

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ**

**IMPLEMENTACIÓN DE UN SOFTWARE PARA BÚSQUEDA DE
PUBLICACIONES CIENTÍFICAS EN BASES DE DATOS
ESTRUCTURADAS MEDIANTE DATOS ENLAZADOS**

Tesis para optar por el Título de Ingeniero Informático que presenta el bachiller:

Iván Renato Uribe Canchanya

ASESOR: Dr. Héctor Andrés Melgar Sasieta

Lima, febrero del 2018

ANEXOS

Contenido

1. Anexo1: Documento de Análisis.....3
2. Anexo2: Documento de Arquitectura de Software.....10



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
ESPECIALIDAD DE INGENIERÍA INFORMÁTICA



SGPUCP

Documento de Análisis y Diseño
Versión 1.0

PROPUESTO POR: Iván Uribe ivan.uribe@pucp.edu.pe

ELABORADO POR: Iván Uribe ivan.uribe@pucp.edu.pe

Lima, 08 de mayo del 2017

Historial de Revisiones

Historial de revisiones				
Ítem	Fecha	Versión	Descripción	Equipo
1	05/04/2017	1.0	Versión inicial.	Iván Uribe



Tabla de Contenido

1. Objetivo	6
2. Diagrama de Clases	6
3. Diccionario de Datos	7



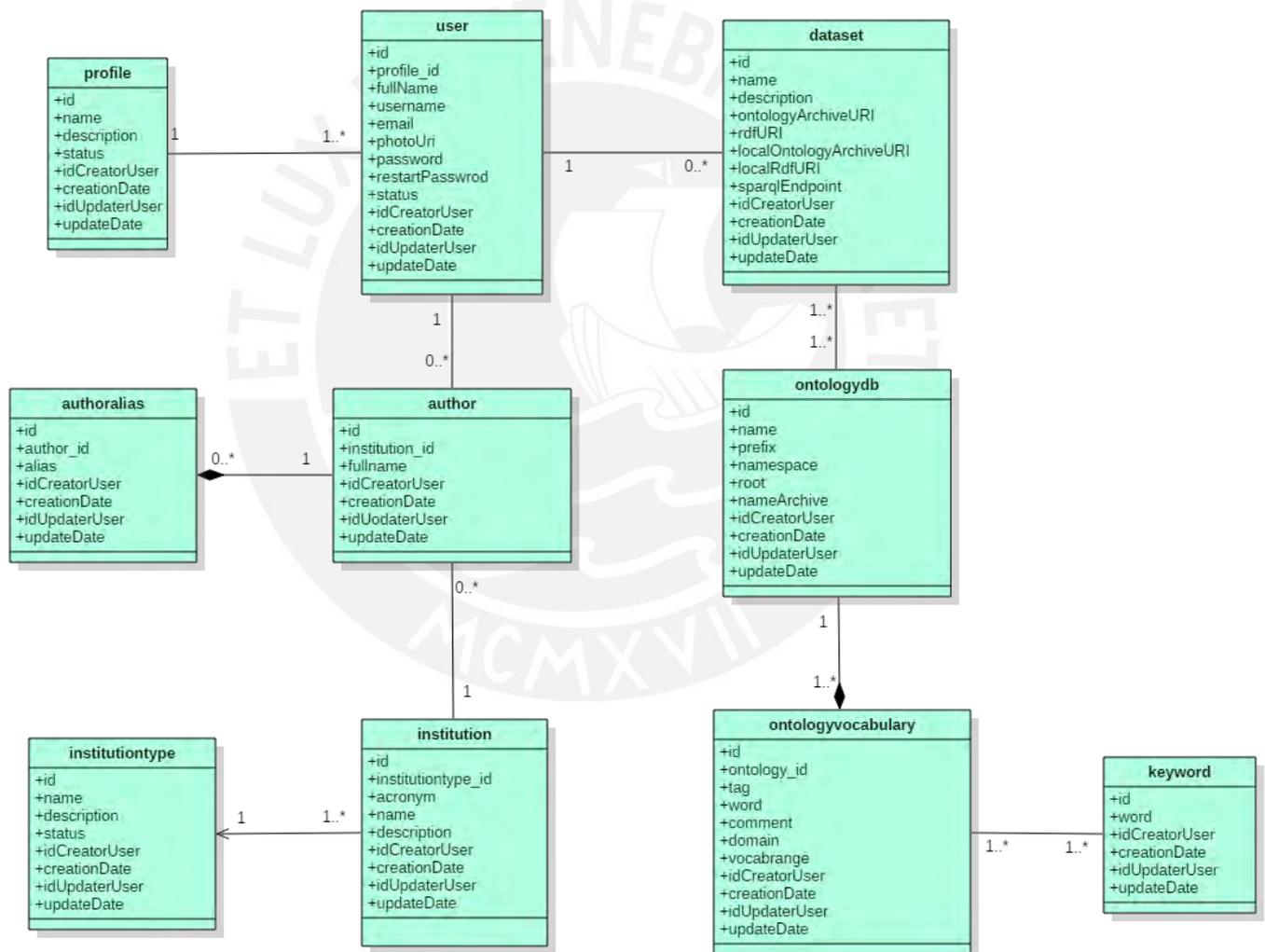
Objetivo

Este documento tiene como propósito principal describir y representar las distintas relaciones que existen entre las entidades que definirán el sistema a ser implementado.

El documento consta de dos partes:

- Diagrama de clases: relación entre las clases propuestas
- Diccionario de datos: descripción de cada clase y sus atributos.

Diagrama de Clases



Diccionario de Datos

Tabla	Descripción	Campo	Tipo	Longitud	Descripción
Profile	Entidad que representa a los distintos perfiles que se manejan para los usuarios del sistema.	id	numeric	20,0	Código interno.
		name	varchar	100	Nombre del perfil.
		description	varchar	250	Detalle del perfil.
		status	varchar	10	Estado actual: Activo o Inactivo.
		idCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos del perfil.
		creationDate	datetime		Fecha del primer ingreso de los datos del perfil.
		idUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos del perfil.
		updateDate	datetime		Fecha de última actualización de los datos del perfil.
User	Entidad que representa a cualquier usuario del sistema.	id	numeric	20,0	Código interno.
		profile_id	numeric	20,0	Código del perfil del usuario.
		fullName	varchar	100	Nombre personal del usuario.
		username	varchar	50	Nombre de usuario en el sistema.
		email	varchar	50	Correo del usuario.
		photoUri	varchar	250	ruta de la foto del usuario
		password	varchar	50	Contraseña para el inicio de sesión.
		restartPassword	varchar	1	Indicador para restablecimiento de contraseña
		status	varchar	10	Estado actual: Activo o Inactivo.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos del usuario.
		CreationDate	datetime		Fecha del primer ingreso de los datos del usuario.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos del usuario.
		UpdateDate	datetime		Fecha de última actualización de los datos del usuario.
Institutiontype	Entidad que representa a los tipos de instituciones.	id	numeric	20,0	Código interno.
		name	varchar	100	Nombre de la institución.
		description	varchar	250	Descripción del tipo de institución.
		status	varchar	12	Fecha requerida para el movimiento
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos del tipo de institución.
		CreationDate	datetime		Fecha del primer ingreso de los datos del tipo de institución.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos del tipo de institución.
		UpdateDate	datetime		Fecha de última actualización de los datos del tipo de institución.
Institution	Entidad que representa a las instituciones.	id	numeric	20,0	Código interno.

		institutiontype_id	numeric	20,0	Código del tipo de institución.
		acronym	varchar	40	Siglas de la institución
		name	varchar	1	Nombre de la institución.
		description	varchar	250	Descripción de la institución
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos de la institución.
		CreationDate	datetime		Fecha del primer ingreso de los datos de la institución.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos de la institución.
		UpdateDate	datetime		Fecha de última actualización de los datos de la institución.
Author	Entidad que representa a los autores.	id	numeric	20,0	Código interno.
		institution_id	numeric	20,0	Código de la institución del autor.
		fullname	varchar	200	Nombre del autor.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos del autor.
		CreationDate	datetime		Fecha del primer ingreso de los datos del autor.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos del autor.
		UpdateDate	datetime		Fecha de última actualización de los datos del autor.
Authoralias	Entidad que representa a la lista de alias que puede tener un autor.	id	numeric	20,0	Código interno.
		author_id	varchar	20,0	Código del autor al que pertenece el alias.
		alias	varchar	100	Alias del autor.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos del alias.
		CreationDate	datetime		Fecha del primer ingreso de los datos del alias.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos del alias.
		UpdateDate	datetime		Fecha de última actualización de los datos del alias.
dataset	Entidad que representa a un dataset.	id	numeric	20,0	Código interno.
		name	varchar	100	Nombre del dataset.
		description	numeric	250	Descripción del dataset.
		ontologyArchiveURI	numeric	250	URI de internet donde se ubica el archivo RDF de la ontología.
		rdfURI	varchar	250	URI de internet donde se ubica el dataset RDF.
		localOntologyArchiveURI	varchar	250	URI local donde se ubica el archivo RDF de la ontología.
		localRdfURI	datetime	250	URI local donde se ubica el archivo RDF (dataset).
		sparqlEndPoint	datetime	250	URI del sparql Endpoint.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos del dataset.
		CreationDate	datetime		Fecha del primer ingreso de los datos del dataset.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos del dataset.

		UpdateDate	datetime		Fecha de última actualización de los datos del dataset.
ontologydb	Entidad que representa a una ontología.	id	numeric	20,0	Código interno.
		name	varchar	100	Nombre de la ontología.
		prefix	numeric	250	Prefijo de la ontología.
		namespace	numeric	250	URI del espacio de nombres (prefijo).
		root	varchar	250	Ruta raíz donde se ubica el archivo de la ontología.
		namearchive	varchar	250	Nombre del archivo de la ontología.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos de la ontología.
		CreationDate	datetime		Fecha del primer ingreso de los datos de la ontología.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos de la ontología.
UpdateDate	datetime		Fecha de última actualización de los datos de la ontología.		
ontology vocabulary	Entidad que representa el vocabulario de una ontología.	id	numeric	20,0	Código interno.
		ontology_id	numeric	20,0	Código de la ontología a la que pertenece la etiqueta.
		tag	varchar	30	Etiqueta (prefijo y vocablo).
		word	varchar	30	Vocablo de la etiqueta.
		domain	varchar	30	Dominio de la etiqueta.
		vocabrang	varchar	30	Rango de la etiqueta.
		comment	varchar	150	Comentarios sobre la etiqueta.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos de la ontología.
		CreationDate	datetime		Fecha del primer ingreso de los datos de la ontología.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos de la ontología.
UpdateDate	datetime		Fecha de última actualización de los datos de la ontología.		
keyword	Entidad que representa al corpus de palabras claves.	id	numeric	20,0	Código interno.
		word	varchar	50	Palabra clave.
		IdCreatorUser	numeric	20,0	Código interno del usuario que ingresó por primera vez los datos de la ontología.
		CreationDate	datetime		Fecha del primer ingreso de los datos de la ontología.
		IdUpdaterUser	numeric	20,0	Código interno del usuario que actualizó por última vez los datos de la ontología.
		UpdateDate	datetime		Fecha de última actualización de los datos de la ontología.

**FACULTAD DE CIENCIAS E INGENIERÍA
ESPECIALIDAD DE INGENIERÍA INFORMÁTICA**



SGPUCP

**Documento de Arquitectura de Software
Versión 1.0**

PROPUESTO POR: Iván Uribe ivan.uribe@pucp.edu.pe
ELABORADO POR: Iván Uribe ivan.uribe@pucp.edu.pe

Lima, 08 de mayo del 2017

Tabla de Contenido

1.	Introducción	12
1.1.	Propósito	12
1.2.	Alcance	12
1.3.	Definiciones, Acrónimos y Abreviaturas	12
1.3.1.	Definiciones	12
1.3.2.	Acrónimos	12
1.4.	Visión General del Documento	13
2.	Representación de la Arquitectura	13
3.	Metas y Restricciones de la Arquitectura	14
3.1.	Metas	14
3.2.	Restricciones	14
4.	Vista de Casos de Uso	15
4.1.	Realizaciones de Casos de Uso	15
4.1.1.	Mantener usuarios	15
4.1.2.	Mantener autores	15
4.1.3.	Mantener Instituciones	15
4.1.3.	Mantener Datasets	16
4.1.4.	Mantener Ontologías	16
4.1.5.	Mantener Vocabulario Ontológico	16
4.1.6.	Consultar publicaciones	16
4.1.7.	Consultar estadísticas	17
5.	Vista Lógica	17
5.1.	Descripción	17
5.2.	Paquetes de Diseños Significativos	18
5.2.1.	Paquete Interfaz de Usuarios	18
5.2.2.	Paquete Controller	18
5.2.3.	Paquete Model	18
5.2.4.	Paquete Service	19
5.2.5.	Paquete Util	19
6.	Vista de Despliegue	19
7.	Vista de Implementación	20
7.1.	Descripción	20
7.2.	Niveles o Capas	21
8.	Vista de Datos	22
9.	Tamaño y Performance	23
10.	Calidad	23

1. Introducción

1.1. Propósito

El presente documento tiene como propósito proporcionar una vista de la arquitectura básica del sistema de gestión de publicaciones científicas peruanas que se va a desarrollar; de esta manera, se describirá la estructura de alto nivel del sistema y sus propiedades globales. A lo largo del documento, se mostrará una serie de vistas arquitectónicas, las cuales servirán para representar los diferentes aspectos del sistema.

1.2. Alcance

El documento pretende explicar los principios arquitectónicos sobre los cuales se llevará a cabo el desarrollo del sistema SGPUCP. Entre estos principios arquitectónicos, se encuentran los modelos y descripciones correspondientes a cada una de las vistas arquitectónicas, tales como la vista de casos de uso, la vista lógica, la vista de implementación y la de despliegue. Además, se presenta el modelo de base de datos propuesto para almacenar la data generada en los procesos core del software.

1.3. Definiciones, Acrónimos y Abreviaturas

1.3.1. Definiciones

- StarUML: Herramienta que permite realizar el modelado de los diferentes diagramas presentados en el documento.
- HTML: Lenguaje de marcado de hipertexto; estándar para la elaboración de páginas web.
- Jena: Framework para aplicaciones que manejan Datos Enlazados.
- Spring: Framework para desarrollo de aplicaciones web.

1.3.2. Acrónimos

- RUP: Rational Unified Process
- UML: Unified Modeling Language
- POO: Programación Orientada a Objetos
- API: Application Programming Interface
- JPA: Java Persistence API
- ERS: Especificación de Requisitos de Software
- BD: Base de Datos
- MVC: Modelo - Vista - Controlador

- LAN: Local Area Network

1.4 Visión General del Documento

El documento muestra de manera clara la arquitectura propuesta para la implementación de los casos de uso escogidos para el sistema. El lenguaje utilizado es sencillo y preciso; además, a lo largo del documento se presentan gráficos y vistas de acuerdo a la metodología escogida.

2. Representación de la Arquitectura

Los estilos arquitectónicos que se seguirán para la implementación del sistema son:

- **Orientado a objetos**

El software será desarrollado mediante un enfoque orientado a objetos, a través del uso de Java y el framework "Play".

- **Organizado por capas**

El software será desarrollado mediante el uso del patrón arquitectónico MVC.

- **Cliente-Servidor**

El software tendrá un servidor de aplicaciones centralizado, el cual almacenará la aplicación web. El cliente será el navegador web en cualquier computadora conectada a la página de la empresa.

- **Repositorio**

Toda la data generada durante los procesos core será guardada en una base de datos centralizada (MySQL).

- **Intérpretes**

Se utilizará la máquina virtual de Java (JVM) para compilar el código fuente; y el lenguaje HTML para representar la página web en los distintos navegadores de manera casi idéntica.

- **Modelo "4+1"**

Este modelo será utilizado para documentar la arquitectura del software a implementar. Dicho modelo incluye las siguientes vistas: lógica, de implementación, de procesos, de despliegue y de casos de uso.

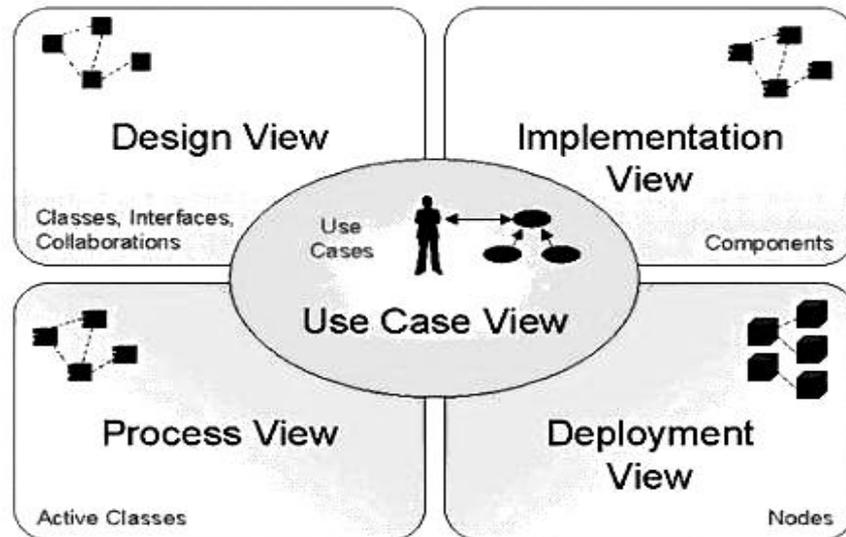


Figura 1: Modelo "4+1", el cual incluye las vistas lógica, de implementación, de procesos, de despliegue y de casos de uso.

3. Metas y Restricciones de la Arquitectura

La meta principal de la arquitectura propuesta para el sistema a desarrollar es representar los aspectos y componentes que influirán a lo largo del ciclo de vida del desarrollo del software.

3.1 Metas

- Los usuarios podrán acceder al sistema desde cualquier computadora con conexión a internet por medio de un navegador web.
- Para mantener el control de seguridad de acceso, los usuarios requerirán de un nombre de usuario y contraseña personalizados.
- Dependiendo del perfil de cada usuario se habilitará, o no, determinadas funcionalidades con el fin de abarcar cada una de las funcionalidades de acuerdo a la necesidad de cada rol de usuario.
- Todos los requerimientos definidos en el catálogo de requisitos deberán formar parte de la arquitectura propuesta para el sistema.

3.2 Restricciones

El sistema será soportado por los siguientes navegadores:

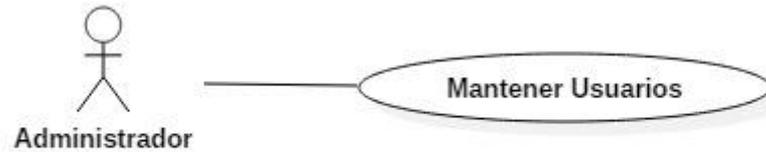
- Google Chrome
- Mozilla Firefox (desde la versión 10.4 en adelante)
- Internet Explorer (desde la versión 11.0 en adelante)

4. Vista de Casos de Uso

4.1. Realizaciones de Casos de Uso

4.1.1. Mantener usuarios

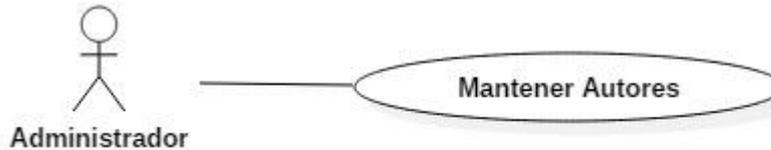
Esta funcionalidad permitirá realizar la consulta de usuarios viendo sus datos específicos, la creación de un usuario nuevo, y la modificación de los datos de un usuario. Esto puede realizarlo el administrador.



Caso de uso Mantener usuarios del paquete Administración del Sistema.

4.1.2. Mantener autores

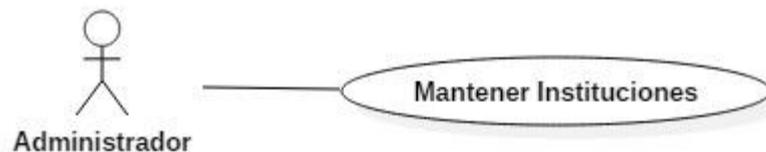
Esta funcionalidad permite al administrador realizar la consulta, actualización y creación de autores peruanos.



Caso de uso Mantener Autores del paquete Administración del Sistema.

4.1.3 Mantener Instituciones

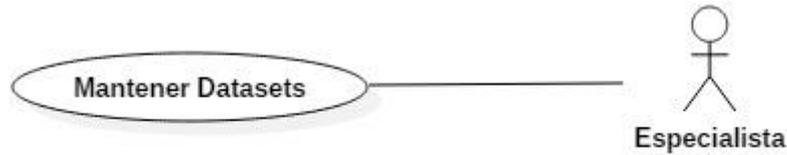
Esta funcionalidad permitirá registrar instituciones nacionales a los que están asociados los autores. Se podrá consultar, actualizar y agregar nuevas instituciones, así como el tipo de institución.



Caso de uso Mantener Instituciones del paquete Administración del Sistema.

4.1.3. Mantener Datasets

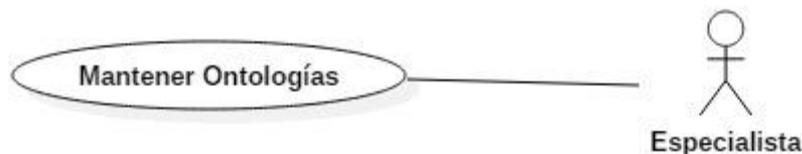
Esta funcionalidad permitirá al especialista poder mantener una lista de datasets relacionados a una ontología.



Caso de uso Mantener Datasets del paquete Mantenimientos Especializados.

4.1.4. Mantener Ontologías

Esta funcionalidad permite al especialista poder mantener una lista de ontologías presentes en la web y que son relevantes para el dominio de publicaciones y autores.



Caso de uso Mantener Ontologías del paquete Mantenimientos Especializados

4.1.5. Mantener Vocabulario Ontológico

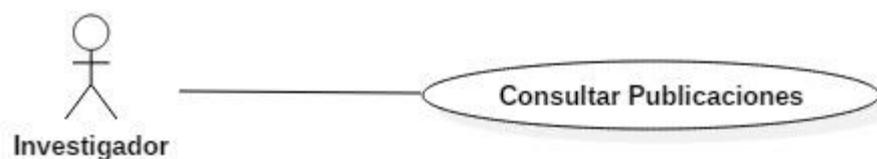
Esta funcionalidad permite al especialista mantener un vocabulario de recursos (propiedades y clases) de las ontológicas, el cual sirve como input para el procesamiento de búsqueda de información en los datasets.



Caso de uso Mantener Vocabulario del paquete Mantenimientos Especializados

4.1.6. Consultar publicaciones

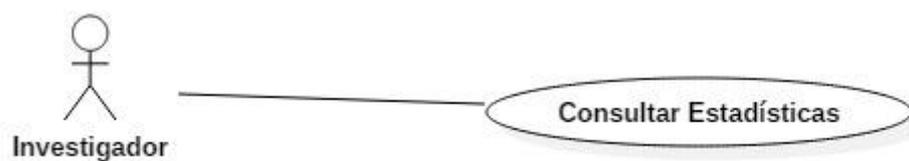
El caso de uso permitirá la consulta de publicaciones por autor. Se visualizará la información relevante (metadatos) de cada publicación y, de ser posible, el documento de la publicación.



Caso de uso Consultar Publicaciones del paquete Consultas.

4.1.7. Consultar estadísticas

El caso de uso permitirá generar un reporte estadístico sobre las publicaciones de los autores por año, así como las publicaciones más citadas de los autores. El actor involucrado es el investigador.



Caso de uso Consultar Estadísticas del paquete Consultas.

5. Vista Lógica

5.1. Descripción

La vista lógica proporciona la base para la comprensión de la estructura interna y organización de los elementos que conformarán el software a implementar.

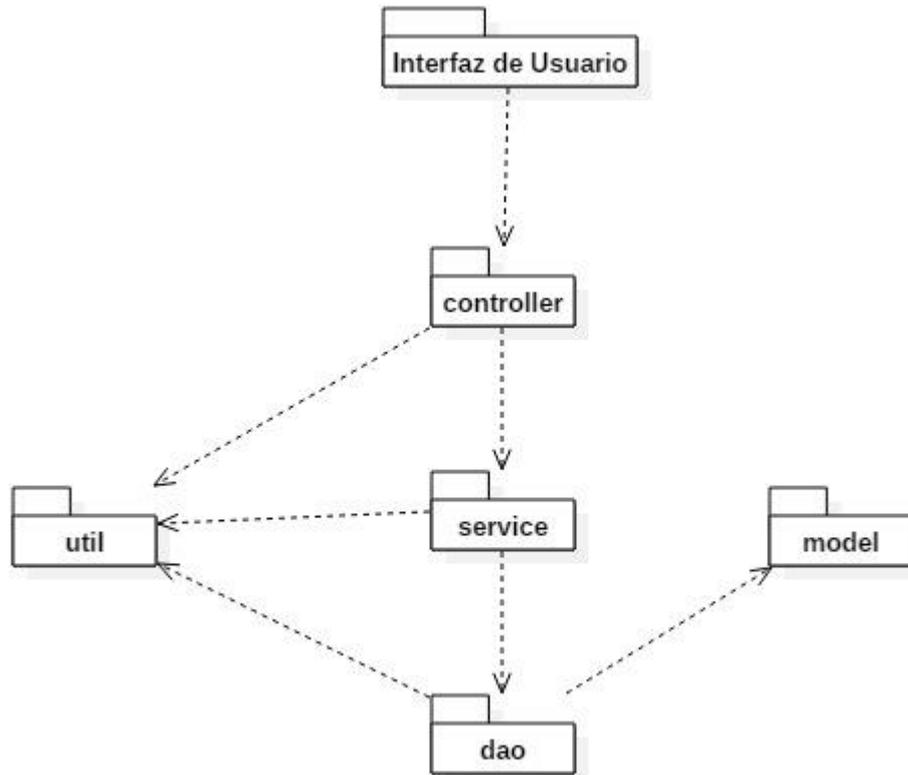


Figura 2: Vista lógica que muestra la estructura interna y organización de los elementos que conformarán el software.

5.2. Paquetes de Diseños Significativos

5.2.1. Paquete Interfaz de Usuarios

La interfaz de usuario contiene los archivos HTML de cada una de las páginas web que formarán parte del sistema. Esta capa provee cada una de las vistas a través de las cuales el usuario podrá interactuar y comunicarse con el sistema. Este paquete depende de los controladores.

5.2.2. Paquete Controller

Esta capa contiene todas las clases controladoras que representan los casos de uso y manejan el comportamiento del software. Esta capa provee la comunicación entre la interfaz y las entidades del dominio.

5.2.3. Paquete Model

Este paquete contiene todas las clases que representan a las entidades que componen el dominio del sistema; tales como autor, institución, ontología, dataset, entre otros.

5.2.4. Paquete Service

El paquete Service es la capa Middleware que proporciona soporte para el acceso e interacción con la base de datos (BD). Provee la comunicación entre las entidades de la aplicación y la BD. Para el sistema a implementar, el Middleware será el JPA, el cual estará bajo el control e implementación del framework Hibernate.

5.2.5. Paquete Util

Este paquete brinda funciones utilitarias que son utilizadas por los controladores. Asimismo, se encuentran las funcionalidades implementadas bajo el framework de Jena para el manejo de Ontologías y Grafos RDF.

6. Vista de Despliegue

En esta vista se detalla la disposición física de los elementos y nodos que formarán parte del sistema a desarrollar; así como la distribución de los componentes sobre los nodos.

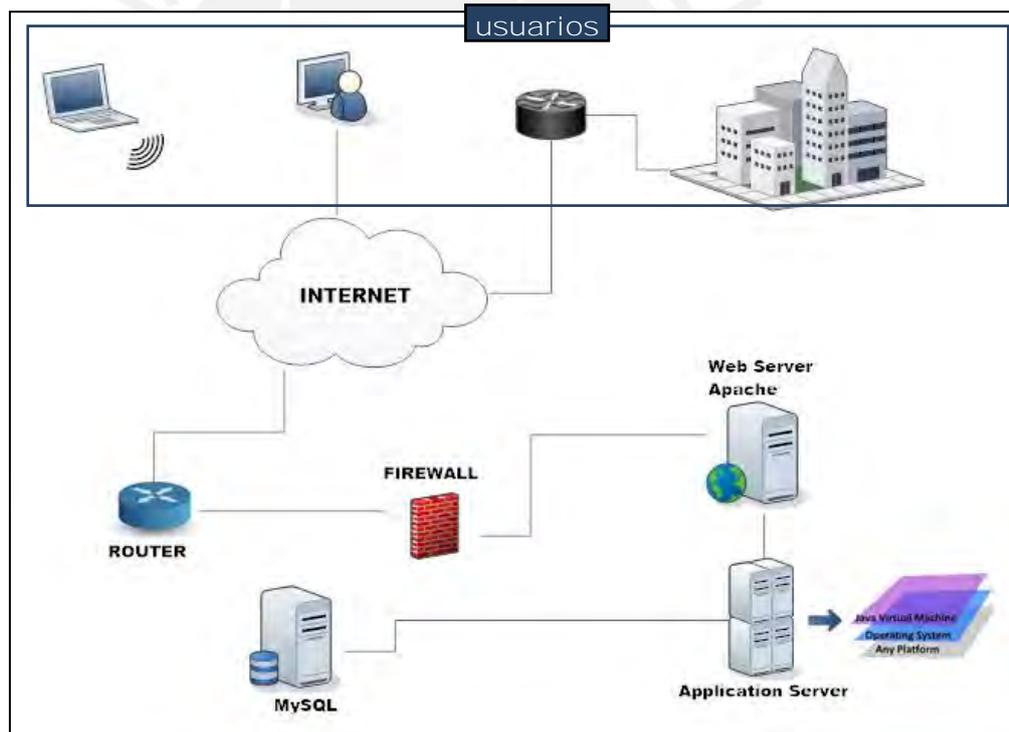


Figura 3: Diagrama de despliegue que muestra la disposición física de los elementos y nodos que formarán parte del sistema.

- **Usuarios**

Todos los usuarios podrán acceder al sistema a través de un navegador en una computadora con conexión a internet. Según el perfil asignado,

cada usuario tendrá limitaciones en cuanto a las acciones que pueda realizar dentro del sistema.

- **Web Server Apache**

Es el servidor web para la aplicación, donde se manejará cada una de las páginas web, así como los servicios que conforman todo el sistema.

- **Servidor de Aplicaciones**

Es el servidor que almacena el software a ser implementado. Toda PC cliente se conecta al sistema mediante protocolos TCP/IP (Internet).

- **Servidor de Base de Datos**

Es el servidor que contiene la base de datos del sistema. El software accede a este servidor mediante una red LAN.

7. Vista de Implementación

7.1. Descripción

Esta vista describe la manera en la cual será implementada la arquitectura propuesta para el sistema.

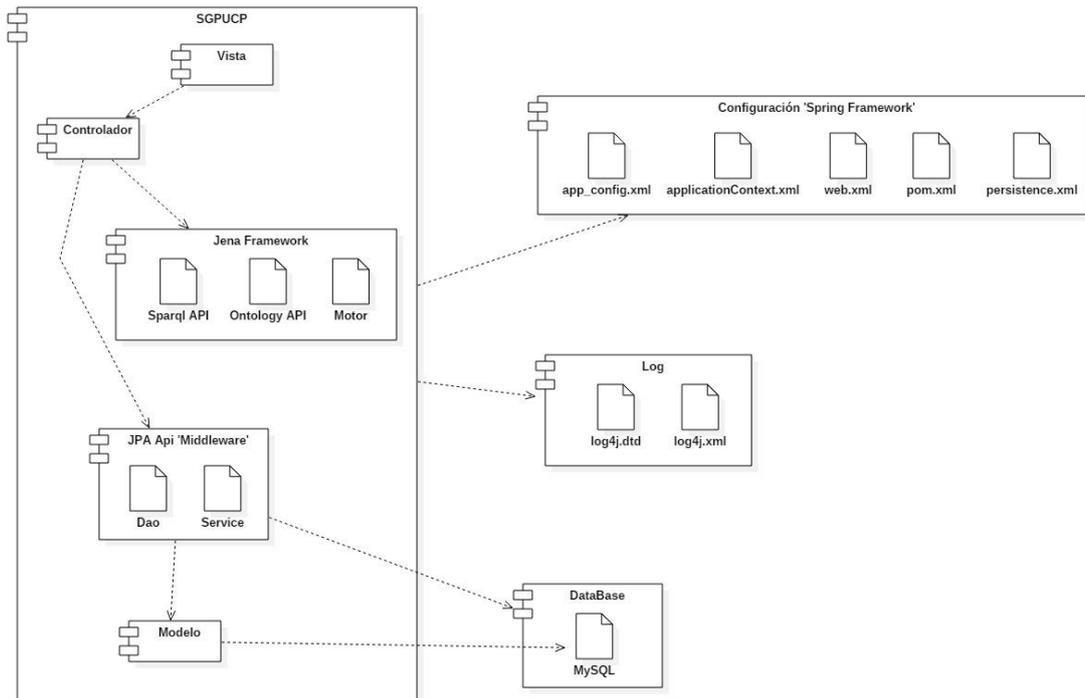


Figura 4: Vista de implementación que muestra la manera en la cual será implementada la arquitectura propuesta para el sistema.

7.2. Niveles o Capas

7.2.1. Resources

Este componente contiene las librerías usadas para las interfaces gráficas del sistema. Incluye el framework de Bootstrap para las hojas de estilo (CSS) del html, y JQuery.

7.2.2. Configuration

Este componente contiene 3 archivos necesarios para la configuración del sistema:

- **web.xml**

Es el archivo de configuración principal para la aplicación. Especifica el contexto de la aplicación.

- **app-config.xml**

Es el archivo provee la configuración para el mapeo de los componentes de software.

- **applicationContext.xml**

Este archivo provee la configuración del JPA y las propiedades de conexión a la base de datos.

- **pom.xml**

Este archivo contiene la configuración de las librerías que se utilizan en el proyecto. Mediante esta configuración los componentes son gestionados remotamente con el repositorio de Maven.

- **persistence.xml**

Este archivo contiene la configuración del Hibernate, framework con el cual se implementa la interacción y control de la base de datos MySQL.

7.2.3. Controllers

Contiene los componentes de lógica de negocio del sistema.

7.2.4. Models

Contiene los componentes de Data Access Object del sistema. Es el conjunto de entidades que interactúan directamente con la base de datos.

7.2.5. Views

Contiene los componentes de interfaz gráfica del sistema.

8. Vista de Datos

En la Figura 5 se muestra la estructura interna de las entidades de la base de datos, así como las relaciones entre estas.

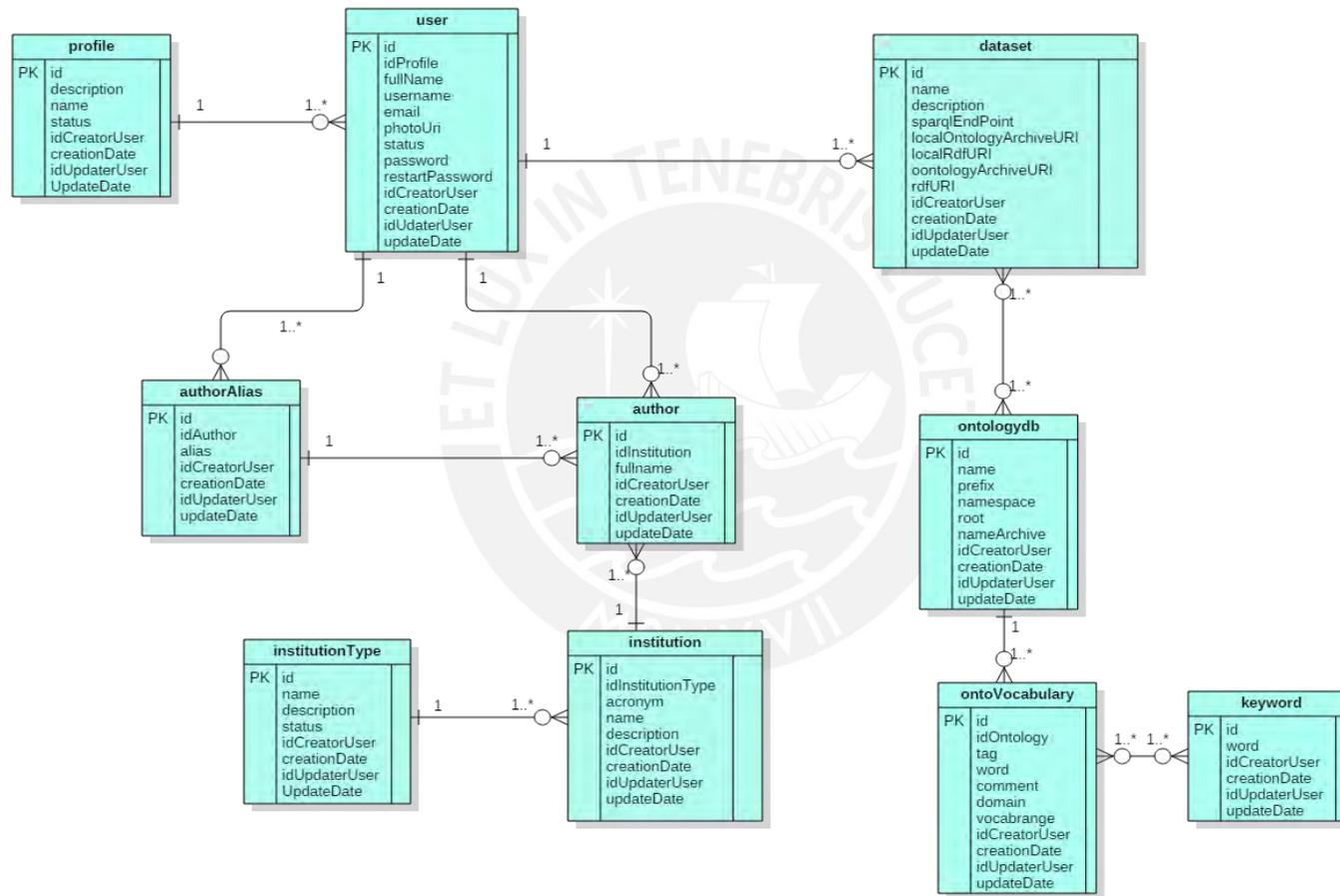


Figura 5: Modelo de base de datos.

9. Tamaño y Performance

- El sistema deberá estar disponible las 24 horas del día, 7 días a la semana.
- Se mantendrá un respaldo de la base de datos, el cual se actualizará de manera diaria.

10. Calidad

- Los usuarios podrán acceder a la interfaz del sistema mediante el uso de los siguientes navegadores web: Google Chrome y Firefox. Adicionalmente, se podrá hacer uso de Internet Explorer; pero esto último está condicionado a las limitaciones del navegador.
- El sistema responderá y soportará la concurrencia y peticiones de los distintos tipos de usuarios con una disponibilidad de 24 horas al día y 7 días a la semana.
- El sistema deberá realizar los cálculos y controles necesarios según las políticas del mismo.
- El sistema deberá tener la capacidad de responder ante fallos y/o excepciones que no forman parte de los servicios que brinda; tales como el ingreso de información errónea y fallas por conexión a internet, entre otros. Se emitirán los mensajes de informes, así como el guardado del log de errores. Estos aspectos darán al sistema la característica de tolerancia a fallos.
- El usuario podrá reconocer el resultado de sus acciones mediante mensajes de confirmación.
- El sistema permitirá la administración de usuarios y perfiles como parte de la seguridad del sistema. Se realizará la autenticación de usuario por medio del nombre/código y contraseña de usuario. Asimismo, toda contraseña pasa por un proceso de encriptación para su almacenamiento en la base de datos.
- La interfaz de usuario será gráfica y su diseño contará con todos los elementos necesarios que permitan el uso intuitivo del mismo, incluso para usuarios con conocimientos básicos en computación.
- Las opciones del menú variarán según el tipo de usuario que ha iniciado sesión, lo cual limita al usuario a visualizar solamente aquellas funcionalidades para las cuales se tiene acceso.
- Los componentes gráficos del sistema seguirán un estándar de colores, tipos de menús y botones en cada una de las páginas del sistema; lo cual permitirá un fácil uso y una curva de aprendizaje rápida de acuerdo a las necesidades y exigencias del negocio.