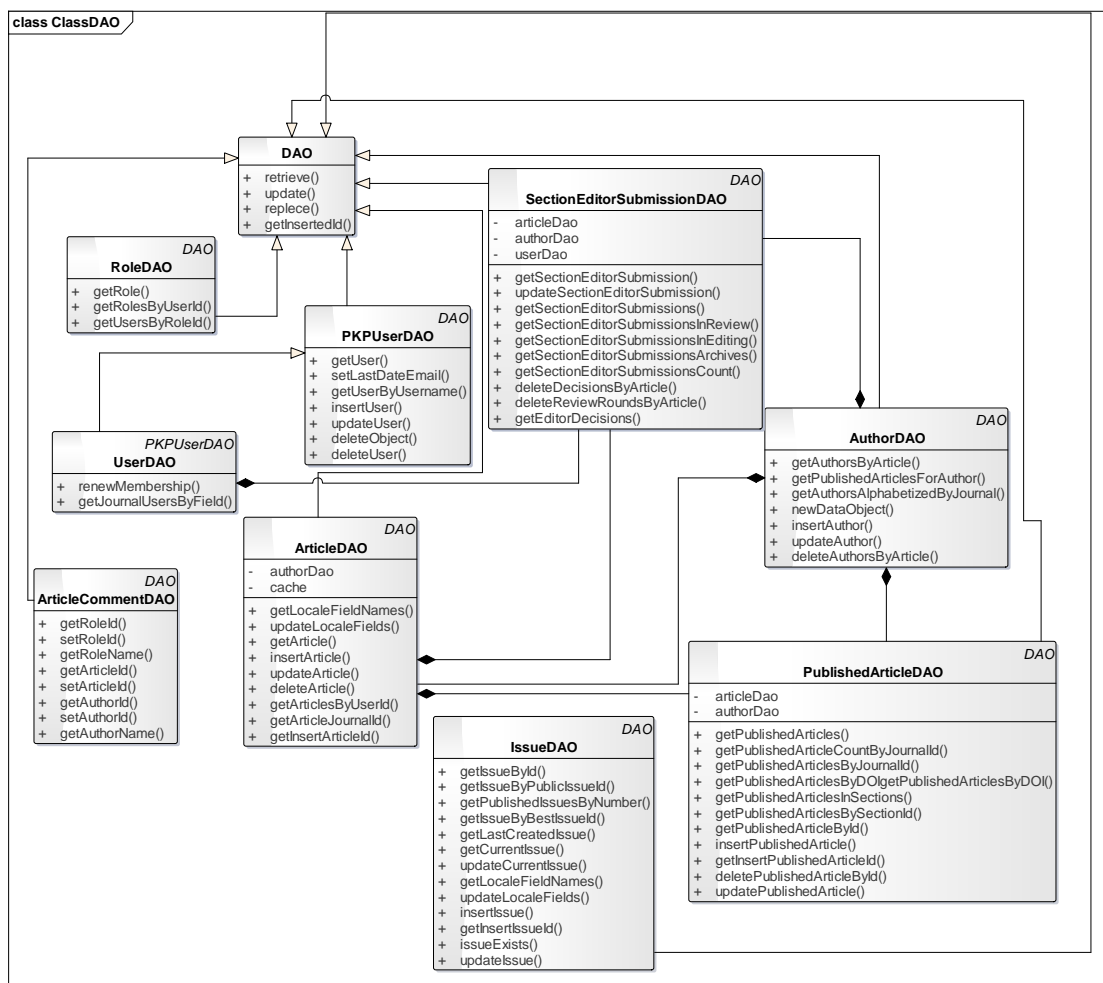


Ilustración 16 - Clases Core

2.3.1.2 Clases DAO

Las clases DAO (Data Access Object) son las clases de acceso a datos, son las encargadas de realizar la conexión con la base de datos, ofrecen de manera general operaciones directas con componentes de la base. A continuación se muestra el diagrama de clases DAO con las más importantes. [15]



Ilustracion 18 - Clases DAO

2.3.1.3 Clases de formulario

Las clases de formulario constituyen aquellas encargadas de renderizar al usuario toda la salida que generan las clases que manejan el procesamiento. Estas clases reciben el request del web browser, delega el procesamiento a la clase adecuada y a través de un template genera una respuesta. [15]

Estas clases y sus subclases son utilizadas por el sistema para administrar una sección, centralizar la implementación de tareas comunes tales como validación y manejo de errores.

A continuación se encuentran las clases más importantes utilizadas por el aplicativo.

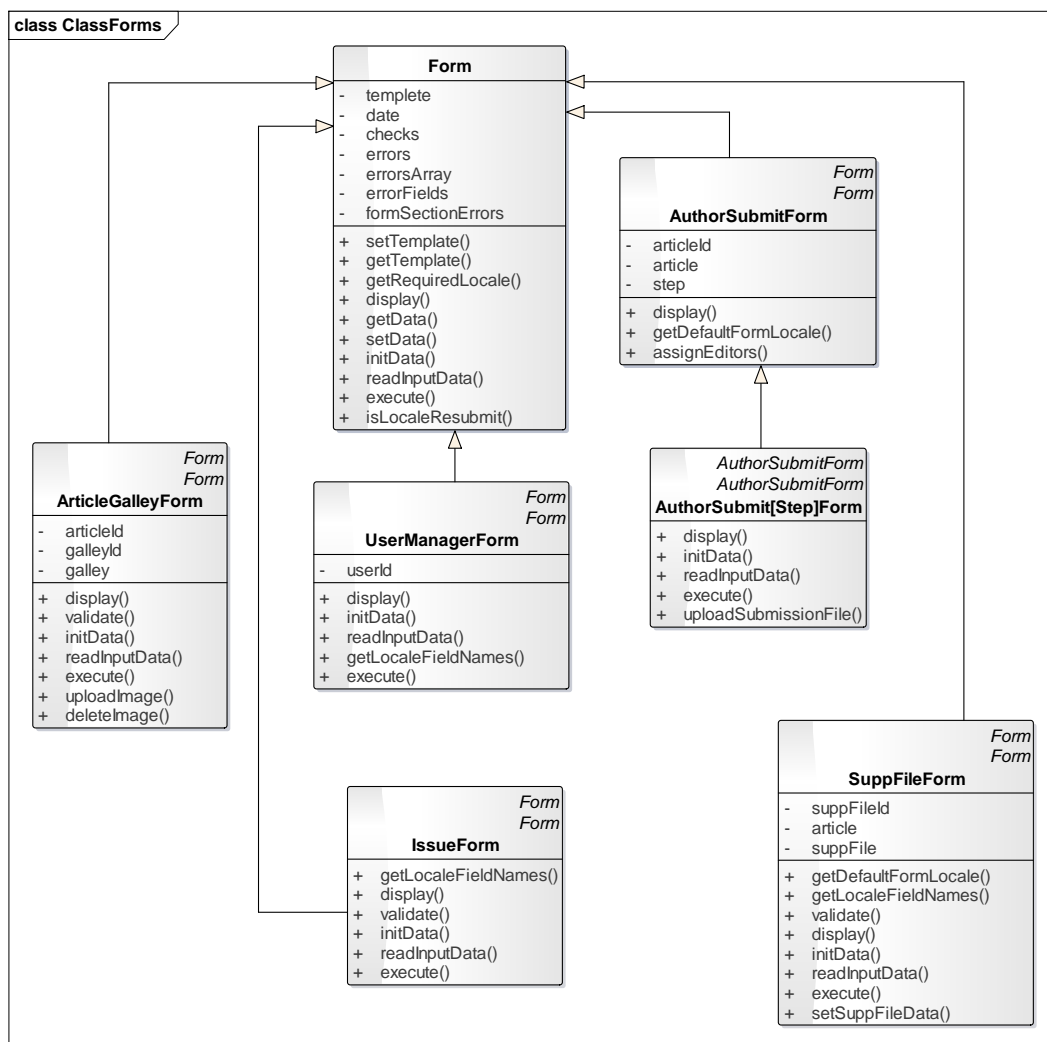


Ilustración 19 - Clases de formulario

2.3.1.4 Clases administrativas

Estas clases consisten en los tipos listados a continuación (Public Knowledge Project):

- **Acción:** Son usadas por las clases *Handler* para realizar procedimientos de negocio cada uno de los roles que son más complejos tienen su propia clase de acción.
- **Manager:** Se encargan del manejo de datos en el sistema de archivos, es decir guardar, modificar y acceder a los distintos archivos y sus versiones.

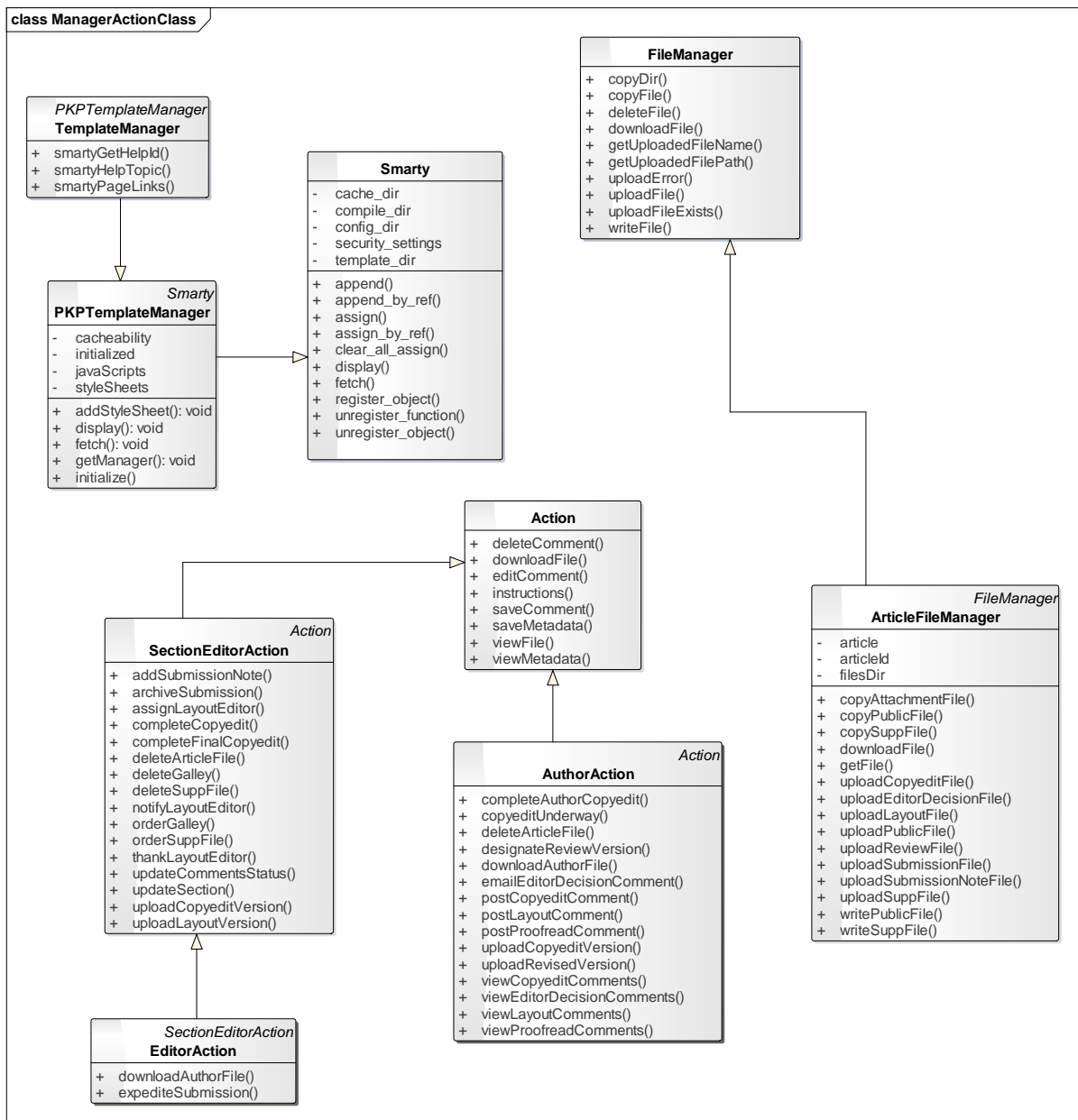


Ilustración 20 - Clases administrativas

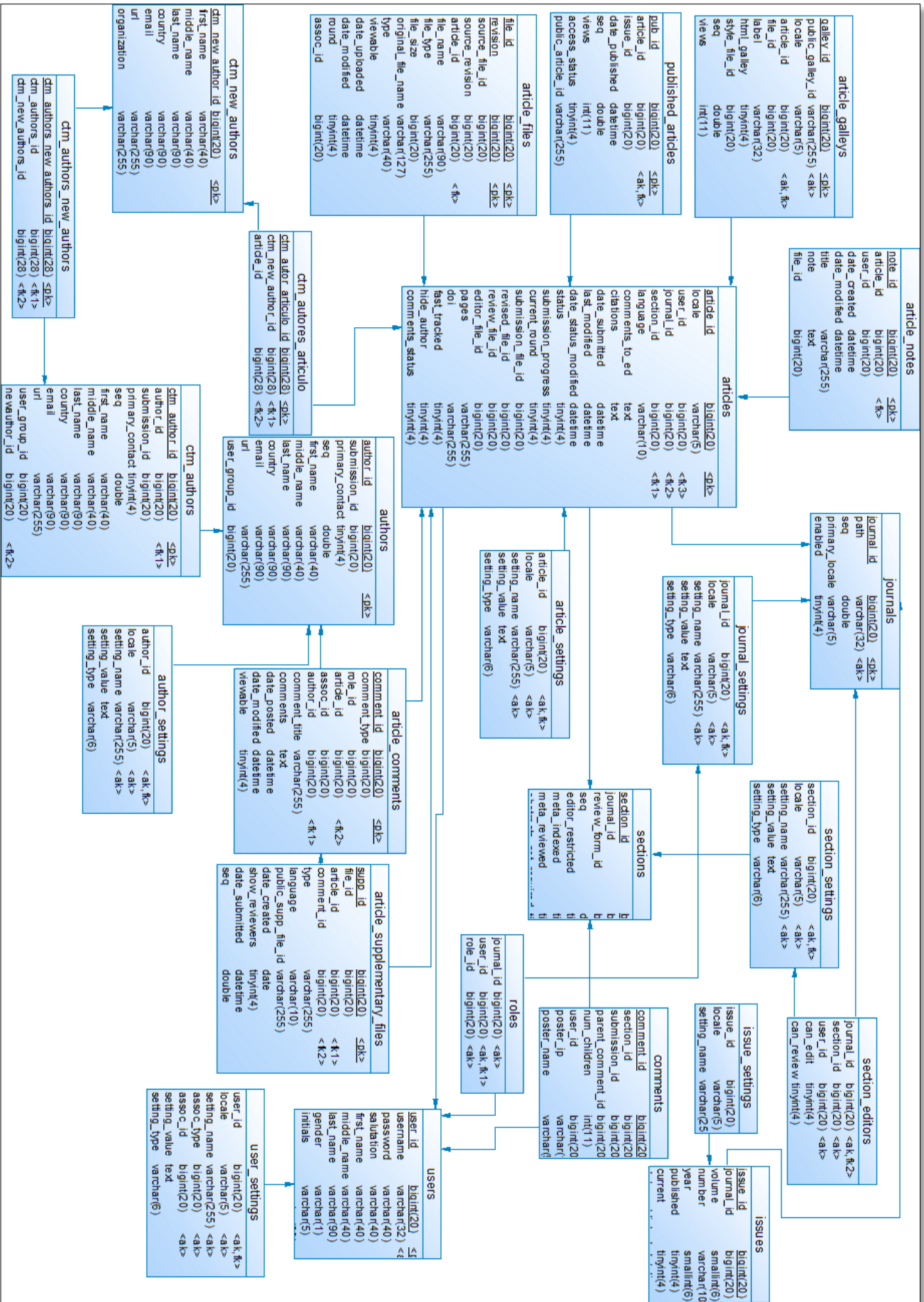


Ilustración 21 - Modelo de base de datos

2.3.2 MODELO DE BASE DE DATOS

En el modelo mostrado se encuentran las tablas más importantes que componen la gestión del *Open Journal Systems*, además existen las tablas con prefijo ctm, mismas que fueron elaboradas como un componente adicional para la indexación semántica.

2.3.3 DISEÑO DE INTERFACES

2.3.3.1 Interfaz principal

La siguiente pantalla corresponde a la primera que el usuario verá cuando ingrese al aplicativo, está compuesta en bloques de los cuales el único que cambia a lo largo del aplicativo es el bloque de CONTENIDO

No se detallan los bloques SUSCRIPCION y ANUNCIOS, debido a que su contenido es una sola línea de texto.

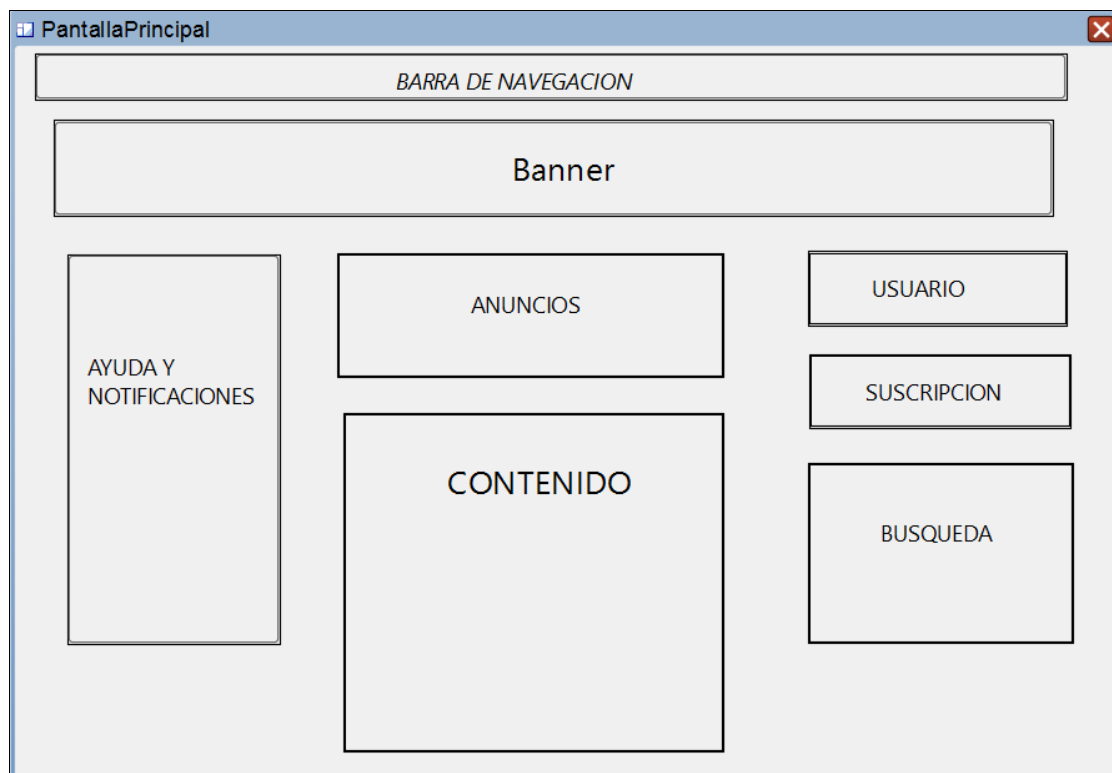


Ilustración 22 - Interfaz Principal

2.3.3.2 Bloque contenido

En este bloque se ubicará la mayoría de información que el aplicativo vaya mostrando en sus distintos módulos, las pantallas que se pueden apreciar en este bloque son:

2.3.3.2.1 Menú administrador

Esta pantalla es la encargada de recibir los datos de ingreso del usuario administrador

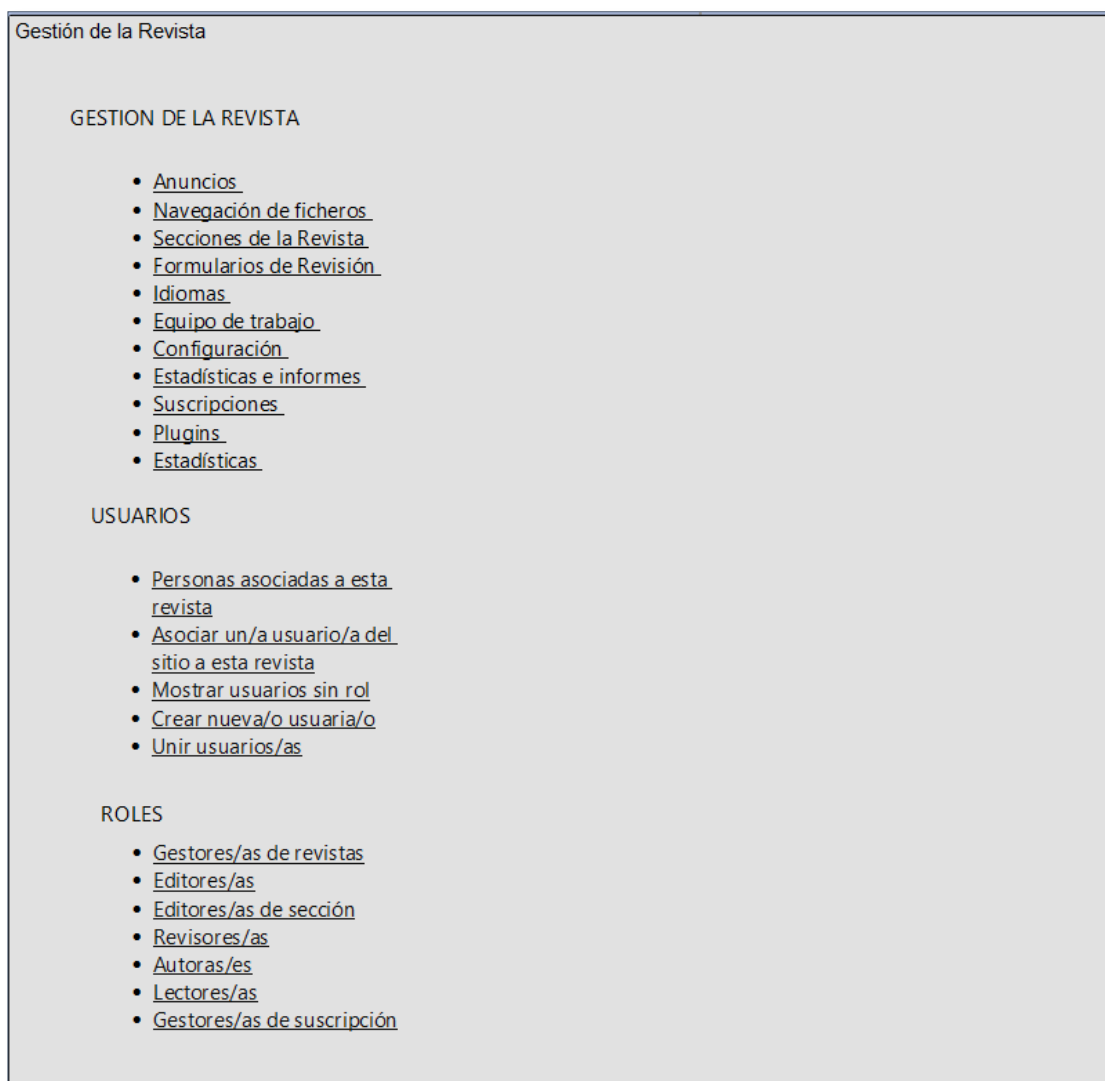


Ilustración 23 – Interfaz Menú administrador

2.3.3.2.2 Menú de configuración

Esta pantalla contiene la funcionalidad para la administración de configuración general de la revista

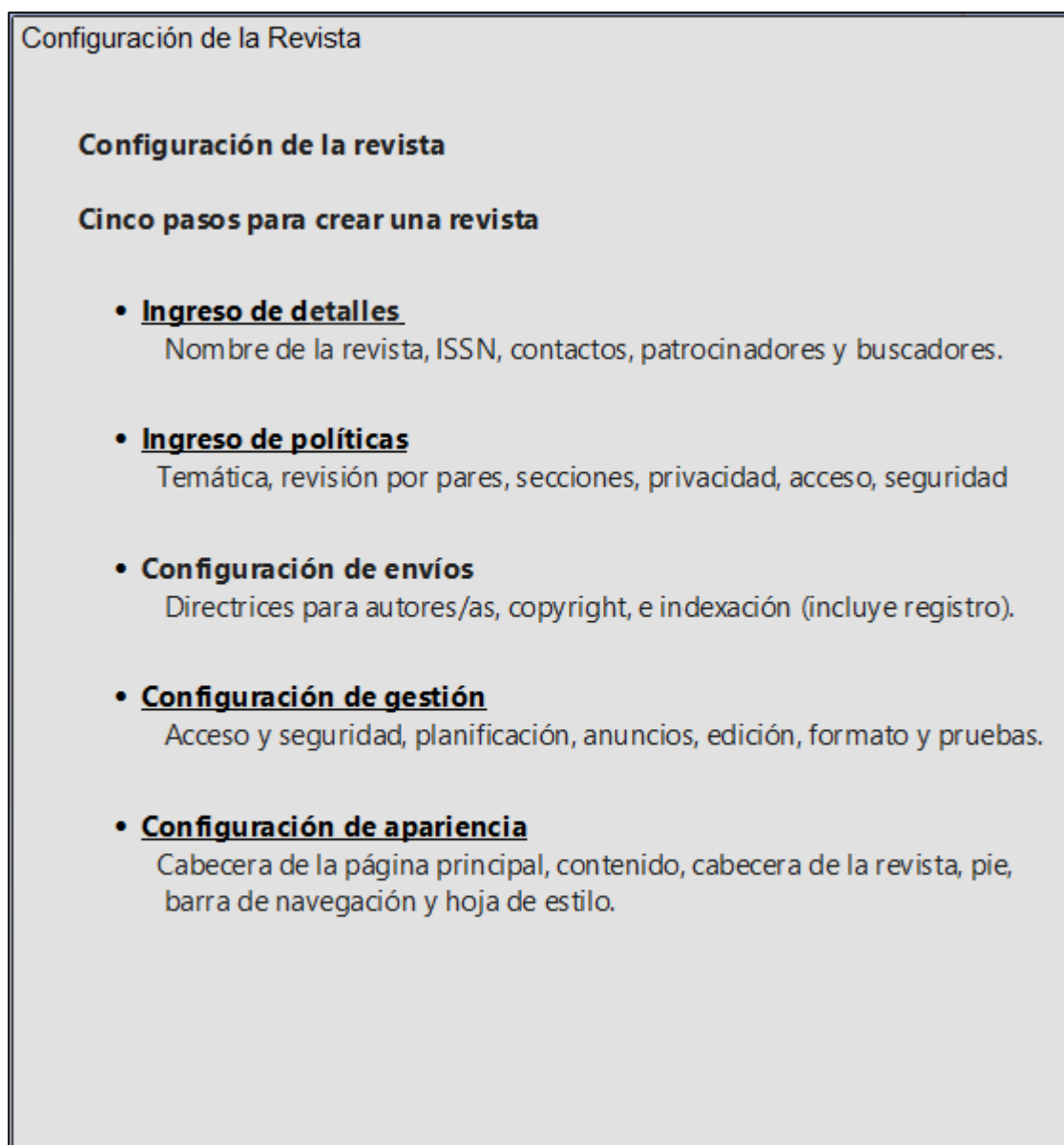


Ilustración 24 – Interfaz menú de configuración

2.3.3.2.3 Detalles de configuración

Cada una de las configuraciones que deban ser aplicadas por el usuario deberá mostrarse de manera descendente.

Cada ítem contendrá su información correspondiente, al final de la pantalla se muestran dos botones, *Guardar* y *Cancelar*

The image shows a software interface titled "Detalle" (Detail). It features a list of configuration items. The first item, "Item1", is expanded to show three input fields: "Información 1 Item 1", "Información 2 Item 1", and "Información n item 1". Below this list are three bullet points and another item labeled "Item n". At the bottom of the interface are two buttons: "Guardar" (Save) and "Cancelar" (Cancel).

Ilustración 25 - Interfaz de detalle de configuración

2.3.3.2.4 Configuración de usuarios: pantalla principal

La siguiente pantalla permite la administración de los usuarios del sistema:

Administración de Usuarios

Inscripción

Personas Asociadas

Roles Nombre, Apellido Opciones

A B C.....Z

- [Gestores/as de revistas](#)
- [Editores/as](#)
- [Editores/as de sección](#)
- [Revisores/as](#)
- [Autoras/es](#)
- [Lectores/as](#)
- [Gestores/as de suscripción](#)

	NOMBRE USUARIO	NOMBRE	CORREO E.	ACCION
<input type="checkbox"/>	username1	Usuario 1	user1@dominio.com	EDITAR/DESUSCRIPCION
<input type="checkbox"/>	username2	Usuario 2	user2@dominio.com	EDITAR/DESUSCRIPCION

Ilustración 26 – Interfaz principal de configuración de usuarios

2.3.3.2.5 Configuración de usuarios: editar perfil

En la siguiente gráfica se puede ver la pantalla de edición de perfil de usuario

The screenshot shows a web form titled "EditarUsuario" with the following fields and controls:

- Personas**
 - Editar Perfil**
- Tratamiento**: text input field
- Primer Nombre**: text input field
- Segundo Nombre**: text input field
- Apellidos**: text input field
- Genero**: dropdown menu
- Nombre de Usuario**: text input field with value "username1"
- Contraseña**: text input field
- Repita la contraseña**: text input field
- Notas Privadas**: text input field
- Dirección Postal**: text input field
- País**: dropdown menu
- Notas Bibliográficas**: large text area
- Filiación**: text input field
- Firma**: text input field
- Correo Electrónico**: text input field
- Url**: text input field
- telefono**: text input field
- Fax**: text input field
- Intereses**: text input field

At the bottom right, there are two buttons: **Guardar** (black background) and **Cancelar** (white background).

Ilustración 27 – Interfaz de edición de usuario

2.3.3.2.6 Creación de usuario

La siguiente interface corresponde a la creación de usuario:

The image shows a web form titled "Crear Usuario" (Create User). The form is organized into two columns of input fields. The left column contains fields for personal information: "Tratamiento", "Primer Nombre", "Segundo Nombre", "Apellidos", "Genero" (with a dropdown arrow), "Inscribirse como" (with a dropdown menu showing "ROL"), "Nombre de Usuario", "Contraseña", "Repita la contraseña", "Filiación", "Firma", "Correo Electrónico", "Url", "telefono", "Fax", and "Intereses". The right column contains fields for additional information: "Notas Privadas", "Dirección Postal", "País" (with a dropdown arrow), and "Notas Bibliográficas" (with a large text area). At the bottom right of the form, there are two buttons: "Guardar" (Save) and "Cancelar" (Cancel).

Ilustración 28 - Interfaz de creación de usuario

El resto de diseños se los puede encontrar el *Anexo 4: Diseño de Interfaces*

CAPITULO 3 CONSTRUCCIÓN Y TRANSICIÓN

Esta fase tiene por objetivo el alcanzar el desarrollo del producto final de manera incremental. [10]

Durante el transcurso de esta fase se desarrolla la implementación, integración y pruebas de todos los componentes que son parte del sistema y así poder obtener una versión para que los usuarios puedan probar el mismo.

3.1 DESARROLLO

3.1.1 ESTÁNDARES DE CODIFICACIÓN

Para el presente proyecto se utilizará las siguientes herramientas:

- Open Journal Systems, como CMS para el alojamiento del sitio web
- Gestor de Base de Datos MySql para alojar la información del sistema
- Triplify, para la transformación de datos a formato RDF
- Pubby, Front End para la muestra de resultados semánticos en un punto SPARQL
- OpenLink Virtuoso, servidor encargado de alojar el punto SPARQL
- Apache webserver, Servidor web encargado de alojar la aplicación de Open Journal Systems
- Apache Tomcat, Servidor Web Java encargado de alojar la aplicación web Pubby.
- Java JDK, lenguaje de programación sobre el cual se basa Pubby
- PHP lenguaje de programación sobre el cual se basa el resto de Herramientas

El *Open Journal System* tiene las siguientes convenciones [16]

- El código fuente, nombres de archivos, URL y nombres de clase están presentados en letra tipo *courier*.

- Se utilizará *corchetes* en código, archivos o URLs que indiquen un valor, por ejemplo [texto]Handler.inc.php. Puede ser interpretado como cualquier archivo que termine en *Handler.inc.php*.
- Los directorios se nombran utilizando la convención *lowerCamelCase*.
- Siempre y cuando sea posible las variables globales y funciones fuera de las clases deben ser evitadas.
- Los nombres de archivo deben coincidir en lo posible con las clases implementadas en los mismos.
- Los nombres de clase y variables deben ser capitalizadas de la siguiente manera: las clases deben usar *CamelCase* y las instancias *lowerCamelCase*.
- siempre que sea posible la instancia debe coincidir con el nombre de la clase.
- Las validaciones deben ser hechas en tanto en las pantallas como en la base de datos.

3.1.2 ARQUITECTURA

La arquitectura en que se basa la aplicación de Open Journal Systems es en el Modelo MVC, Model – View – Controller (Modelo-Vista-Controlador) [16]

Este patrón de diseño tiene como principal característica la separación de componentes por funcionalidad. Hace una división entre los componentes que son las Interfaces gráficas de Usuario, el procesamiento de datos y el acceso a los datos en 3 capas de separación, esto permite un mejor mantenimiento del código

Generalmente el proceso de las interacciones en el sistema cuando se aplica el modelo MVC es el siguiente:

- El usuario realiza una acción en la interfaz
- El aplicativo (dependiendo la configuración a realizar) mapea la petición del usuario hacia el controlador.

- El controlador tiene métodos para poder manejar cada petición que hace el cliente.
- El controlador notificará al modelo la acción a tomar en base a los datos de entrada del cliente.
- Una vez realizado esto el controlador redireccionará a otra vista o enviará datos de vuelta a la vista correspondiente en donde se renderizará esta información en el cliente para que pueda ser vista por el usuario.
- El usuario puede seguir ingresando nueva información y así sucesivamente. El diseño del *Open Journal Systems* por tanto fue implementado pensando

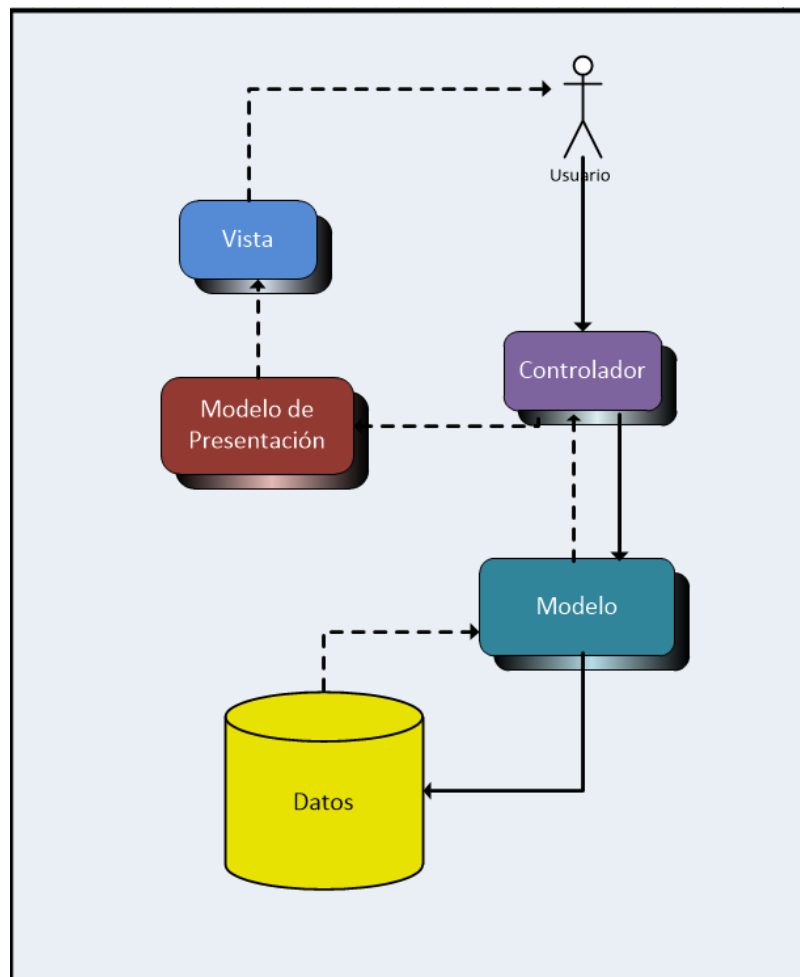


Ilustración 29 - Modelo Vista Controlador

en su mantenibilidad, flexibilidad y robustez.

A continuación, listamos las diferentes clases que tiene el *Open Journal Systems* de front end a back end.

- **Templates Smarty:** Estas Clases son las encargadas de ensamblar las páginas HTML que se muestran a los usuarios.
- **Clases Page:** Reciben los *Requests* del Browser, delegan el procesamiento requerido a varias clases, y llaman a la Template Smarty apropiada, para generar la respuesta.
- **Clases Action:** Son usadas por las clases Page para desempeñar procesamiento no trivial de los requests
- **Clases Model:** Implementan los objetos PHP representando las varias entidades del sistema tales como *Usuarios, Artículos y Revistas*.
- **Data Access Objects (DAO):** Proveen de manera general operaciones CRUD del modelo asociado de datos, son responsables de toda la interacción con la base de datos.
- **Clases de Soporte:** Proveen funcionalidades misceláneas, clases comunes, funciones, etc.

3.1.3 DIAGRAMA DE DESPLIEGUE

A continuación, se muestra como están constituidos los distintos nodos que componen el sistema, la organización de los componentes sobre los nodos.

Un nodo es un recurso de ejecución, puede ser un computador, un dispositivo, procesadores, memoria. Los nodos pueden contener instancias de componentes software como procesos u objetos.

La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

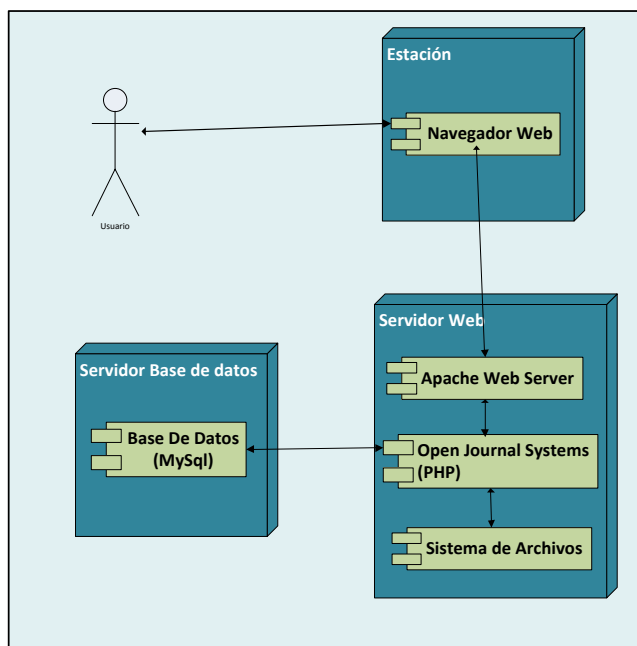


Ilustración 30 - Diagrama de despliegue

A continuación, también podemos ver el diagrama de despliegue del punto SPARQL

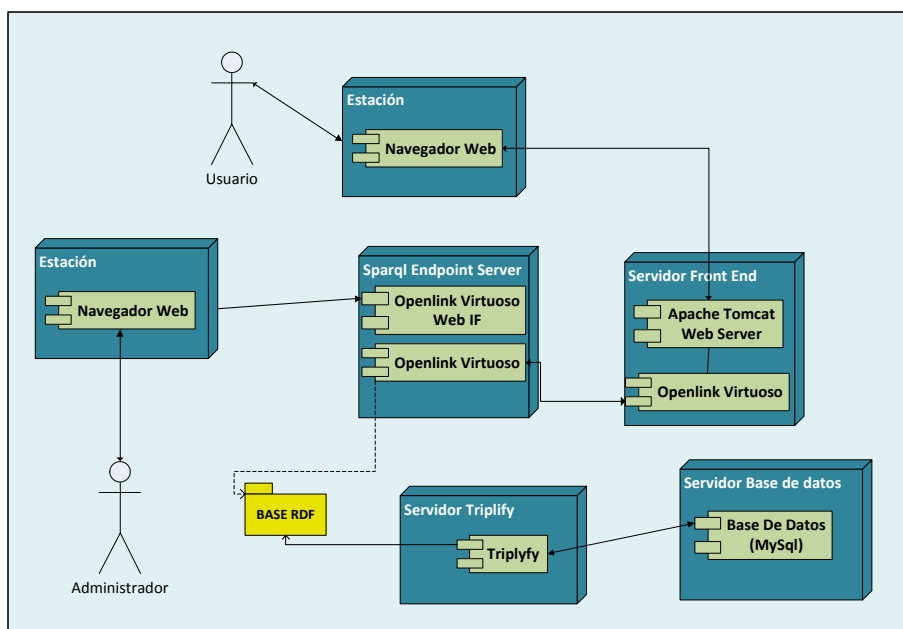


Ilustración 31 - Diagrama de despliegue punto SPARQL

3.1.4 DIAGRAMA DE COMPONENTES [16]

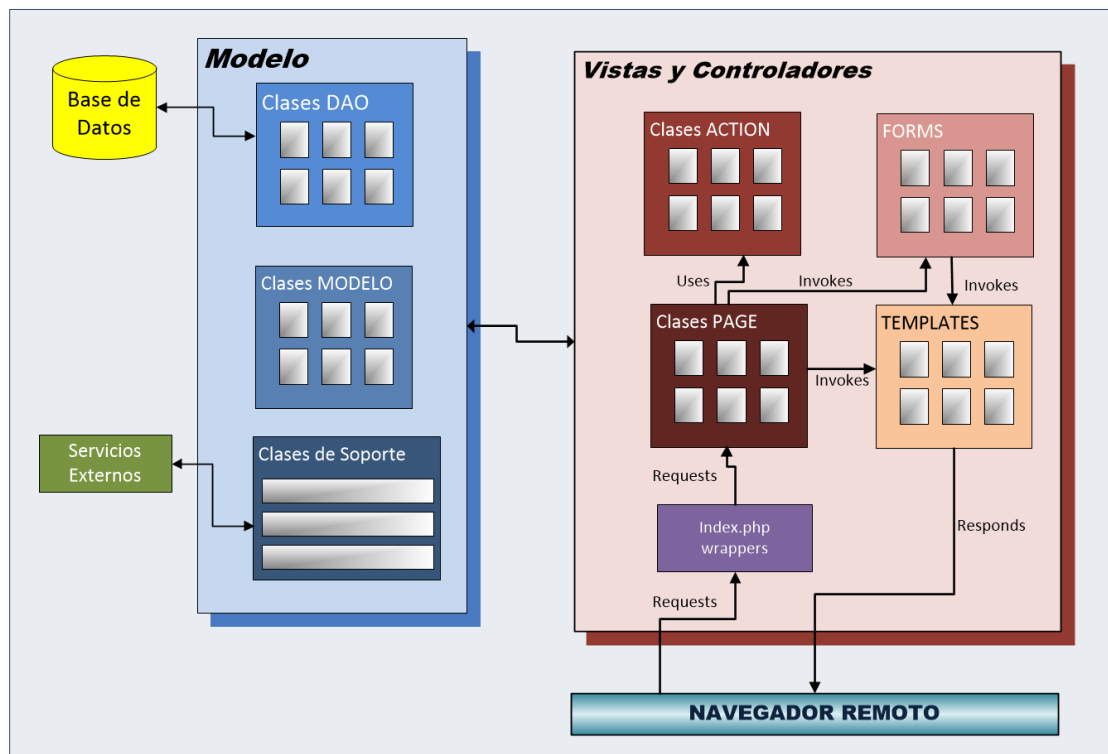


Ilustración 32 - Diagrama de componentes Open Journal Systems

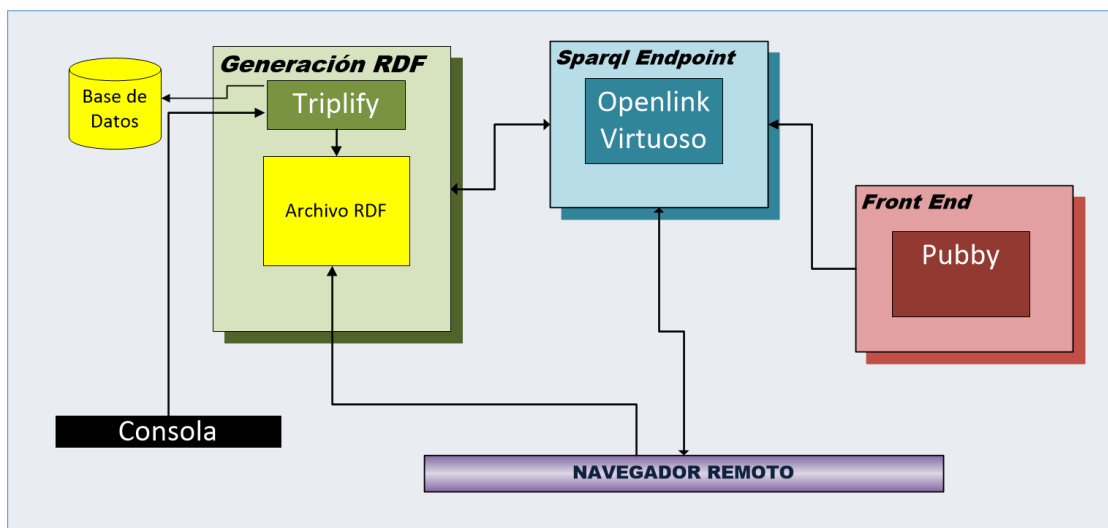


Ilustración 33- Diagrama de componentes SPARQL

3.1.5 COMPONENTES

Se ha dividido el sistema de la Revista en los siguientes módulos:

- Publicación y Administración de la Revista
- Publicación SPARQL

3.1.5.1 Publicación y administración de la revista

Comprende todo lo que es el entorno de la Revista Científica, su ingreso al sistema, la recepción edición y publicación de artículos. En este módulo tenemos los siguientes componentes:

3.1.5.1.1 Navegador web

A continuación, el detalle del componente:

Tabla 17 - Componente navegador web

MODULO	Revista, Publicación SPARQL
COMPONENTE	Navegador Web
ID	CP01
DESCRIPCION	El navegador web es una herramienta que debe ser instalada en la estación del cliente, sirve para que el usuario pueda interactuar con la aplicación dar información y recibir información del aplicativo a través de ella.

3.1.5.1.2 Servidor web

A continuación, el detalle del componente:

Tabla 18 - Componente servidor web

MODULO	Revista
COMPONENTE	Servidor web: Apache web server
ID	CP02
DESCRIPCION	El servidor web es el encargado de alojar el sitio web, brindando una respuesta http hacia los navegadores, se encarga de administrar los http request y envía información HTML hacia el cliente. Este servidor APACHE estará encargado de alojar el sitio de la Revista Científica.

3.1.5.1.3 Open Journal Systems

A continuación, la información general del componente CMS a utilizar

Tabla 19 - Componente Open Journal Systems

MODULO	Revista
COMPONENTE	CMS: Open Journal Systems
ID	CP03
DESCRIPCION	El Open Journal Systems es el CMS encargado de gestionar todo lo relacionado a la revista Científica, desde la autenticación hasta la gestión de artículos y publicación de ediciones, este CMS está hecho en lenguaje de programación PHP, Necesita de mínimo un usuario para poder gestionar todos los módulos correspondientes a

	creación de usuarios y configuración en general.
--	--

3.1.5.1.4 Base de datos revista

A continuación, el cuadro de detalle de componente:

Tabla 20 - Componente de base de datos

MODULO	Revista
COMPONENTE	Base de Datos Revista
ID	CP04
DESCRIPCION	<p>La base de datos es donde se almacena toda la información referente al sistema de la revista, los datos de usuarios, artículos, autores, etc.</p> <p>El motor de base de datos utilizado es MySQL, deberá tener un usuario con acceso de lectura y escritura para que el aplicativo pueda administrar la misma.</p> <p>La base de datos será autogenerada por el CMS, a excepción de las tablas con prefijo ctm, mismas que son diseñadas para la desambiguación de los autores.</p>

3.1.5.1.5 Sistema de Archivos de Almacenamiento

A continuación, el detalle del componente:

Tabla 21 - Componente Sistema de archivos de almacenamiento

MODULO	Revista
---------------	---------

COMPONENTE	Sistema de Archivos de Almacenamiento
ID	CP05
DESCRIPCION	Es un espacio dentro del sistema utilizado para guardar todos los archivos que contienen los respectivos artículos que van a ser publicados en una edición de la revista, cada archivo tendrá una distinta versión, que será guardada de manera jerárquica por el sistema guardando su relación con los autores, revisores, en cada una de las versiones correspondientes.

3.1.5.2 Publicación SPARQL

Ahora se detallará los componentes que se encuentran en el módulo de publicación del punto SPARQL

3.1.5.2.1 Plugin Triplify

Tabla 22 - Componente Plugin Triplify

MODULO	Publicación SPARQL
COMPONENTE	Plugin Triplify
ID	CP06
DESCRIPCION	Triplify es un script PHP que en el presente proyecto se lo utilizará para generar un archivo RDF a partir de la base de datos de la revista, Se deberá proveer con las respectivas configuraciones de

	usuario de la base, nombre de la base de datos, y la respectiva consulta SQL para generar los triples.
--	--

3.1.5.2.2 Servidor SPARQL Virtuoso

A continuación, se muestra el detalle del componente:

Tabla 23 - Componente servidor SPARQL

MODULO	Publicación SPARQL
COMPONENTE	Servidor SPARQL
ID	CP07
DESCRIPCION	El servidor SPARQL se encarga de dos tareas, la una es dar una interfaz web para que el administrador pueda subir un archivo RDF a su entorno, la segunda es poder configurar un punto SPARQL por medio del archivo proporcionado, este archivo tiene que ser un formato <i>RDF-XML</i> o <i>Turtle</i> . ^[1]

3.1.5.2.3 Servidor web Tomcat

El Servlet container para el front end, el detalle de este componente a continuación:

Tabla 24 - Servidor web Tomcat

MODULO	Publicación SPARQL
COMPONENTE	Servidor Web Tomcat

ID	CP08
DESCRIPCION	<p>Este servidor web se encarga de alojar aplicaciones web de tipo java, por tanto, es un requerimiento que el servidor físico donde resida Tomcat tenga instalado el JDK.</p> <p>Apache Tomcat necesita también un espacio físico y permisos de acceso en el sistema operativo.</p>

3.1.5.2.4 Front end para punto SPARQL: Pubby

La herramienta de acceso a datos para visualización de la publicación SPARQL

Tabla 25 - Front end Pubby

MODULO	Publicación SPARQL
COMPONENTE	Front End para el punto SPARQL: Pubby
ID	CP09
DESCRIPCION	<p>El front End permite que el usuario web (no el usuario SPARQL) tenga acceso a los datos publicados en un punto SPARQL, de manera legible por un navegador.</p> <p>Este Front End precisa de dos requisitos, que deben ser configurados en su entorno: Debe existir y tener acceso a un punto SPARQL, en este caso este punto será proporcionado por <i>Openlink Virtuoso</i>. Segundo, necesita de un servidor web Apache Tomcat, ya que este está desarrollado en java.</p>

3.2 PRUEBAS DEL SISTEMA

La fase de pruebas es una de las más importantes del desarrollo del sistema permite que se pueda establecer las técnicas, herramientas y actividades relacionadas con la ejecución y validación de la solución.

En esta fase dispondremos de los recursos y prerrequisitos que deben ser considerados en cada una de las pruebas, esto permitirá que se garantice el cumplimiento de los requerimientos planteados.

Se muestra a continuación cada uno de los casos de prueba

3.2.1.1 Ingreso Al Sistema

Tabla 26 - Caso de Prueba Ingreso al sistema

Caso de Prueba: Ingreso al Sistema	
Nombre del Proceso	Acceso al Sistema
Descripción del Proceso	Ingreso del usuario al sistema
Escenario	El usuario se ingresará en el sistema con el módulo correspondiente.
Entradas	Nombre de Usuario: admin Contraseña: admin01#
Pre-Condiciones	Pantalla Principal

Procedimientos	<ul style="list-style-type: none"> - Se ingresa el nombre de usuario en el campo correspondiente. - Se ingresa la contraseña en el campo "Clave"
Resultado	El sistema muestra que el usuario está identificado
Observaciones	El sistema muestra la pantalla por defecto según el rol del usuario.

3.2.1.2 Configuración de la revista

Tabla 27 - Caso de prueba Configuración de la revista

Caso de Prueba: Configuración de la Revista	
Nombre del Proceso	Configuración de la Revista
Descripción del Proceso	El usuario ingresa los parámetros de configuración de la revista científica
Escenario	El Sistema muestra la pantalla de ingreso de "Detalles" de configuración en la revista
Entradas	<p>Información general</p> <p>Título: Revista Politécnica</p> <p>Iniciales de la Revista: RP</p>

	<p>ISSN-Impresión: 3333-2222</p> <p>Sufijo DOI: "Usar patrón por defecto"</p> <p>Dirección: Av. Toledo y Madrid</p> <p>Contacto Principal</p> <p>Nombre: Revista Politécnica</p> <p>Filiación: Escuela Politécnica Nacional</p> <p>Correo: correo1@gmail.com</p> <p>Teléfono: 2222222</p> <p>Contacto Soporte Técnico</p> <p>Correo: correo1@gmail.com</p> <p>Nombre: Soporte Técnico</p>
Pre-Condiciones	<ul style="list-style-type: none"> - El usuario a probar debe tener rol de administrador - Menú de Navegación "Configuración" - > "Detalles"
Procedimientos	<ul style="list-style-type: none"> - El usuario ingresa los datos correspondientes en cada campo. - El usuario presiona el botón "Guardar"
Resultados	<p>El sistema muestra un mensaje de configuración actualizada</p>

Observaciones	Se puede leer el mensaje de aviso de actualización de configuración una vez terminado el proceso
----------------------	--

3.2.1.3 Crear usuario

Tabla 28 - Caso de prueba Crear usuario

Caso de Prueba: Crear Usuario	
Nombre del Proceso	Administrar Usuario
Descripción del Proceso	<ul style="list-style-type: none"> - El Usuario ingresa los datos de usuario a crear en los campos correspondientes. - Presiona el botón Guardar
Escenario	El sistema muestra la pantalla de Ingreso de Datos de Usuario
Entradas	Tratamiento: Sr Primer Nombre: José Segundo Nombre: Manuel Apellidos: Leiva Género: Masculino Iniciales:JML Inscribir Usuario Como: "Autor"

	<p>Nombre de Usuario: jmleiva</p> <p>Contraseña: jmleiva</p> <p>Correo Electrónico: jmleiva@hotmail.com</p>
Pre-Condiciones	<p>El usuario debe tener perfil de administrador.</p> <p>Navegar Menú “Creación Revista” -> “Crear Usuario”</p>
Procedimientos	<ul style="list-style-type: none"> - El usuario debe ingresar cada uno de los datos a ingresar en el campo correspondiente - Una vez ingresados los datos presionar el botón Guardar
Resultados	<p>El sistema no informa de ningún error y muestra la pantalla principal</p>
Observaciones	<p>A pesar que el sistema no muestre un mensaje de éxito al guardar, el sistema guarda el registro correspondiente, se lo puede apreciar luego en la pantalla de búsqueda</p>

3.2.1.4 Modificar usuario

Tabla 29 - Caso de prueba Modificar usuario

Caso de Prueba: Modificar Usuario	
Nombre del Proceso	Administrar Usuario

Descripción del Proceso	<ul style="list-style-type: none"> - El usuario modifica los datos de un usuario existente - Presiona el botón Guardar
Escenario	El sistema muestra la pantalla de Ingreso de Datos de Usuario
Entradas	<p>Tratamiento: Sra.</p> <p>Primer Nombre: Josefina</p> <p>Segundo Nombre: Manuela</p> <p>Apellidos: Larco</p> <p>Género: Femenino</p> <p>Iniciales:JML</p> <p>Inscribir Usuario Como: "Revisor"</p> <p>Contraseña: jmleiva</p> <p>Correo Electrónico: jmlarco@prueba.com</p>
Pre-Condiciones	<ul style="list-style-type: none"> - El usuario debe tener perfil de administrador. - Navegar Menú "Gestión Revista" -> "Personas Asociadas a esta revista" - Buscar por apellido: Leiva - Elegir usuario: jmleiva -> Editar
Procedimientos	<ul style="list-style-type: none"> - El usuario debe ingresar cada uno de

	<p>los datos a ingresar en el campo correspondiente</p> <ul style="list-style-type: none"> - Una vez ingresados los datos presionar el botón Guardar - Buscar por apellido - Buscar "larco" - El sistema muestra el usuario modificado
Resultados	El sistema muestra el usuario con los datos modificados
Observaciones	Se puede apreciar que los datos de usuario han cambiado luego de realizar el procedimiento

3.2.1.5 Deshabilitar usuario

Tabla 30 - Caso de prueba Deshabilitar usuario

Caso de Prueba: Deshabilitar Usuario	
Nombre del Proceso	Administrar Usuario
Descripción del Proceso	<ul style="list-style-type: none"> - El administrador buscar al usuario a deshabilitar - Sobre el usuario deseado da clic en el link de deshabilitar
Escenario	El sistema muestra la pantalla de Búsqueda de Personas relacionadas

Entradas	nombre de usuario: jmleiva
Pre-Condiciones	<ul style="list-style-type: none"> - El usuario debe tener perfil de administrador. - Navegar Menú “Gestión Revista” -> “Personas Asociadas a esta revista” - Buscar por apellido: Leiva - Elegir usuario: jmleiva -> Desactivar
Procedimientos	<ul style="list-style-type: none"> - Se debe ingresar el nombre de usuario en la búsqueda - En el combo de opciones de búsqueda seleccionar “Nombre de Usuario” - Clic en Desactivar - El sistema preguntará nuevamente si se quiere desactivar dicho usuario - Dar clic en desactivar - El sistema mostrará la pantalla principal
Resultados	El sistema muestra el usuario con datos como desactivado
Observaciones	Luego de ejecutar el proceso se puede apreciar el estado del usuario como desactivado

El resto de casos de prueba se los puede encontrar en el *Anexo 5: Casos de Prueba*

3.3 IMPLANTACIÓN

La solución a implantar consta de algunos componentes necesarios para su funcionamiento entre ellos sitios web, servidores web, navegador y base de datos.

A continuación, se detalla todos los elementos que se necesitan para el correcto funcionamiento de la solución.

3.3.1 REQUISITOS DEL SISTEMA

Los siguientes son los requerimientos de servidor por cada componente, sin embargo, esto no quiere decir que tenga que instalarse cada uno de una manera separada, es decir, bien podría instalarse toda la solución en un mismo servidor, con tal que este cumpla con todos los requisitos establecidos.

3.3.1.1 Revista

- **Sistema Operativo:** Linux Debian 6
- **Lenguaje de Programación:** PHP, versión 4.2
- **Servidor Web:** Apache2

3.3.1.2 Base de datos

- **Sistema Operativo:** Linux Debian 6
- **Espacio en Disco:** 500 MB mínimo

3.3.1.3 Triplify

- **Lenguaje de Programación:** PHP
- **Sistema Operativo:** Linux Debian 6

3.3.1.4 Punto SPARQL Virtuoso

- **Sistema Operativo:** Linux Debian 6

3.3.1.5 Front end SPARQL

- **Sistema Operativo:** Linux Debian 6
- **Servidor Web:** Apache Tomcat
- **Lenguaje de Programación:** Java

3.3.2 PLAN DE INSTALACIÓN

3.3.2.1 Preparación De Software Requerido

3.3.2.1.1 Lenguaje de programación

Para la solución actual se instalarán dos lenguajes de programación PHP y JAVA, PHP es necesario para la mayoría de software a instalar mientras que JAVA se necesita para que corra el front end el punto SPARQL.

3.3.2.1.2 Base de datos

Solo se trabajará con el motor MySql. Este es necesario para guardado de datos de la revista científica digital.

3.3.2.1.3 Servidor web

Se trabajará con dos servidores web, el uno es Apache2 para el poder alojar el sistema web de la Revista, mientras que Apache Tomcat es necesario para poder correr aplicaciones Java, el servidor Front end para el punto SPARQL.

3.3.2.1.4 Gestor de contenido para la revista científica

Es necesario instalar el CMS Open Journal Systems, este sistema se encarga de alojar los artículos y hacer la gestión necesaria de los mismos para su posterior publicación.

3.3.2.1.5 Extractor de datos RDF de la base de Datos

Para poder obtener los datos de la revista en formato RDF se necesitará *Triplify*.

3.3.2.1.6 Servidor para punto SPARQL

Este componente usa el servidor SPARQL *Virtuoso Opensource* para alojar el punto SPARQL.

3.3.2.1.7 Front end para punto SPARQL

Para que los usuarios puedan leer los datos que provee el punto SPARQL se utilizará un software que pueda ser interpretado por un navegador, esta herramienta se llama *Pubby*, para que funcione correctamente *Pubby* necesita que en el servidor donde este se aloja esté instalado Java y Apache Tomcat.

3.3.2.2 Instalación Y Configuración De Software Requerido

A continuación, se muestra los pasos para la instalación del Software que la solución necesita:

- Instalación del Sistema Operativo
- Instalación de PHP
- Instalación de Apache2
- Instalación de MySql Server
- Instalación y configuración del CMS Open Journal Systems
- Instalación y configuración de Triplify
- Instalación y configuración de Virtuoso Opensource
- Instalación del JDK (java)
- Instalación y configuración de Pubby

La guía detallada de Instalación y Configuración se encuentra en el *Anexo 6 Manual de Instalación y Configuración*.

CAPITULO 4 ANÁLISIS DE RESULTADOS

4.1.1 OBTENCIÓN DE INFORMACIÓN

Para poder evaluar los resultados del uso del aplicativo se procede a obtener una encuesta del personal a cargo del sistema.

El objetivo será tener una respuesta por parte del usuario en aspectos del sistema, así se podrá dar un seguimiento a las funciones y elementos de la solución.

A continuación, se presenta el formato de Encuesta con el que se procede a hacer la evaluación a los usuarios del sistema:

A continuación por favor llene las preguntas del 1 al 5, siendo 5 lo mejor (siempre, si, todo, etc.) y siendo 1 lo peor (no, nunca, nada, etc.)

Tabla 31 – Muestra de encuesta

Nro	Pregunta	1	2	3	4	5
1	Recomendaría el software a sus colegas					
2	El software se ejecuta rápidamente					
3	No se detiene de manera repentina					
4	Es fácil de usar					
5	Disfruta su manejo					
6	Le parece que el sistema es intuitivo					

7	Encuentra la ayuda que necesita dentro del sistema					
8	La documentación en línea cumple con la información que necesita					
9	La interfaz es amigable					
10	La información que presenta es coherente					
11	Cumple con lo que necesita el proceso de revista					
12	El sistema está siempre disponible					

Las Encuestas realizadas a los usuarios se las podrá encontrar en el *Anexo 7: Encuestas Realizadas*

4.1.2 EVALUACIÓN DE INFORMACIÓN OBTENIDA

Luego de que los usuarios han llenado la encuesta se procede a calcular cada uno de los puntos en base a una calificación total.

Para esto nos valdremos de la siguiente tabla en donde se puede apreciar cada uno de los puntos analizados, su calificación individual y la calificación general.

Tabla 32 - Resultados encuesta de evaluación

PREGUNTA	USUARIO 1	USUARIO 2	USUARIO 3	TOTAL	CALIFICACIÓN INDIVIDUAL
1	3	4	4	12	80
2	5	5	5	15	100
3	5	5	5	15	100
4	4	4	3	11	73.33
5	4	4	3	11	73.33
6	3	4	3	10	66.67
7	3	5	4	12	80
8	4	5	4	13	86.67

9	4	5	3	12	80
10	4	5	5	14	93.33
11	4	4	3	11	73.33
12	4	5	5	14	93.33
				PORCENTAJE TOTAL	
					83.33

4.1.3 INTERPRETACIÓN DE LOS RESULTADOS

Se puede apreciar que en los aspectos generales el software cumple con los requerimientos de los usuarios, ya que tiene una calificación general de 8.3 sobre 10.

El software no tiene puntos bajos en ningún punto evaluado a excepción del punto 6, los temas de evaluación de software tienen calificaciones altas.

El punto 6 pregunta si el sistema es intuitivo. En base a los resultados esto se cumple medianamente, debido a que el sistema es complejo, una persona con experiencia en procesos de gestión de revistas podría entenderlo más rápido, sin embargo personas con experiencia media o sin ninguna experiencia necesitan una inducción para poder realizar su actividad en el sistema.

Por esta misma complejidad del sistema es que no es totalmente fácil de usar, ya que en si los procesos de la revista son numerosos y sobre todo dependen de algunas personas para su gestión.

Como puntos más altos se puede ver que la efectividad y coherencia del sistema son lo mejor del producto, el brinda veracidad y rendimiento deseados por el usuario.

Se requiere manuales bien explicados de los procesos y capacitación, para disminuir los tiempos de familiarizarse con el sistema.

Al momento el sistema ya se encuentra en funcionamiento un tiempo y tiene gran acogida, al momento se está trabajando en un proyecto que cuenta con conteo de visitas al sitio.

CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El software implantado cumple con todos los requerimientos que necesita un sistema de gestión para revistas científicas.
- El Acceso abierto permite el acceso a artículos de parte de los lectores aprovechando el alcance que los mismos tienen a internet.
- Los Artículos ingresados en la revista cumplen con los datos necesarios para utilizar un vocabulario *dublin core* para su indexación semántica.
- El uso de un software para la administración de los procesos de la revista permite que esta tenga más facilidad para cumplir sus procesos de edición publicación y difusión.
- El formato RDF y el lenguaje de consulta SPARQL permiten la correcta interconexión entre distintas fuentes de datos con información entendible a nivel de software.
- Debido a la complejidad del proceso de recepción, edición y publicación de artículos, la familiarización de usuarios con el sistema requiere un tiempo.
- El Sistema de la revista maneja una arquitectura MVC, Modelo Vista Controlador, esto permite la correcta división del software a través de módulos. Separando una capa de la otra.
- La implementación del Open Journal Systems permite el almacenamiento de información de autores y artículos, sin embargo existe en su base de datos una duplicación de autores al momento de asignar mas de un autor a un artículo, para la publicación de datos enlazados se procedió a implementar procedimientos de desambiguación de autores, dichos procedimientos solo afectan a la publicación SPARQL, más no al sistema de la revista en si.

5.2 RECOMENDACIONES

- Se recomienda que los editores de la revista definan los criterios de corrección para los artículos y los publiquen, debido a que las correcciones se basan en herramientas como versiones de corrección de Microsoft Word.
- Especificar las políticas de uso de los artículos de la revista de acuerdo a normativas acceso abierto.
- Actualizar el grafo de datos RDF en cada publicación de la revista.
- Es recomendable que la persona a cargo de la extracción de datos esté familiarizada con el lenguaje de consulta SQL debido a que el manejo de la extracción se basa en este tipo de consultas a la base.
- Se recomienda hacer respaldos periódicos de la información guardada por parte de la revista tanto en el sistema de archivos como en la base de datos.
- Si algún momento se planea una actualización el administrador deberá hacer una migración de tablas personalizadas sobre la base de la revista y analizar cómo realizar la migración de datos de una manera conveniente.

BIBLIOGRAFIA

1. Becket, D., & Berners Lee, T. (25 de febrero de 2014). *RDF 1.1 Turtle*.
Obtenido de <http://www.w3.org/TR/turtle>
2. Bizer, C., & Cyganiak, R. (13 de mayo de 2014). *Pubby - A Linked Data Frontend for SPARQL Endpoints*. Obtenido de <http://wifo5-03.informatik.uni-mannheim.de/pubby/>
3. Carpenter, L. (2 de abril de 2014). *OAI para principiantes*. Obtenido de <http://travesia.mcu.es/portalanb/jspui/html/10421/1823/intro.htm>
4. Crawford, W. (2011). *Open Access What You Need To Know Now*. Chicago: American Library Association.
5. Dietzold, S. (24 de abril de 2014). *Triplify - About*. Obtenido de Triplify: <http://triplify.org/About#h1-9>
6. Dublin Core Metadata Initiative (DCMI). (5 de abril de 2014). *www.dublincore.org*. Obtenido de Dublin Core: <http://dublincore.org/>
7. Heath, T., & Bizer, C. (2011). *Linked Data Envolving The Web Into A Global Data Space*. Morgan & Claypool.
8. Hidalgo Navarro, A. (4 de abril de 2014). *Que es un "corrector de estilo"*. Obtenido de <http://www.uv.es/ahidalgo>
9. Inter-Research;. (22 de Agosto de 2014). *Electronic Publishing in Science: Changes and Risks*. Obtenido de <http://www.int-res.com/abstracts/meps/v180/editorial/>
10. Kroll, P., & Krutchen, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. Addison Wesley.
11. MySQL. (9 de agosto de 2014). *What is MySQL*. Obtenido de MySQL: <http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>

12. Open Archives Initiative. (25 de febrero de 2014). *Open Archives Initiative*.
Obtenido de <https://www.openarchives.org/>
13. Paskin, N. (12 de mayo de 2014). *Digital Object Identifier (DOI) System*.
Obtenido de DOI: https://www.doi.org/overview/DOI_article_ELIS3.pdf
14. PHP Development Team. (22 de marzo de 2014). *¿Qué es PHP? - Manual*.
Obtenido de <http://php.net/manual/es/intro-what-is.php>
15. Public Knowledge Project. (10 de abril de 2014). *Developer Documentation - PKP Wiki*. Obtenido de
https://pkp.sfu.ca/wiki/index.php?title=Developer_Documentation
16. Public Knowledge Project. (n.d.). *Open Journal Systems - Technical Reference*. California: Simon Fraser University Library.
17. Seaden, M. (13 de mayo de 2014). *Introducing Linked Data And The Semantic Web*. Obtenido de <http://www.linkeddatatools.com/semantic-web-basics>
18. The Apache Software Foundation. (2014, 07 15). *Apache - Httpd Server Project*. Retrieved from http://httpd.apache.org/ABOUT_APACHE.html
19. Torres Salinas, D., & Cabezas Clavijo, A. (22 de agosto de 2014). *Como publicar en revistas científicas de impacto: Consejos y reglas sobre publicación*. Obtenido de [https://ec3metrics.com/wp-content/uploads/2013/07/2-ART%
c3%8dCULO-C%c3%b3mo-Publicar-en-Revistas-Cient%
c3%adficas-de-Impacto_Consejos-y-Reglas-sobre-Publicaci%
c3%b3n-Cient%
c3%adfica.pdf](https://ec3metrics.com/wp-content/uploads/2013/07/2-ART%c3%8dCULO-C%c3%b3mo-Publicar-en-Revistas-Cient%c3%adficas-de-Impacto_Consejos-y-Reglas-sobre-Publicaci%c3%b3n-Cient%c3%adfica.pdf)
20. Willinsky, J., Stranack, K., Smecher, A., & MacGregor, J. (2010). *Open Journal Systems: A Complete Guide to Online Publishing* (2nd ed.). California: Simon Fraser University Library.

GLOSARIO

Revisión Por Pares: Método usado para validar trabajos escritos y solicitudes de financiación con el fin de evaluar su calidad, originalidad, factibilidad, rigor científico, etcétera, antes de su publicación.

Formato ODT: es un formato de archivo abierto y estándar para el almacenamiento de documentos ofimáticos tales como hojas de cálculo, textos, gráficas y presentaciones.

URI: Es una cadena de caracteres que identifica los recursos de una red de forma unívoca.

Servlet Container: Servidor de aplicaciones que proporciona la funcionalidad de correr Java Servlets.

Servlet: es una clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web

Punto SPARQL: Servicio que dispone acceso a datos a través de lenguaje SPARQL tanto para lectura como actualización.

SPARQL: Es un acrónimo recursivo del inglés SPARQL Protocol and RDF Query Language. Se trata de un lenguaje estandarizado para la consulta de grafos RDF.

Request: En un modelo cliente-servidor, un request es un requerimiento, petición o solicitud que le hace un cliente a un servidor. Por lo general los request suelen hacerse a través de una red

CMS: Content Management System, Sistema de Gestor de contenidos, permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web, por parte de los administradores, editores, participantes y demás usuarios.

SQL: es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

Triples: En web semántica representa la relación entre 3 elementos, sujeto, objeto y predicado.

Rdf-Xml: Formato de representación de un grafo RDF mediante lenguaje XML.

Turtle: Formato de representación de un grafo RDF mediante un lenguaje sencillo a través de declaraciones simples (y repetitivas) de sujeto, predicado y objeto.