

SEMANTIC PHI-BASE WEB INTERFACE

Tesis de Máster

Autor:

Juan Camilo Mesa Polo

Director:

Alejandro Rodríguez González



Universidad Politécnica de Madrid
Master Universitario en Ingeniería Informática

AGRADECIMIENTOS

La realización de este trabajo ha sido posible gracias a todos los conocimientos y experiencias que he tenido durante la realización del Máster Universitario en Ingeniería Informática. Haber cursado satisfactoriamente todas las asignaturas aplicando y entendiendo los conceptos estudiados, ha hecho posible que todo este trabajo haya sido llevado a cabo satisfactoriamente.

Quiero agradecer, primero a mis padres, quienes han sido las personas que me han apoyado y me han ayudado a formarme y a ser la persona y profesional que soy hoy. Sin ellos, nada de esto habría sido posible. También quiero agradecer profundamente a mi novia, Dania Carolina Ortiz Acosta, quién ha sido la persona que me ha impulsado desde un principio a iniciar mis estudios de Máster en España y que me ha hecho ver que no existen límites de todo lo que somos y podemos hacer.

Finalmente, quiero agradecer a todos mis compañeros y profesores que han estado conmigo durante el Máster, ya que de ellos también he aprendido mucho, no sólo a nivel académico, sino también a nivel personal.

RESUMEN

PHI-BASE (Pathogen-Host-Interaction) es una base de datos XML que ha sido creada por el instituto de investigación agrario en Reino Unido “Rothamsted Research”, la cual contiene información sobre las interacciones que existen entre Hosts (Seres Vivos) y Patógenos (Enfermedades, Virus, etc.). La estructura, generación de consultas y la posible integración de esta base de datos es compleja debido al formato en que se encuentra. Por esta razón, los investigadores del **CBGP** “Centro de Biotecnología y Genómica de Plantas” junto con investigadores de la **ETSIINF** “Escuela Técnica Superior de Ingenieros Informáticos”, han construido **Semantic PHI-BASE**, una base de conocimientos creada con tecnologías de Web Semántica, con la que la generación de consultas para obtener información valiosa y generar conocimiento se puede hacer de una mejor manera, además de permitir una integración natural con otras bases de conocimiento en ámbitos de investigación científica. Sin embargo, la generación de consultas hacia **Semantic PHI-BASE**, requiere conocimientos técnicos elevados que no son poseídos por todos los investigadores que consumirán los datos disponibles. Para solucionar esto, se ha desarrollado un sistema totalmente web, llamado **Semantic PHI-BASE Web Interface**, el cual permite a los usuarios generar consultas dinámicas automáticamente para obtener información valiosa sobre los datos contenidos en **Semantic PHI-BASE**. El sistema cuenta con un módulo completo de administración de usuarios, perfiles, consultas de ingreso al sistema y de acciones realizadas, así como la integración con un módulo externo de visualización gráfica de datos, el cual permite a los usuarios, de una manera interactiva, navegar por los datos disponibles en la base de conocimientos. Adicionalmente a las funcionalidades desarrolladas, se han dejado establecidas las bases para seguir creciendo la aplicación con módulos para la adición de nuevos datos a la base de conocimientos, edición de datos existentes y la generación de consultas federadas utilizando otras bases de conocimientos relacionadas.

Palabras Clave: Semantic PHI-BASE, Base de Conocimiento, Ontología, SPARQL, Query, Web Semántica.

ABSTRACT

PHI-BASE (Pathogen-Host-Interaction) is a XML database that has been created by the research institute in the UK called “Rothamsted Research”, which contains information about the interactions that exists between Hosts (living creatures) and Pathogens (diseases, viruses, etc). The Structure, query generation and the possible integration of this database with other databases is complex due to its format. For this reason, researchers at the **CBGP** “Centro de Biotecnología y Genómica de Plantas” along with researchers at the **ETSIINF** “Escuela Técnica Superior de Ingenieros Informáticos” have built **Semantic PHI-BASE**, which is a knowledge base created using Semantic Web technologies, that allows better query generation to get valuable information and generate knowledge, and additionally it allows a natural integration with other knowledge bases belonging to other research areas. However, the generation of queries to retrieve knowledge from Semantic PHI-BASE requires elevated technical knowledges that not all the researchers that consume the data available have. To solve this, it has been developed a 100% web application, that allows the dynamic query generation automatically in order to obtain valuable information about the data contained in **Semantic PHI-BASE**. The application has a complete module for the user and profile administration, consultation of the logins and actions made by the users, and the integration with an external module for graphical data visualization, which allows in an interactive way, to navigate through all the data available in the knowledge base. In addition to the developed functionalities, is has been established all the bases to keep evolving the application with modules to insert and edit data in the knowledge base and the generation of federated queries using other related knowledge bases.

Keywords: Semantic PHI-BASE, Knowledge Base, Ontology, SPARQL, Query, Semantic Web.

TABLA DE CONTENIDO

CAPÍTULO I: INTRODUCCIÓN Y OBJETIVOS	1
CAPÍTULO II: ESTADO DEL ARTE	7
CAPÍTULO III: EVALUACIÓN DE RIESGOS.....	11
Identificación de Riesgos	11
Estimación de Riesgos.....	12
Plan de Riesgos	12
CAPÍTULO IV: DESARROLLO	15
Necesidad Usuario.....	15
Tareas a Desarrollar.....	16
Diagrama de Gantt.....	19
Tecnologías Usadas.....	20
Java	20
Eclipse Kepler.....	20
MySQL Database – MySQL Workbench.....	21
Apache Jena	21
ICEfaces Software	22
Spring & Hibernate Framework	22
Amazon Web Services AWS.....	23
Arquitectura de Software.....	24
Diseño de Base de datos	25
Convenciones.....	25
Tablas Base de Datos.....	26
Modelo Entidad-Relación.....	28
Especificación Básica de Casos de Uso	29
Actores del Sistema	29
Descripción de Casos de Uso	29
Diseño de Componentes	32
Capa de Persistencia	32
Capa de Servicios (Backing Beans).....	34
Interfaz Gráfica.....	37

Arquitectura de Despliegue	39
CAPÍTULO V: RESULTADOS	41
Inicio Aplicación	41
Menú Principal	42
Mantenimiento Estructura	43
Consulta Semantic PHI-BASE	44
Manual de Usuario	46
CAPÍTULO VI: CONCLUSIONES	47
CAPÍTULO VII: LÍNEAS FUTURAS	49
Generación Consultas Federadas	49
Inserción Controlada de Datos	49
Edición de Datos Actuales	49
Publicación en Revista “Journal of Web Semantics”	49
REFERENCIAS	51
ANEXO A – DISEÑO DE BASE DE DATOS	53
Estándar Nombre de Tablas y Campos	53
Descripción Tablas Base de Datos	54
ANEXO B – ESPECIFICACIÓN DETALLADA DE CASOS DE USO	59
ANEXO C – MANUAL DE DESPLIEGUE	77
Pre-Requisitos	78
Instalación y Configuración de Aplicación	79
Mantenimiento y Revisión Funcionamiento	82
ANEXO D – MANUAL DE DESARROLLADOR	83
Proyecto PhiBaseWeb – Eclipse Kepler	83
Adicionar Nueva Tabla en Base de Datos	88
Creación nueva Pantalla	89

LISTADO DE FIGURAS

Imagen 1 - Modelo Semantic PHI-BASE	8
Imagen 2 - Linked Data	10
Imagen 3 - Diagrama de Gantt.....	19
Imagen 4 - Ícono Java	20
Imagen 5 - Ícono Eclipse	20
Imagen 6 - Ícono MySQL	21
Imagen 7 - Ícono Apache Jena.....	21
Imagen 8 - Ícono ICEfaces.....	22
Imagen 9 - Ícono Spring-Hibernate	22
Imagen 10 - Ícono Amazon Web Services AWS.....	23
Imagen 11 - Arquitectura de Software.....	24
Imagen 12 - Modelo Entidad-Relación.....	28
Imagen 13 - Diagrama Clases Capa Persistencia.....	32
Imagen 14 - Diagrama Clases Capa Servicios	34
Imagen 15 - Mapa de Navegación	37
Imagen 16 - Arquitectura de Despliegue	39
Imagen 17 - Pantalla Inicio Aplicación.....	41
Imagen 18 - Pantalla Menú Principal.....	42
Imagen 19 - Mantenimiento Servicio del Sistema	43
Imagen 20 - Pantalla Consulta Semantic PHI-BASE (Tab I)	44
Imagen 21 - Pantalla Consulta Semantic PHI-BASE (Tab II).....	44
Imagen 22 - Pantalla Consulta Semantic PHI-BASE (Tab III)	45
Imagen 23 - Pantalla Manual de Usuario	46
Imagen 24 - Estructura de Ficheros	77
Imagen 25 - Puerto Escucha Tomcat	79
Imagen 26 - Export variables Java.....	79
Imagen 27 - Encripción con herramienta dbkeygen.jar	81
Imagen 28 - Reinicio Servidor Tomcat.....	82
Imagen 29 - Proyecto Eclipse PhiBaseWeb.....	83
Imagen 30 - Paquetes Proyecto y Archivos Configuración	84
Imagen 31 - Contenido WebContent	86

LISTADO DE TABLAS

Tabla 1 - Objetivos Específicos	1
Tabla 2 - Conceptos Clave	3
Tabla 3 - Conceptos Semánticos	9
Tabla 4 - Identificación Riesgos	11
Tabla 5 - Estimación de Riesgos	12
Tabla 6 - Plan de Riesgos	12
Tabla 7 - Tareas a Desarrollar	16
Tabla 8 - Tablas Base de Datos	26
Tabla 9 - Descripción Casos de Uso	29
Tabla 10 - Descripción Clases Capa Servicios	34
Tabla 11 - Plantillas Web	37
Tabla 12 - Descripción Tabla CSPPERFU	54
Tabla 13 - Descripción Tabla CSPSERSI	54
Tabla 14 - Descripción Tabla CSPSERPF	54
Tabla 15 - Descripción Tabla CSPPARSI	55
Tabla 16 - Descripción Tabla USMUSUAR	55
Tabla 17 - Descripción Tabla USDBLOQU	55
Tabla 18 - Descripción Tabla USDLOGQU	56
Tabla 19 - Descripción Tabla USDLOGIN	56
Tabla 20 - Descripción Tabla SEPCONCE	56
Tabla 21 - Descripción Tabla SEPRELCO	57
Tabla 22 - Descripción Tabla SEDCONCE	57
Tabla 23 - Descripción Tabla SEDRELCO	58
Tabla 24 - EDCU CU-001	59
Tabla 25 - EDCU CU-002	60
Tabla 26 - EDCU CU-003	61
Tabla 27 - EDCU CU-004	62
Tabla 28 - EDCU CU-005	63
Tabla 29 - EDCU CU-006	65
Tabla 30 - EDCU CU-007	66
Tabla 31 - EDCU CU-008	68
Tabla 32 - EDCU CU-009	68
Tabla 33 - EDCU CU-010	69
Tabla 34 - EDCU CU-011	70
Tabla 35 - EDCU CU-012	71
Tabla 36 - EDCU CU-013	72
Tabla 37 - EDCU CU-014	73
Tabla 38 - EDCU CU-015	73

Tabla 39 - Estructura de Ficheros	77
Tabla 40 - Configuración config.properties	79
Tabla 41 - Paquetes Proyecto y Archivos Configuración	84
Tabla 42 - Contenido WebContent	86

CAPÍTULO I: INTRODUCCIÓN Y OBJETIVOS

El instituto de investigación agrario en Reino Unido “Rothamsted Research”, ha generado una base de datos en formato XML, llamada **PHI-BASE**, la cual contiene información sobre las interacciones que existen entre Host (Anfitrión) y Pathogen (Patógeno). Host hace referencia a un ser vivo, tal como un ser humano, un animal o una planta, mientras que Pathogen hace referencia a las enfermedades, virus o bacterias que se alojan y afectan al Host en cuestión.

Los investigadores del CBGP “Centro de Biotecnología y Genómica de Plantas” junto con investigadores de la ETSIINF “Escuela Técnica Superior de Ingenieros Informáticos”, han generado una base de conocimientos semántica partiendo de la base de datos XML **PHI-BASE**, llamada **Semantic PHI-BASE**¹.

Para generar esta base de datos semántica, se han filtrado los datos, seleccionado únicamente los Hosts que son de tipo plantas, ya que son los de interés para el CBGP. Esta base de datos semántica, se encuentra desplegada en la plataforma tecnológica *Virtuoso*, por lo que puede ser inspeccionada a través de consultas **SPARQL**, las cuales permiten obtener información valiosa sobre las relaciones que existen en el modelo planteado.

Debido a que la construcción de consultas **SPARQL** para obtener información valiosa sobre las interacciones Host-Pathogen es una labor muy técnica y compleja, surge la necesidad de construir una herramienta que permita a los usuarios finales generar automáticamente estas consultas de acuerdo al filtro de búsqueda deseado y que los resultados sean desplegados de una manera legible, amigable e interactiva.

Por lo anterior, el objetivo general de este trabajo de fin de Máster, es construir una aplicación web, que permita consultar la base de conocimientos **Semantic PHI-BASE**, de manera que los usuarios no tengan que conocer la sintaxis y estructura de las consultas **SPARQL** requeridas para hacerlo. La aplicación Web construida, por lo tanto, tendrá el nombre de **Semantic PHI-BASE Web Interface**.

Para cumplir con el objetivo general de la construcción de la aplicación web, se tienen como objetivos específicos los siguientes:

Tabla 1 - Objetivos Específicos

<u>Objetivo</u>	<u>Descripción Objetivo</u>
Construcción de interfaz amigable al usuario final	La construcción de la interfaz gráfica se debe hacer cumpliendo con los principales estándares establecidos en la construcción de interfaces web. Por esto, debe ser amigable e intuitiva, de manera que los usuarios que hagan uso de la aplicación, sepan desde un principio cómo usarla sin tener la necesidad

¹ Alejandro Rodríguez-Iglesias, Alejandro Rodríguez-González, Alistair G. Irvine, Ane Sesma, Martin Urban, Kim E. Hammond-Kosack and Mark D. Wilkinson. “Publishing FAIR Data: An Exemplar Methodology Utilizing PHI-Base” [Online] <<http://journal.frontiersin.org/article/10.3389/fpls.2016.00641/full>> [Fecha de Consulta: 24-05-2016]

	de leer por completo el manual de usuario o de recibir formaciones especializadas.
Construcción de módulo de administración de usuarios	La aplicación debe contar con un módulo que permita la administración de los usuarios que podrán acceder al sistema. Debe ser posible la adición de usuarios, activación o inactivación de sus cuentas, así como consultar sus ingresos y sus acciones.
Generar diseño de aplicación extensible	El sistema a desarrollar debe ser diseñado de tal forma que sea posible adicionar nuevas funcionalidades sin tener que realizar modificaciones al núcleo principal de la aplicación.
Código generado documentado	La totalidad del código que se escriba para el sistema debe estar debidamente documentado, de tal manera que sea de fácil comprensión a la hora de realizar modificaciones.
Generar manual de desarrollador	Así como el código generado debe estar debidamente documentado, también se debe contar con un manual de desarrollador, que ayude a entender a otros desarrolladores cómo ha sido la construcción del software y qué deben hacer para realizar modificaciones y adicionar nuevas funcionalidades al sistema.
Generar manual de usuario final	Se debe tener también un manual para el usuario final, para que pueda ser consultado, en caso de tenga dudas o confusiones acerca del funcionamiento de la aplicación o de cómo realizar alguna acción específica en el sistema.
Integración con módulo de visualización de datos	El producto generado debe ser capaz de integrarse con un módulo de visualización externo, el cual se encargará de representar gráficamente los resultados obtenidos a partir de una consulta generada automáticamente por el sistema.

De acuerdo a los objetivos propuestos anteriormente, el contenido de este documento presentará, en detalle, el contexto del escenario actual, así como el desarrollo de la solución propuesta y los resultados obtenidos tras realizar el trabajo. Más concretamente los capítulos que siguen a este son los siguientes:

- I. CAPÍTULO II: ESTADO DEL ARTE: En este capítulo se hablará sobre el contexto actual del proyecto más a detalle, comentando las necesidades existentes, así mismo, se hablará de las tecnologías involucradas en la solución y cómo serán usadas.
- II. CAPÍTULO III: EVALUACIÓN DE RIESGOS: Aquí se realizará un análisis de los posibles riesgos que se han encontrado durante la realización del proyecto y de las acciones que se llevarán a cabo tanto como para evitar que ocurran como para tratar en caso de que se presenten.
- III. CAPÍTULO IV: DESARROLLO: Este es el capítulo central del documento, en donde se mostrará el diseño completo de la solución implementada, así como las tecnologías y buenas prácticas usadas.
- IV. CAPÍTULO V: RESULTADOS: En este capítulo se mostrarán los resultados obtenidos al realizar el desarrollo de la aplicación web *Semantic PHI-BASE Web Interface*. Se podrán

ver pantallas de la aplicación construida, así como ejemplos de algunas de las queries automáticas generadas por el sistema que permitirán a los usuarios finales extraer información valiosa de la base de conocimientos de Virtuoso.

- V. CAPÍTULO VI: CONCLUSIONES: Contendrá todas las conclusiones obtenidas a lo largo de la realización de este trabajo de fin de Máster.
- VI. CAPÍTULO VII: LÍNEAS FUTURAS: En este capítulo se comentará sobre las líneas futuras que se podrán lograr al concluir con este trabajo de fin de Máster. Se dejará la base para posibles extensiones al software ya desarrollado.

Además de los capítulos principales del documento, también existen unos apartados anexos que complementan la información. Los anexos contenidos en este documento son los siguientes:

- I. ANEXO A – DISEÑO DE BASE DE DATOS: En este anexo, se mostrará detalladamente el diseño de la base de datos relacional que da soporte a la aplicación web.
- II. ANEXO B – ESPECIFICACIÓN DETALLADA DE CASOS DE USO: Aquí, se detallará cada uno de los casos de uso que hacen parte de la solución diseñada. Cada caso de uso ha sido escrito utilizando una plantilla estándar para especificación detallada de casos de uso.
- III. ANEXO C – MANUAL DE DESPLIEGUE: Este anexo contiene toda la información relevante y necesaria para que sea posible realizar el despliegue de la aplicación web en un servidor con distribución Linux, utilizando una base de datos MySQL.
- IV. ANEXO D – MANUAL DE DESARROLLADOR: Aquí se encontrará un manual diseñado específicamente para una persona técnica que comprenda todos los conceptos de desarrollo de software. El propósito de este documento es explicar los pasos a seguir para modificar o adicionar nuevas funcionalidades a la aplicación web.

Para entender de una manera más clara todos los contenidos, la tabla a continuación, contiene una lista de los conceptos clave con su definición y uso en el contexto de la tesis de Máster.

Tabla 2 - Conceptos Clave

<u>Concepto Clave</u>	<u>Definición</u>
Semantic PHI-BASE Web Interface	Es el nombre de la aplicación Web desarrollada que se encargará de generar automáticamente las consultas a Semantic PHI-BASE para que los usuarios puedan extraer información sobre datos almacenados.
XML	Lenguaje que sirve para almacenar datos de forma legible. El significado de sus siglas en inglés es “eXtensible Language Markup”. Es el lenguaje utilizado para la creación de la base de datos PHI-BASE .
RDF	Se trata de un modelo para la descripción de recursos Web. El significado de sus siglas en inglés es “Resource Description Framework”. La base de conocimientos Semantic PHI-BASE ha sido definida con RDF por lo que los recursos que se definan puedan ser relacionados con recursos de bases de conocimientos externas.

PHI-BASE	Es la base de datos inicial desarrollada por el instituto de investigación agrario en Reino Unido “Rothamsted Research”. Su formato es XML y ha sido la base inicial para la creación de Semantic PHI-BASE .
Semantic PHI-BASE	Es la base de conocimientos desarrollada utilizando técnicas de web semántica para representar los datos contenidos en PHI-BASE en un formato que pueda ser consultado más fácilmente. Adicionalmente, con la creación de Semantic PHI-BASE se han solucionado algunos de los problemas de normalización de la base de datos PHI-BASE .
SPARQL	Es un lenguaje de consulta similar al SQL , el cual permite realizar consultas a bases de conocimiento que han sido creadas con RDF como lo ha sido Semantic PHI-BASE . Con este lenguaje es que se generan automáticamente las consultas que son enviadas al sistema de Virtuoso .
Virtuoso	Se trata de un servidor de base de datos que es capaz de almacenar bases de conocimiento en formato RDF . Es el servidor que contiene todos los datos generados para Semantic PHI-BASE .
Apache Jena – Jena	Se trata de un framework Java que sirve para manipular Web Semántica como RDF . Es el framework que ha sido utilizado para generar automáticamente las consultas que se envían al sistema de Virtuoso .
ICEfaces	Framework de Java utilizado para construir aplicaciones RIA (Rich Internet Application). Con este framework se ha construido toda la interfaz gráfica de Semantic PHI-BASE Web Interface .
Consulta – Query	Término utilizado para indicar las peticiones que se realizan al servidor de Virtuoso para obtener información sobre la base de conocimientos de Semantic PHI-BASE . Las consultas o queries se realizan en el lenguaje de consulta SPARQL .
Endpoint	Hace referencia al conjunto de Servidor-Dirección al que se le podrán enviar las peticiones de consultas SPARQL . Por ejemplo, el Endpoint de Semantic PHI-BASE actualmente es: http://linkeddata.systems:8890/sparql
Linked Data	Datos que se encuentran relacionados entre sí y que son representados en forma de grafo. Los nodos hacen referencia a un dato en particular y las aristas reflejan las relaciones que tienen con otro dato de la misma base o de otra base de conocimientos.
Ontología	Hace referencia a una definición formal de tipos, propiedades y relaciones entre entidades que se encuentran definidas dentro de un dominio particular.
Host (Anfitrión)	Se trata de la planta que hace parte de la base de conocimientos de Semantic PHI-BASE .

Pathogen (Patógeno)	Se trata de la enfermedad, virus o bacteria que afecta y se aloja en un Host determinado.
Interacción	Es la relación que existe entre un Host y un Pathogen. De esta relación se pueden obtener muchas más propiedades que serán vistas más adelante.

CAPÍTULO II: ESTADO DEL ARTE

La base de datos **PHI-BASE** en formato XML tiene una estructura compleja, lo cual dificulta la realización de consultas para extraer información. Además de esto, la base de datos de **PHI-BASE** adolece de varios problemas entre los que se encuentran la normalización y curación de los datos, así como la falta de esquemas de representación de conocimiento común, entre otros. Debido a esto, existen muchos registros duplicados, o registros diferentes con variaciones mínimas pero que representan al mismo objeto.

Con la creación de *Semantic PHI-BASE*, es posible realizar consultas complejas utilizando SPARQL y obtener información valiosa sobre los datos almacenados. Adicionalmente, se han solucionado parcialmente los problemas de normalización provenientes de **PHI-BASE**. Sin embargo, aún existen registros duplicados que hacen referencia a un mismo objeto. Por ejemplo, existe una propiedad llamada “Invitro Growth” perteneciente a los alelos que están relacionados a una interacción. Existen muchos registros de tipo “Invitro Growth” con el valor “Wild Type” pero escrito de diferentes maneras como: WildType, Wild Type, wildtype, wild-type, WT, etc.

Otra de las ventajas que ha traído la implementación de *Semantic PHI-BASE*, es que los datos se han podido relacionar e integrar junto con otros conceptos semánticos que existen en bases de conocimiento externas a través de vocabularios controlados existentes en la propia área biomédica. Esto trae una gran ventaja y es que es posible realizar consultas “federadas” que son capaces de combinar los datos de distintos Endpoints para producir un único resultado de salida. Algunas de las bases de conocimientos y ontologías ya existentes con las que se han creado relaciones son:

- ✓ Semantic Science [<http://semanticscience.org>]: Es una comunidad para apoyar el desarrollo de tecnologías de Web Semántica que facilitará la integración y el descubrimiento conocimiento científico.
- ✓ Edamontology [<http://edamontology.org>]: Se trata de una ontología de conceptos relacionados con la Bioinformática. Su nombre proviene de “EMBRACE Data and Methods”.
- ✓ Linked Life Data [<http://linkedlifedata.com>]: Es una plataforma del estilo “Data as a Service” que da acceso a más de 25 bases de datos de información biomédica a través de un único Endpoint.
- ✓ Uniprot [<http://purl.uniprot.org>]: Es una base de conocimientos que contiene información biomédica.
- ✓ Otras bases de conocimiento como **RDFS** y **FOAF** que son muy comúnmente utilizados en el ámbito de la Web Semántica.

Para entender un poco más el modelo de definición de la base de conocimientos **Semantic PHI-BASE**, a continuación, se presenta el esquema del grafo:

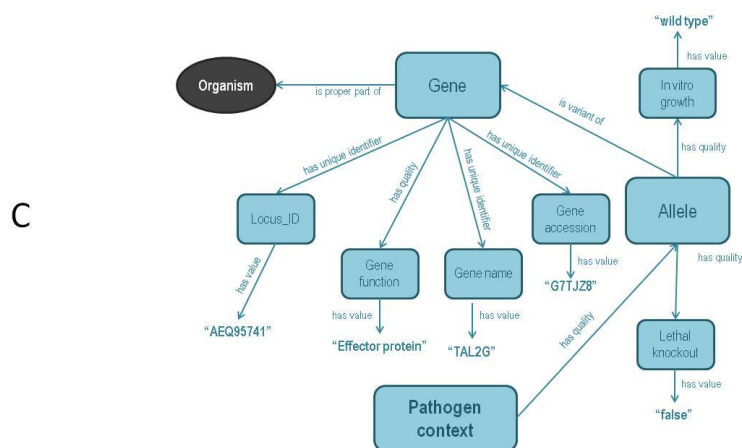
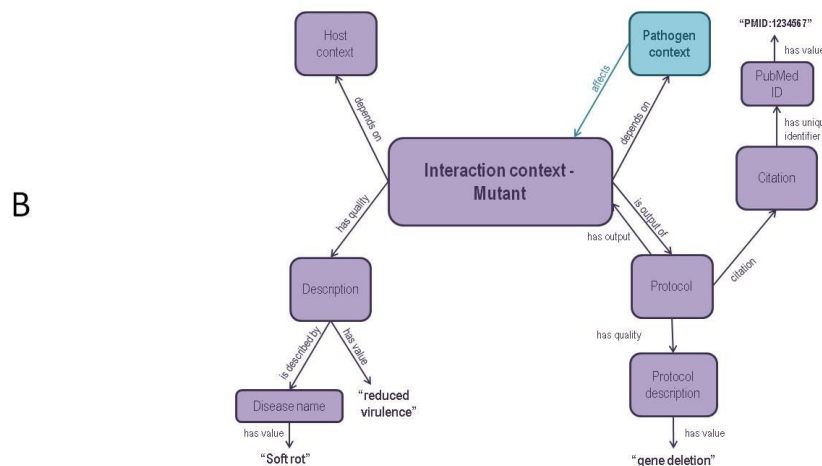
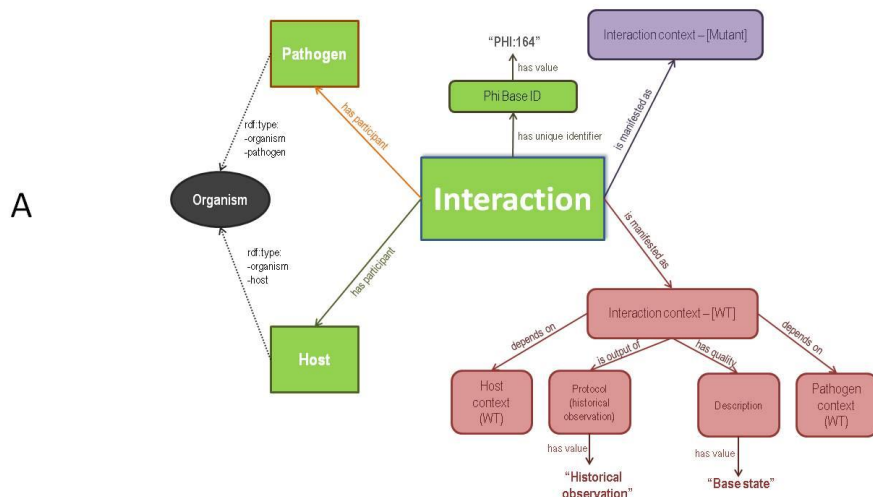


Imagen 1 - Modelo Semántico PHI-BASE

Los conceptos más relevantes y los que serán utilizados para generar las consultas automáticas son los siguientes:

Tabla 3 - Conceptos Semánticos

<u>Concepto Semántico</u>	<u>Descripción Concepto</u>
Pathogen	Organismo que afecta a un Host. Es de tipo “Organism” y está relacionada con la interacción a través de la relación “Has Participant”.
Host	Planta que participa en la interacción. Es de tipo “Organism” y está relacionada con la interacción a través de la relación “Has Participant”.
Interaction Context - [Mutant] Description	Descripción del contexto mutante de la interacción. Está relacionado con la clase “Interaction Context - [Mutant]” a través de la relación “Has Quality”. La clase “Interaction Context - [Mutant]” está directamente relacionada con la interacción a través de la relación “Is Manifested As”.
Disease Name	Nombre de la enfermedad que describe a “Interaction Context - [Mutant] Description”. Se encuentra relacionado directamente con la clase “Interaction Context - [Mutant] Description” a través de la relación “Is Described By”.
Citation	Hace referencia al documento científico en donde se encuentra descrito el concepto “Protocol” de “Interaction Context - [Mutant]”. Tiene un identificador único llamado “PubMedId”.
Protocol Description	Descripción del “Protocol” de “Interaction Context - [Mutant]”
Allele	Alelo que se encuentra relacionado con el “Pathogen Context” del cual depende directamente “Interaction Context - [Mutant]”.
In vitro Growth	Propiedad del “Allele” que indica el crecimiento in vitro. Relacionado con “Allele” a través de la propiedad “Has Quality”.
Lethal Knockout	Propiedad booleana del “Allele” que indica un Knockout letal. Relacionado con “Allele” a través de la propiedad “Has Quality”.
Gene	Variante del “Allele”. Está directamente relacionado con “Allele” a través de la propiedad “Is Variant Of”.
Locus_ID	Identificador único del Gen.
Gene Function	Función que realiza el Gen.
Gene Name	Identificador único del Gen.
Gene Accession	Identificador único del Gen.

La idea es, entonces, generar una interfaz gráfica amigable al usuario en la que sea posible seleccionar uno o varios filtros de búsqueda, de acuerdo a los conceptos semánticos anteriores, y con base en esto, el sistema sea capaz de generar automáticamente una consulta **SPARQL** que será ejecutada en el Endpoint de Virtuoso. Al realizar la ejecución de la consulta, se obtendrán uno o varios registros y se construirá dinámicamente una tabla de resultados en donde el usuario no sólo podrá visualizar los resultados obtenidos en forma de tabla, sino que además al interactuar con cualquiera de ellos, la aplicación se integrará con un servicio de visualización de grafos desde el cual podrá navegar a través de los resultados obtenidos y verlos de una manera más clara.

La siguiente imagen, que es un ejemplo de los datos disponibles en la base de conocimientos de “Linked Life Data”, muestra cómo es la visualización de los datos relacionados en un grafo de acuerdo al concepto de **Linked Data**:

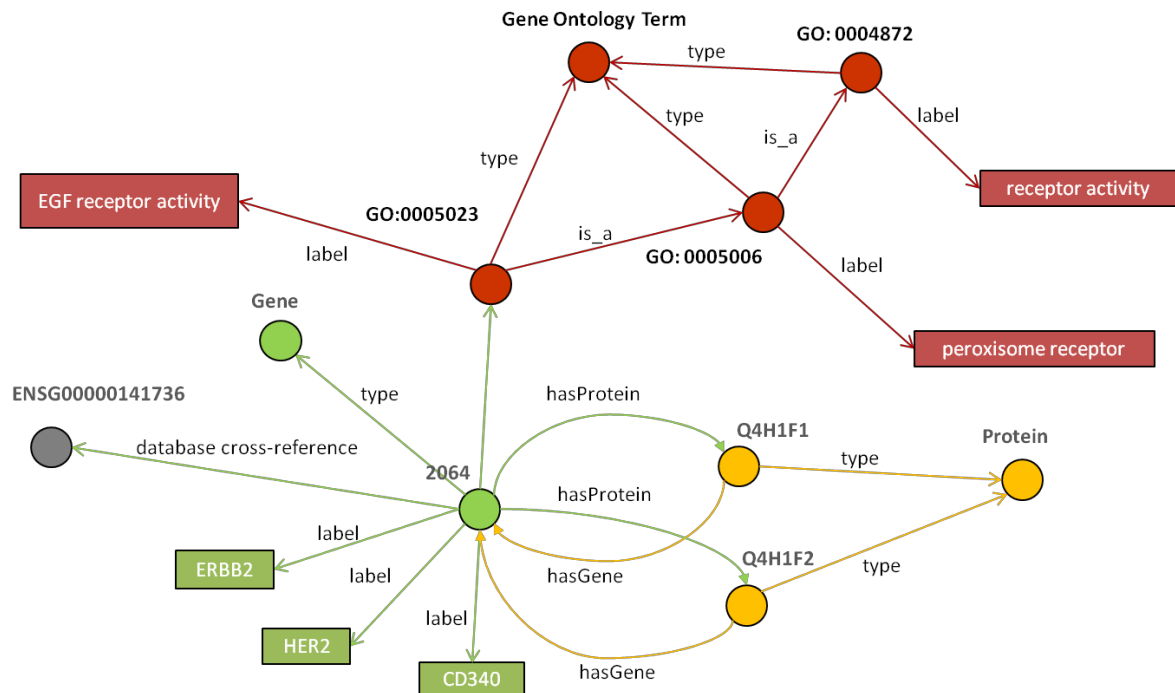


Imagen 2 - Linked Data

Con la construcción de **Semantic PHI-BASE Web Interface**, los usuarios finales serán capaces de ver cómo están relacionados todos los datos entre sí y encontrar y generar nuevo conocimiento y patrones en los datos que podrán ser útiles para las investigaciones que se llevan a cabo en los ámbitos de la biotecnología.

CAPÍTULO III: EVALUACIÓN DE RIESGOS

Debido a que la construcción de *Semantic PHI-BASE Web Interface* se puede tratar como un proyecto de ingeniería de software tradicional, se llevará a cabo una evaluación de riesgos siguiendo un enfoque tradicional, en el cual, primero se realizará una identificación de los posibles riesgos que puedan afectar al proyecto, seguido de una estimación de los riesgos en donde se asignará una probabilidad y un impacto para cada riesgo identificado, y finalmente, se elaborará un plan de riesgos para tratar cada riesgo identificado. Es importante comentar que se tomarán como riesgo, únicamente, los eventos que puedan afectar de forma negativa el desarrollo normal del proyecto.

Identificación de Riesgos

A continuación, se presenta una tabla en donde están contenidos todos los riesgos identificados que pueden ocurrir durante la realización del proyecto *Semantic PHI-BASE Web Interface*:

Tabla 4 - Identificación Riesgos

<u>Riesgo</u>	<u>Descripción Riesgo</u>
Entendimiento incorrecto del problema	Se trata de que no se realice una correcta comprensión del problema a solucionar y por lo tanto no se planteen las soluciones adecuadas.
Tecnología usada incorrecta	Ocurre cuando con la tecnología en la que se pretende solucionar el problema, no sea posible realizar la construcción de todos los componentes necesarios que hacen parte de la solución.
Falta de experiencia en tecnologías	Se trata de que en el momento de realizar el desarrollo no se tenga la experiencia suficiente para cumplir con los objetivos y por lo tanto se presenten atrasos y/o no cumplimiento.
Cambios de requerimientos previamente definidos	Ocurre cuando ya una vez establecidos los requerimientos y objetivos a realizar, se presenten cambios en los requerimientos y necesidades, por lo que es necesario volver a plantear propuestas de soluciones al problema general.
Cambios en diseño estructural durante el desarrollo	Esto ocurre cuando ya durante la fase de desarrollo, existan cambios estructurales ya sea a nivel de base de datos o a nivel de componentes, lo cual obliga a una re-escritura de código y por lo tanto un atraso en los tiempos del proyecto.
Problemas con la integración del módulo de visualización	Riesgo en el cual no sea posible realizar una correcta integración con el módulo de visualización de grafos externo.

Estimación de Riesgos

Para la estimación de la probabilidad y el impacto asociado a cada riesgo, se utilizará una escala cualitativa de cinco (5) valores: Muy Bajo, Bajo, Medio, Alto, Muy Alto. Con la combinación de la probabilidad de ocurrencia y el impacto en el proyecto, se obtendrá una calificación del riesgo con la que se podrá crear un plan de tratamiento.

Tabla 5 - Estimación de Riesgos

<u>Riesgo</u>	<u>Probabilidad</u>	<u>Impacto</u>	<u>Calificación</u>
Entendimiento incorrecto del problema	Baja	Muy Alto	Alta
Tecnología usada incorrecta	Muy Baja	Muy Alto	Media
Falta de experiencia en tecnologías	Media	Medio	Baja
Cambios de requerimientos previamente definidos	Media	Alto	Media
Cambios en diseño estructural durante el desarrollo	Media	Medio	Baja
Problemas con la integración del módulo de visualización	Baja	Muy Alto	Alta

De acuerdo a las calificaciones obtenidas, es posible realizar una priorización de los riesgos y del plan de tratamiento que se debe llevar a cabo para evitar o mitigar cada riesgo.

Plan de Riesgos

Para cada uno de los riesgos identificados, de acuerdo a su estimación, se propone el siguiente plan de tratamiento:

Tabla 6 - Plan de Riesgos

<u>Riesgo</u>	<u>Tratamiento Riesgo</u>
Entendimiento incorrecto del problema	Para mitigar este riesgo, se ha tomado el tiempo suficiente para leer toda la documentación disponible sobre Semantic PHI-BASE . Adicionalmente, luego de estudiar todos los conceptos disponibles, se han realizado reuniones con el tutor del trabajo de fin de Máster para corroborar que el planteamiento del problema era correcto y que las soluciones propuestas eran óptimas.
Tecnología usada incorrecta	Debido a que existen muchas tecnologías para desarrollo Web, se ha hecho un estudio previo antes de iniciar con las actividades de diseño y desarrollo, para determinar qué tecnología usar con la que se puedan desarrollar todas las funcionalidades requeridas.

Falta de experiencia en tecnologías	Con el mismo estudio sobre la elección de la tecnología, se ha tenido en cuenta la experiencia previa y la curva de aprendizaje necesaria para realizar desarrollos. La conclusión ha sido desarrollar en Java utilizando el framework ICEfaces para la construcción de la interfaz gráfica.
Cambios de requerimientos previamente definidos	Debido a que esto es un problema muy común en los proyectos de ingeniería de software, se ha utilizado una metodología iterativa para el desarrollo, de manera que se vayan verificando los requisitos constantemente y se vaya ajustando cada vez el desarrollo en caso de pequeños cambios. De igual manera, se han establecido inicialmente los requerimientos a desarrollar y se ha adquirido un compromiso de ambas partes (Tutor y Alumno) para llevar a cabo dichos requerimientos.
Cambios en diseño estructural durante el desarrollo	Es también común que en pleno desarrollo surjan cambios a nivel estructural para dar soporte a las funcionalidades requeridas. Se ha diseñado un procedimiento a seguir en caso de que surjan cambios a nivel estructural. Este procedimiento consiste en, primero realizar los cambios en los documentos de diseño, posteriormente generar los nuevos scripts para creación de la base de datos y finalmente realizar los cambios a nivel de código. De esta manera siempre habrá una consistencia entre todas las partes donde está involucrado el diseño estructural del sistema.
Problemas con la integración del módulo de visualización	Ya que es muy importante que la integración con el módulo de visualización pueda funcionar correctamente, desde un inicio se han establecido diversas posibilidades para realizar la integración entre ambos sistemas. Al tener varias posibilidades de integración, se reduce la probabilidad de que existan problemas al momento de integrar los sistemas.

CAPÍTULO IV: DESARROLLO

A lo largo de este capítulo, se mostrará como desde una necesidad del usuario final, se ha planteado y se ha construido una solución software utilizando diferentes tecnologías. Se explicará cada una de las tecnologías usadas para posteriormente entrar directamente en el diseño completo de la solución, viendo cómo se ha construido, desde la capa de persistencia, hasta la interfaz gráfica.

Necesidad Usuario

Como ya ha sido mencionado, se parte del hecho que se ha creado la base de conocimientos llamada **Semantic PHI-BASE**, la cual contiene información relativa a “Pathogen-Host-Interaction”.

La aplicación Web a crear tendrá como objetivo principal, permitir a los usuarios realizar consultas sobre la base de conocimientos **Semantic PHI-BASE**, creando criterios de búsqueda de manera flexible y dinámica. Adicionalmente, se podrán visualizar los resultados no sólo a manera de tabla de resultados, sino que también se podrán ver de manera gráfica, haciendo uso del módulo externo de visualización desarrollado específicamente para navegar sobre el grafo representado en la base de conocimientos.

Sin embargo, estas funcionalidades no pueden existir solas sin un soporte de administración de la aplicación. Es por esto, que también se requiere la existencia de un módulo de administración de usuarios y perfiles, así como también un registro total de las actividades que son realizadas por los usuarios finales. De acuerdo a esto, se contará también con pantallas de consulta en donde se podrán ver los ingresos de los usuarios al sistema y se tendrá un registro detallado de las consultas que han sido ejecutadas. Con este registro de consulta, será posible volver a ejecutar directamente las consultas y ver los resultados, o cargar el filtro de búsqueda seleccionado para poderlo editar y lanzar una nueva consulta.

Tareas a Desarrollar

Para cumplir con la necesidad el usuario, se plantean las siguientes tareas, pertenecientes a un proyecto de ingeniería de software:

Tabla 7 - Tareas a Desarrollar

<u>Nombre de tarea</u>	<u>Descripción</u>
FORMACIÓN	
Lectura documentación inicial	Consiste en la primera lectura de la documentación entregada por el tutor del proyecto.
Entendimiento conceptos clave Semantic PHI-Base	En esta tarea se hará un análisis más profundo de la documentación para entender los conceptos involucrados y tener una idea más clara sobre el proyecto y como cumplir los objetivos.
Investigación y selección de tecnología para desarrollo	Luego de haber entendido de una mejor manera la base del proyecto, se debe investigar sobre tecnologías disponibles para el desarrollo y seleccionar una de acuerdo a las capacidades y conocimientos previos y que además permita cumplir con lo que se requiere en el proyecto.
DISEÑO SOFTWARE	
Especificación funcionalidades a desarrollar	Esta tarea consiste en especificar cada una de las funcionalidades que va a tener el sistema a desarrollar. Se debe saber por cada funcionalidad, sus pre-condiciones, pos-condiciones y el flujo de eventos normales y de eventos alternos.
Diseño modelo de datos	Aquí se debe diseñar un modelo de base de datos que vaya de acuerdo con las funcionalidades especificadas y que de soporte a la aplicación.
Diseño arquitectura de software y servicios	Se trata de generar la arquitectura de software del sistema. Saber qué componentes software se van a desarrollar y cómo se van a construir y comunicar entre ellos.
Diseño integración externa	En esta actividad se diseñará la integración con el sistema que se encargará de la visualización de los resultados obtenidos de acuerdo a las queries ejecutadas por los usuarios del sistema.
Diseño interfaz gráfica aplicación	Se trata de realizar un diseño de la interfaz gráfica de la aplicación. Generar plantillas para las pantallas y saber cómo será el estilo de la

	aplicación. También se diseñará la interacción de usuario sistema para que sea lo más amigable y natural posible.
IMPLEMENTACION SOFTWARE	
Construcción base de datos	Se trata de la construcción de la base de datos diseñada previamente. Se deben generar todos los scripts para construir la base de datos y para realizar una población inicial base que permita una correcta ejecución del sistema más adelante.
Construcción capa de datos y servicios	Se trata de construir la capa de aplicación que se encarga de la comunicación con la base de datos, con el Endpoint de SPARQL y de todos los servicios que serán consumidos por el nivel de interfaz de usuario.
Construcción interfaz gráfica	Aquí se construirán todas las pantallas de la aplicación siguiendo estándares para el desarrollo Web y técnicas de usabilidad.
Despliegue aplicación en entorno web	En este punto se realizará el despliegue de la aplicación Web en un entorno real para posteriormente poder realizar las pruebas necesarias.
PRUEBAS	
Pruebas unitarias software	Se trata de realizar pruebas por separado de cada uno de los componentes que hacen parte de la aplicación.
Pruebas de integración	Aquí se probarán todos los componentes en conjunto de la aplicación y además se probará la correcta integración con el módulo de visualización.
DOCUMENTACIÓN	
Creación manual de desarrollador	Aquí se creará un manual para desarrollador en donde se indicarán todos los componentes, técnicas y tecnologías usadas en el desarrollo y lo que se debe hacer para cambiar o adicionar nuevos componentes en el sistema en el momento en que ya se encuentre en producción.
Creación manual de usuario	Se realizará la creación de un manual de usuario final para que sea posible consultarlo, si se presenta algún problema, a la hora de usar la aplicación.
MEMORIA FINAL	

Creación plantilla memoria final	Se trata de crear la plantilla inicial con todos los apartados que tendrá la memoria final del proyecto desarrollado.
Contenido memoria final	Aquí se pondrá todo el contenido de la memoria final en una primera versión.
Revisión memoria	Aquí se revisará toda la memoria final y se realizarán las correcciones que sean pertinentes.

Diagrama de Gantt

De acuerdo a las tareas definidas, el siguiente es el diagrama de Gantt en el que se puede apreciar la duración y la precedencia de cada una de ellas:




























		Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1			▸ DESARROLLO DE INTERFACES WEB PARA SEMANTIC PHI-BASE	484 horas	lun 15/02/16	vie 17/06/16	
2			▸ FORMACIÓN	24 horas	lun 15/02/16	jue 18/02/16	
3			Lectura documentación inicial	6 horas	lun 15/02/16	lun 15/02/16	
4			Entendimiento conceptos clave Semantic PHI-Base	10 horas	mar 16/02/16	mié 17/02/16	3
5			Investigación y selección de tecnología para desarrollo	8 horas	mié 17/02/16	jue 18/02/16	4
6			▸ DISEÑO SOFTWARE	90 horas	vie 19/02/16	vie 11/03/16	
7			Especificación funcionalidades a desarrollar	20 horas	vie 19/02/16	mié 24/02/16	5
8			Diseño modelo de datos	15 horas	mié 24/02/16	vie 26/02/16	7
9			Diseño arquitectura de software y servicios	20 horas	vie 26/02/16	jue 03/03/16	8
10			Diseño integración externa	15 horas	jue 03/03/16	mar 08/03/16	9
11			Diseño interfaz gráfica aplicación	20 horas	mar 08/03/16	vie 11/03/16	10
12			▸ IMPLEMENTACION SOFTWARE	160 horas	lun 14/03/16	mié 27/04/16	
13			Construcción base de datos	20 horas	lun 14/03/16	jue 17/03/16	11
14			Construcción capa de datos y servicios	50 horas	jue 17/03/16	mié 06/04/16	13
15			Construcción interfaz gráfica	80 horas	mié 06/04/16	lun 25/04/16	14
16			Despliegue aplicación en entorno web	10 horas	mar 26/04/16	mié 27/04/16	15
17			▸ PRUEBAS	60 horas	mié 27/04/16	jue 12/05/16	
18			Pruebas unitarias software	40 horas	mié 27/04/16	lun 09/05/16	16
19			Pruebas de integración	20 horas	lun 09/05/16	jue 12/05/16	18
20			▸ DOCUMENTACIÓN	60 horas	jue 12/05/16	vie 27/05/16	
21			Creación manual de desarrollador	30 horas	jue 12/05/16	vie 20/05/16	19
22			Creación manual de usuario	30 horas	vie 20/05/16	vie 27/05/16	21
23			▸ MEMORIA FINAL	90 horas	vie 27/05/16	vie 17/06/16	
24			Creación plantilla memoria final	3 horas	vie 27/05/16	lun 30/05/16	22
25			Contenido memoria final	75 horas	lun 30/05/16	mié 15/06/16	24
26			Revisión memoria	12 horas	mié 15/06/16	vie 17/06/16	25

Imagen 3 - Diagrama de Gantt

Tecnologías Usadas

Es importante conocer las tecnologías que han sido utilizadas para la construcción de **Semantic PHI-BASE Web Interface**, ya que con esto se podrá determinar las capacidades actuales que tiene el sistema, así como las posibilidades de escalar con nuevas funcionalidades y con número de usuarios.

Java

Java es una tecnología, actualmente perteneciente a la compañía Oracle, ampliamente usada a nivel mundial para el desarrollo de aplicaciones. Su sintaxis es similar a la utilizada por el lenguaje C. Es importante no confundir Java con JavaScript, que es una tecnología, lenguaje de programación, utilizado en el desarrollo de páginas web y es ejecutado únicamente en el navegador web.

Java cuenta con muchas funcionalidades y Frameworks que permiten que el desarrollo de aplicaciones se pueda hacer de manera más sencilla y más eficiente. Es por esto que se ha seleccionado este lenguaje como principal lenguaje para la construcción de **Semantic PHI-BASE Web Interface**.



Imagen 4 - Ícono Java

Eclipse Kepler

Eclipse es más que un IDE (Integrated Development Environment), es una comunidad *Open Source* de herramientas, proyectos y grupos de trabajo colaborativos. El IDE de Eclipse, ha sido desarrollado inicialmente para la construcción de aplicaciones Java, pero hoy en día, es posible desarrollar aplicaciones en C/C++ y PHP también.

Para la construcción de **Semantic PHI-BASE Web Interface**, se ha utilizado la edición “Eclipse Kepler”, que es la 4.3. Se ha decidido utilizar esta edición y no la última (Eclipse Mars al inicio del desarrollo), ya que para esta edición existe un soporte adicional que ayuda en el desarrollo de aplicaciones **RIA** (Rich Internet Application), utilizando el framework de **ICEfaces**.

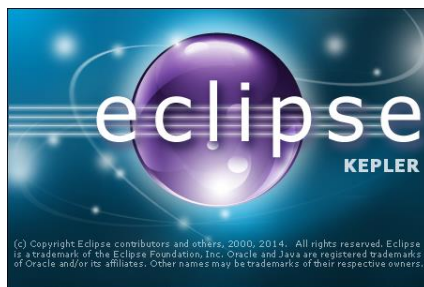


Imagen 5 - Ícono Eclipse

MySQL Database – MySQL Workbench

MySQL es la base de datos más popular de tipo *Open Source*. Hoy en día, también pertenece a Oracle y tiene varias ediciones de pago. Para la construcción de **Semantic PHI-BASE Web Interface** se ha utilizado la edición estándar que sigue siendo *Open Source*. Existen dos tipos de motores de bases de datos dentro de MySQL:

- ✓ MyISAM: Es un motor, en el cual las tablas creadas no tienen las propiedades de verificación de integridad referencial. Esto es bastante útil para construir tablas que van a ser utilizadas únicamente para consultas. Por ejemplo, para la construcción de Data Warehouses.
- ✓ InnoDB: Es el motor por defecto de MySQL. Este motor si realiza todas las verificaciones en los momentos de realizar Update, Insert y Delete en la base de datos. Es el motor que se está utilizando para dar soporte a las tablas de **Semantic PHI-BASE Web Interface**.

Para realizar el diseño y los scripts de creación y población inicial de la base de datos, se ha utilizado la herramienta *MySQL Workbench*, la cual permite de una manera muy gráfica crear los modelos e interactuar directamente con la base de datos.



Imagen 6 - Ícono MySQL

Apache Jena

Es un framework Java de tipo *Open Source*, que permite construir aplicaciones que utilizan Web Semántica y datos enlazados (Linked Data). Este framework ha sido utilizado dentro de la construcción de **Semantic PHI-BASE Web Interface** para realizar la comunicación entre la aplicación Web y la base de conocimientos de Virtuoso.

Con el uso de este framework, tanto la ejecución de las consultas, como la obtención de los resultados, se hacen de una manera muy sencilla y transparente, por lo que lo más complicado es la creación automática del texto de la consulta a ejecutar.



Imagen 7 - Ícono Apache Jena

ICEfaces Software

ICEfaces es también un framework de tipo *Open Source*, utilizado para el desarrollo de **RIA (Rich Internet Application)**, dentro de **JavaEE (Java Enterprise Edition)**. Con ICEfaces es posible crear aplicaciones Web que se ejecutan tanto en equipos de escritorio, como en teléfonos inteligentes o tabletas.

ICEfaces provee un gran número de componentes integrados, que hace que el desarrollo de las interfaces gráficas sea muy sencillo. Además, las aplicaciones desarrolladas con ICEfaces utilizan siempre AJAX para la comunicación con el servidor, por lo que el usuario no verá recargas de página molestas.



Imagen 8 - Ícono ICEfaces

Spring & Hibernate Framework

Spring junto Hibernate son ambos Frameworks Java que ayudan en el desarrollo de las aplicaciones empresariales, cada uno con ventajas para permitir una mayor eficiencia, reducción del código a escribir y por lo tanto mejora en los tiempos de desarrollo.

- ✓ Hibernate: Es un framework ORM (Object Reational Mapping), el cual ayuda a la conversión entre registros de la base de datos a objetos concretos de Java. También elimina la necesidad de escribir SQL directo en la aplicación por lo que incrementa el bajo acoplamiento entre la capa de persistencia de la aplicación Java y el motor de bases de datos.
- ✓ Spring: Es un framework ampliamente utilizado en Java, cuyo principal objetivo es permitir la inyección de dependencias para mantener un bajo acoplamiento entre las capas de la aplicación. Inyección de dependencias hace referencia a que en el nivel superior se trabaja con Interfaces Java y en momento de ejecución es cuando se inyectan las clases concretas que son las que tienen implementadas las funcionalidades a ejecutar.



Imagen 9 - Ícono Spring-Hibernate

Amazon Web Services AWS

Amazon Web Services es una plataforma de Cloud Computing, por lo que ofrece servicios de computo, almacenamiento, bases de datos, redes, etc. AWS provee más de 50 servicios, los cuales pueden ser administrados a través de la Web o de APIs de programación.

Al tratarse de un Cloud público, los costes se van generando dinámicamente de acuerdo a los servicios que se van consumiendo y no son fijos.

Se ha seleccionado esta plataforma como alternativa de despliegue para **Semantic PHI-BASE Web Interface**, debido a su gran potencial, alta disponibilidad y a la capacidad de escalar automáticamente cuando sea necesario.



Imagen 10 - Ícono Amazon Web Services AWS

Arquitectura de Software

Al tratarse de una aplicación Web, se ha seguido el esquema tradicional de tres capas para la construcción de la misma. La siguiente imagen, ilustra cómo es la arquitectura de software de **Semantic PHI-BASE Web Interface**:

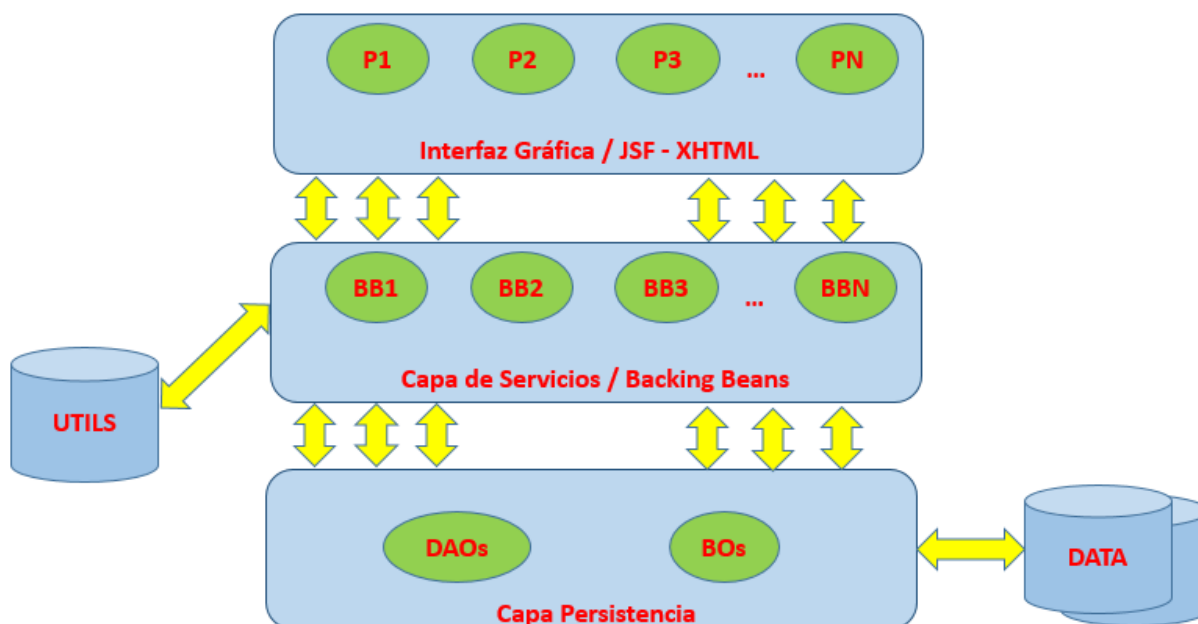


Imagen 11 - Arquitectura de Software

A continuación, se explica cada una de las capas de la arquitectura, comenzando desde la parte inferior:

- ✓ **Capa Persistencia:** En este nivel se encuentran las clases BO (Business Objects), las cuales son una representación a nivel de clases de las tablas definidas en la base de datos. Adicionalmente, se tienen las clases llamadas DAO (Data Access Object), las cuales son las encargadas de hacer directamente el acceso a la base de datos y de realizar la conversión de registros de la base de datos a BOs y viceversa.
- ✓ **Capa de Servicios / Backing Beans:** Esta capa contiene toda la lógica de negocio y los componentes necesarios para atender a los llamados que se hacen desde la interfaz gráfica. Los **Backing Beans** son clases que están directamente relacionadas con la interfaz y que reaccionan a los eventos ejecutados desde pantalla. De la misma manera, esta capa de servicios necesita los componentes de utilidades (UTILS) con los que se hacen operaciones específicas. Por ejemplo, en UTILS, se encuentran las clases necesarias para ejecutar las consultas a **Semantic PHI-BASE**, así como también hay utilidades para manejo de fechas, criptografía, escritura en log, etc.
- ✓ **Interfaz Gráfica / JSF-XHTML:** En esta capa, se encuentran definidas todas las páginas web de la aplicación, las plantillas, scripts de JavaScript y hojas de estilo CSS. Las páginas **JSF (Java Server Faces)**, son las que se comunican directamente con los **Backing Beans** para obtener los datos a visualizar en pantalla y lanzar la ejecución de eventos.

Diseño de Base de datos

En esta sección, se explicará, de una manera breve, el diseño de base de datos que se ha realizado para dar soporte de persistencia a la aplicación **Semantic PHI-BASE Web Interface**. El diseño completo de la base de datos se explica en el [Anexo A – Diseño de Base de Datos](#).

Convenciones

Para la definición de los nombres de las tablas y campos de la base de datos, se ha definido el siguiente estándar:

Estándar nombre de las tablas (Longitud de 8 caracteres):

- ✓ Primeros dos caracteres: Módulo del sistema. Se han definido los siguientes módulos:
 - Configuración Sistema: CS
 - Usuarios: US
 - Semántica: SE
- ✓ Tercer carácter:
 - P: Parámetros
 - D: Detalle
 - M: Maestro
- ✓ Siguiendo cinco caracteres: Nombre significativo para la tabla.

Estándar nombre de Campos (Longitud de 9 caracteres):

- ✓ Primeros 3 caracteres: Prefijo único de la tabla.
- ✓ Siguiendo 4 caracteres: Libre para nombre de campo.
- ✓ Siguiendo carácter (carácter 8): Indicador de tipo de campo.
 - A: Alfanumerico
 - N: Numérico
 - M: Monetario
 - V: Multivalor
 - S: Booleano
 - F: Fecha
 - T: Hora
 - P: Porcentaje
- ✓ Último carácter (carácter 9): Indicador de campo o llave.
 - F: Campo
 - K: Llave

Tablas Base de Datos

A continuación, se presenta la definición y objetivo de cada una de las tablas de la base de datos:

Tabla 8 - Tablas Base de Datos

<u>Nombre Tabla</u>	<u>Descripción Tabla</u>
CSPPERFU – PERFIL USUARIO	Tabla de configuración para crear los diferentes perfiles de usuario que habrá en el sistema.
CSPSERSI – SERVICIO SISTEMA	Tabla de configuración para tener los servicios que ofrecerá el sistema. Esto permitirá que se vayan adicionando dinámicamente nuevos servicios conforme la aplicación vaya evolucionando.
CSPSERPF – SERVICIO DE PERFIL DE USUARIO	Esta tabla permitirá la asociación de servicios del sistema a los perfiles. De esta manera se podrá saber qué servicios podrán ser ejecutados por los usuarios que tengan un determinado perfil.
CSPPARSI – PARÁMETRO SISTEMA	Tabla base para almacenar los parámetros de configuración del sistema. Se almacenarán parámetros como por ejemplo el Endpoint de Virtuoso, el número mínimo de caracteres en una contraseña, etc.
USMUSUAR – USUARIO SISTEMA	Tabla maestra para almacenar los usuarios registrados en el sistema. Únicamente los usuarios que se encuentren registrados en esta tabla son los que podrán acceder al sistema y ejecutar los servicios que tengan asociados de acuerdo a su perfil de usuario.
USDBLOQU – HISTORICO BLOQUEOS USUARIO	En esta tabla se almacenarán los bloqueos que tenga un usuario en su cuenta cuando excede el número máximo de intentos de login de una manera no satisfactoria.
USDLOGQU – LOG QUERY USUARIO	Tabla para almacenar todas las queries que hayan sido ejecutadas por cada uno de los usuarios del sistema. Se almacenarán todos los datos necesarios para poder ejecutar nuevamente la consulta o para cargar los criterios de búsqueda seleccionados.
USDLOGIN – LOG INGRESO USUARIO	En esta tabla se almacenarán todos los intentos de login que haga un usuario dentro del sistema. Se almacenará fecha y hora del login junto con la dirección IP de la máquina desde donde se está realizando el ingreso.

SEPCONCE – CONCEPTO SEMÁNTICO	Tabla de configuración para almacenar los conceptos semánticos de la ontología de Semantic PHI-BASE . Servirá para poder representar y almacenar en la base de datos los conceptos que se han definido con RDF . Un ejemplo de un concepto semántico es “Host”, “Pathogen” o “Gene”.
SEPRELCO – RELACIÓN CONCEPTO SEMÁNTICO	Tabla de configuración para almacenar las relaciones que pueden tener los diferentes conceptos semánticos definidos previamente. Un ejemplo de una relación puede ser “has participant” entre “Interaction” y “Host”. Adicionalmente, también se contempla en esta tabla el rango de la relación para en un futuro poder realizar una inserción controlada de nuevas tripletas en la base de conocimiento.
SEDCONCE – DETALLE CONCEPTO SEMÁNTICO	Contiene información de un concepto propio de la base de conocimientos. Es decir, representa una instancia concreta del modelo. Servirá para poder mostrar los filtros de búsqueda en la pantalla principal de consulta a Semantic PHI-BASE .
SEDRELCO – DETALLE RELACIÓN CONCEPTO SEMÁNTICO	Contiene el valor de una relación entre instancias de conceptos semánticos. Por ejemplo, contendrá el valor “ <i>PHI:164</i> ” de un concepto tipo “ <i>Phi base ID</i> ” y la relación “ <i>has value</i> ”.

Modelo Entidad-Relación

De acuerdo a las tablas definidas anteriormente, el siguiente es el modelo Entidad-Relación en donde se puede apreciar de una mejor manera el diseño realizado:

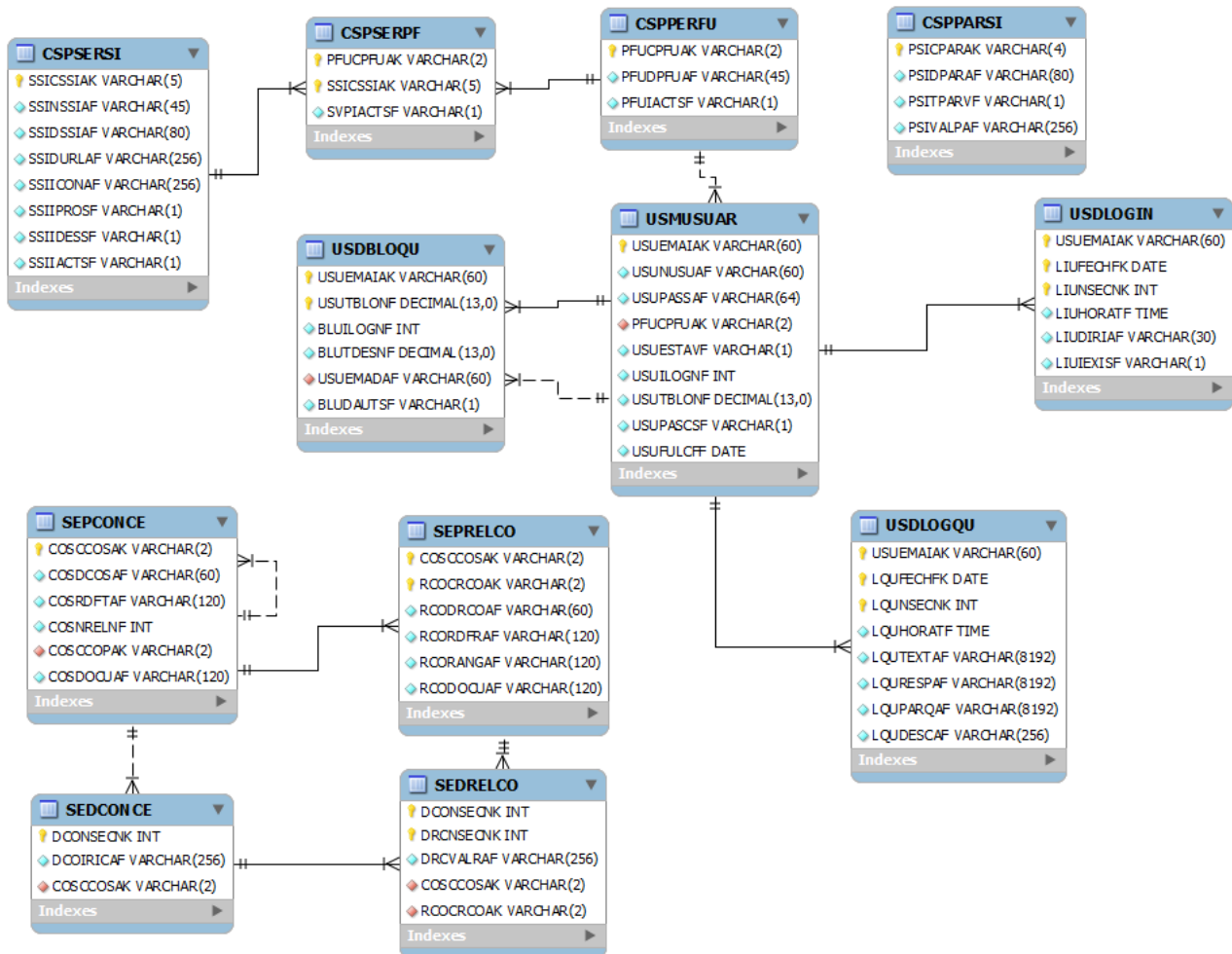


Imagen 12 - Modelo Entidad-Relación

Especificación Básica de Casos de Uso

Actores del Sistema

Debido a que el sistema contará con un diseño dinámico para el ajuste de perfiles de usuario y sus funcionalidades, sólo se contará con un tipo de actor que será el usuario del sistema. El usuario del sistema ejecutará las acciones a las que esté autorizado de acuerdo a su perfil.

Descripción de Casos de Uso

A continuación, se presenta la lista de los casos de uso que se desarrollarán para cumplir con los objetivos de la construcción de una interfaz Web para **Semantic PHI-BASE**. Es importante mencionar que los casos de uso con la palabra “Mantenimiento” hacen referencia a el conjunto de pantallas para administrar una tabla de configuración del sistema (“CRUD” – Create, Read, Update, Delete). La especificación detallada de casos de uso (EDCU – Especificación Detallada de Caso de Uso) se encuentra en el [Anexo B – Especificación Detallada de Casos de Uso](#).

Tabla 9 - Descripción Casos de Uso

<u>Identificador</u>	<u>Nombre</u>	<u>Descripción</u>
CU-001	Inicio de sesión en el sistema	Esta funcionalidad será la pantalla inicial de la aplicación en la que se solicitará el Id de usuario (su email) y su contraseña para iniciar sesión. Dependiendo de si el login es satisfactorio o no se irá a la pantalla del menú principal de la aplicación o a la misma pantalla desplegando mensaje de error.
CU-002	Mantenimiento Servicio Sistema	Se trata de mantener la tabla que contiene los servicios del sistema. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.
CU-003	Mantenimiento Perfil de Usuario	Se trata de mantener la tabla que contiene los perfiles de usuario. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.
CU-004	Mantenimiento Parámetro Sistema	Se trata de mantener la tabla que contiene parámetros del sistema. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.

CU-005	Mantenimiento Concepto Semántico – Relación Concepto	Se trata de mantener las tablas que contienen los conceptos semánticos y sus relaciones. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.
CU-006	Mantenimiento Detalle Concepto Semántico – Detalle Relación Concepto	Se trata de mantener las tablas que contienen los detalles de conceptos semánticos junto con sus relaciones. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.
CU-007	Mantenimiento Usuario Sistema	Se trata de mantener la tabla que contiene los usuarios del sistema. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD. Adicionalmente a las funcionalidades básicas, se permitirá realizar el desbloqueo manual de un usuario y la regeneración de su contraseña.
CU-008	Consulta Histórico Bloqueos Usuario	Se trata de proveer la funcionalidad de consulta de la tabla de histórico de bloqueos de los usuarios. Se podrá consultar por rango de fechas y por un usuario en específico.
CU-009	Consulta Log Ingreso Usuario	Se trata de proveer la funcionalidad de consulta de la tabla de log de ingreso de los usuarios. Se podrá consultar por rango de fechas y por un usuario en específico.
CU-010	Consulta Log Query Usuario	Se trata de proveer la funcionalidad de consulta de la tabla de log de queries ejecutadas por los usuarios. Se podrá consultar por rango de fechas y por un usuario en específico.
CU-011	Consulta Semantic PHI-BASE	Es la funcionalidad principal del sistema en donde se podrán crear queries SPARQL dinámicamente que se ejecutarán en el Endpoint de Virtuoso y se desplegarán los resultados en una tabla pudiendo también ser consultados por un módulo de visualización de grafo externo.

CU-012	Proceso Desbloqueo Automático de Usuarios	Es un proceso que debe correr periódicamente para desbloquear los usuarios que han sido bloqueados por intentos fallidos en el inicio de sesión.
CU-013	Proceso Cargue Automático Detalle Concepto Semántico	Es un proceso que servirá para cargar automáticamente los datos de las estructuras de Detalle Concepto Semántico y sus relaciones de acuerdo a las configuraciones previamente realizadas en los conceptos semánticos.
CU-014	Actualización Contraseña Usuario	Funcionalidad que permitirá al usuario actualizar su contraseña en el momento en que lo desee o cuando el sistema le indique que deba realizarlo.
CU-015	Asociar Servicio a Perfil de Usuario	Permite asociar servicios del sistema a los perfiles de usuario para saber a qué servicios tiene acceso un determinado perfil.

Diseño de Componentes

Como ya se ha mencionado, **Semantic PHI-BASE Web Interface** cuenta con una arquitectura de software clásica de tres niveles, en los que cada nivel tiene su función específica. A continuación, se explica detalladamente el diseño de cada uno de los componentes que forman parte de cada capa de la aplicación.

Capa de Persistencia

La capa de persistencia cuenta con los BOs (Business Objects) y con los DAOs (Data Access Object) que son los encargados del acceso a los datos. Por cada tabla de la base de datos existe un BO correspondiente que lo representa a manera de clase junto con una clase DAO encargada de realizar las operaciones de acceso a datos para la tabla específica. Es importante mencionar, que se ha realizado un diseño basado en interfaces, de manera que cuando en la capa de servicios se requiere instanciar un DAO específico, esto se realiza a través de la inyección de dependencias que permite el framework de Spring. A continuación, se muestra un diagrama de clases resumido, en el que se puede ver el diseño con interfaces para el uso de la inyección de dependencias (No se muestra el diseño completo por cuestiones de espacio):

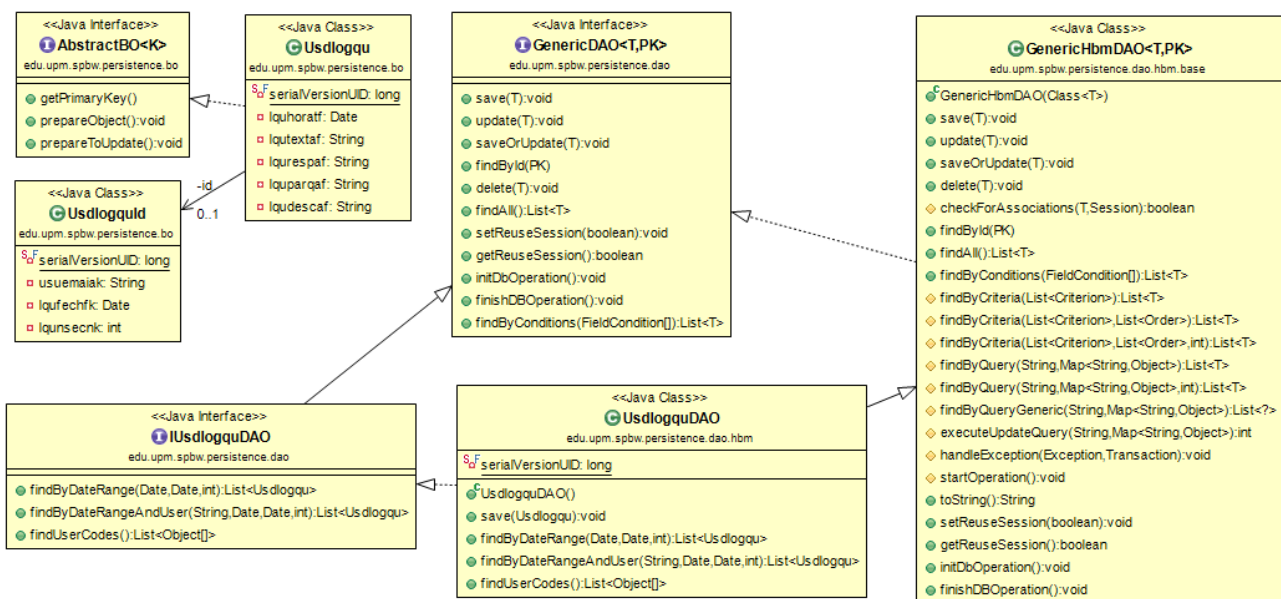


Imagen 13 - Diagrama Clases Capa Persistencia

Como se puede ver en el diagrama de clases, a nivel de BOs, existe una clase con el mismo nombre de la tabla en la base de datos (para el ejemplo Usdlogqu – Log Query Usuario), la cual, en este caso, tiene una clase adicional que representa su llave primaria UsdlogquId. Es necesario tener una clase adicional para la llave primaria en los casos en que ésta sea compuesta. También se puede observar que la clase BO implementa la interfaz **AbstractBO**, la cual define unos métodos básicos que todos los BOs deben implementar para el buen funcionamiento de la aplicación.

En cuanto a los DAOs, se puede observar en el diagrama, que existe una interfaz llamada **GenericDAO**, la cual define los métodos básicos que debe tener todo DAO que se encuentre en la aplicación. Estos métodos son:

- ✓ `save(obj)`: Se encarga de guardar un objeto en la base de datos.
- ✓ `update(obj)`: Actualiza el objeto en la base de datos.
- ✓ `delete(obj)`: Elimina el objeto de la base de datos.
- ✓ `findById(objId)`: Busca un objeto en la base de datos, dado su llave primaria.
- ✓ `findAll()`: Busca todos los elementos pertenecientes a una tabla de la base de datos.
- ✓ Operaciones para reutilización de las sesiones de la base de datos.

También existe una clase abstracta llamada **GenericHmbDAO**, la cual representa al DAO genérico del framework de Hibernate. Este DAO genérico de Hibernate ya implementa todas las operaciones básicas definidas por **GenericDAO** y adiciona unas operaciones nuevas para facilitar la generación de consultas con Hibernate.

Ya propio de cada tabla de la base de datos, existe su interfaz DAO, que para el caso del modelo de ejemplo sería **IUsdlogquDAO**, la cual implementa la interfaz de **GenericDAO**. Finalmente, la clase concreta para el ejemplo es **UsdlogquDAO**, la cual extiende de **GenericHmbDAO** para tener todas sus operaciones e implementa la interfaz **IUsdlogquDAO**, lo que le obligará la implementación de las operaciones específicas para la tabla USDLOGQU.

Al realizar la implementación de la capa de persistencia de esta manera, es posible realizar una inyección de dependencias transparente en el nivel superior, utilizando el framework de Spring.

Por ejemplo, para solicitar una nueva instancia del DAO para manejar las operaciones de base de datos con la tabla USDLOGQU, basta con ejecutar la siguiente línea de código:

```
usdlogquDao = applicationContext.getBean(IUsdlogquDAO.class);
```

Como se puede ver, lo único que se debe hacer es solicitar al contexto de la aplicación de Spring que nos dé una instancia concreta de “IUsdlogquDAO”. En el momento en que se realiza la petición, Spring se da cuenta que la clase concreta que realiza la implementación de “IUsdlogquDAO” es la clase “UsdlogquDAO” que utiliza el framework de Hibernate para realizar el acceso a los datos.

Capa de Servicios (Backing Beans)

La capa de servicios es la encargada de recibir las peticiones que vienen desde la interfaz gráfica, realizar el procesamiento requerido y realizar las modificaciones al modelo para que se actualice la vista. La actualización de la interfaz se realiza de acuerdo al patrón MVC (Model View Controller), en el que la Vista (Interfaz Gráfica), depende del estado del modelo, el cual puede cambiar de acuerdo a las acciones realizadas por el Controller (Controlador). Las clases encargadas de recibir las peticiones son las llamadas “Backing Beans”, que significa los Java Beans que se encuentran en el Back-End de una aplicación recibiendo las notificaciones.

El siguiente modelo, ilustra las clases más importantes pertenecientes a la capa de servicios:

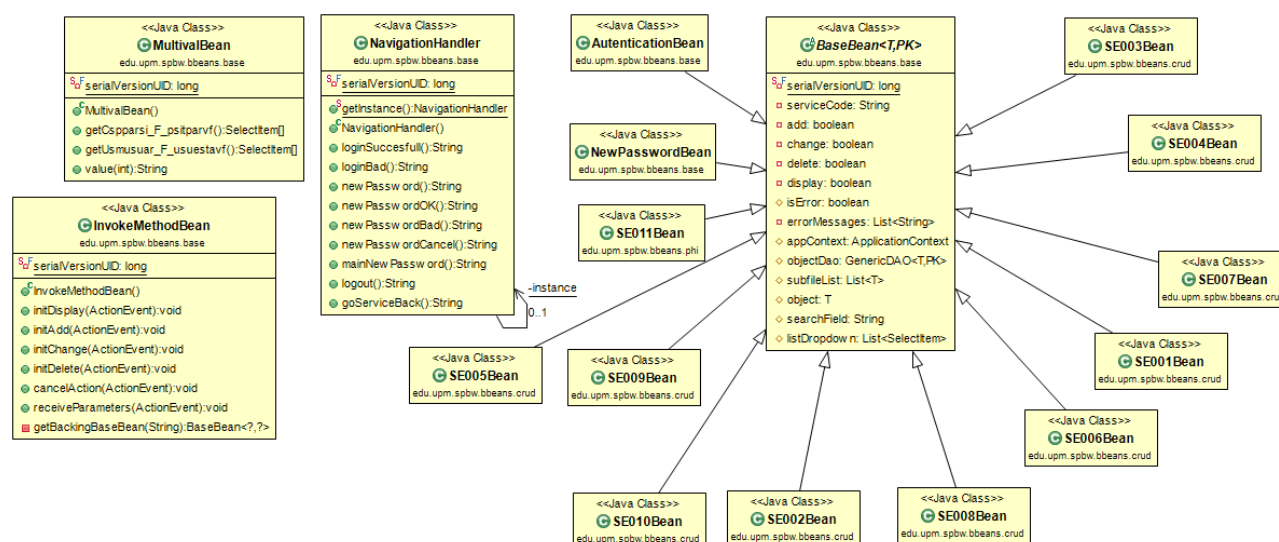


Imagen 14 - Diagrama Clases Capa Servicios

A continuación, se describe cada una de las clases pertenecientes a la capa de servicios:

Tabla 10 - Descripción Clases Capa Servicios

<u>Clase</u>	<u>Descripción</u>
BaseBean	Es el Backing Bean base para todos los Backing Beans que soportan servicios en el sistema. Contiene todos los atributos básicos que deben tener todos los Backing Beans, así como la implementación de las operaciones estándar de cada uno de ellos. Algunos de sus atributos básicos son el código de servicio, indicador de mostrar pantalla de adicionar, indicador de mostrar pantalla de actualizar, lista de tabla principal en pantalla, etc. En cuanto a sus operaciones básicas, algunas son: Revisar si el usuario está autenticado, adicionar un registro, actualizar un registro, eliminar un registro, recibir parámetros de otro Backing Bean, etc.
AutenticationBean	Es el Bean que da soporte a la autenticación del sistema. Contiene la información básica de autenticación como el usuario en sesión, la fecha

	de ingreso, lista de servicios consultados, etc. Tiene operaciones para autenticar y/o autorizar a un usuario a un servicio específico.
NewPasswordBean	Bean encargado de las operaciones relacionadas con la actualización de la contraseña de un usuario. Realiza las validaciones necesarias para determinar si es posible actualizar o no la contraseña de un usuario y de ser posible lo hace.
InvokeMethodBean	Bean especial para realizar la invocación de métodos en los Beans de servicios. Invoca los métodos para iniciar una adición, modificación, eliminación o detalle de un registro, así como también invoca el método para recibir parámetros que se pasan de un Bean a otro.
MultivalBean	Bean que contiene los valores de los campos tipo “multivalor” de las tablas de la base de datos. Los campos multivalor son los que tienen valores fijos como por ejemplo “Tipo de parámetro” en la tabla CSPPARSI – Parámetro del Sistema. Sus valores fijos son: ‘1’ – Alfanumérico, ‘2’ Numérico.
SE001Bean	Bean relativo al servicio encargado del mantenimiento de la tabla CSPSERSI – Servicio del Sistema. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros. Ver Especificación detallada de Casos de Uso EDCU CU-002 .
SE002Bean	Bean relativo al servicio encargado del mantenimiento de la tabla CSPPERFU – Perfil de Usuario. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros. Ver Especificación detallada de Casos de Uso EDCU CU-003 .
SE003Bean	Bean relativo al servicio encargado del mantenimiento de la tabla CSPSERPF – Servicios de Perfil de Usuario. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros. Ver Especificación detallada de Casos de Uso EDCU CU-015 .
SE004Bean	Bean relativo al servicio encargado del mantenimiento de la tabla CSPPARSI – Parámetro del Sistema. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros. Ver Especificación detallada de Casos de Uso EDCU CU-004 .
SE005Bean	Bean relativo al servicio encargado del mantenimiento de la tabla USMUSUAR – Usuario del Sistema. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros. También se encarga de llevar las operaciones para la regeneración de las contraseñas de los usuarios y su desbloqueo manual. Ver Especificación detallada de Casos de Uso EDCU CU-007 .

SE006Bean	Bean relativo al servicio encargado de la consulta de la tabla USDLOGIN – Log de Ingreso Usuario. Ver Especificación detallada de Casos de Uso EDCU CU-009 .
SE007Bean	Bean relativo al servicio encargado de la consulta de la tabla USDBLOQU –Bloqueos Usuario. Ver Especificación detallada de Casos de Uso EDCU CU-008 .
SE008Bean	Bean relativo al servicio encargado de la consulta de la tabla USDLOGQU –Log Query Usuario. Es posible ejecutar nuevamente las consultas del log o volver a cargar el filtro de búsqueda previamente seleccionado. Ver Especificación detallada de Casos de Uso EDCU CU-010 .
SE009Bean	Bean relativo al servicio encargado del mantenimiento de las tablas SEPCONCE – Concepto Semántico y SEPRELCO – Relación Concepto Semántico. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros de cada una de las tablas. Ver Especificación detallada de Casos de Uso EDCU CU-005 .
SE010Bean	Bean relativo al servicio encargado del mantenimiento de las tablas SEDCONCE – Detalle Concepto Semántico y SEDRELCO – Detalle Relación Concepto Semántico. Maneja las operaciones para adicionar, modificar, eliminar o visualizar los registros de cada una de las tablas. Ver Especificación detallada de Casos de Uso EDCU CU-006 .
SE011Bean	Bean que da soporte a la pantalla de Consulta Semantic PHI-BASE. Este Bean se encarga de obtener los datos necesarios para desplegar el filtro de búsqueda en la pantalla, así como de enviar los parámetros necesarios para la creación de la consulta a ejecutar y procesar los resultados obtenidos. Adicionalmente, este Bean también maneja la integración con el servicio externo de visualización de grafo. Ver Especificación detallada de Casos de Uso EDCU CU-011 .
NavigationHandler	Es una clase de soporte que se encarga de manejar los eventos de navegación entre pantallas.

Cada uno de los Beans definidos en la tabla anterior hace uso de los DAOs requeridos para el acceso a datos y de las clases de utilidades para realizar operaciones genéricas de la aplicación. Algunas de las clases de utilidades son:

- ✓ Utilitario de fechas.
- ✓ Utilitario de log.
- ✓ Utilitario ejecución Consultas SPARQL. Utilizado únicamente por SE011Bean.
- ✓ Utilitario de Criptografía.
- ✓ Utilitario de envío de correos.

Interfaz Gráfica

El desarrollo de la interfaz gráfica se ha hecho utilizando el framework de ICEfaces, con el cual se sigue una implementación de JSF (Java Server Faces). Las pantallas Web tienen la extensión XHTML o JSF.

El siguiente es el mapa de navegación entre las pantallas desarrolladas:

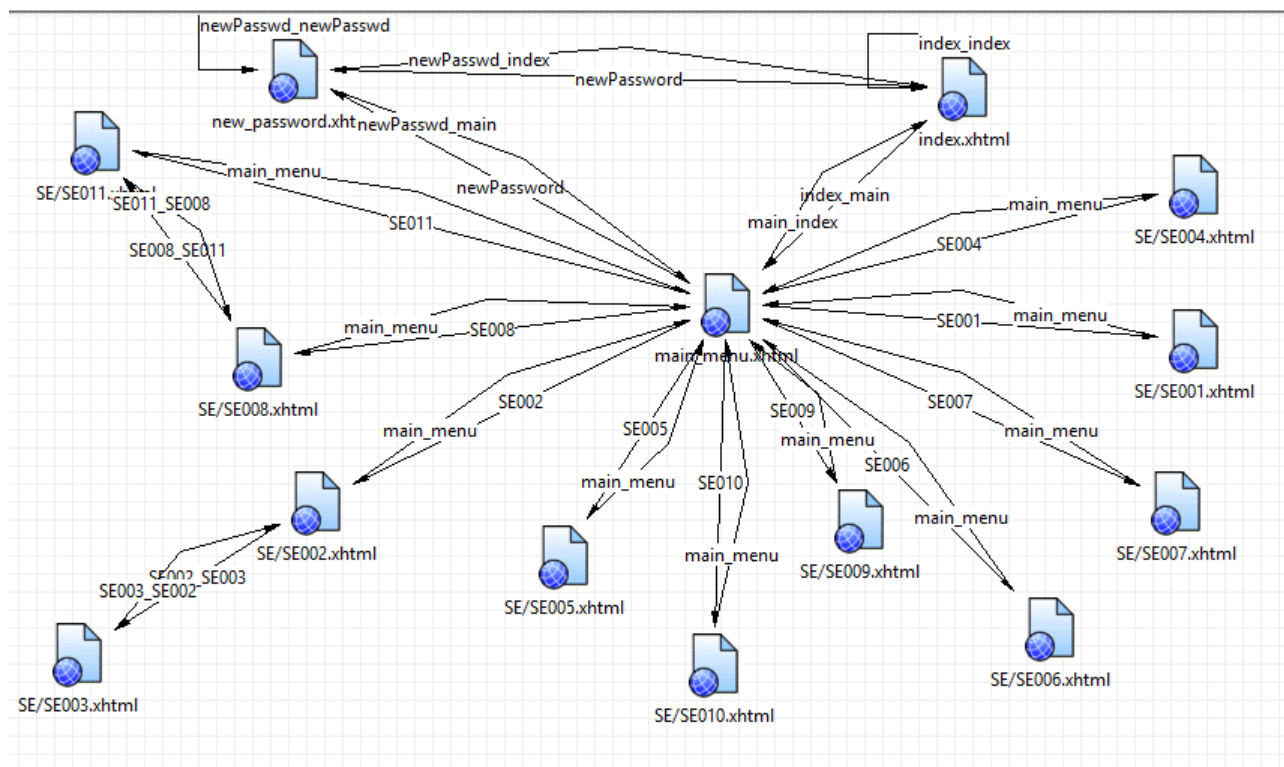


Imagen 15 - Mapa de Navegación

Como se puede observar en la imagen, para cada uno de los Backing Beans previamente definidos, existe una pantalla Web que se encarga de visualizar el modelo contenido. También es posible observar, que desde el menú principal es posible navegar hacia todos los servicios del sistema y a las pantallas principales de inicio de sesión y de actualización de contraseña.

Para facilitar el desarrollo de las pantallas Web, se han creado las siguientes plantillas que ayudan a una correcta reutilización de código:

Tabla 11 - Plantillas Web

<u>Plantilla XHTML</u>	<u>Descripción Plantilla</u>
template/help_index_template.xhtml	Plantilla diseñada para la pantalla principal del manual de usuario.
template/help_service_template.xhtml	Plantilla diseñada para cada uno de las páginas de ayuda del manual de usuario.
template/index_template.xhtml	Plantilla que contiene el diseño de las pantallas de inicio de sesión y de actualización de contraseñas.

template/main_footer.xhtml	Footer que aparece en todas las pantallas del sistema.
template/main_header.xhtml	Header que aparece en todas las pantallas del sistema.
template/main_template.xhtml	Plantilla con el diseño para los servicios del sistema.
weblets/authorization_popup.xhtml	PopUp que está integrado en todas las pantallas y se despliega en el momento en el que un usuario no esté autenticado o autorizado a utilizar el servicio.
weblets/list_paginator.xhtml	Paginador que se despliega en la parte inferior de las listas que muestran registros en pantalla. Permite navegar entre las diferentes páginas de las listas en pantalla.
weblets /std_add.xhtml	Botón que inicia la adición de un nuevo registro en el sistema.
weblets /std_change.xhtml	Botón que se despliega en una lista, relacionado a un registro específico para iniciar su modificación.
weblets /std_delete.xhtml	Botón que se despliega en una lista, relacionado a un registro específico para eliminarlo.
weblets /std_display.xhtml	Botón que se despliega en una lista, relacionado a un registro específico para desplegar su detalle completo.
weblets /std_goback.xhtml	Botón que sirve para navegar de un servicio hacia su servicio anterior.
weblets /std_panel_popup_add.xhtml	PopUp de plantilla para la pantalla de adición de un nuevo registro en el sistema.
weblets /std_panel_popup_change.xhtml	PopUp de plantilla para la modificación de un registro en el sistema.
weblets /std_panel_popup_display.xhtml	PopUp de plantilla para desplegar el detalle completo de un registro en el sistema.
weblets /std_panel_popup.xhtml	PopUp genérico para desplegar información adicional en una pantalla.

Adicionalmente a las plantillas definidas, también se han definido hojas de estilo CSS para mantener un diseño uniforme en toda la interfaz del sistema.

Arquitectura de Despliegue

Semantic PHI-BASE Web Interface se ha desplegado en el Cloud de Amazon (AWS – Amazon Web Services), de manera que sea posible aprovechar todas las capacidades de la nube y que la aplicación esté disponible la mayor parte del tiempo posible. La siguiente figura ilustra la arquitectura de despliegue diseñada sobre AWS:

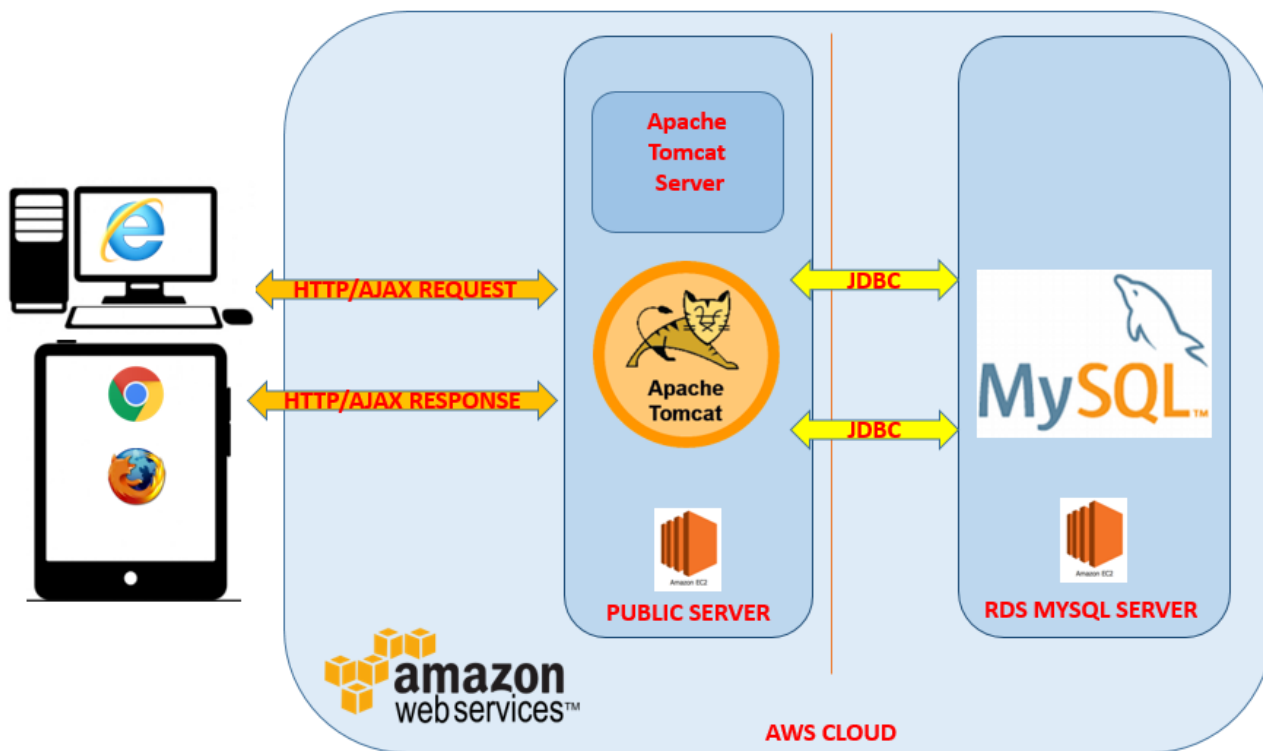


Imagen 16 - Arquitectura de Despliegue

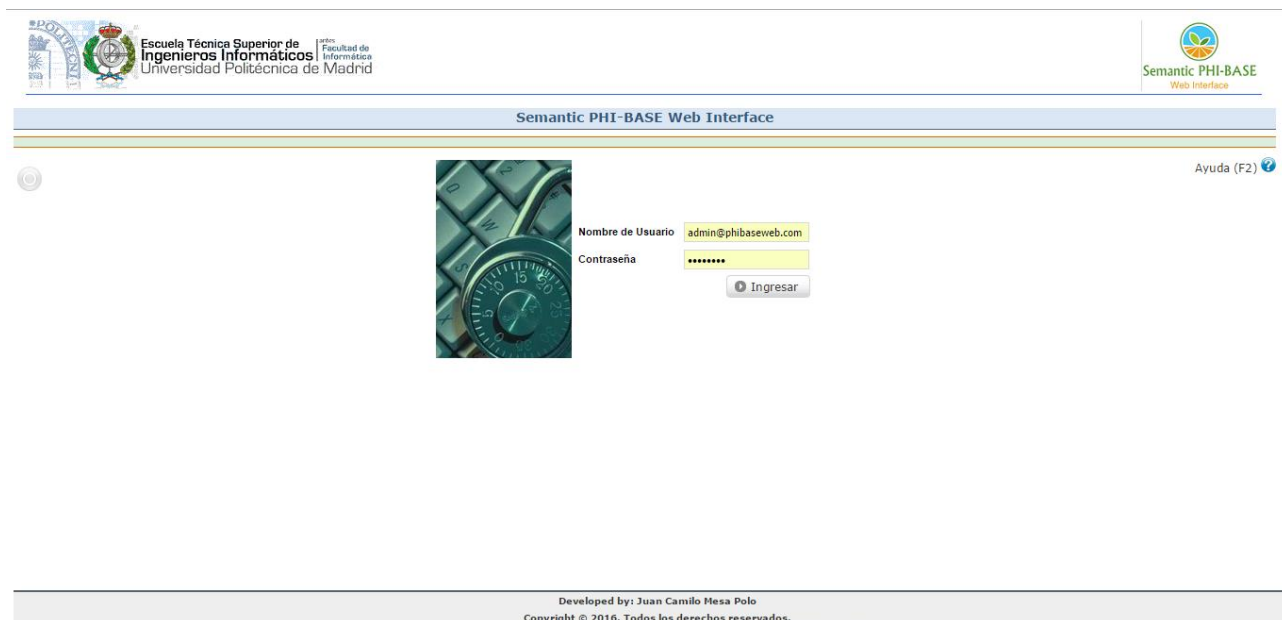
De acuerdo a la imagen anterior, el usuario final accederá a la aplicación a través de un navegador web estándar desde el que se enviarán peticiones HTTP y AJAX hacia el servidor público que contiene un servidor de aplicaciones Apache Tomcat para procesarlas. La aplicación contenida en Apache Tomcat se comunicará con el servidor de base de datos a través del protocolo JDBC (Java Data Base Connection) para obtener registros, insertar, actualizar y eliminar información.

Al realizar el despliegue en el Cloud de Amazon, será posible configurar un auto escalado a futuro de acuerdo al crecimiento en el uso de la aplicación por parte de los usuarios finales.

CAPÍTULO V: RESULTADOS

Después de las tareas de formación, diseño, implementación, y pruebas de software, se ha generado una aplicación web que, principalmente, permite a los usuarios generar consultas dinámicas hacia la base de conocimientos de **Semantic PHI-BASE**. A continuación, se presentan algunas de las pantallas que se han obtenido como resultados de todas las funcionalidades realizadas.

Inicio Aplicación



The screenshot displays the login interface of the Semantic PHI-BASE Web Interface. At the top, there is a header bar with the logo of the Escuela Técnica Superior de Ingenieros Informáticos (Universidad Politécnica de Madrid) on the left and the Semantic PHI-BASE Web Interface logo on the right. Below the header, a blue bar contains the text "Semantic PHI-BASE Web Interface". The main content area features a large image of a green wristwatch on the left. To the right of the watch, there are two input fields: "Nombre de Usuario" with the value "admin@phibaseweb.com" and "Contraseña" with a masked password "*****". Below these fields is a button labeled "Ingresar". In the top right corner of the main area, there is a link for "Ayuda (F2)" with a help icon. At the bottom of the page, a footer bar contains the text "Developed by: Juan Camilo Mesa Polo" and "Copyright © 2016. Todos los derechos reservados."

Imagen 17 - Pantalla Inicio Aplicación

Como se puede ver en la imagen, para el inicio de la aplicación se debe tener un nombre de usuario y una contraseña. Esto le da al sistema un nivel superior de seguridad y permite además identificar las acciones que realiza cada usuario una vez ya se encuentra autenticado.

Menú Principal

El menú principal de la aplicación se desplegará una vez que el usuario haya sido autenticado correctamente. Las opciones que se despliegan en la pantalla dependerán de los servicios a los que el usuario se encuentre autorizado, de acuerdo a su perfil de usuario asociado.

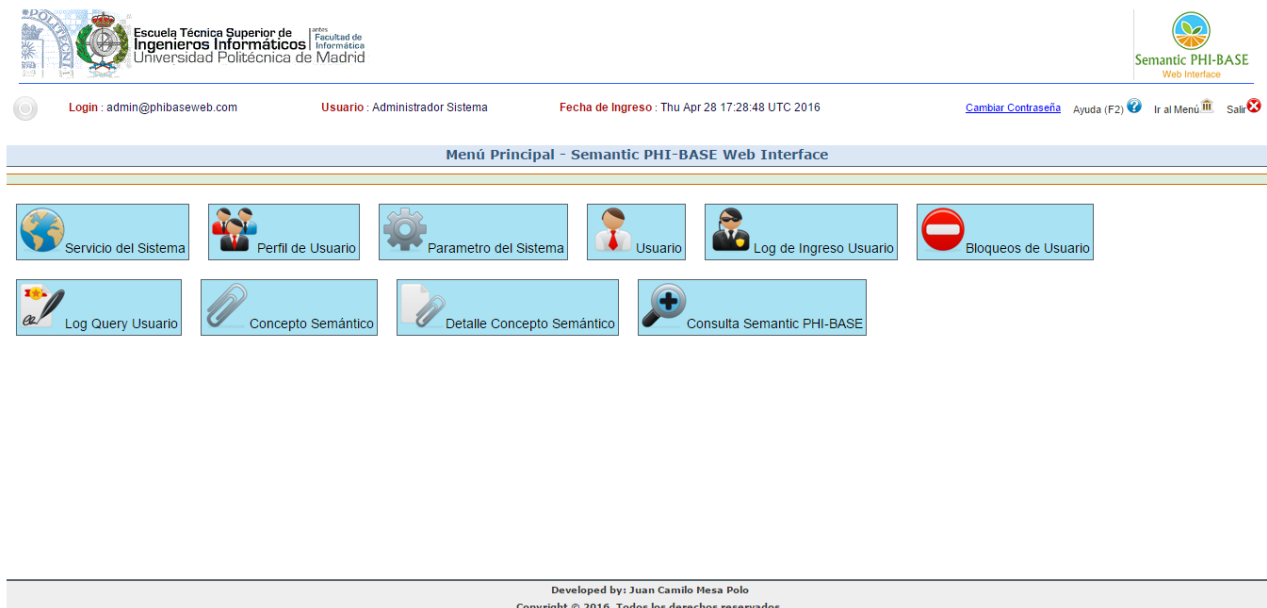


Imagen 18 - Pantalla Menú Principal

En la imagen anterior, es posible visualizar todas las funcionalidades del sistema, ya que se ha realizado la autenticación con un usuario que tiene un perfil de administrador. Las funcionalidades son:

- ✓ Servicio del Sistema: Mantenimiento de los servicios del sistema.
- ✓ Perfil de Usuario: Mantenimiento de los perfiles de usuario. Permite también asociar servicios del sistema a un perfil de usuario.
- ✓ Parámetro del Sistema: Mantenimiento de los parámetros del sistema.
- ✓ Usuario: Mantenimiento de los usuarios del sistema.
- ✓ Log de Ingreso Usuario: Consulta de los registros de log de ingreso de los usuarios al sistema.
- ✓ Bloqueos Usuario: Consulta de los bloqueos de los usuarios por intentos fallidos de login.
- ✓ Log Query Usuario: Consulta de las Queries (Consultas) ejecutadas por todos los usuarios del sistema. Permite también volver a ejecutar la consulta o cargar el filtro de búsqueda seleccionado.
- ✓ Concepto Semántico: Mantenimiento de las estructuras de conceptos semánticos y sus relaciones.
- ✓ Detalle Concepto Semántico: Mantenimiento del detalle de los conceptos semánticos junto con sus relaciones. Estas estructuras serán pobladas automáticamente por el proceso de carga automático de conceptos.
- ✓ Consulta Semantic PHI-BASE: Pantalla principal del sistema. Permite la generación dinámica de consultas hacia **Semantic PHI-BASE**.

Mantenimiento Estructura

El sistema permite realizar el mantenimiento de todas las estructuras (tablas de la base de datos). El mantenimiento hace referencia a adicionar, modificar, eliminar y visualizar registros. A continuación, se visualiza un ejemplo de un mantenimiento de una estructura del sistema.

Mantenimiento Servicio del Sistema

Servicio del Sistema:

Código Servicio del Sistema	Nombre Servicio	Descripción Servicio	URL Servicio	Icono Servicio	Servicio Propio	Desplegable en Menu	Servicio Activo	Cambiar	Eliminar
SE001	Servicio del Sistema	Mantenimiento de los Servicios del Sistema	SE001	icons/normal/48/world_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE002	Perfil de Usuario	Mantenimiento de los perfiles de usuario	SE002	icons/normal/48/group_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE003	Asociar Servicios a Perfil	Permite asociar servicios del sistema a los perfiles de usuario	SE003	icons/normal/48/tool_48.png	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE004	Parametro del Sistema	Mantenimiento de los Parametros del Sistema	SE004	icons/normal/48/gear_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE005	Usuario	Mantenimiento de Usuarios	SE005	icons/normal/48/administrator_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE006	Log de Ingreso Usuario	Consulta del Log de Ingreso de un Usuario	SE006	icons/normal/48/watchman_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE007	Bloqueos de Usuario	Consulta de los bloqueos de un Usuario por Password Invalido	SE007	icons/normal/48/delete_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
SE008	Log Query Usuario	Consulta de las queries ejecutadas por los usuarios	SE008	icons/normal/48/contract_48.png	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Developed by: Juan Camilo Mesa Polo
Copyright © 2016. Todos los derechos reservados.

Imagen 19 - Mantenimiento Servicio del Sistema

Como estándar del sistema, siempre se tendrá el mismo encabezado, en el que es posible visualizar en la parte superior izquierda el logo de la Escuela Técnica Superior de Ingenieros Informáticos y en la parte superior derecha el logo de **Semantic PHI-BASE Web Interface**. Un poco más abajo, se visualizará la información del inicio de sesión y se tendrán los controles para cambiar la contraseña, consultar la ayuda, ir al menú principal y salir del sistema.

Debajo del encabezado, se tendrá siempre el título de la pantalla seguido de lo específico de cada pantalla en sí. Para todos los mantenimientos, se tendrá siempre un buscador que dependerá de cada estructura, con el cual se podrán filtrar los registros que aparecerán en la tabla inferior. Para cada registro de la tabla, normalmente, será posible editar y/o eliminar el registro. Finalmente, en la parte inferior de la pantalla, se tendrá un botón de adicionar un nuevo registro a la estructura.

Pueden existir variaciones y opciones adicionales dependiendo de la estructura que se está manteniendo o consultando. Sin embargo, al usar plantillas para la interfaz gráfica, se mantendrá siempre similitud entre las diferentes pantallas, haciendo que el uso del sistema sea más intuitivo para el usuario final.

Consulta Semantic PHI-BASE

Esta es la funcionalidad más importante del sistema. Está compuesta de cuatro Tabs (Pestañas):

- ✓ *Tab I (Selección Filtro Consulta):* Tab dedicado a seleccionar los criterios por los que se quiere obtener interacciones. Los criterios se seleccionarán utilizando la técnica de arrastrar y soltar, junto con autocompletar. Será posible también seleccionar los operadores lógicos AND y OR entre los diferentes conceptos seleccionados.


Imagen 20 - Pantalla Consulta Semantic PHI-BASE (Tab I)

- ✓ *Tab II (Mis Queries):* En este Tab, será posible consultar las queries previamente ejecutadas para volver a ejecutarlas o para cargar el filtro de búsqueda seleccionado en el Tab I.

Descripción Query	Tiempo de Ejecución	Cargar Parámetros de Query	Ejecutar Query y Ver Resultados
Citation - CITATION_00003	2016-04-28 17:01:10		
Sin Información	2016-04-27 13:45:13		
Citation - CITATION_00003	2016-04-27 13:23:40		
Sin Información	2016-04-21 12:00:31		
Citation - CITATION_00003	2016-04-21 12:00:23		
Citation - CITATION_00003	2016-04-21 11:59:38		

Imagen 21 - Pantalla Consulta Semantic PHI-BASE (Tab II)


- ✓ *Tab III (Visualización Resultados - Tabla):* Este es el Tab donde se mostrará la tabla de resultados obtenida al realizar una consulta. Será posible seleccionar las columnas a desplegar de acuerdo a las preferencias del usuario. También, al dar click sobre alguno de los conceptos obtenidos como resultado, se navegará hacia el Tab IV de visualización gráfica.



Escuela Técnica Superior de Ingenieros Informáticos

Facultad de Informática

Universidad Politécnica de Madrid



Semantic PHI-BASE Web Interface


Login: admin@phibaseweb.com Usuario: Administrador Sistema Fecha de Ingreso: Thu Apr 28 18:37:02 UTC 2016 [Cambiar Contraseña](#) [Ayuda \(F2\)](#) [Ir al Menú](#) [Salir](#)

Consulta Semantic PHI-BASE

[Selección Filtro Consulta](#)
[Mis Queries](#)
[Visualización Resultados - Tabla](#)
[Visualización Resultados - Grafo](#)

Selección de Columnas a Desplegar

Interacción	Host	Interaction Context Mutant Desc
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - Sc Sak2/Sak22 homolog	Host - Triticum - HOST_00951	Reduced virulence
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - Sc Sch9 homolog	Host - Triticum - HOST_00961	Reduced virulence
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - HbF1	Host - Triticum - HOST_00921	Reduced virulence
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - FGS6_07812	Host - Triticum - HOST_01013	Unaffected pathogenicity
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - FGS6_07812	Host - Triticum - HOST_01013	Unaffected pathogenicity
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - FGS6_07812	Host - Triticum - HOST_01013	Unaffected pathogenicity
Interaction - Gibberella zeae (related: Fusarium graminearum) / Wheat [Fusarium ear blight] - FGS6_07812	Host - Triticum - HOST_01013	Unaffected pathogenicity



Developed by: Juan Camilo Mesa Polo
Copyright © 2016. Todos los derechos reservados.

Imagen 22 - Pantalla Consulta Semantic PHI-BASE (Tab III)

- ✓ *Tab IV (Visualización Resultados - Grafo):* Este Tab contiene el módulo de visualización externo, desarrollado para poder ver de una manera gráfica el grafo que representa **Semantic PHI-BASE**.

Manual de Usuario

Para el manual de usuario, se ha decidido desarrollar pantallas Web integradas al sistema, que contengan toda la ayuda e instrucciones necesarias para que un usuario final pueda usar el sistema correctamente. De esta manera, no hará falta que el usuario tenga un documento a parte para consultar sobre el uso del sistema y además se garantizará que siempre esté disponible para cuando sea necesario.

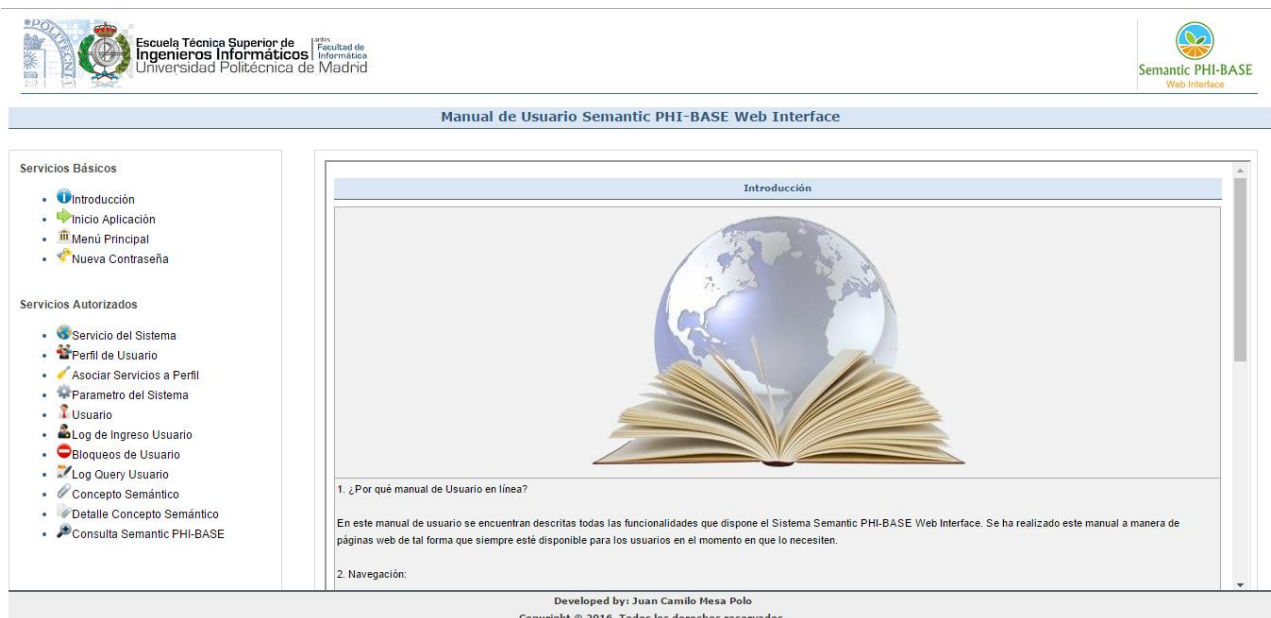


Imagen 23 - Pantalla Manual de Usuario

Se puede ver en la imagen, que existe una ayuda básica, llamada “Servicios Básicos”, los cuales son una introducción a la ayuda, información sobre el inicio de la aplicación, el menú principal y la actualización de la contraseña. La otra parte de la ayuda, se compone de los “Servicios Autorizados”, por lo que se desplegará la ayuda únicamente para los servicios a los cuales el usuario tiene permiso. De esta manera, se hace más eficiente el uso de la ayuda y los usuarios no verán ayuda relacionada a funcionalidades que no pueden ejecutar.

CAPÍTULO VI: CONCLUSIONES

- ✓ El diseño propuesto para **Semantic PHI-BASE Web Interface**, se ha realizado de tal manera que sea posible su escalabilidad y creación de nuevas funcionalidades, sin tener que afectar las funcionalidades ya existentes, por lo que los tiempos de desarrollo y adaptaciones serán menores a los tiempos promedio.
- ✓ La construcción de **Semantic PHI-BASE Web Interface**, permitirá a los usuarios finales realizar consultas complejas, hacia la base de conocimientos de **Semantic PHI-BASE**, de una manera sencilla, lo que les ayudará a extraer información importante y generar conocimiento gracias a los datos obtenidos con los resultados de las consultas.
- ✓ Para un uso más óptimo del sistema construido, es necesario diseñar procesos de depuración y de normalización de los datos, de manera que la generación de consultas y resultados no tenga datos duplicados y/o diferentes, pero con el mismo significado. Para lograr esto, se debe realizar un análisis completo de los datos que identificar patrones y a partir de ellos construir los procesos de depuración con los que se creará una nueva **Semantic PHI-BASE** normalizada.
- ✓ El uso de las tecnologías de Web Semántica (RDF, SPARQL) es muy importante en ámbitos de investigación, ya que la integración entre diferentes ambientes y plataformas se puede lograr naturalmente y de esta manera se tendrá una base de conocimientos consolidada, de la cual será posible extraer conocimiento con herramientas como **Semantic PHI-BASE Web Interface**.
- ✓ La integración del sistema con un módulo de visualización externo, con el cual se realizan comunicaciones estándar HTTP hace posible que se puedan hacer nuevas implementaciones y mejoras en dicho módulo, sin tener que tocar el código utilizado para lograr la integración. Sin embargo, se crea una dependencia del buen funcionamiento de este módulo para lograr una visualización gráfica en el sistema.

CAPÍTULO VII: LÍNEAS FUTURAS

Al realizar la construcción de **Semantic PHI-BASE Web Interface**, se ha posibilitado al usuario realizar consultas sobre la información contenida en la base de conocimientos creada a partir de la base de datos XML inicial. Sin embargo, es posible aún, extender mucho más los criterios de búsqueda e incluir nuevas opciones que permitan a los usuarios generar nuevo conocimiento. Los siguientes son los módulos o funcionalidades que se podrán incluir en el sistema para dar al usuario una funcionalidad más extendida y más posibilidades de trabajo con los datos de **Semantic PHI-BASE**.

Generación Consultas Federadas

Las consultas federadas, son consultas que se realizan accediendo no sólo a los datos disponibles en la base de conocimientos propia **Semantic PHI-BASE**, sino que sea posible obtener información y realizar cruces con otras bases de conocimientos ya existentes y disponibles en la Web. Por ejemplo, sería posible preguntar por información de las bases de conocimiento de *Edamontology*, *Linked Life Data* o *Uniprot* que se encuentra directamente relacionada con los datos de **Semantic PHI-BASE**. Esto permitirá un rango de búsqueda mucho más amplio, lo que le dará a los usuarios la posibilidad de generar aún más conocimiento y de extraer información valiosa.

Inserción Controlada de Datos

Con el desarrollo que se ha hecho hasta el momento, se han dejado las bases necesarias para poder un módulo de inserción controlada de datos. Esto permitirá que se incluyan nuevas relaciones en el grafo de **Semantic PHI-BASE**, complementando así, la información ya existente.

Edición de Datos Actuales

Se trata de un módulo adicional al módulo de inserción, ya que este módulo lo que permitirá específicamente, es editar los valores de los datos que se encuentran actualmente en la base de conocimientos de **Semantic PHI-BASE**.

Publicación en Revista “Journal of Web Semantics”

“Journal of Web Semantics” es una revista que está basada en la investigación y aplicación de varias áreas que contribuyen al desarrollo de sistemas inteligentes y basados en conocimiento. Actualmente, esta revista está invitando a investigadores a que presenten ideas (Papers) sobre problema de visualización de datos en Ontologías y Linked Data. Se pretende, entonces, presentar todo el trabajo realizado con **Semantic PHI-BASE Web Interface**, junto con el módulo encargado de la visualización gráfica de los datos.

REFERENCIAS

- [1] Frontiers in Plant Science. Technical Advances in Plant Science. “Publishing FAIR Data: An Exemplar Methodology Utilizing PHI-Base”. Authors: Alejandro Rodríguez-Iglesias, Alejandro Rodríguez-González, Alistair G. Irvine, Ane Sesma, Martin Urban, Kim E. Hammond-Kosack and Mark D. Wilkinson [Online]
<<http://journal.frontiersin.org/article/10.3389/fpls.2016.00641/full>> [Fecha de Consulta: 24-05-2016]
- [2] Wikipedia, The Free Encyclopedia. Virtuoso Universal Server [Online]
<https://en.wikipedia.org/wiki/Virtuoso_Universal_Server> [Fecha de Consulta: 16-02-2016]
- [3] Stackoverflow, Set up RDF Ontology with Virtuoso [Online]
<<http://stackoverflow.com/questions/5531224/setup-rdf-ontology-with-virtuoso>> [Fecha de Consulta: 16-02-2016]
- [4] Apache Jena, SPARQL Tutorial [Online] <<https://jena.apache.org/tutorials/sparql.html>> [Fecha de Consulta: 20-02-2016]
- [5] ICEfaces Software, Tutorials [Online] <<http://www.icesoft.org/wiki/display/ICE/Tutorials>> [Fecha de Consulta: 03-03-2016]
- [6] IBM Developer Works, Search RDF data with SPARQL [Online]
<<http://www.ibm.com/developerworks/xml/library/j-sparql/>> [Fecha de Consulta: 03-03-2016]
- [7] Adictos al Trabajo, JCapcha – Generación de Captchas en Java [Online]
<<http://www.adictosaltrabajo.com/tutoriales/jcaptcha/>> [Fecha de Consulta: 05-03-2016]
- [8] Docs Oracle, Class Timer [Online]
<<https://docs.oracle.com/javase/7/docs/api/java/util/Timer.html>> [Fecha de Consulta: 06-03-2016]
- [9] Stackoverflow, How do I Schedule a task to run at periodic intervals? [Online]
<<http://stackoverflow.com/questions/4544197/how-do-i-schedule-a-task-to-run-at-periodic-intervals>> [Fecha de Consulta: 06-03-2016]
- [10] W3C, SPARQL Query Language for RDF [Online] <<https://www.w3.org/TR/rdf-sparql-query/>> [Fecha de Consulta: 08-03-2016]
- [11] Korrekt ORG, The State of the UNION: Why no SPARQL condition should be Second Class [Online]
<http://korrekt.org/page/Note:The_State_of_the_UNION:_Why_no_SPARQL_Condition_Should_be_Second_Class> [Fecha de Consulta: 08-03-2016]
- [12] ICEfaces Software, ICEfaces component showcase [Online] <<http://component-showcase.icesoft.org/component-showcase/showcase.iface>> [Fecha de Consulta: 12-03-2016]
- [13] ICEfaces Software Showcase, Autocomplete Entry Bean [Online] <<http://icefaces-showcase.icesoft.org/showcase.jsf?grp=aceMenu&exp=autoCompleteEntryBean>> [Fecha de Consulta: 12-03-2016]
- [14] ICEfaces Software Showcase, Datatable Dynamic Column [Online] <<http://icefaces-showcase.icesoft.org/showcase.jsf?grp=ace:dataTable&exp=Dynamic%20Column>> [Fecha de Consulta: 12-03-2016]

- [15] Tecmint, Linux How To Guide, 20 Command Line tools to monitor Linux Performance [Online] <<http://www.tecmint.com/command-line-tools-to-monitor-linux-performance/>> [Fecha de Consulta: 02-04-2016]
- [16] Wikipedia, The Free Encyclopedia. Host Pathogen Interaction [Online] <https://en.wikipedia.org/wiki/Host%E2%80%93pathogen_interaction> [Fecha de Consulta: 16-04-2016]
- [17] Wikipedia, The Free Encyclopedia. SPARQL [Online] <<https://es.wikipedia.org/wiki/SPARQL>> [Fecha de Consulta: 16-04-2016]
- [18] Apache Jena. Apache Jena Overview [Online] <<https://jena.apache.org/>> [Fecha de Consulta: 19-04-2016]
- [19] Google Code. Semantic Science [Online] <<https://code.google.com/archive/p/semanticsscience/>> [Fecha de Consulta: 20-04-2016]
- [20] Edamontology. Edamontology Definition [Online] <<http://edamontology.org/page#Definition>> [Fecha de Consulta: 20-04-2016]
- [21] Linked Life Data. What is Linked Life Data? [Online] <<http://edamontology.org/page#Definition>> [Fecha de Consulta: 20-04-2016]
- [22] Wikilibros, Libros libres para un mundo libre. Dirección de Proyectos / Planificación de los Riesgos del Proyecto [Online] <https://es.wikibooks.org/wiki/Direcci%C3%B3n_de_Proyectos/Planificaci%C3%B3n_de_los_riesgos_del_proyecto> [Fecha de Consulta: 20-04-2016]
- [23] Java. ¿Qué es Java? [Online] <https://www.java.com/es/about/whatis_java.jsp> [Fecha de Consulta: 23-04-2016]
- [24] MySQL. MySQL Editions [Online] <<https://www.mysql.com/products/>> [Fecha de Consulta: 23-04-2016]
- [25] Icesoft Technologies Inc. ICEfaces Overview [Online] <<http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>> [Fecha de Consulta: 23-04-2016]
- [26] Amazon Web Services AWS. ¿Qué es la Cloud Computing? [Online] <https://aws.amazon.com/es/what-is-cloud-computing/?nc1=f_cc> [Fecha de Consulta: 26-04-2016]
- [27] Eclipse. Eclipse Kepler SR2 Packages [Online] <<http://www.eclipse.org/downloads/packages/release/Kepler/SR2>> [Fecha de Consulta: 10-05-2016]
- [28] ICEfaces Technologies INC. ICEfaces downloads [Online] <<http://www.icesoft.org/java/downloads/detail-new.jsf>> [Fecha de Consulta: 10-05-2016]
- [29] Journal of Web Semantics, cfp: Special Issue on Visualization and Interaction for Ontologies and Linked Data [Online] <<http://journalofwebsemantics.blogspot.fr/2016/05/cfp-special-issue-on-visualization-and.html>> [Fecha de Consulta: 10-05-2016]

ANEXO A – DISEÑO DE BASE DE DATOS

Estándar Nombre de Tablas y Campos

Estándar nombre de las tablas (Longitud de 8 caracteres):

- ✓ Primeros dos caracteres: Módulo del sistema. Se han definido los siguientes módulos:
 - Configuración Sistema: CS
 - Usuarios: US
 - Semántica: SE
- ✓ Tercer carácter:
 - P: Parámetros
 - D: Detalle
 - M: Maestro
- ✓ Siguietes cinco caracteres: Nombre significativo para la tabla.

Estándar nombre de Campos (Longitud de 9 caracteres):

- ✓ Primeros 3 caracteres: Prefijo único de la tabla.
- ✓ Siguietes 4 caracteres: Libre para nombre de campo.
- ✓ Siguiente carácter (carácter 8): Indicador de tipo de campo.
 - A: Alfanumerico
 - N: Numérico
 - M: Monetario
 - V: Multivalor
 - S: Booleano
 - F: Fecha
 - T: Hora
 - P: Porcentaje
- ✓ Último carácter (carácter 9): Indicador de campo o llave.
 - F: Campo
 - K: Llave

Descripción Tablas Base de Datos

A continuación, se presenta la descripción detallada de cada una de las tablas de la base de datos:

Tabla 12 - Descripción Tabla CSPPERFU

Nombre Tabla	CSPPERFU	Prefijo Tabla	PFU
Descripción	PERFIL USUARIO		
Uso	Tabla de configuración para crear los diferentes perfiles de usuario que habrá en el sistema.		
Campos	<ul style="list-style-type: none">▪ Código Perfil Usuario (K) (PFUCPFUAK)▪ Descripción Perfil (PFUDPFUAF)▪ Perfil Activo (PFUIACTSF)		

Tabla 13 - Descripción Tabla CSPSERSI

Nombre Tabla	CSPSERSI	Prefijo Tabla	SSI
Descripción	SERVICIO SISTEMA		
Uso	Tabla de configuración para tener los servicios que ofrecerá el sistema. Esto permitirá que se vayan adicionando dinámicamente nuevos servicios conforme la aplicación vaya evolucionando.		
Campos	<ul style="list-style-type: none">▪ Código Servicio Sistema (K) (SSICSSIAK)▪ Nombre Servicio Sistema (SSINSSIAF)▪ Descripción Servicio Sistema (SSIDSSIAF)▪ URL Servicio (SSIDURLAF)▪ Icono Servicio (SSIICONAF)▪ Indicador servicio propio (SSIIPROSF)▪ Desplegable en el menú (SSIIDESSF)▪ Servicio Activo (SSIIACTSF)		

Tabla 14 - Descripción Tabla CSPSERPF

Nombre Tabla	CSPSERPF	Prefijo Tabla	SVP
Descripción	SERVICIO DE PERFIL DE USUARIO		
Uso	Esta tabla permitirá la asociación de servicios del sistema a los perfiles. De esta manera se podrá saber qué servicios podrán ser ejecutados por los usuarios que tengan un determinado perfil.		
Campos	<ul style="list-style-type: none">▪ Código Perfil Usuario (K) (PFUCPFUAK)▪ Código Servicio Sistema (K) (SSICSSIAK)▪ Servicio Activo en Perfil (SVPIACTSF)		

Tabla 15 - Descripción Tabla CSPPARSI

Nombre Tabla	CSPPARSI	Prefijo Tabla	PSI
Descripción	PARAMETRO SISTEMA		
Uso	Tabla base para almacenar los parámetros de configuración del sistema. Se almacenarán parámetros como por ejemplo el Endpoint de Virtuoso, el número mínimo de caracteres en una contraseña, etc.		
Campos	<ul style="list-style-type: none"> ▪ Código Parámetro Sistema (K) (PSICPARAK) ▪ Descripción Parámetro (PSIDPARAF) ▪ Tipo Parámetro (PSITPARVF) <ul style="list-style-type: none"> ○ '1' Alfanumérico ○ '2' Numérico ▪ Valor Parámetro (PSIVALPAF) 		

Tabla 16 - Descripción Tabla USMUSUAR

Nombre Tabla	USMUSUAR	Prefijo Tabla	USU
Descripción	USUARIO SISTEMA		
Uso	Tabla maestra para almacenar los usuarios registrados en el sistema. Únicamente los usuarios que se encuentren registrados en esta tabla son los que podrán acceder al sistema y ejecutar los servicios que tengan asociados de acuerdo a su perfil de usuario.		
Campos	<ul style="list-style-type: none"> ▪ Email Usuario (K) (USUEMAIAK) ▪ Nombre Usuario (USUNUSUAF) ▪ Contraseña Usuario (USUPASSAF) ▪ Código Perfil Usuario (PFUCPFUAK) ▪ Estado Usuario (USUESTAVF) <ul style="list-style-type: none"> ○ '1' Activo ○ '2' Inactivo ○ '3' Inactivo Intentos Fallidos ▪ Intentos Fallidos Usuario (USUILOGNF) ▪ Timestamp Bloqueo Usuario (USUTBLONF) ▪ Contraseña caducada (USUPASCSF) ▪ Fecha último cambio contraseña (USUFULCFF) 		

Tabla 17 - Descripción Tabla USDBLOQU

Nombre Tabla	USDBLOQU	Prefijo Tabla	BLU
Descripción	HISTORICO BLOQUEOS USUARIO		
Uso	En esta tabla se almacenarán los bloqueos que tenga un usuario en su cuenta cuando excede el número máximo de intentos de login de una manera no satisfactoria.		

Campos	<ul style="list-style-type: none"> Email Usuario (K) (USUEMAIAK) Timestamp Bloqueo Usuario (K) (USUTBLONF) Intentos Fallidos Usuario (BLUILOGNF) Timestamp Des-Bloqueo Usuario (BLUTDESNF) Usuario que realiza Desbloqueo (USUEMADAF) Desbloqueo Automático (BLUDAUTSF)
---------------	---

Tabla 18 - Descripción Tabla USDLOGQU

Nombre Tabla	USDLOGQU	Prefijo Tabla	LQU
Descripción	LOG QUERY USUARIO		
Uso	Tabla para almacenar todas las queries que hayan sido ejecutadas por cada uno de los usuarios del sistema. Se almacenarán todos los datos necesarios para poder ejecutar nuevamente la consulta o para cargar los criterios de búsqueda seleccionados.		
Campos	<ul style="list-style-type: none"> Email Usuario (K) (USUEMAIAK) Fecha Query (K) (LQUFECHFK) Secuencia Query (K) (LQUNSECNK) Hora Query (LQUHORATF) Texto Query (LQUTEXTAF) Resultado Query (LQURESPAF) Parámetros Query (LQUPARQAF) Descripción Query (LQUDESCAF) 		

Tabla 19 - Descripción Tabla USDLOGIN

Nombre Tabla	USDLOGIN	Prefijo Tabla	LIU
Descripción	LOG INGRESO USUARIO		
Uso	En esta tabla se almacenarán todos los intentos de login que haga un usuario dentro del sistema. Se almacenará fecha y hora del login junto con la dirección IP de la máquina desde donde se está realizando el ingreso.		
Campos	<ul style="list-style-type: none"> Email Usuario (K) (USUEMAIAK) Fecha Ingreso (K) (LIUFECHFK) Secuencia Ingreso (K) (LIUNSECNK) Hora ingreso (LIUHORATF) Dirección Ingreso (LIUDIRIAF) Ingreso Exitoso (LIUIEXISF) 		

Tabla 20 - Descripción Tabla SEPCONCE

Nombre Tabla	SEPCONCE	Prefijo Tabla	COS
---------------------	----------	----------------------	-----

Descripción	CONCEPTO SEMÁNTICO
Uso	Tabla de configuración para almacenar los conceptos semánticos de la ontología de Semantic PHI-BASE . Servirá para poder representar y almacenar en la base de datos los conceptos que se han definido con RDF . Un ejemplo de un concepto semántico es “Host”, “Pathogen” o “Gene”.
Campos	<ul style="list-style-type: none"> ▪ Código Concepto (K) (COSCCOSAK) ▪ Descripción Concepto (COSDCOSAF) ▪ Tipo RDF (rdf:type) (COSRDFTAF) ▪ Número de relaciones (Atributos) (COSNRELNF) ▪ Código concepto Padre (COSCCOPAK) ▪ Documentación Concepto (COSDOCUAF)

Tabla 21 - Descripción Tabla SEPRELCO

Nombre Tabla	SEPRELCO	Prefijo Tabla	RCO
Descripción	RELACIÓN CONCEPTO SEMÁNTICO		
Uso	Tabla de configuración para almacenar las relaciones que pueden tener los diferentes conceptos semánticos definidos previamente. Un ejemplo de una relación puede ser “has participant” entre “Interaction” y “Host”. Adicionalmente, también se contempla en esta tabla el rango de la relación para en un futuro poder realizar una inserción controlada de nuevas triplas en la base de conocimiento.		
Campos	<ul style="list-style-type: none"> ▪ Código Concepto (K) (COSCCOSAK) ▪ Código Relación Concepto (K) (RCOCRCOAK) ▪ Descripción Relación (RCODRCOAF) ▪ RDF Relación (RCORDFRAF) ▪ Rango Relación (RCORANGAF) ▪ Documentación Relación (RCODOCUAF) 		

Tabla 22 - Descripción Tabla SEDCONCE

Nombre Tabla	SEDCONCE	Prefijo Tabla	DCO
Descripción	DETALLE CONCEPTO SEMÁNTICO		
Uso	Contiene información de un concepto propio de la base de conocimiento. Es decir, representa una instancia concreta del modelo. Servirá para poder mostrar los filtros de búsqueda en la pantalla principal de consulta a Semantic PHI-BASE .		
Campos	<ul style="list-style-type: none"> ▪ Secuencia Detalle Concepto (K) (DCONSECNK) ▪ IRI Detalle Concepto (DCOIRICAF) ▪ Código Concepto (COSCCOSAK) 		

Tabla 23 - Descripción Tabla SEDRELCO

Nombre Tabla	SEDRELCO	Prefijo Tabla	DRC
Descripción	DETALLE RELACIÓN CONCEPTO SEMÁNTICO		
Uso	Contiene el valor de una relación entre instancias de conceptos semánticos. Por ejemplo, contendrá el valor “ <i>PHI:164</i> ” de un concepto tipo “ <i>Phi base ID</i> ” y la relación “ <i>has value</i> ”.		
Campos	<ul style="list-style-type: none"> ▪ Secuencia Detalle Concepto (K) (DCONSECNK) ▪ Secuencia Detalle Relación Concepto (K) (DRCNSECNK) ▪ Valor Relación (DRCVALRAF) ▪ Código Concepto (COSCCOSAK) ▪ Código Relación Concepto (RCOCCRCOAK) 		

ANEXO B – ESPECIFICACIÓN DETALLADA DE CASOS DE USO

A continuación, se presenta la especificación detallada de cada uno de los casos de uso del sistema:

Tabla 24 - EDCU CU-001

Caso de Uso	CU-001	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	CU-014
Nombre caso de uso	Inicio de sesión en el sistema		
Descripción caso de uso	Esta funcionalidad será la pantalla inicial de la aplicación en la que se solicitará el Id de usuario (su email) y su contraseña para iniciar sesión. Dependiendo de si el login es satisfactorio o no se irá a la pantalla del menú principal de la aplicación o a la misma pantalla desplegando mensaje de error.		
Pre-condición	Ninguna		
Post-condición	El usuario ha ingresado satisfactoriamente en el sistema y se ha cargado un menú con las opciones disponibles para él, según su perfil de usuario asociado.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. Ingresa por el navegador web de su preferencia a la dirección de inicio de la aplicación Semantic PHI-BASE Web Interface.		2. El sistema despliega la pantalla de inicio en donde se solicita un nombre de usuario (email) y una contraseña asociada.	
3. El usuario ingresa su nombre de usuario y su contraseña y pulsa sobre el botón de ingresar.		4. El sistema valida la autenticidad del usuario y despliega la pantalla de menú principal con los servicios del sistema asociados al perfil del usuario ingresado. El sistema registra el Log de ingreso exitoso de usuario en la base de datos.	
5. El usuario da click sobre alguna de las funcionalidades disponibles según su perfil asociado.		6. El sistema navega hacia la pantalla de la funcionalidad seleccionada por el usuario	
Flujos Alternos			
4a. El sistema valida la autenticidad del usuario y detecta que el nombre de usuario o la contraseña son inválidos. El sistema despliega nuevamente la pantalla de inicio de sesión con un mensaje de error indicando que el nombre de usuario o la contraseña no es válido. En caso de que el nombre de usuario ingresado exista en la base de datos, se debe registrar en el log un ingreso fallido por parte del usuario y al mismo tiempo incrementar el número de intentos fallidos del usuario. En caso			

de que el número de intentos fallidos del usuario alcance el definido en el parámetro del sistema, se bloqueará al usuario por el tiempo en minutos indicado en otro parámetro del sistema.

4b. En caso de que sea válido el nombre de usuario y la contraseña, pero el usuario tenga la contraseña caducada, el sistema navegará hacia la funcionalidad de actualización de contraseña iniciando así el caso de uso CU-014.

Tabla 25 - EDCU CU-002

Caso de Uso	CU-002	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Mantenimiento Servicio Sistema		
Descripción caso de uso	Se trata de mantener la tabla que contiene los servicios del sistema. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de la estructura de Servicios del Sistema.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de mantenimiento de servicios del sistema.		2. El sistema despliega la pantalla principal de la funcionalidad de servicios del sistema. Se carga una lista en pantalla con los servicios del sistema existentes y se provee funcionalidad para filtrar por un servicio del sistema específico.	
3. El usuario da click sobre el botón de adicionar un nuevo servicio del sistema.		4. El sistema despliega un pop-up solicitando los datos para ingresar un nuevo servicio del sistema.	
5. El usuario ingresa los datos correspondientes al nuevo servicio del sistema y da click sobre el botón adicionar.		6. El sistema valida y registra el nuevo servicio del sistema. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de servicios del sistema (paso 2).	
7. El usuario da click sobre el botón de actualizar un servicio del sistema específico.		8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a un servicio del sistema con sus valores actuales.	
9. El usuario actualiza los valores que desea cambiar sobre el servicio del sistema actual y posteriormente da click sobre el botón actualizar.		10. El sistema valida los nuevos valores ingresados sobre el servicio del sistema y actualiza los datos en la base de datos. Si todo es correcto, el sistema	

	despliega nuevamente la pantalla principal de la funcionalidad de servicios del sistema (paso 2).
11. El usuario da click sobre el botón de eliminar un servicio del sistema.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de la funcionalidad de servicios del sistema (paso 2).
Flujos Alternos	
6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir. 13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de la funcionalidad de servicios del sistema (paso 2).	

Tabla 26 - EDCU CU-003

Caso de Uso	CU-003	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Mantenimiento Perfil de Usuario		
Descripción caso de uso	Se trata de mantener la tabla que contiene los perfiles de usuario. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de la estructura de Perfiles de Usuario.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de mantenimiento de perfiles de usuario.		2. El sistema despliega la pantalla principal de la funcionalidad de perfiles de usuario. Se carga una lista en pantalla con los perfiles de usuario existentes y se provee funcionalidad para filtrar por un perfil de usuario específico.	
3. El usuario da click sobre el botón de adicionar un nuevo perfil de usuario.		4. El sistema despliega un pop-up solicitando los datos para ingresar un nuevo perfil de usuario.	

5. El usuario ingresa los datos correspondientes al nuevo perfil de usuario y da click sobre el botón adicionar.	6. El sistema valida y registra el nuevo perfil de usuario. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de perfiles de usuario (paso 2).
7. El usuario da click sobre el botón de actualizar un perfil de usuario específico.	8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a un perfil de usuario con sus valores actuales.
9. El usuario actualiza los valores que desea cambiar sobre el perfil de usuario actual y posteriormente da click sobre el botón actualizar.	10. El sistema valida los nuevos valores ingresados sobre el perfil de usuario y actualiza los datos en la base de datos. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de perfiles de usuario (paso 2).
11. El usuario da click sobre el botón de eliminar un perfil de usuario.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de la funcionalidad de perfiles de usuario (paso 2).
Flujos Alternos	
6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir. 13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de la funcionalidad de perfiles de usuario (paso 2).	

Tabla 27 - EDCU CU-004

Caso de Uso	CU-004	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Mantenimiento Parámetro Sistema		
Descripción caso de uso	Se trata de mantener la tabla que contiene parámetros del sistema. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de la estructura de Parámetros del Sistema.		
Flujo Normal de Eventos			

Usuario	Sistema
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de mantenimiento de parámetros del sistema.	2. El sistema despliega la pantalla principal de la funcionalidad de parámetros del sistema. Se carga una lista en pantalla con los parámetros del sistema existentes y se provee funcionalidad para filtrar por un parámetro del sistema específico.
3. El usuario da click sobre el botón de adicionar un nuevo parámetro del sistema.	4. El sistema despliega un pop-up solicitando los datos para ingresar un nuevo parámetro del sistema.
5. El usuario ingresa los datos correspondientes al nuevo parámetro del sistema y da click sobre el botón adicionar.	6. El sistema valida y registra el nuevo parámetro del sistema. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de parámetros del sistema (paso 2).
7. El usuario da click sobre el botón de actualizar un parámetro del sistema específico.	8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a un parámetro del sistema con sus valores actuales.
9. El usuario actualiza los valores que desea cambiar sobre el parámetro del sistema actual y posteriormente da click sobre el botón actualizar.	10. El sistema valida los nuevos valores ingresados sobre el parámetro del sistema y actualiza los datos en la base de datos. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de parámetros del sistema (paso 2).
11. El usuario da click sobre el botón de eliminar un parámetro del sistema.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de la funcionalidad de parámetros del sistema (paso 2).
Flujos Alternos	
6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir. 13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de la funcionalidad de parámetros del sistema (paso 2).	

Tabla 28 - EDCU CU-005

Caso de Uso	CU-005	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Mantenimiento Concepto Semántico – Relación Concepto		

Descripción caso de uso	Se trata de mantener las tablas que contienen los conceptos semánticos y sus relaciones. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En inglés: CRUD.
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de las estructuras de Conceptos Semánticos y su tabla hija de Relaciones Concepto.
Flujo Normal de Eventos	
Usuario	Sistema
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de mantenimiento de Conceptos Semánticos y sus relaciones.	2. El sistema despliega la pantalla principal de la funcionalidad de Conceptos Semánticos y sus relaciones. Se carga una lista en pantalla con los Conceptos Semánticos existentes y se provee funcionalidad para filtrar por un Concepto específico.
3. El usuario da click sobre el botón de adicionar un nuevo Concepto Semántico.	4. El sistema despliega un pop-up solicitando los datos para ingresar un nuevo concepto semántico.
5. El usuario ingresa los datos correspondientes al nuevo Concepto Semántico y da click sobre el botón adicionar. Al dar el atributo de número de relaciones se solicitarán inmediatamente los datos de las relaciones para ser registrados junto con el concepto semántico.	6. El sistema valida y registra el nuevo concepto semántico junto con sus relaciones asociadas. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de conceptos semánticos y sus relaciones (paso 2).
7. El usuario da click sobre el botón de actualizar un concepto semántico específico.	8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a un concepto semántico con sus valores actuales y sus relaciones asociadas.
9. El usuario actualiza los valores que desea cambiar sobre el concepto semántico actual y posteriormente da click sobre el botón actualizar.	10. El sistema valida los nuevos valores ingresados sobre el concepto y sus relaciones y procede a actualizar los datos en la base de datos. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad conceptos semánticos y sus relaciones (paso 2).
11. El usuario da click sobre el botón de eliminar un concepto semántico.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de conceptos semánticos y sus relaciones (paso 2).

15. El usuario da click sobre el botón de visualizar un concepto semántico.	16. El sistema despliega un pop-up con todos los datos del concepto semántico y todas sus relaciones asociadas.
Flujos Alternos	
6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir. 13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de conceptos semánticos y sus relaciones (paso 2).	

Tabla 29 - EDCU CU-006

Caso de Uso	CU-006	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Mantenimiento Detalle Concepto Semántico – Detalle Relación Concepto		
Descripción caso de uso	Se trata de mantener las tablas que contienen los detalles de conceptos semánticos junto con sus relaciones. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de las estructuras de Detalles de Conceptos Semánticos y su tabla hija de Detalle de Relaciones Concepto.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de mantenimiento de Detalle de Conceptos Semánticos y sus relaciones.		2. El sistema despliega la pantalla principal de la funcionalidad de Detalle Conceptos Semánticos y sus relaciones. Se carga una lista en pantalla con los Conceptos Semánticos existentes y se provee funcionalidad para filtrar por un Concepto específico o por un nombre de detalle parcial o total.	
3. El usuario da click sobre el botón de adicionar un nuevo Detalle de Concepto Semántico.		4. El sistema despliega un pop-up solicitando los datos para ingresar un nuevo detalle de concepto semántico y sus relaciones.	
5. El usuario ingresa los datos correspondientes al nuevo Detalle de Concepto Semántico junto con todas las		6. El sistema valida y registra el nuevo concepto semántico junto con sus relaciones asociadas. Si todo es correcto, el sistema despliega nuevamente la	

relaciones que se deben ingresar de acuerdo a la parametrización realizada y da click sobre el botón adicionar.	pantalla principal de la funcionalidad de detalle de conceptos semánticos y sus relaciones (paso 2).
7. El usuario da click sobre el botón de actualizar un detalle de concepto semántico específico junto con sus relaciones.	8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a un detalle de concepto semántico con sus valores actuales y sus relaciones asociadas.
9. El usuario actualiza los valores que desea cambiar sobre el detalle de concepto semántico actual y posteriormente da click sobre el botón actualizar.	10. El sistema valida los nuevos valores ingresados sobre el concepto y sus relaciones y procede a actualizar los datos en la base de datos. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de detalle de conceptos semánticos y sus relaciones (paso 2).
11. El usuario da click sobre el botón de eliminar un detalle de concepto semántico.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de detalle de conceptos semánticos y sus relaciones (paso 2).
15. El usuario da click sobre el botón de visualizar un detalle de concepto semántico.	16. El sistema despliega un pop-up con todos los datos del detalle de concepto semántico y todas sus relaciones asociadas.
Flujos Alternos	
<p>6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir.</p> <p>13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de detalle de conceptos semánticos y sus relaciones (paso 2).</p>	

Tabla 30 - EDCU CU-007

Caso de Uso	CU-007	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Mantenimiento Usuario Sistema		
Descripción caso de uso	Se trata de mantener la tabla que contiene los usuarios del sistema. Funcionalidades de Adicionar, Leer, Actualizar, Borrar. En ingles: CRUD. Adicionalmente a las funcionalidades básicas, se permitirá realizar el desbloqueo manual de un usuario y la regeneración de su contraseña.		

Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de la estructura de Usuarios del Sistema.
Flujo Normal de Eventos	
Usuario	Sistema
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de mantenimiento de parámetros del sistema.	2. El sistema despliega la pantalla principal de la funcionalidad de usuarios del sistema. Se carga una lista en pantalla con los usuarios del sistema existentes y se provee funcionalidad para filtrar por un usuario del sistema específico.
3. El usuario da click sobre el botón de adicionar un nuevo usuario del sistema.	4. El sistema despliega un pop-up solicitando los datos para ingresar un nuevo usuario del sistema.
5. El usuario ingresa los datos correspondientes al nuevo usuario del sistema y da click sobre el botón adicionar.	6. El sistema valida y registra el nuevo usuario del sistema. Si todo es correcto, el sistema envía un correo electrónico al usuario indicándole la bienvenida al sistema, su nombre de usuario y una contraseña Random asignada que deberá ser cambiada en el primer inicio de sesión. A continuación, se despliega nuevamente la pantalla principal de la funcionalidad de usuarios del sistema (paso 2).
7. El usuario da click sobre el botón de actualizar un usuario del sistema específico.	8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a un usuario del sistema con sus valores actuales.
9. El usuario actualiza los valores que desea cambiar sobre el usuario del sistema actual y posteriormente da click sobre el botón actualizar.	10. El sistema valida los nuevos valores ingresados sobre el usuario del sistema y actualiza los datos en la base de datos. Si el usuario se encontraba inactivo por intentos fallidos y se cambia su estado, entonces se registra un histórico de bloqueos de usuario indicando que el desbloqueo ha sido manual y el usuario que ha realizado el desbloqueo. Posteriormente, si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de usuarios del sistema (paso 2).
11. El usuario da click sobre el botón de eliminar un usuario del sistema.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de la funcionalidad de usuarios del sistema (paso 2).

Flujos Alternos
<p>6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir.</p> <p>13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de la funcionalidad de usuarios del sistema (paso 2).</p>

Tabla 31 - EDCU CU-008

Caso de Uso	CU-008	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Consulta Histórico Bloqueos Usuario		
Descripción caso de uso	Se trata de proveer la funcionalidad de consulta de la tabla de histórico de bloqueos de los usuarios. Se podrá consultar por rango de fechas y por un usuario en específico.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han visualizado los registros de la estructura de Histórico de Bloqueos de Usuario.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de consulta de histórico de bloqueos de usuario.		2. El sistema despliega la pantalla en la que se despliega una lista con los resultados de los últimos históricos de bloqueos de usuario realizados. Se permite filtrar por nombre de usuario y por un rango de fechas específico.	
Flujos Alternos			
No se cuenta con flujos alternos en este caso de uso.			

Tabla 32 - EDCU CU-009

Caso de Uso	CU-009	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna

Nombre caso de uso	Consulta Log Ingreso Usuario	
Descripción caso de uso	Se trata de proveer la funcionalidad de consulta de la tabla de log de ingreso de los usuarios. Se podrá consultar por rango de fechas y por un usuario en específico.	
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.	
Post-condición	Se han visualizado los registros de la estructura de Log de Ingreso de Usuarios.	
Flujo Normal de Eventos		
Usuario	Sistema	
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de consulta de log de ingreso de usuarios.	2. El sistema despliega la pantalla en la que se despliega una lista con los resultados de los últimos registros de ingresos de usuarios en el sistema. Se permite filtrar por nombre de usuario y por un rango de fechas específico.	
Flujos Alternos		
No se cuenta con flujos alternos en este caso de uso.		

Tabla 33 - EDCU CU-010

Caso de Uso	CU-010	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Consulta Log Query Usuario		
Descripción caso de uso	Se trata de proveer la funcionalidad de consulta de la tabla de log de queries ejecutadas por los usuarios. Se podrá consultar por rango de fechas y por un usuario en específico.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han visualizado los registros de la estructura de Log de Queries de Usuarios.		
Flujo Normal de Eventos			
Usuario		Sistema	

1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para acceder a la funcionalidad de consulta de log de queries de usuario.	2. El sistema despliega la pantalla en la que se despliega una lista con los resultados de los últimos registros de queries ejecutadas por los usuarios. Se permite filtrar por nombre de usuario y por un rango de fechas específico. El sistema permitirá realizar una reconstrucción de la tabla de resultados obtenida, así como de llamar al servicio externo de visualización.
Flujos Alternos	
No se cuenta con flujos alternos en este caso de uso.	

Tabla 34 - EDCU CU-011

Caso de Uso	CU-011	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	Ninguna
Nombre caso de uso	Consulta Semantic PHI-BASE		
Descripción caso de uso	Es la funcionalidad principal del sistema en donde se podrán crear queries Sparql dinámicamente que se ejecutarán en el Endpoint de Virtuoso y se desplegarán los resultados en una tabla pudiendo también ser consultados por un módulo de visualización de grafo externo.		
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad.		
Post-condición	Se han ejecutado Queries de Sparql sobre el Servidor Virtuoso de acuerdo a los filtros seleccionados por el usuario y se han obtenido los resultados y han sido desplegados en una tabla que podrá ser visualizada por una funcionalidad externa para ver de una manera gráfica el grafo de resultado.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. El usuario se encuentra en el menú principal de la aplicación y da click sobre el enlace para ingresar a la funcionalidad de consulta Semantic PHI-BASE.		2. El sistema despliega una pantalla en donde se cargan a manera de árbol los detalles de conceptos semánticos con sus respectivas relaciones para que puedan ser seleccionadas por el usuario como filtros en el criterio de búsqueda.	
3. El usuario procede a seleccionar los filtros de su elección. Los filtros se realizarán arrastrando los elementos del árbol que desee y podrá seleccionar si desea usar conjunción o disyunción entre		4. El sistema construye automáticamente la query Sparql de acuerdo a los criterios de búsqueda ingresados por el usuario y procede a consultar el servidor de Virtuoso con la query construida. El sistema obtiene los resultados de la query y construye	

elementos de la misma clase, así como entre los diferentes grupos de filtro. Luego de seleccionar los filtros, el usuario da click sobre el botón de ejecutar consulta.	una tabla de salida que será desplegada en la pantalla del usuario. Se realiza un registro en la estructura de Log de query de usuario para llevar cuenta de todas las queries que han sido ejecutadas por los usuarios y los resultados que se han obtenido.
5. El usuario da click sobre uno de los elementos de la tabla de resultados obtenida al ejecutar la query.	6. El sistema obtiene el identificador del objeto sobre el cual el usuario ha pulsado y procede a consultar el servicio externo de visualización de grafo. Los resultados de la consulta al sistema externo de visualización se despliegan en un iframe que se encuentra embebido en la pantalla de consulta del usuario.
Flujos Alternos	
No hay flujos alternos para este caso de uso.	

Tabla 35 - EDCU CU-012

Caso de Uso	CU-012	Tipo Caso de Uso	Primario
Actores	Sistema	Referencias	Ninguna
Nombre caso de uso	Proceso Desbloqueo Automático de Usuarios		
Descripción caso de uso	Es un proceso que debe correr periódicamente para desbloquear los usuarios que han sido bloqueados por intentos fallidos en el inicio de sesión.		
Pre-condición	Se debe encontrar un usuario de la base de datos bloqueado por intentos fallidos de Login. El proceso a ejecutar tiene como parámetro de entrada el identificador (email) del usuario a desbloquear.		
Post-condición	Se ha desbloqueado automáticamente el usuario luego del tiempo parametrizado en la estructura de parámetros del sistema.		
Flujo Normal de Eventos			
Usuario		Sistema	
No hay interacción del usuario en este caso de uso ya que es un proceso que corre automáticamente un tiempo determinado después de que un usuario ha sido bloqueado por intentos fallidos de login		1. Se obtienen los datos del usuario del sistema que ha sido bloqueado. 2. Se verifica que el usuario siga inactivo por intentos fallidos de login. En caso de que no se encuentre en este estado se cancela la ejecución del proceso. 3. Se toma el Timestamp de bloqueo del usuario y se crea registro de histórico de bloqueos de usuario con ese valor y con el numero intentos fallidos realizados.	

	<p>4. Se actualiza el estado del usuario a activo y se reinicia el contador de intentos fallidos.</p> <p>5. Finaliza el proceso exitosamente.</p>
Flujos Alternos	
No hay flujos alternos dentro de este proceso.	

Tabla 36 - EDCU CU-013

Caso de Uso	CU-013	Tipo Caso de Uso	Primario
Actores	Sistema	Referencias	Ninguna
Nombre caso de uso	Proceso Cargue Automático Detalle Concepto Semántico		
Descripción caso de uso	Es un proceso que servirá para cargar automáticamente los datos de las estructuras de Detalle Concepto Semántico y sus relaciones de acuerdo a las configuraciones previamente realizadas en los conceptos semánticos.		
Pre-condición	Se deben tener cargadas correctamente las estructuras de Conceptos Semánticos y sus relaciones para poder así obtener las tripletas correspondientes de la base de Virtuoso.		
Post-condición	Se han cargado automáticamente las estructuras de Detalle Concepto Semántico y sus Relaciones de acuerdo a la parametrización establecida.		
Flujo Normal de Eventos			
Usuario		Sistema	
Este caso de uso no cuenta con una acción del usuario ya que será un proceso que podrá ser programado para ejecutar todos los días a una determinada hora, de manera que se tenga la base de datos del sistema actualizada de acuerdo a los registros que se tienen en la base de conocimientos de Virtuoso.		1. Se limpian las estructuras de detalle concepto semántico y detalle relación concepto para no obtener registros duplicados. 2. El sistema obtiene todos los conceptos semánticos de la base de datos. 3. Por cada concepto ejecuta una query Sparql para obtener sus elementos y cargarlos como registros de detalle de concepto semántico. 4. El sistema obtiene cada una de las relaciones de cada concepto semántico y por cada relación construye queries Sparql para obtener sus valores y así poblar la estructura de detalle de relación concepto semántico.	
Flujos Alternos			
No hay flujos alternos dentro de este proceso.			

Tabla 37 - EDCU CU-014

Caso de Uso	CU-014	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	CU-001
Nombre caso de uso	Actualización Contraseña Usuario		
Descripción caso de uso	Funcionalidad que permitirá al usuario actualizar su contraseña en el momento en que lo desee o cuando el sistema le indique que deba realizarlo.		
Pre-condición	El usuario debe estar autenticado en el sistema.		
Post-condición	Se ha actualizado la contraseña del usuario.		
Flujo Normal de Eventos			
Usuario		Sistema	
1. El usuario se encuentra en cualquier pantalla del Sistema y da click sobre el enlace para actualizar la contraseña.		2. El sistema despliega una pantalla en la que se solicita la contraseña actual, y la nueva contraseña con su verificación correspondiente.	
3. El usuario ingresa los datos solicitados y da click en el botón de actualizar.		4. El sistema valida que los datos sean correctos y que se cumplan las especificaciones de la nueva contraseña (longitud de la contraseña). En caso de ser todo correcto el sistema actualiza la contraseña del usuario y navega nuevamente hacia el menú principal de la aplicación desplegando un mensaje de actualización correcta.	
Flujos Alternos			
4a. El sistema detecta un error en los datos ingresados. Despliega mensaje de error y se retorna al paso 3 para que el usuario corrija los valores ingresados.			

Tabla 38 - EDCU CU-015

Caso de Uso	CU-015	Tipo Caso de Uso	Primario
Actores	Usuario	Referencias	CU-003
Nombre caso de uso	Asociar Servicio a Perfil de Usuario		

Descripción caso de uso	Permite asociar servicios del sistema a los perfiles de usuario para saber a qué servicios tiene acceso un determinado perfil.
Pre-condición	El usuario debe estar autenticado en el sistema y debe tener autorización para ejecutar esta funcionalidad. Se debe tener como parámetro el perfil de usuario al que se le van a asociar los servicios del sistema
Post-condición	Se han adicionado, visualizado, actualizado o eliminado registros de la estructura de asociación de Servicios del Sistema con perfiles de usuario.
Flujo Normal de Eventos	
Usuario	Sistema
1. El usuario se encuentra en la pantalla principal de Mantenimiento de Perfil de Usuario y selecciona la opción de Asociar Servicios del Sistema a un Perfil de Usuario específico.	2. El sistema despliega la pantalla principal de la funcionalidad de asociar servicios del sistema a un perfil de usuario. Se carga una lista en pantalla con los servicios del sistema ya asociados al perfil de usuario recibido por parámetro y se permite filtrar por un servicio del sistema específico.
3. El usuario da click sobre el botón de asociar un servicio del sistema al perfil de usuario.	4. El sistema despliega un pop-up solicitando los datos para realizar la asociación de un servicio del sistema al perfil de usuario.
5. El usuario selecciona el servicio del sistema a asociar al perfil de usuario y los datos adicionales y click sobre el botón adicionar.	6. El sistema valida y registra la asociación entre el servicio del sistema y el perfil de usuario. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de asociar servicios del sistema a perfil de usuario (paso 2).
7. El usuario da click sobre el botón de actualizar un servicio del sistema específico asociado al perfil de usuario.	8. El sistema despliega un pop-up en donde se muestran todos los campos correspondientes a la asociación del perfil de usuario con el servicio del sistema junto con sus valores actuales.
9. El usuario actualiza los valores que desea cambiar y posteriormente da click sobre el botón actualizar.	10. El sistema valida los nuevos valores ingresados sobre la asociación del servicio del sistema con el perfil de usuario y actualiza los datos en la base de datos. Si todo es correcto, el sistema despliega nuevamente la pantalla principal de la funcionalidad de asociar servicios del sistema a perfil de usuario (paso 2).
11. El usuario da click sobre el botón de eliminar un servicio del sistema de la asociación con el perfil de usuario.	12. El sistema despliega confirmación de eliminación del registro seleccionado.
13. El usuario acepta la confirmación de la eliminación del registro.	14. El sistema elimina el registro seleccionado de la base de datos y si todo es correcto se despliega nuevamente la pantalla principal de la funcionalidad

	de asociar servicios del sistema a perfil de usuario (paso 2).
Flujos Alternos	
<p>6a, 10a. En caso de que haya un dato incorrecto, el sistema desplegará mensaje de error indicándole al usuario los datos que debe corregir.</p> <p>13a. El usuario no acepta la confirmación de eliminación del registro. El sistema despliega nuevamente la pantalla principal de la funcionalidad de asociar servicios del sistema a perfil de usuario (paso 2).</p>	

ANEXO C – MANUAL DE DESPLIEGUE

Para desplegar la **Semantic PHI-BASE Web Interface**, se utilizará un ambiente Linux (Ubuntu 14.04 LTS) recién instalado, una base de datos MySQL y un servidor de aplicaciones Apache Tomcat. Se indicarán, primero, los pre-requisitos que se deben cumplir para poder instalar la aplicación, posteriormente se explicará la instalación y configuración de la aplicación y finalmente se comentará el mantenimiento y revisión del buen funcionamiento.

Para realizar el despliegue, es importante conocer los ficheros que componen el software completo de **Semantic PHI-BASE Web Interface**. La estructura de ficheros es la siguiente:

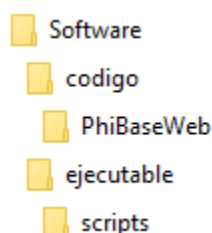


Imagen 24 - Estructura de Ficheros

Tabla 39 - Estructura de Ficheros

<u>Carpeta</u>	<u>Descripción</u>
Software	Carpeta principal que contiene a las carpetas de código y ficheros ejecutables descritas a continuación.
Código	Contiene el código de la aplicación.
PhiBaseWeb	Carpeta que contiene un proyecto de Eclipse Kepler , en el que se encuentra todo el código del sistema.
Ejecutable	Carpeta en donde se encuentran los binarios de la aplicación. Aquí se encuentra una carpeta que contiene scripts, un archivo .jar (Ejecutable Java dbkeygen.jar) para generar claves encriptadas y el archivo .war (PhiBaseWeb.war), en el cual se encuentra contenido todo lo necesario para el despliegue de la aplicación en Apache Tomcat.
Scripts	Contiene los siguientes archivos: <ul style="list-style-type: none">- Parametrizaciones.xlsx: Excel en el que se encuentran las parametrizaciones de todas las tablas básicas del sistema.- ScriptCreacionBD.sql: Archivo SQL en el que está todo lo necesario para crear el esquema de base de datos y las tablas necesarias.- ScriptPoblacionBD.sql: Archivo SQL que contiene todas las sentencias insert necesarias para realizar una población inicial de la base de datos. Ha sido generado de acuerdo a Parametrizaciones.xlsx

Pre-Requisitos

Los siguientes pasos se deben seguir para preparar el ambiente en el que correrá **Semantic PHI-BASE Web Interface**:

1. En un terminal ejecutar los siguientes comandos:
 - a. `sudo apt-get update` (Obtiene la lista de paquetes para saber si hay versiones nuevas que instalar)
 - b. `sudo apt-get dist-upgrade` (Se encarga de actualizar el software en el servidor)
2. Instalar Java en el servidor. Se recomienda instalar la versión de Oracle ya que es más estable que la versión Open JDK que se instala por defecto en Linux. Para esto, se debe realizar lo siguiente:
 - a. Descargar **Java Server JRE** para Linux x86 o x64 dependiendo de la arquitectura del servidor. Se puede descargar del siguiente enlace:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - b. Descomprimir archivo y dejar la carpeta resultante en una ruta conocida que será utilizada más adelante
3. Instalar y configurar base de datos MySQL:
 - a. Ejecutar: `sudo apt-get install mysql-server`
 - b. Ingresar contraseña para el usuario root
 - c. Conectarse a la base de datos: `mysql -u root -p`
 - d. Crear un usuario y otorgarle privilegios sobre el esquema llamado **PHIBASEDB** que se creará más adelante. El siguiente ejemplo muestra la creación de un usuario llamado PHIBASEDB con contraseña PHIBASEDB. Por seguridad, se recomienda que la contraseña no sea el mismo nombre de usuario:
 - i. `CREATE USER 'PHIBASEDB'@'%' IDENTIFIED BY 'PHIBASEDB';`
 - ii. `GRANT ALL PRIVILEGES ON PHIBASEDB.* TO 'PHIBASEDB'@'%';`
4. Descargar y descomprimir el servidor de aplicaciones de Apache Tomcat. Se puede descargar de la siguiente página: <http://tomcat.apache.org/download-70.cgi>. Se recomienda la última versión CORE de Tomcat 7, debido a que los ejecutables de la aplicación están optimizados para correr en esta versión. Sin embargo, la aplicación correrá en cualquier otro Tomcat desde la versión 6 en adelante. Al descomprimir el archivo tar.gz, se recomienda renombrar la carpeta a únicamente **Tomcat**, con la finalidad de tener mayor facilidad a la hora de realizar las configuraciones siguientes.

Instalación y Configuración de Aplicación

Primero, se debe realizar la configuración del Tomcat realizando los siguientes pasos:

1. Editar el archivo **Tomcat/conf/server.xml** para poner el puerto de escucha a 80:

```
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

Imagen 25 - Puerto Escucha Tomcat

2. Exportar las variables **JAVA_HOME** y **JRE_HOME** en el script **Tomcat/bin/catalina.sh**. Se deben poner de acuerdo a la ruta en donde se haya descomprimido el Java Server JRE:

```
export JAVA_HOME=/opt/jdk1.8.0_77/
export JRE_HOME=/opt/jdk1.8.0_77/jre/
```

Imagen 26 - Export variables Java

3. Copiar el ejecutable de la aplicación **PhiBaseWeb.war** dentro de la carpeta de aplicaciones del Tomcat en **Tomcat/webapps**.
4. Ejecutar script de arranque del Tomcat **Tomcat/bin/startup.sh**. En este momento se inicia el despliegue de la aplicación. Se debe esperar un momento hasta que termine de crear y descomprimir todo el contenido dentro de **Tomcat/webapps/PhiBaseWeb**.
5. Cuando se haya terminado de desplegar, ejecutar el script para finalizar el Tomcat y poder proceder a las siguientes configuraciones. Ejecutar **Tomcat/bin/shutdown.sh**.

Ahora, se deben realizar las configuraciones propias de los archivos de **Semantic PHI-BASE Web Interface** para el correcto funcionamiento de la aplicación. Primero se debe configurar el archivo de propiedades **Tomcat/webapps/PhiBaseWeb/WEB-INF/conf/config.properties**:

Tabla 40 - Configuración config.properties

<u>Propiedad</u>	<u>Configuración / Descripción</u>
edu.upm.spbw.db.driver	Se trata del driver de la base de datos. Al ser MySQL, el valor que trae por defecto será válido.
edu.upm.spbw.db.url	Es la URL para acceder a la base de datos. Se debe encontrar en formato de dirección JDBC. Para el ejemplo de configuración en el que la base de datos es local, sería válido tener el siguiente valor: <i>jdbc:mysql://127.0.0.1:3306/PHIBASEDB</i>
edu.upm.spbw.db.username	Nombre del usuario de la base de datos. Es el nombre que se ha creado en los pre-requisitos al momento de instalar la base de datos. Este valor

	debe estar encriptado utilizando el programa dbkeygen.jar .
edu.upm.spbw.db.passwd	Contraseña del usuario de la base de datos. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .
edu.upm.spbw.db.dialect	Dialecto de la base de datos. Al usar MySQL, el valor que viene por defecto es el correcto.
filecfg	Ruta del fichero de configuración del log. Este fichero se encuentra en la misma ruta donde se encuentra el config.properties .
edu.upm.spbw.sparql.prefix	Ruta del fichero que contiene los prefijos para realizar las consultas SPARQL al Endpoint de virtuoso. Este fichero se encuentra también en la misma ruta donde se encuentra el config.properties .
edu.upm.spbw.login.link	Se trata de la dirección de ingreso a la aplicación. Este parámetro será usado para el envío de los correos automáticos en la creación de usuarios del sistema.
edu.upm.spbw.mail.logo	Ruta de donde se encuentra el logo de la aplicación para poderlo adjuntar en los correos electrónicos.
edu.upm.spbw.mail.account	Cuenta de correo desde donde se enviarán los correos electrónicos del sistema. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .
edu.upm.spbw.mail.password	Contraseña de la cuenta de correo. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .
edu.upm.spbw.mail.host	Dirección del host SMTP del servidor de correos. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .
edu.upm.spbw.mail.port	Puerto del servidor SMTP para envío de correos. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .
edu.upm.spbw.mail.auth	Indicador de si se requiere autenticación en el momento de envío de correos. Su valor será 1 o 0. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .

edu.upm.spbw.mail.tls	Indicador de si el envío de correos se debe realizar utilizando el protocolo de seguridad TLS. Su valor será 1 o 0. Este valor debe estar encriptado utilizando el programa dbkeygen.jar .
-----------------------	---

Para los valores que se deben encriptar utilizando el programa **dbkeygen.jar**, se deben seguir los siguientes pasos:

1. Abrir una consola de comandos y situarse en la ubicación en donde se encuentra el ejecutable **dbkeygen.jar**.
2. Ejecutar el comando `java -jar dbkeygen.jar <texto-a-encriptar>`. Por ejemplo, una ejecución sería la siguiente:

```
C:\temp>java -jar dbkeygen.jar prueba
-----
U9GrLOpuXZo=
-----
prueba
-----
```

Imagen 27 - Encripción con herramienta dbkeygen.jar

3. Copiar el texto que se muestra en la primera fila como salida y pegarlo en la propiedad correspondiente. El texto para el ejemplo sería **U9GrLOpuXZo=**.

El siguiente archivo a configurar es el archivo **Tomcat/webapps/PhiBaseWeb/WEB-INF/conf/logger.properties**. La mayoría de los parámetros de este archivo son funcionales por defecto. El único valor que se debe modificar es el del parámetro **log4j.appender.fileappender.File**. Su valor debe tener la ruta de donde se quiere que se generen los archivos de Log de la aplicación.

Finalmente, al tener todos los archivos configurados correctamente, será posible arrancar nuevamente el servidor de aplicaciones y comprobar que todo funciona bien. Para arrancar el servidor de aplicaciones Tomcat se debe ejecutar el script de **Tomcat/bin/startup.sh**.

Mantenimiento y Revisión Funcionamiento

Luego de que la aplicación se encuentre en funcionamiento, es importante, realizar revisiones periódicas de los archivos de Log para ver que todo se encuentre bien. Para revisar los archivos de Log de la aplicación, basta con consultarlos en la ruta que se ha configurado en el archivo **logger.properties**. Siempre se generarán los Logs de la aplicación en la ruta indicada en el parámetro **log4j.appender.fileappender.File** de este archivo.

Por otra parte, los ejecutables de las clases que componen el proyecto **PhiBaseWeb**, se encuentran en la ruta **Tomcat/webapps/PhiBaseWeb/WEB-INF/classes**. En caso de ser necesario realizar modificaciones sobre alguna de las clases, se debe reemplazar el archivo .class correspondiente y posteriormente dar un reinicio al servidor de aplicaciones Tomcat.

Para reiniciar el servidor, se deben ejecutar los scripts de **shutdown.sh** y posteriormente **startup.sh**. La siguiente imagen muestra un reinicio del servidor:

```
ubuntu@ip-10-0-0-9:/opt/tomcat/bin$ sudo ./shutdown.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /opt/jdk1.8.0_77/jre/
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
ubuntu@ip-10-0-0-9:/opt/tomcat/bin$ sudo ./startup.sh
Using CATALINA_BASE:   /opt/tomcat
Using CATALINA_HOME:   /opt/tomcat
Using CATALINA_TMPDIR: /opt/tomcat/temp
Using JRE_HOME:        /opt/jdk1.8.0_77/jre/
Using CLASSPATH:       /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Tomcat started.
```

Imagen 28 - Reinicio Servidor Tomcat

Es importante verificar que el servidor ha arrancado nuevamente sin problema alguno.

Finalmente, los archivos de estilos y pantallas xhtml, podrán ser modificados directamente sin tener que reiniciar el servidor o realizar cambios adicionales. Esto permitirá que se actualicen dinámicamente las pantallas en la aplicación sin tener que detener el servicio.

ANEXO D – MANUAL DE DESARROLLADOR

En este apartado se explica la estructura del proyecto **Semantic PHI-BASE Web Interface**, de manera que cualquier otro desarrollador pueda adicionar y modificar las funcionalidades ya existentes sin tener ningún problema. Para esto, se explicará primero como montar el proyecto sobre el IDE de Eclipse y ejecutarlo a manera de pruebas, luego se comentará como adicionar una nueva tabla en la base de datos, junto con todas sus implicaciones, y finalmente se explicará cómo se debe hacer para crear una nueva pantalla/funcionalidad dentro del sistema.

Proyecto PhiBaseWeb – Eclipse Kepler

Como ya se ha mencionado anteriormente, el proyecto **PhiBaseWeb**, ha sido desarrollado utilizando el IDE de Eclipse Kepler. Para importarlo correctamente se debe realizar lo siguiente:

1. Descargar y descomprimir la versión de Eclipse Kepler de la página:
<http://www.eclipse.org/downloads/packages/release/Kepler/SR2>
2. Instalar el Plugin de ICEfaces 3.3 para Eclipse Kepler, siguiendo las instrucciones que se encuentran en la página:
<http://www.icesoft.org/java/downloads/detail-new.jsf>
3. Importar el proyecto **PhiBaseWeb** en Eclipse. Para esto, ir a File → Import → General → Existing Projects into Workspace. Seleccionar la carpeta **PhiBaseWeb** e importarlo.

En el momento en que haya finalizado la importación del proyecto, se debe ver una estructura como la siguiente en el IDE:

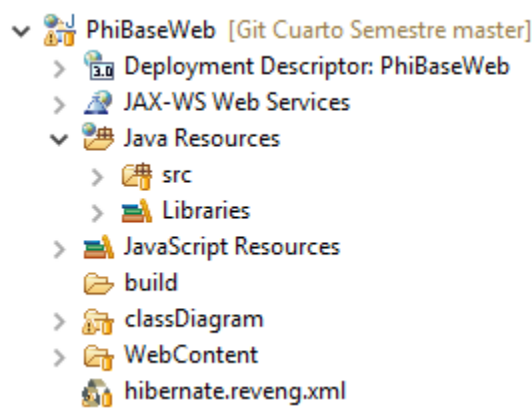


Imagen 29 - Proyecto Eclipse PhiBaseWeb

Las carpetas más importantes son **src**, perteneciente a **Java Resources**, la cual contiene todas las clases Java y algunos archivos de configuración, y **WebContent**, que es la carpeta que contiene todas las plantillas, scripts, hojas de estilo y pantallas Web de la aplicación.

Los paquetes y archivos que componen el código de la aplicación son los siguientes:

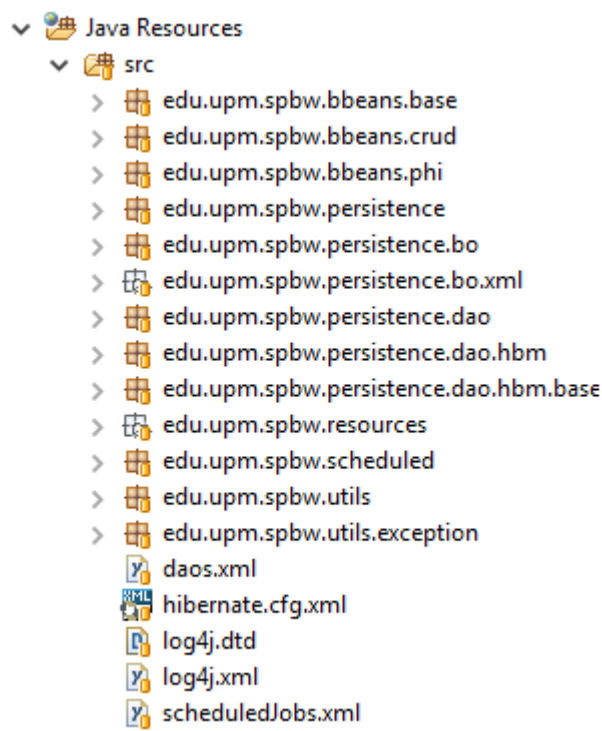


Imagen 30 - Paquetes Proyecto y Archivos Configuración

A continuación, se describe cada uno de ellos:

Tabla 41 - Paquetes Proyecto y Archivos Configuración

<u>Paquete / Archivo Configuración</u>	<u>Descripción</u>
edu.upm.spbw.bbeans.base	Este paquete contiene todas las clases básicas para la implementación de los servicios del sistema o Backing Beans. Se encuentran clases como el Bean de autenticación, el manejador de navegación, el Bean base, etc.
edu.upm.spbw.bbeans.crud	Aquí se encuentran todos los backing beans correspondientes a los servicios CRUD (Create-Read-Update-Delete) del sistema.
edu.upm.spbw.bbeans.phi	Contiene el Backing Bean que se encarga de manejar los eventos de la pantalla principal de consulta Semantic PHI-BASE. Además, se encuentran todas las clases auxiliares que ayudan a la generación de las consultas automáticas.
edu.upm.spbw.persistence	Paquete base de persistencia de la aplicación. Contiene una clase que es capaz de manejar conexiones directas JDBC con la base de datos, una clase de excepción de base de datos y finalmente una clase para ejecutar consultas indicando condiciones en los campos de una tabla.

edu.upm.spbw.persistence.bo	Contiene todos los Business Objects que representan a las tablas de la base de datos en objetos.
edu.upm.spbw.persistence.bo.xml	Contiene los archivos XML en donde se realiza el mapeo entre campos de la base de datos y atributos de las clases BO definidas previamente.
edu.upm.spbw.persistence.dao	Contiene la definición de las interfaces encargadas del acceso a cada una de las tablas de la base de datos.
edu.upm.spbw.persistence.dao.hbm	Este paquete tiene la implementación concreta de las interfaces definidas previamente, utilizando el framework de Hibernate.
edu.upm.spbw.persistence.dao.hbm.base	Contiene clases básicas de Hibernate para el acceso a la base de datos. Se encuentra la fábrica de sesiones y la definición del DAO genérico de Hibernate.
edu.upm.spbw.resources	En este paquete se encuentran los archivos de mensajes de la aplicación en los diferentes idiomas en que se quiera que se despliegue. Actualmente se encuentra definido únicamente el lenguaje español.
edu.upm.spbw.scheduled	En este paquete se encontrarán todos los procesos que deban correr de una manera periódica en el sistema. Actualmente solo existe un proceso, el cual es el encargado de obtener todos los conceptos semánticos de la base de conocimientos y cargarlos directamente en la base de datos para poder generar las consultas automáticamente.
edu.upm.spbw.utils	Paquete en el que se encuentran clases para utilidades del sistema. Utilidades puede ser por ejemplo el manejo de fechas, criptografía, escritura de log, envío de correos, etc.
edu.upm.spbw.exceptions	Paquete para definir las posibles excepciones que la aplicación puede arrojar.
daos.xml	Archivo de configuración de Spring en donde se indica la implementación de los DAOs concretos que serán inyectados en tiempos de ejecución.
hibernate.cfg.xml	Archivo de configuración de Hibernate en donde se indican las propiedades para la conexión con la base de datos y los archivos de mapeos entre tablas y objetos.
log4j.dtd	Archivo de definición del formato XML de la configuración de Log4j. No debe ser modificado.

log4j.xml	Archivo en donde se definen las propiedades de configuración para Log4j. No debe ser modificado.
scheduledJobs.xml	Archivo de configuración de Spring en donde se definen los procesos que deben ejecutarse periódicamente.

El contenido de la carpeta **WebContent** es el siguiente:

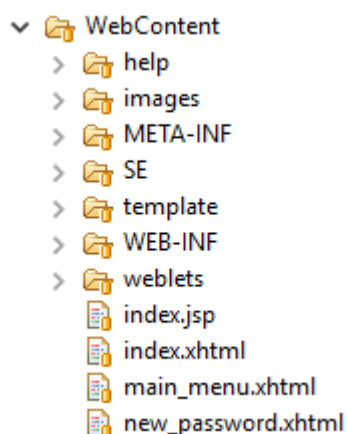


Imagen 31 - Contenido WebContent

Tabla 42 - Contenido WebContent

<u>Carpeta / Archivo</u>	<u>Descripción</u>
help	Carpeta que contiene todas las pantallas de manual de usuario del sistema. Interior a esta carpeta, se encuentra una carpeta llamada docs , en la cual se sigue la misma estructura de pantallas principal para poder invocar dinámicamente las pantallas de ayuda de acuerdo a la ubicación del usuario en el mapa de navegación.
images	Esta carpeta contiene todas las imágenes que son utilizadas en las pantallas del sistema.
META-INF	Carpeta que contiene el archivo de Manifest usado en el despliegue de la aplicación. No debe ser modificado su contenido
SE	Carpeta que contiene todas las pantallas de la definición de los servicios del sistema. Los nombres de los servicios siguen un consecutivo y están de acuerdo a la definición de los Backing Beans que lo soportan.
template	Carpeta en la que se encuentran definidas todas las plantillas para una definición estándar de las pantallas. Además de las plantillas, se encuentran definidos aquí las hojas de estilo CSS y los scripts de JavaScript necesarios para la operación del sistema.

WEB-INF	Es una de las carpetas más importantes de la aplicación, ya que aquí se encuentran definidos los archivos principales de configuración y las librerías adicionales necesarias para la correcta ejecución del sistema.
weblets	En esta carpeta se encuentran definidas plantillas de componentes Web que son usados en las pantallas. Por ejemplo, un weblet puede ser considerado el botón de adicionar de las pantallas CRUD, o la definición del Paginador de las listas de pantalla.
index.jsp	Pantalla de entrada al sistema. Esta pantalla sólo tiene una redirección hacia la pantalla principal index.xhtml .
index.xhtml	Pantalla principal del sistema. Esta pantalla presenta el formulario para realizar el inicio de sesión.
main_menu.xhtml	Pantalla en la que se realiza la carga dinámica del menú principal de la aplicación. Las opciones desplegadas en esta pantalla dependerán de los permisos otorgados al perfil de usuario.
new_password.xhtml	Pantalla encargada de la funcionalidad de la actualización de contraseña de un usuario.

Adicionar Nueva Tabla en Base de Datos

Para adicionar una tabla nueva en la base de datos, se deben seguir los siguientes pasos:

1. Crear la tabla en el motor de base de datos utilizando la sentencia `CREATE TABLE` correspondiente.
2. Crear el objeto BO (Business Object) y el archivo de mapeo de manera correspondiente en los paquetes **edu.upm.spbw.persistence.bo** y **edu.upm.spbw.persistence.bo.xml**. Esto es posible hacerlo de manera manual, o automáticamente utilizando Hibernate Tools. Para Hibernate Tools:
 - a. Instalar y configurar Hibernate Tools en Eclipse Kepler.
 - b. Configurar la conexión a la base de datos de acuerdo a las propiedades definidas en el archivo de configuración **hibernate.cfg.xml**.
 - c. Incluir la tabla en el archivo **hibernate.reveng.xml**.
 - d. Ejecutar **Hibernate Code Generation** indicándole que se quiere generar el BO y el archivo de mapeo.
3. Revisar que el objeto creado y el archivo de mapeo tienen todas las propiedades definidas de manera correcta, de acuerdo a la definición de la tabla en la base de datos.
4. Crear interfaz en el paquete **edu.upm.spbw.persistence.dao**, en donde se definirán las operaciones necesarias adicionales para interactuar con la nueva tabla creada.
5. Crear clase concreta en el paquete **edu.upm.spbw.persistence.dao.hbm**, la cual implementa las operaciones definidas en la interfaz anterior.
6. Declarar la tabla y la clase DAO en los archivos de **hibernate.cfg.xml** y **daos.xml** correspondientemente.

Para verificar que todos los pasos se han realizados correctamente, se pueden revisar clases que soportan las tablas que han sido definidas previamente y ver que se siguen los mismos estándares.

Creación nueva Pantalla

Para crear una nueva pantalla en el sistema se deben seguir los siguientes pasos:

1. Crear nueva clase de Backing Bean que herede de la clase BaseBean definida en el paquete **edu.upm.spbw.bbeans.base**. Esta clase se encargará de soportar los servicios necesarios en la nueva pantalla y recibir las peticiones que llegan directamente desde la interfaz.
2. Registrar el Backing Bean en el archivo de configuración **faces-config.xml** que se encuentra en **WebContent/WEB-INF/**.
3. Crear pantalla XHTML usando las plantillas definidas y siguiendo el formato estándar de la aplicación. Se puede tomar como ejemplo cualquiera de las otras pantallas que ya haya sido definida previamente en el sistema.
4. Incluir la pantalla en el mapa de navegación del archivo **faces-config.xml**. Es importante que se haga este paso de manera correcta y con nombres estándar de servicios ya que de esto depende que se pueda navegar correctamente desde y hacia la pantalla creada.
5. Crear servicio del sistema en la tabla CSPSERSI – SERVICIO DEL SISTEMA. Se debe seguir el estándar de códigos de nombre definido previamente. Posteriormente, asociar el servicio al perfil o perfiles de usuario que podrán acceder al nuevo servicio creado.
6. Verificar funcionamiento de la nueva pantalla.