

**COMPONENTE WEB PARA EL REGISTRO Y ANÁLISIS DE LA INFORMACIÓN
DE USUARIOS EN UN SISTEMA DE INFORMACIÓN PARA LA
AUTOEVALUACIÓN DE PROGRAMAS DE POSGRADO EN UNIVERSIDADES
PÚBLICAS.**

**JONATHAN ARIAS SÁNCHEZ
OSCAR DAVID DUITAMA GARCÍA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2014**

**COMPONENTE WEB PARA EL REGISTRO Y ANÁLISIS DE LA INFORMACIÓN
DE USUARIOS EN UN SISTEMA DE INFORMACIÓN PARA LA
AUTOEVALUACIÓN DE PROGRAMAS DE POSGRADO EN UNIVERSIDADES
PÚBLICAS.**

**JONATHAN ARIAS SÁNCHEZ
OSCAR DAVID DUITAMA GARCÍA**

**TRABAJO DE INVESTIGACIÓN PRESENTADO COMO REQUISITO PARA
OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS E INFORMÁTICA**

**DIRECTORA
PhD. SONIA CRISTINA GAMBOA SARMIENTO
DOCTORA EN EDUCACIÓN**

**CODIRECTOR
MSc. CARLOS HUMBERTO CARREÑO DÍAZ
MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2014**

DEDICATORIA

Agradezco primeramente a Dios por darme la fe y las fuerzas necesarias para culminar esta etapa de mi vida.

A mis padres y a toda mi familia, por el apoyo incondicional brindado y por estar siempre a mi lado en cada momento de mi vida.

En especial, dedico este logro a mi hermano, el cual ha sido mi inspiración y me ha dado el aliento para no rendirme nunca.

Jonathan Arias Sánchez

AGRADECIMIENTOS

Agradecimientos especiales:

A el Magíster Carlos Humberto Carreño Díaz que, nos ha orientado, apoyado y corregido durante el transcurso de este proyecto.

A la Doctora Sonia Cristina Gamboa Sarmiento por brindarnos la confianza para la realización de este proyecto.

A la Universidad Industrial de Santander y a la Escuela de Ingeniería de Sistemas e Informática, los cuales contribuyeron en nuestra formación como Ingenieros.

A todos nuestros profesores, amigos y compañeros de la Universidad, de los cuales se aprendió y compartió durante el paso por la Universidad.

CONTENIDO

	Pág.
INTRODUCCIÓN	16
1. JUSTIFICACIÓN	17
2. OBJETIVOS	18
2.1 OBJETIVO GENERAL	18
2.2 OBJETIVOS ESPECÍFICOS.....	18
3. MARCO DE REFERENCIA.....	19
3.1 AUTOEVALUACIÓN	19
3.2 ARQUITECTURAS EJB (ENTERPRISE JAVA BEAN).....	19
3.3 MONGODB.....	20
3.4 LIBRERÍAS UTILIZADAS.....	21
4. METODOLOGÍA.....	23
5. ANÁLISIS, REQUERIMIENTOS Y CASOS DE USO DEL COMPONENTE	25
5.1 ESPECIFICACIÓN DE REQUISITOS SOFTWARE.....	25
5.2 PROPÓSITO	25
5.3 ALCANCE DEL SISTEMA	25
5.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	26
5.5 PERSPECTIVA DEL PRODUCTO	27
5.6 FUNCIONES DEL PRODUCTO	27
5.7 CARACTERÍSTICAS DEL USUARIO.....	28
5.8 RESTRICCIONES.....	28
5.9 SUPOSICIONES Y DEPENDENCIAS.....	28
5.10 REQUISITOS FUTUROS.....	28
5.11 REQUERIMIENTOS FUNCIONALES.....	29
5.12 REQUERIMIENTOS NO FUNCIONALES	32
5.13 ACTORES DEL SISTEMA.....	34
5.14 DESCRIPCIÓN DE ACTORES.....	34
5.15 PRINCIPALES CASOS DE USO IDENTIFICADOS.....	35
5.16 DIAGRAMA GENERAL DE CASOS DE USO Y ACTORES.....	36

5.17	DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO POR ACTORES.....	37
5.18	DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO	38
6.	DISEÑO DE OBJETOS, ESTRUCTURAS DE DATOS Y COMPONENTES	49
6.1	DISEÑO DE LA BASE DE DATOS.....	49
6.2	DISEÑO DE ENTIDADES UTILIZANDO <i>MORPHIA FRAMEWORK</i>	53
6.3	DIAGRAMA DE CLASES.....	54
6.4	IMPLEMENTACIÓN DE DISEÑO DE CLASES	56
7.	DESARROLLO.....	64
7.1	MODELO VISTA CONTROLADOR.....	64
7.2	DESCRIPCIÓN DE LOS COMPONENTES	64
7.3	INTERACCIÓN DE COMPONENTES.....	65
7.4	USO DE PATRONES DE DISEÑO: FACADE.....	65
7.5	USO DE PATRONES DE DISEÑO: MODELO VISTA CONTROLADOR (MVC)	67
8.	PRUEBAS DEL COMPONENTE	69
8.1	PRUEBAS JUNIT	72
9.	CONCLUSIONES.....	76
10.	SUGERENCIAS Y RECOMENDACIONES.....	77
	BIBLIOGRAFÍA	78
	ANEXOS	80
	ANEXO A - MANUAL DE USUARIO	81

LISTADO DE TABLAS

	Pág.
Tabla 1. Requerimiento Funcional 01 – Consultar Información de Usuarios	29
Tabla 2. Requerimiento Funcional 02 – Actualizar Usuario	29
Tabla 3. Requerimiento Funcional 03 – Consultar Propiedad Intelectual	29
Tabla 4. Requerimiento Funcional 04 – Modificar Propiedad Intelectual	30
Tabla 5. Requerimiento Funcional 05 – Agregar Propiedad Intelectual	30
Tabla 6. Requerimiento Funcional 06 – Eliminar Propiedad Intelectual	30
Tabla 7. Requerimiento Funcional 07 – Consultar Parámetros	31
Tabla 8. Requerimiento Funcional 08 – Modificar Parámetros	31
Tabla 9. Requerimiento Funcional 09 – Generar Reportes	31
Tabla 10. Requerimiento No Funcional 01 – Interfaz del Sistema	32
Tabla 11. Requerimiento No Funcional 02 – Mantenimiento del Sistema	32
Tabla 12. Requerimiento No Funcional 03 – Confiabilidad del Sistema	32
Tabla 13. Requerimiento No Funcional 04 - Interoperabilidad	33
Tabla 14. Requerimiento No Funcional 05 – Facilidad para hacer pruebas	33
Tabla 15. Requerimiento No Funcional 06 – Validación de Datos	33
Tabla 16. Requerimiento No Funcional 07 - Arquitectura	33
Tabla 17. Requerimiento No Funcional 08 – Ingreso de Datos	34
Tabla 18. Descripción de Actores - Coordinador	34
Tabla 19. Descripción de Actores – Operador Administrativo	35
Tabla 20. Descripción de Casos de Uso – Consultar Información de Usuarios	40
Tabla 21. Descripción de Casos de Uso – Actualizar Usuario	41
Tabla 22. Descripción de Casos de Uso – Consultar Propiedad Intelectual	42
Tabla 23. Descripción de Casos de Uso – Modificar Propiedad Intelectual	43
Tabla 24. Descripción de Casos de Uso – Agregar Propiedad Intelectual	44
Tabla 25. Descripción de Casos de Uso – Eliminar Propiedad Intelectual	45
Tabla 26. Descripción de Casos de Uso – Consultar Parámetros	46
Tabla 27. Descripción de Casos de Uso – Modificar Parámetros	47
Tabla 28. Descripción de Casos de Uso – Ver y Generar Reportes	48
Tabla 29. Clases Asociadas al Paquete Modelo	58
Tabla 30. Clases Asociadas al Paquete Dao	58
Tabla 31. Clases Asociadas al Paquete Controller	61
Tabla 32. Pruebas Unitarias – Crear Persona	70
Tabla 33. Pruebas Unitarias – Crear Propiedad Intelectual	71
Tabla 34. Pruebas Unitarias – Generar Reportes	71

LISTADO DE FIGURAS

	Pág.
Figura 1. Arquitectura de los EJB	20
Figura 2. Descripción general de los modelos UWE	23
Figura 3. Actores del Sistema	34
Figura 4. Diagrama general de Casos de Uso y Actores	36
Figura 5. Diagrama de Casos de Uso – Operador Administrativo	37
Figura 6. Diagrama de Casos de Uso – Coordinador	38
Figura 7. Diagrama de Casos de Uso – Gestionar Personas	38
Figura 8. Diagrama de Casos de Uso – Gestionar Propiedad Intelectual	41
Figura 9. Diagrama de Casos de Uso – Administrar Parámetros	45
Figura 10. Diagrama de Casos de Uso – Ver y Generar Reportes	47
Figura 11. Esquema de la Base de Datos – Tablas Personas, Estudios e Idiomas	50
Figura 12. Esquema de la Base de Datos – Tablas Grupos Investigación, Programa Académico y Proyectos Investigación	51
Figura 13. Esquema de la Base de Datos – Tablas Experiencia Profesional, Propiedad Intelectual, Distinciones y Líneas Investigación	52
Figura 14. Modelo Conceptual de la Base de Datos	52
Figura 15. Diagrama de BaseEntity	53
Figura 16. Diagrama de la @Entidad Personas	54
Figura 17. Diagrama de Clases	55
Figura 18. Clases de Modelo de Negocio	56
Figura 19. Conexión de los Paquetes Dao, Facade y Controller para la clase ProgramaAcadémico	59
Figura 20. Vista del Paquete Controller en NetBeans	60
Figura 21. Diagrama de Actividades – Administrar Parámetros	61
Figura 22. Diagrama de Actividades – Gestionar Usuarios	62
Figura 23. Diagrama de Actividades – Generar Reportes	62
Figura 24. Diagrama de Navegación del Componente	63
Figura 25. Interacción de Componentes	65
Figura 26. Estructura sin clase Facade vs. Estructura con clase Facade	66
Figura 27. Estructura de la clase AbstractFacade<T>	67
Figura 28. Representación de aplicación del patrón de diseño MVC	67
Figura 29. Bosquejo final de la página principal	68
Figura 30. Pruebas JUnit – Prueba 1: crearPersona	72
Figura 31. Pruebas JUnit – Prueba 2: crearPersona	72
Figura 32. Pruebas JUnit – Prueba 3: crearPersona	73
Figura 33. Pruebas JUnit – Prueba 4: crearPropiedadIntelectual	73
Figura 34. Pruebas JUnit – Prueba 5: crearPropiedadIntelectual	74
Figura 35. Pruebas JUnit – Prueba 6: generarDiagramaCircular	74

Figura 36. Pruebas JUnit – Prueba 7: generarDiagramaLineal..... 74

Figura 37. Pruebas JUnit – Prueba 8: generarDiagramaBarra..... 75

RESUMEN

TÍTULO

COMPONENTE WEB PARA EL REGISTRO Y ANÁLISIS DE LA INFORMACIÓN DE USUARIOS EN UN SISTEMA DE INFORMACIÓN PARA LA AUTOEVALUACIÓN DE PROGRAMAS DE POSGRADO EN UNIVERSIDADES PÚBLICAS.

AUTORES

JONATHAN ARIAS SÁNCHEZ
OSCAR DAVID DUITAMA GARCÍA

PALABRAS CLAVE

Portales Web, Portales de Conocimiento, Sociedad del Conocimiento, Conocimiento Científico, Grupos de Investigación.

DESCRIPCIÓN

Los programas académicos de postgrado, especialmente maestrías y doctorados, requieren tener visibilidad en la web con información actualizada acerca de sus propias características que permitan hacer una permanente verificación, coordinación y un análisis constante de los procesos educativos para identificar fortalezas, debilidades, amenazas y oportunidades que ayuden al mejoramiento continuo de la calidad de enseñanza, dicho proceso se denomina autoevaluación. Asimismo, se requiere mantener registro de la información concerniente tanto a sus actores como a los procesos académicos.

Para ello es necesario disponer de ambientes virtuales en los cuales los actores de estos programas puedan interactuar entre ellos mismos, y llevar a cabo procesos académicos. Este proyecto busca diseñar e implementar una aplicación web mediante la cual se ofrezca algunos servicios en un portal de programa de postgrado. Se desarrollará un componente web que permitirá la captación de diferentes tipos de datos de los usuarios de programas de posgrado, lo cual dará vía libre al tratamiento de los datos para obtener información (reportes) que ayudaran con el proceso de autoevaluación en los diferentes programas. Dicho componente debe estar integrado al portal de posgrados, debe permitir captación, almacenamiento y tratamiento de los diferentes datos y generar diferentes reportes de la información procesada.

*Trabajo de grado modalidad de investigación.

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director: Sonia Cristina Gamboa Sarmiento, PhD

ABSTRACT

TITLE

WEB COMPONENT FOR REGISTRATION AND ANALYSIS OF USERS' INFORMATION IN AN INFORMATION SYSTEM FOR SELF-ASSESSMENT OF POSTGRADUATE PROGRAMS IN PUBLIC UNIVERSITIES.

AUTHORS

JONATHAN ARIAS SÁNCHEZ
OSCAR DAVID DUITAMA GARCÍA

KEY WORDS

Web portals, knowledge portals, knowledge society, research groups, scientific knowledge.

DESCRIPTION

The postgraduate academic programs, especially masters and doctoral degrees, need to have access to the web and find out updated information about the features of the program. In addition, it is required to maintain registers of the relevant information from both its participants and academic processes in order to make the continuous verification, coordination and constant analysis of the educative processes, identifying strengths, weaknesses, threats and opportunities and helping to the continuous improvement of the quality of teaching, this process is called self assessment.

Therefore, it is necessary to have virtual environments in which the participants of those programs could interact among themselves and carry out academic processes. Hence, this project seeks to design and implement a web application which offers some services on a web portal that contains a web component that will allow the collection of different types of data from users of postgraduate programs. It will give way for the processing of data to obtain concerning information (reports) to help with the process of self-assessment in the programs involved. This component must be integrated into the postgraduate portal so that it allows collection, storage and processing of different data and generate diverse reports of the processed information.

*Bachelor thesis.

**Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas. Director: Sonia Cristina Gamboa Sarmiento, PhD

INTRODUCCIÓN

En el campo académico, es necesario hacer una permanente verificación, coordinación y un análisis constante de los procesos educativos para identificar fortalezas, debilidades, amenazas y oportunidades que ayuden al mejoramiento continuo de la calidad de enseñanza. Dicho proceso se denomina autoevaluación.

En la Universidad Industrial de Santander existen numerosos programas de posgrado que tienen la necesidad de implementar un sistema que promueva la autoevaluación para garantizar la calidad de estos. Por esta razón, la escuela de Ingeniería de Sistemas e Informática propone el diseño de un portal web para programas de posgrado que facilite la interacción de docentes y estudiantes, en donde encuentren información asertiva, oportuna y actualizada cada vez que así lo requieran, de manera que el acceso sea fácil y ayude a dichos programas académicos a llevar un registro de cada uno de los participantes y los procesos que van desarrollando para obtener resultados que permitan mostrar la eficacia y buen desempeño del programa.

Durante la presentación de este proyecto se presentará el marco de referencia con conceptos clave que ayudarán al entendimiento de proceso de Autoevaluación en el contexto académico, y cómo fue desarrollado un componente software que soporte dicho proceso, incluyendo la metodología seleccionada, así como la descripción detallada del proceso de ingeniería de software, desde el análisis, diseño, desarrollo y finalizando con las pruebas realizadas.

1. JUSTIFICACIÓN

Durante los últimos años se ha podido notar que los programas académicos de posgrado requieren tener visibilidad en la web para facilitar la interacción con información actualizada acerca de sus propios programas de maestría y doctorado. Esta información es un valioso recurso en los programas académicos, a partir de la cual es posible encontrar patrones, tendencias u otros conocimientos que favorezcan los procesos de toma de decisiones, planificación y autoevaluación.

Para ello se requiere mantener registro de la información referente tanto a sus actores como a los procesos académicos, haciendo necesario disponer de ambientes virtuales en los cuales los actores de estos programas puedan interactuar entre ellos mismos, alimentando el sistema con información que soporte los diferentes procesos académicos que se llevan a cabo. De esta manera se propiciará la agilización de procesos y la toma de decisiones a nivel académico y administrativo de acuerdo a la información de los estudiantes de posgrado, profesores y directivos.

Este proyecto busca diseñar e implementar una aplicación web mediante la cual se ofrezca algunos servicios en un portal de programa de postgrado. Se desarrollará un componente web que permitirá la captación de diferentes tipos de datos de los usuarios de programas de posgrado, lo cual dará vía libre al tratamiento de los datos para obtener información (reportes) que ayudaran con el proceso de autoevaluación en los diferentes programas. Dicho componente debe estar integrado al portal de posgrados, debe permitir captación, almacenamiento y tratamiento de los diferentes datos y generar diferentes reportes de la información procesada.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e implementar un componente web para el registro y análisis de la información de usuarios, para autoevaluación de programas de posgrado en universidades públicas.

2.2 OBJETIVOS ESPECÍFICOS

- Especificar los requerimientos principales del sistema, determinando los diferentes actores, escenarios y servicios que prestarán.
- Realizar el diseño de objetos, estructuras de datos e interfaces que se van a utilizar de acuerdo a los requerimientos previamente establecidos.
- Implementar el componente software para el registro y análisis de los datos de los diferentes usuarios de un portal web de autoevaluación, de manera que permita:
 - Generar campos dinámicos en los formularios de registro de información del sistema.
 - Elaborar reportes estadísticos a partir de la información académica y financiera de los usuarios registrados en sistema de información.
 - Diseñar y desarrollar un conjunto de servicios que permitan la comunicación con los demás componentes del sistema de información de autoevaluación.
 - Administrar los parámetros de configuración del componente, de manera que sea modificable una vez sea implantado en el servidor de aplicaciones.
- Diseñar y aplicar un conjunto de pruebas en cada uno de los módulos del componente, utilizando el *framework* JUnit.

3. MARCO DE REFERENCIA

3.1 AUTOEVALUACIÓN

Se entiende por autoevaluación (en el contexto de programas académicos) al proceso en el cual se hace una permanente verificación, diagnóstico, análisis, acción y retroalimentación que permiten identificar las fortalezas y debilidades que tienen los programas, buscando siempre el mejoramiento continuo de estos. Actualmente en el mercado se pueden identificar diferentes sistemas de información que facilitan todos estos procesos que se requieren para la autoevaluación de los programas académicos, los cuales son muy útiles debido a que esta tarea devenga de manejo y análisis de grandes cantidades de datos, que difícilmente se podrían lograr sin ayuda de estos.

Un ejemplo de un sistema de autoevaluación que se encuentra es el de la Universidad Nacional de Colombia que se puede encontrar en el siguiente enlace: <http://www.autoevaluacion.unal.edu.co/>. En este, se tienen diferentes menús en los cuales se muestra la documentación de cómo se va a llevar a cabo el proceso con diversa información que compete a la autoevaluación de los programas, y por último se encuentra los formatos de recolección de datos y un informe final. Este sistema se centra en presentar un conjunto de herramientas que le permitirán a los usuarios/actores del programa enterarse de este proceso y participar en cierta medida en él pero no hace parte activa del proceso, que es lo que se busca en el nuestro.

3.2 ARQUITECTURAS EJB (ENTERPRISE JAVA BEAN)

Cuando se construye una aplicación, son muchas las responsabilidades que se deben tener en cuenta, como por ejemplo, la seguridad, las transacciones, el control de concurrencia, entre otras. El estándar EJB permite centrarse en el código de la lógica de negocio del problema que se desea solucionar y deja el resto de responsabilidades al contenedor de aplicaciones donde se ejecutará la aplicación.

El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. Además, es posible reutilizar un mayor nivel de

funcionalidades e incluso es posible modificar estas funcionalidades y adaptarlas a cada entorno de trabajo particular sin tocar el código del componente desarrollado.

La arquitectura EJB se encuentra definida en un modelo de capas (véase Figura 1), siendo la más externa la interfaz con el cliente (Client Machine), la intermedia la de la lógica del negocio (Java EE Server) y la más profunda la del acceso a datos (Database Server). De forma sencilla, un cliente hace una petición de búsqueda del EJB y una vez se ha hecho la petición, el contenedor de EJB alojado en el servidor de aplicaciones busca el nombre del EJB a través de la organización de los EJB 's mediante JNDI (Java Naming and Directory Interface). Después de encontrado se le va devolviendo a las capas superiores la interfaz remota del EJB con la que interactuará el cliente, invocando los métodos que considere oportunos. Así, las distintas peticiones van pasando por las capas expuestas.

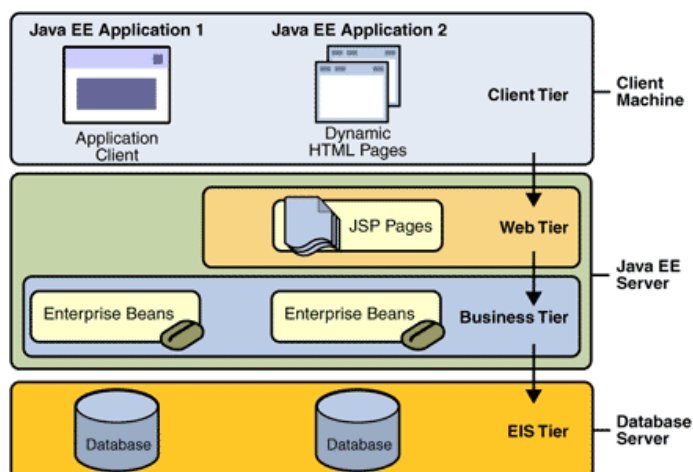


Figura 1. Arquitectura de los EJB

3.3 MONGODB

Es un sistema de base de datos no relacional orientado a documentos, está licenciado como GNU AGPL 3.0, es decir, es un software de licencia libre. Funciona en sistemas operativos Windows, Linux, Mac OS X y Solaris.

MongoDB se destaca por su gran velocidad y por su sencillo sistema de consulta de los contenidos de la base de datos. Alcanza un balance perfecto entre rendimiento y funcionalidad, incorporando muchos de los tipos de consulta que se utilizan en un sistema relacional común, pero sin sacrificar rendimiento.

3.4 LIBRERÍAS UTILIZADAS

4.4.1 Morphia

Es una biblioteca ligera para el mapeo de objetos Java a/desde MongoDB. Las características que más se destacan de Morphia son que proporciona seguridad en el manejo de los datos, y el apoyo de una interfaz fluida de consulta con validación en tiempo de ejecución. Morphia utiliza anotaciones para que no haya archivos XML para gestionar o actualizar. El ambiente que usa Morphia es muy apropiado para desarrolladores con experiencia en Java Persistence API (JPA).

4.4.2 Java Server Faces (JSF)

Es una herramienta de trabajo que facilita el desarrollo de interfaces de usuario del lado del servidor para aplicaciones Java basadas en web. JSF usa la tecnología Java Server Pages (JSP) la cual permite hacer el despliegue de las páginas. También usa el patrón de arquitectura Modelo Vista Controlador (MVC).

JSF presenta las siguientes principales características:

- Un conjunto de APIs para manejar eventos, validar entradas, navegación de las páginas y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario.
- Administración de estados.

4.4.3 PrimeFaces

Es una biblioteca de componentes de código abierto para Java Server Faces, desarrollada por Prime Technology. Proporciona una colección de componentes en su mayoría visuales (widgets) entre los cuales se encuentra un Editor de HTML, autocompletar, graficas o paneles, entre otros. Una de las ventajas de utilizar PrimeFaces, es que en cuanto a la experiencia de los usuarios finales los componentes son amigables al usuario y cuentan con un diseño innovador.

4.4.4 HTML5 Y XHTML5

Hypertext Markup Language, versión 5 abreviado HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Su objetivo fundamental ha sido mejorar el idioma con soporte para los archivos multimedia mientras continua siendo fácilmente legible por las personas y consistentemente entendido por los ordenadores y dispositivos (navegadores web).

XHTML5, por sus siglas del inglés eXtensible Hypertext Markup Language versión 5, XHTML5 es simplemente una extensión de HTML5 estandarizado (un ejemplo de esto es que no puede tener etiquetas sin cerrar), esto lo hace más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores, entre otros.

4.4.5 GitLab

Es una aplicación gratuita para la administración de repositorios de Git, desarrollada en Ruby on Rails. Es ideal para los desarrolladores que quieren mantener su código privado. GitLab ofrece a los usuarios poder administrar; usuarios, repositorios, gestionar ramas, permisos, ver los commits, ver el código, mostrar estadísticas, entre muchas más características.

Esta herramienta juega un papel muy importante en el desarrollo del proyecto debido a la facilidad que brinda de trabajar remotamente, mediante ramas o (branch) de desarrollo permiten una integración de los demás proyectos que hacen parte del conjunto de autoevaluación de programas de postgrado y mantener copias de seguridad alojadas en la nube en caso de presentarse cualquier eventualidad.

4. METODOLOGÍA

La metodología para el desarrollo del componente software está estructurada en el método de programación **UWE** (UML-based Web Engineering), el cual es un procedimiento de ingeniería web orientado a objetos basado en UML, que se utiliza específicamente para aplicaciones web.

Se seleccionó esta metodología porque cubre todo el ciclo de vida de las aplicaciones Web y su proceso de desarrollo se basa en tres fases principales: la fase de captura de requisitos, la fase de análisis y diseño, y la fase de implementación. Para crear modelos de UWE para aplicaciones Web se puede utilizar cualquier herramienta CASE que soporte extensiones UML, lo cual brinda mayor facilidad.

El meta modelo de UWE consta de seis paquetes:

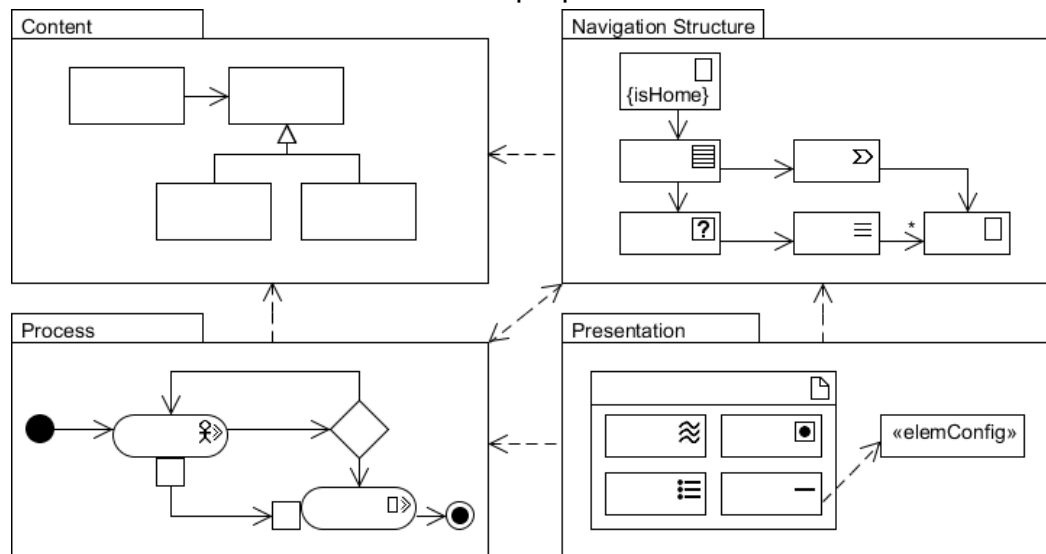


Figura 2. Descripción general de los modelos UWE

- El paquete de 'Requerimientos', que comprende la extensión de UWE, usa Casos de Uso para capturar los requisitos del sistema.
- Los paquetes de 'Navegación' (modelo de usuario, que incluye modelos estáticos y dinámicos) y 'Presentación' (modelo de estructura de presentación y modelo de flujo de presentación) corresponden a los mismos modelos que usa UML.

- Los paquetes de 'Contenido' (modelo del dominio) y 'Procesos' (modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto) actualmente se utilizan para mostrar que UWE permite al diseñador desarrollar modelos de contenido y proceso usando todas las características de UML.
- El paquete de 'Adaptabilidad' representa la característica más importante de las aplicaciones Web.

5. ANÁLISIS, REQUERIMIENTOS Y CASOS DE USO DEL COMPONENTE

5.1 ESPECIFICACIÓN DE REQUISITOS SOFTWARE

Especificación de Requisitos Software (ERS) del componente web para el registro y análisis de la información de usuarios en un sistema de información para la autoevaluación de programas de posgrado en universidades públicas. Ha sido elaborado en colaboración con la Doctora Sonia Cristina Gamboa Sarmiento PhD en Educación, Filosofía y Enseñanza de la filosofía y el Magíster de Ingeniería de Sistemas Carlos Humberto Carreño Díaz.

Este documento se ha realizado en base a las directrices dadas por el estándar “IEEE Recommended Practice for Software Requirement Specifications ANSI/IEEE 830 1998”.

5.2 PROPÓSITO

El objeto de la ERS es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. El documento va dirigido al equipo de desarrollo de software, la dirección del departamento de Maestría en Ingeniería de sistemas de la Universidad Industrial de Santander y los usuarios finales del sistema.

5.3 ALCANCE DEL SISTEMA

Actualmente, el departamento de Postgrado en Ingeniería de Sistemas de la Universidad Industrial de Santander no posee un sistema de información propio, por lo tanto requieren tener visibilidad en la web con información actualizada acerca de sus propias características. Asimismo, se requiere mantener registro de la información concerniente tanto a sus actores como a los procesos académicos. Para ello es necesario disponer de ambientes virtuales en los cuales los actores de estos programas puedan interactuar entre ellos mismos, y llevar a cabo procesos académicos.

El componente web para el registro y análisis de la información de usuarios en un sistema de información para la autoevaluación de programas de posgrado en

universidades públicas, en adelante AutoEvalForms, tiene como objetivo principal facilitar la realización de ciertas tareas de la Escuela de Posgrado, en concreto las siguientes:

- Gestión de diferentes personas que usan el portal.
- Gestión de la propiedad intelectual de los usuarios.
- Administrar algunos parámetros del componente para su previa modificación.
- Proceso de información de los usuarios para ayudar al proceso de la autoevaluación de los programas de posgrado.

AutoEvalForms no realizará ninguna de las siguientes tareas:

- Conexión con el sistema de información interno de la Universidad Industrial de Santander, lo cual es responsabilidad del componente de conexión con base de datos institucional.
- Gestión de notas, lo cual es responsabilidad de la Universidad.
- División jerárquica por roles de acceso del portal.

5.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

AUTOEVALFORMS: Nombre clave dado al proyecto del desarrollo del componente web para el registro y análisis de la información de usuarios en un sistema de información para la autoevaluación de programas de posgrado en universidades públicas.

HTTP: Hypertext Transfer Protocol.

USUARIOS DEL PROGRAMA: Aquellas personas que pueden acceder a los servicios del programa de posgrado. Son: Interesados, Alumnos, Egresados, Profesores, Calificadores y Directivos.

JSF: Java Server Faces.

RF: Requisitos Funcionales.

RNF: Requisitos No Funcionales.

AUTOEVALUACIÓN: La Autoevaluación es un proceso permanente de verificación y diagnóstico, que realizan las instituciones de educación superior, a nivel interno con el fin de identificar sus fortalezas y debilidades.

CASOS DE USO: Descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

5.5 PERSPECTIVA DEL PRODUCTO

AutoEvalForms interaccionará con algunos productos software, estos productos serán adquiridos y otros serán desarrollados. Dichos productos serán:

- Un servidor de WEB, que permitirá el acceso de AutoEvalForms a los servicios del sistema.
- Base de datos. La única exigencia será la utilización de una Base de Datos NO Relacional. Por lo tanto, se utilizará MongoDB para el manejo de los datos.
- Algunos servicios web que se encargaran de algunas tareas del sistema, dentro de los que se encuentran, el servicio de extracción de datos del sistema de la universidad, y el servicio que buscara y recopilara información de hojas de vidas de otros sitios externos.

5.6 FUNCIONES DEL PRODUCTO

6.6.1 Actualización de información de usuarios del programa

La información de los usuarios es la fuente fundamental de los programas de postgrado. Por ello, será necesario realizar los respectivos formularios que permitan la actualización o captación de la información de los usuarios ya registrados por el sistema de información de la Universidad.

6.6.2 Gestión de propiedad intelectual

AutoEvalForms permitirá a los usuarios agregar / actualizar la información de su propiedad intelectual tales como publicaciones, congresos asistidos entre otros.

6.6.3 Administración de parámetros

Dentro del complemento los usuarios podrán modificar opciones del sistema sin que tengan que consultar al experto ni entrar en el código fuente del componente.

6.6.4 Proceso de información de los usuarios para ayudar al proceso de la autoevaluación de los programas de posgrado.

Se realizarán diversos formularios con la finalidad de actualizar la información de todos los usuarios y agregar un módulo que permita plasmar reportes de modo grafico que darán soporte al proceso de autoevaluación del programa.

5.7 CARACTERÍSTICAS DEL USUARIO

Principalmente se trabajarán con dos tipos de usuarios: coordinador y operador administrativo. Cada uno de estos usuarios, por formación y desarrollo, pueden tener una habitualidad muy distinta para con los sistemas software. Por ello, AutoEvalForms deberá ofrecer una interfaz de usuario intuitivo, fácil de aprender y sencillo de manejar.

5.8 RESTRICCIONES

- AutoEvalForms deberá proporcionar exclusivamente una interfaz WEB, lo cual facilitará su utilización en múltiples plataformas, a menor coste.
- La disponibilidad de AutoEvalForms deberá ser absoluta, excepto en el caso de cortes en la red de comunicaciones.
- Lenguaje y tecnologías en uso JAVA (JSF).
- Se desarrollará en NetBeans IDE.
- Se usará una base de datos no relacional: MongoDB.
- Lenguaje de marcado XHTML.
- Framework web PrimeFaces.

5.9 SUPOSICIONES Y DEPENDENCIAS

- Se asume que habrá una conexión a Internet en los equipos donde se desee acceder al sistema.
- La carga de los datos deberá ser realizada por la Universidad. Los datos mínimos que la UIS deberá suministrar algunos datos acerca de las personas matriculadas en el programa con sus respectivos datos personales.
- Los usuarios mencionados anteriormente tendrán acceso al componente AutoEvalForms por medio del portal de autoevaluación de postgrados.

5.10 REQUISITOS FUTUROS

Versión móvil para Smartphone y Tablet.

5.11 REQUERIMIENTOS FUNCIONALES

Identificación	RF01
Nombre	Consultar Información de Usuarios
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad de consultar los datos de un usuario, como por ejemplo, se podrá consultar información básica como dirección, e-mail, idiomas que maneja, estudios realizados, entre otros. El usuario, mediante una interfaz gráfica, realizará la búsqueda (por nombre, por apellidos, No. de Documento) y obtendrá una lista con los resultados y elegirá aquel que quiera consultar detalladamente.

Tabla 1. Requerimiento Funcional 01 – Consultar Información de Usuarios

Identificación	RF02
Nombre	Actualizar Usuario
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad de modificar la respectiva información de cada usuario. Mediante una interfaz gráfica se mostrarán los datos correspondientes a cada usuario. Este deberá completar o modificar de forma correcta e íntegra cada campo de información que se le pida. Una vez hecho esto, apretará el botón para guardar la información y se validará que todos los campos sean correctos.

Tabla 2. Requerimiento Funcional 02 – Actualizar Usuario

Identificación	RF03
Nombre	Consultar Propiedad Intelectual
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad de consultar en cualquier momento toda la propiedad intelectual que el usuario posea.

Tabla 3. Requerimiento Funcional 03 – Consultar Propiedad Intelectual

Identificación	RF04
Nombre	Modificar Propiedad intelectual
Prioridad	Alta

Descripción	Este requisito hace referencia a la posibilidad de editar los diferentes aspectos de la propiedad intelectual como nombre del tema, objetivos, entre otros. El usuario, mediante una interfaz gráfica, obtendrá una lista con todos los aspectos de sus propiedades y elegirá aquel que quiera modificar, deberá completar de forma correcta e íntegra cada campo de información que desee cambiar. Una vez hecho esto, apretará el botón para realizar la modificación y se validará que todos los campos sean correctos.
-------------	--

Tabla 4. Requerimiento Funcional 04 – Modificar Propiedad Intelectual

Identificación	RF05
Nombre	Agregar Propiedad Intelectual
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad de agregar un nuevo elemento de la propiedad intelectual al autor. Mediante una interfaz gráfica se pedirán los datos correspondientes a cada entrada. Este deberá completar de forma correcta e íntegra cada campo de información que se le pida. Una vez hecho esto, apretará el botón para enviar y se validará que todos los campos sean correctos.

Tabla 5. Requerimiento Funcional 05 – Agregar Propiedad Intelectual

Identificación	RF06
Nombre	Eliminar Propiedad Intelectual
Prioridad	Alta
Descripción	Este requisito se refiere a poder eliminar diferentes propiedades intelectuales existentes en el currículum de los usuarios. Este, mediante una interfaz gráfica, obtendrá una lista de todas las publicaciones y elegirá aquella que quiera suprimir.

Tabla 6. Requerimiento Funcional 06 – Eliminar Propiedad Intelectual

Identificación	RF07
Nombre	Consultar Parámetros
Prioridad	Alta

Descripción	Este requisito hace referencia a la posibilidad de consultar las diferentes características del portal Web que están parametrizadas dentro de las que se encuentra los títulos, propiedades de acceso a la base de datos y temas del portal. El usuario podrá visualizar que valores tienen los parámetros actuales.
-------------	--

Tabla 7. Requerimiento Funcional 07 – Consultar Parámetros

Identificación	RF08
Nombre	Modificar Parámetros
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad de modificar las diferentes características parametrizadas del sistema, como por ejemplo, se podrán cambiar los títulos de cada página del portal, entre otras. El usuario podrá hacer esto por medio del componente Web sin necesidad de ingresar al código del mismo. El usuario, mediante una interfaz gráfica, obtendrá una lista con todas las características parametrizadas del sistema y elegirá aquel que quiera modificar. Una vez hecho esto, apretará el botón para realizar la modificación.

Tabla 8. Requerimiento Funcional 08 – Modificar Parámetros

Identificación	RF09
Nombre	Generar Reportes
Prioridad	Alta
Descripción	Desde el componente se tendrá disponible un módulo de reportes estadísticos los cuales deben ser gráficos y permitirán plasmar diferentes consultas, entre las cuales se encuentra la información estadística de los géneros que están matriculados en el programa de maestría, el número de distinciones de profesores que han recibido en los últimos cinco años y el porcentaje de estudiantes admitidos que cumplen con el nivel de dominio mínimo de una lengua, los cuales servirán para el apoyo de la autoevaluación del programa. Dentro del módulo ya se encontraran algunas consultas de acuerdo a la información recopilada de los usuarios del sistema.

Tabla 9. Requerimiento Funcional 09 – Generar Reportes

5.12 REQUERIMIENTOS NO FUNCIONALES

Identificación	RNF01
Nombre	Interfaz del Sistema
Características	Se presentará una interfaz sencilla para que sea de fácil manejo para cualquier tipo de usuario
Prioridad	Alta
Descripción	Tendrá una interfaz de uso sencillo e intuitivo.

Tabla 10. Requerimiento No Funcional 01 – Interfaz del Sistema

Identificación	RNF02
Nombre	Mantenimiento del Sistema
Características	El componente debe contar con un manual de documentación
Prioridad	Alta
Descripción	Para facilitar el mantenimiento del sitio se debe realizar una documentación clara y concisa acerca de cada módulo desarrollado. Además debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante su operación.

Tabla 11. Requerimiento No Funcional 02 – Mantenimiento del Sistema

Identificación	RNF03
Nombre	Confiabilidad del Sistema
Características	El sistema tendrá que estar siempre en funcionamiento, ya que está diseñado para la carga de datos y consulta de los mismos.
Prioridad	Alta
Descripción	La disponibilidad tiene que ser continua para el servicio de los usuarios, garantizando el servicio adecuado que ante cualquier falla en alguno de sus componentes contar con un plan de contingencia.

Tabla 12. Requerimiento No Funcional 03 – Confiabilidad del Sistema

Identificación	RNF04
Nombre	Interoperabilidad
Características	El sistema debe permitir la conexión externa con otro sistema
Prioridad	Alta

Descripción	La interoperabilidad debe notarse en el momento en el que se extraigan los datos del sistema de la universidad y se agreguen a la estructura de datos propia.
-------------	---

Tabla 13. Requerimiento No Funcional 04 - Interoperabilidad

Identificación	RNF05
Nombre	Facilidad para hacer pruebas
Características	El sistema debe contar con facilidades para la identificación de la localización de los errores durante la etapa de pruebas y de operación posterior
Prioridad	Alta
Descripción	Este componente debe contar con métodos apropiados para identificar, prevenir y corregir problemas de datos mediante el uso de componentes automatizados como JUnit.

Tabla 14. Requerimiento No Funcional 05 – Facilidad para hacer pruebas

Identificación	RNF06
Nombre	Validación de Datos
Características	El sistema debe poseer métodos para la validación de sus datos
Prioridad	Alta
Descripción	El sistema valida automáticamente la información contenida en los formularios de ingreso. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad de campos, longitud de caracteres permitida por campo, manejo de tipos de datos, etc.

Tabla 15. Requerimiento No Funcional 06 – Validación de Datos

Identificación	RNF07
Nombre	Arquitectura
Características	La solución debe ser desarrollada en la web y toda la administración debe realizarse desde un navegador
Prioridad	Alta
Descripción	La solución debe tener interfaces gráficas de administración y de operación en ambiente Web, para permitir su utilización a través de exploradores o navegadores de Internet.

Tabla 16. Requerimiento No Funcional 07 - Arquitectura

Identificación	RNF08
Nombre	Ingreso de Datos
Características	Los datos que se deben ingresar a la estructura de datos deben ser de manera controlada para no saturar la red
Prioridad	Alta
Descripción	El ingreso de datos al sistema se debe hacerse por medio de transacciones que permitan el ingreso de los datos de forma parcial; es decir, que el tamaño de las páginas de los formularios sea el adecuado.

Tabla 17. Requerimiento No Funcional 08 – Ingreso de Datos

5.13 ACTORES DEL SISTEMA

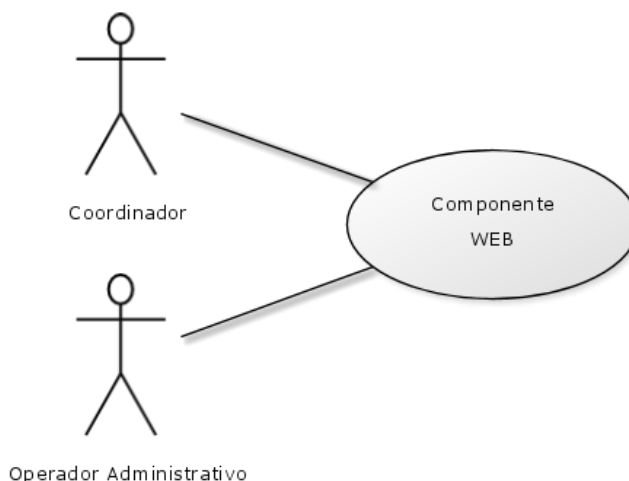


Figura 3. Actores del Sistema

5.14 DESCRIPCIÓN DE ACTORES

Actor	Coordinador
Caso de Uso	Ver y Generar reportes
Descripción	Usuario que tiene la posibilidad de ver y generar diferentes reportes relacionados con la autoevaluación del programa de maestría.
Tipo	Principal

Tabla 18. Descripción de Actores - Coordinador

Actor	Operador Administrador
Caso de Uso	Ver y Generar reportes, Gestionar personas, Gestionar propiedad intelectual, Gestionar parámetros
Descripción	Usuario que tiene el control de acceso a todo el componente web pudiendo realizar todas las operaciones de gestión de personas, propiedad intelectual, parámetros y reportes.
Tipo	Principal

Tabla 19. Descripción de Actores – Operador Administrativo

5.15 PRINCIPALES CASOS DE USO IDENTIFICADOS

- Gestionar Personas
- Gestionar Propiedad Intelectual
- Administrar Parámetros
- Ver y Generar reportes

5.16 DIAGRAMA GENERAL DE CASOS DE USO Y ACTORES

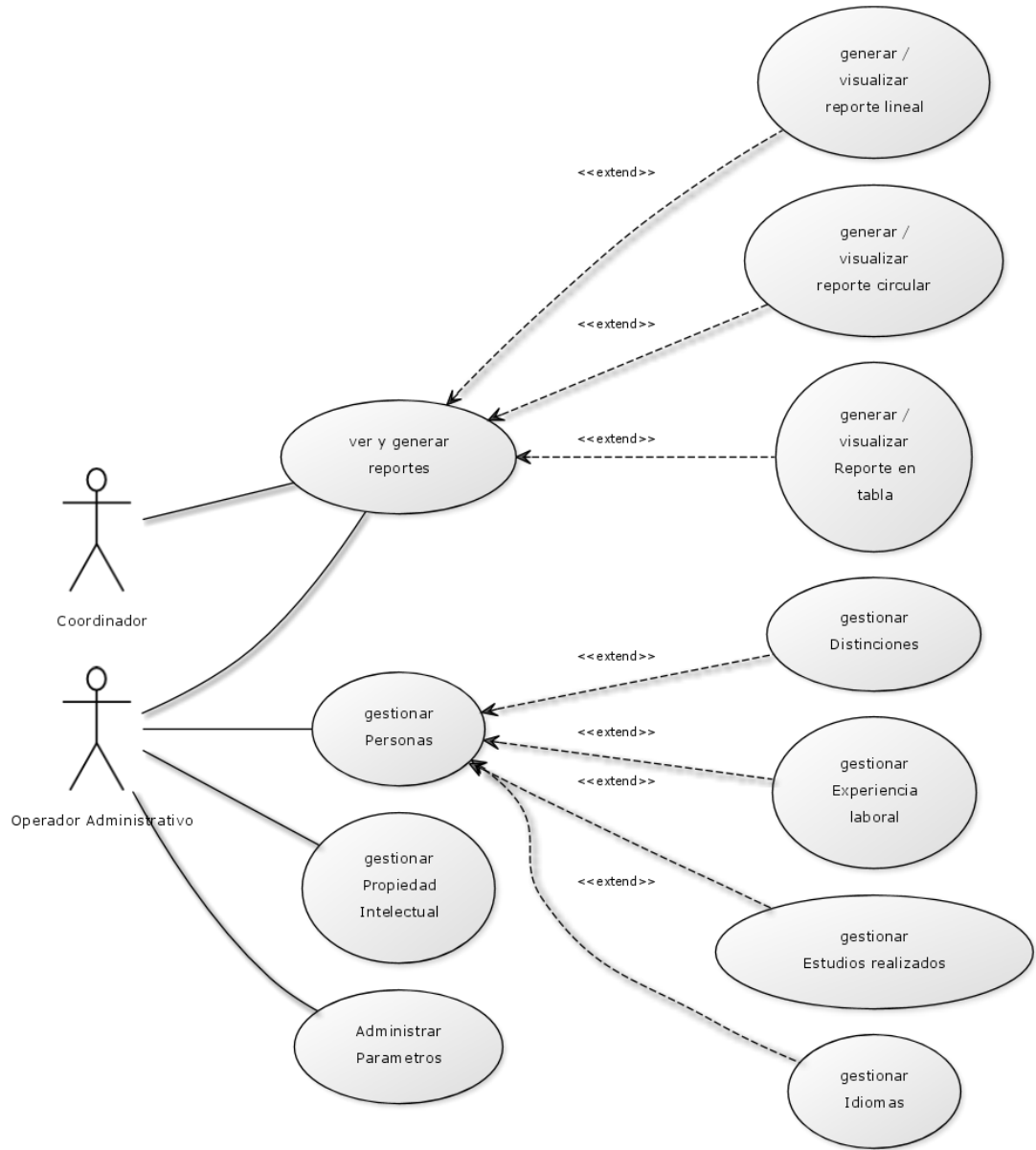


Figura 4. Diagrama general de Casos de Uso y Actores

5.17 DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO POR ACTORES

6.17.1 Operador Administrativo

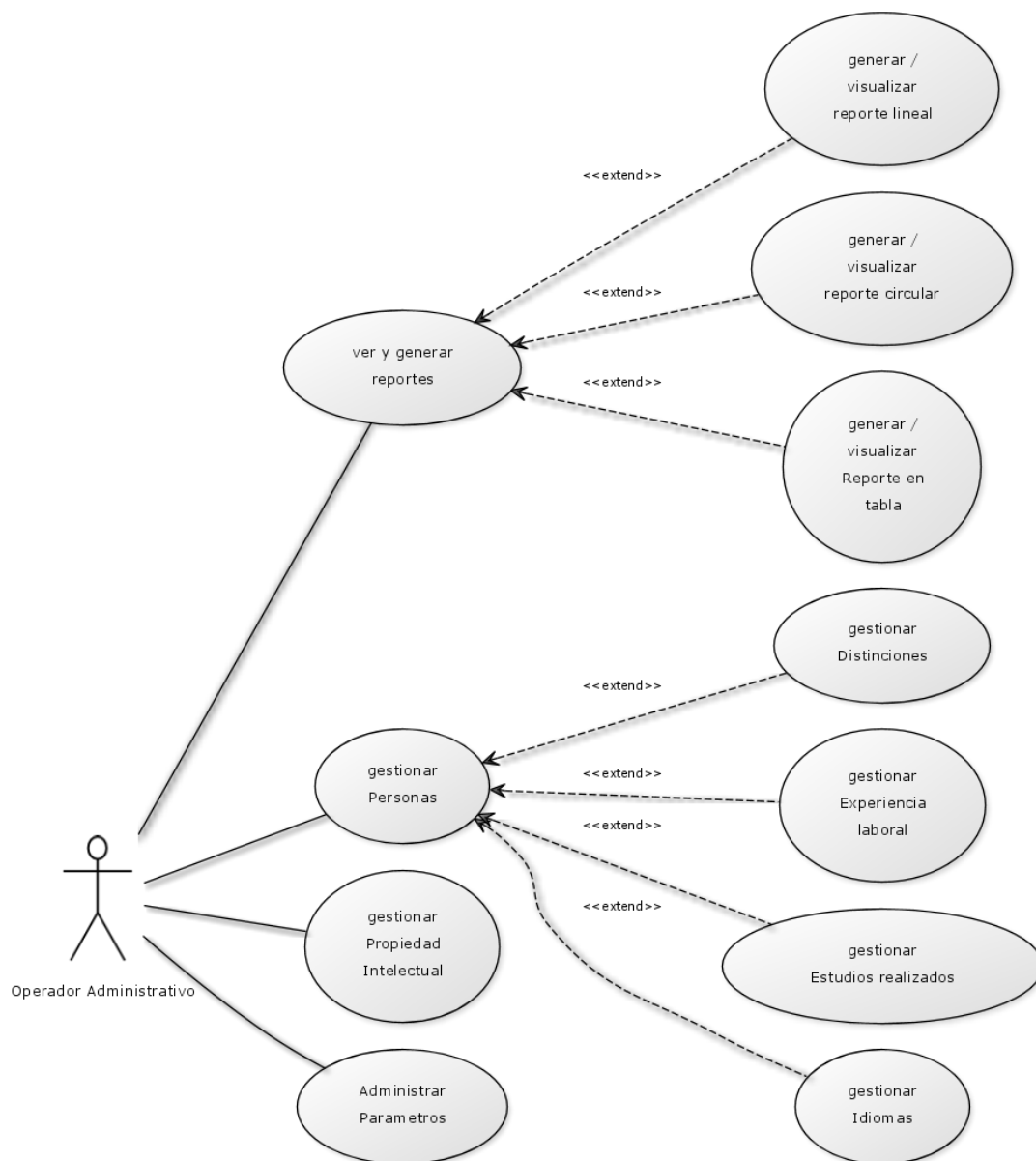


Figura 5. Diagrama de Casos de Uso – Operador Administrativo

6.17.2 Coordinador

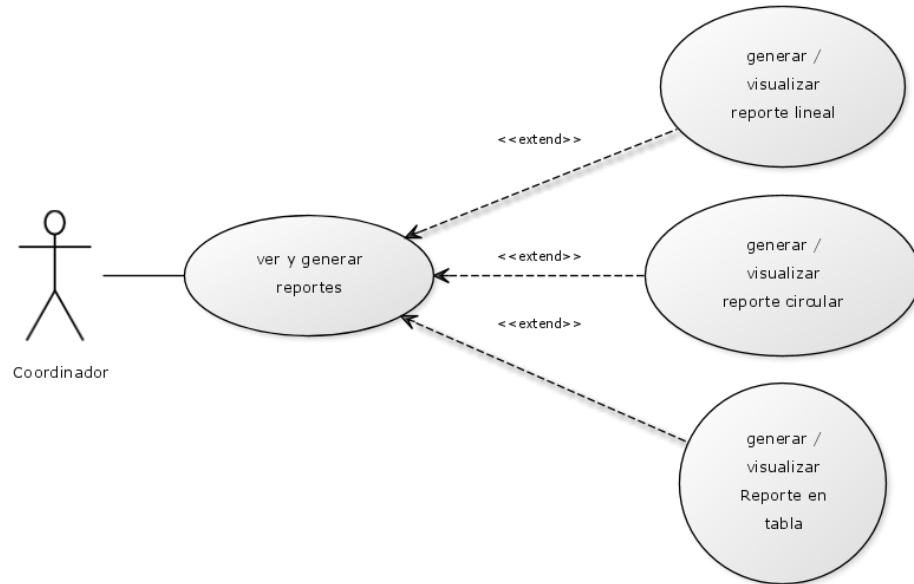


Figura 6. Diagrama de Casos de Uso – Coordinador

5.18 DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO

6.18.1 Gestionar Personas

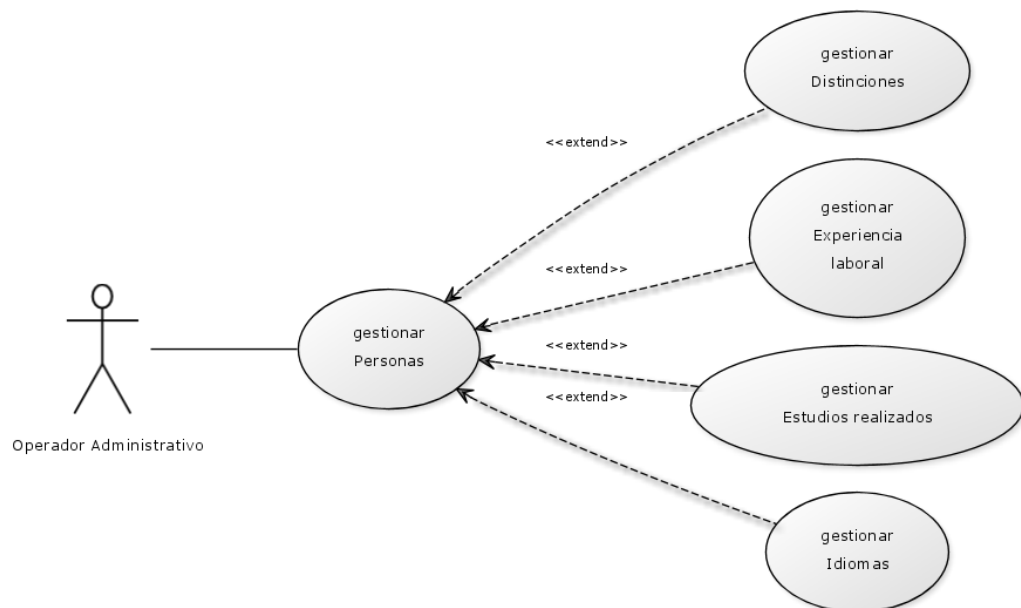


Figura 7. Diagrama de Casos de Uso – Gestionar Personas

Título del Caso de Uso	Consultar Información de Usuarios	ID del Caso de Uso	CU 001
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo cuando seleccione la opción que ofrece el componente web de 'Gestionar Usuario'. Ofrece funcionalidad para consultar la información de determinado usuario.			
Entidades/Actores Involucrados			
Operador Administrativo			
Condiciones de Entrada			
Tener permisos de acceso a la opción 'Gestionar Usuario'.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.	Selecciona la opción 'Gestionar Usuario'.	Muestra una tabla con información básica y resumida de todos los usuarios que están registrados en el sistema. La tabla tendrá los siguientes campos: Nombres, Apellidos, No. de Documento y E-mail. Posee la opción de Filtrar por cada una de estas opciones para facilitar la búsqueda al Operador.	
2.	Selecciona la opción 'Ver/Editar' en determinado usuario.	Muestra un formulario con toda la información del usuario seleccionado. El cual consta de siete pestañas: Tipo, Información Personal, Idiomas, Estudios, Distinciones, Exp. Profesional y Resumen. En la pestaña Tipo se muestran los campos: Tipos de Usuario y Url CvLac. En la pestaña de Información Personal se muestran los siguientes campos: Nombres, Apellidos, Tipo de Documento, No. de Documento, Género, Fecha de Nacimiento, Ciudad de Residencia, País de Origen, Dirección, Estrato, Estado Civil, No. de Hijos, Ingresos Familiares, Vivienda, E-mail y Teléfono de Contacto. En la pestaña de Idiomas se muestra una tabla con los siguientes campos: Nombre, Nivel de Lectura, Nivel de Escritura, Nivel de Escucha,	

	<p>Nivel de Habla y Certificado.</p> <p>En la pestaña Estudios se despliega una tabla con las siguientes columnas: Nivel de Estudio, Título Otorgado, Promedio, Universidad y Trabajo de Grado.</p> <p>En la pestaña Distinciones se observa una tabla con los siguientes campos: Nombre, Motivo, Institución y Fecha.</p> <p>En la pestaña Exp. Profesional se muestra una tabla con las siguientes columnas: Empresa, Cargo, Motivo de Retiro y Rango Salarial.</p> <p>Y en la pestaña Resumen se muestra una breve información de lo digitado por el usuario.</p>
Condiciones de Salida	
Se ha desplegado la información del usuario seleccionado en la pantalla inicial.	
Caminos de Excepción /Flujos alternativos	
Ninguno.	

Tabla 20. Descripción de Casos de Uso – Consultar Información de Usuarios

Título del Caso de Uso	Actualizar Usuario	ID del Caso de Uso	CU 002
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo. Se inicia en el caso de uso CU 001 si el Operador actualiza o modifica la información del usuario. Ofrece funcionalidad para modificar la información de determinado usuario.			
Entidades/Actores Involucrados			
Operador Administrativo			
Condiciones de Entrada			
Se inicia en el caso de uso CU 001. Haber consultado un usuario para realizar la actualización de información.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.	Selecciona y edita el campo que desea modificar o actualizar información según sea necesario.	Muestra los datos modificados.	
2.	En la pestaña Resumen, selecciona la opción Enviar.	Muestra un mensaje de “Operación Exitosa” o de error y redirige a la página principal.	

Condiciones de Salida
Guarda las modificaciones realizadas.
Caminos de Excepción /Flujos alternativos
2.1 Usuario: No selecciona la opción Enviar.
Sistema: No guarda la información modificada.

Tabla 21. Descripción de Casos de Uso – Actualizar Usuario

6.18.2 Gestionar Propiedad Intelectual

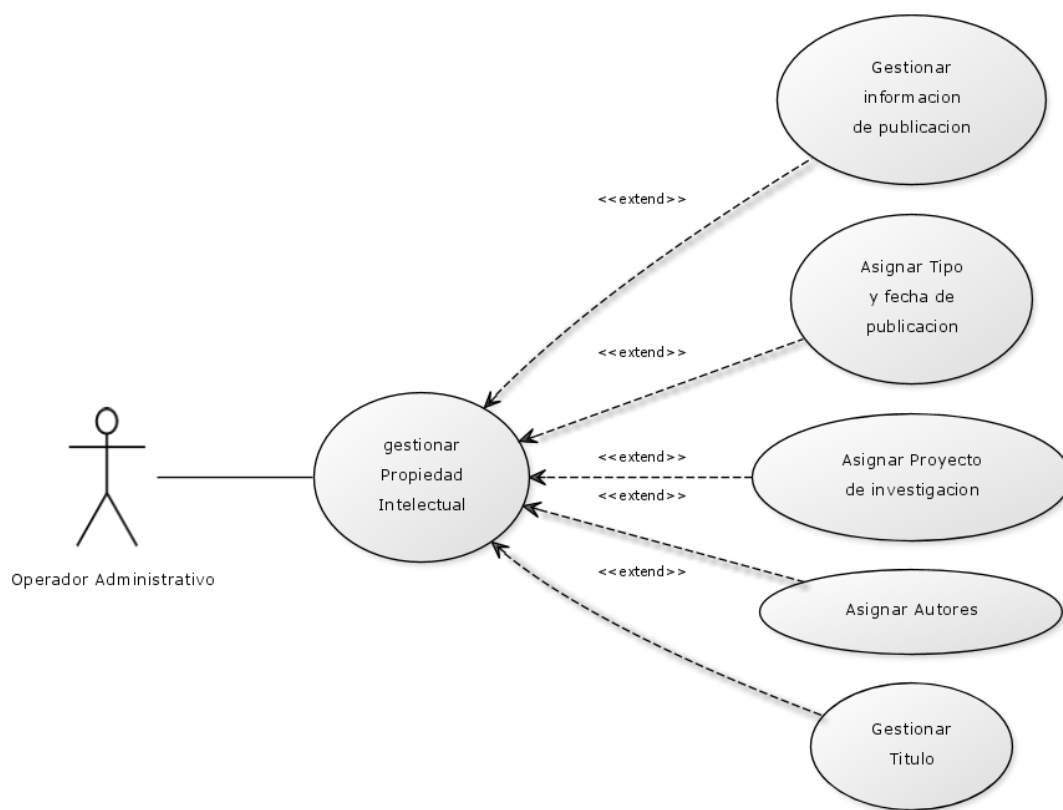


Figura 8. Diagrama de Casos de Uso – Gestionar Propiedad Intelectual

Título del Caso de Uso	Consultar Propiedad Intelectual	ID del Caso de Uso	CU 003
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo cuando seleccione la opción que ofrece el componente web de 'Gestionar Propiedad Intelectual'. Ofrece funcionalidad para consultar determinada propiedad intelectual.			

Entidades/Actores Involucrados		
Operador Administrativo		
Condiciones de Entrada		
Estar autenticado en el sistema. Tener permisos de acceso a la opción 'Gestionar Propiedad Intelectual'.		
Flujo normal de Eventos		
	Operador Administrativo	Sistema
1.	Selecciona la opción 'Gestionar Propiedad Intelectual'.	Muestra una tabla con información básica y resumida de todas las propiedades intelectuales que están registradas en el sistema. La tabla tendrá los siguientes campos: Autores, Tipo de Publicación, Lugar de Publicación y Publicado en. Posee la opción de Filtrar por cada una de estas opciones para facilitar la búsqueda.
2.	Selecciona la opción 'Ver/Editar' en determinada propiedad intelectual.	Muestra un formulario con toda la información de la Propiedad Intelectual seleccionada. El cual consta de cuatro pestañas: Propiedad Intelectual, Proyecto, Descripción y Resumen. En la pestaña Propiedad Intelectual se muestran los campos: Título, Autor(es), Tipo de Publicación, Fecha de Publicación, Palabras Clave, Revista/Libro, Ciudad, Páginas y Formato. En la pestaña de Proyecto se muestra una tabla con las siguientes columnas: Título, Tipo de Proyecto, Problema y Evaluador. En la pestaña Descripción se muestra un Text Área para ingresar un breve resumen de lo que trata la Publicación realizada. Y en la pestaña Resumen se muestra una breve información de lo digitado por el usuario.
Condiciones de Salida		
Se ha desplegado la información de la propiedad intelectual seleccionada en la pantalla inicial.		
Caminos de Excepción /Flujos alternativos		
2.1 Usuario: Selecciona la opción 'Eliminar'.		
Sistema: Se inicia caso de uso CU 006.		

Tabla 22. Descripción de Casos de Uso – Consultar Propiedad Intelectual

Título del Caso de Uso	Modificar Propiedad Intelectual	ID del Caso de Uso	CU 004
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo. Se inicia en el caso de uso CU 003 si el Operador modifica algún aspecto de la propiedad intelectual. Ofrece funcionalidad para modificar determinada propiedad intelectual.			
Entidades/Actores Involucrados			
Operador Administrativo			
Condiciones de Entrada			
Se inicia en el caso de uso CU 003. Haber consultado una propiedad intelectual para realizar la edición.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.	Selecciona y edita el campo del aspecto que desea modificar o actualizar de la propiedad intelectual.	Muestra los datos modificados.	
2.	En la pestaña Resumen, selecciona la opción Enviar.	Muestra un mensaje de “Operación Exitosa” o de error y redirige a la página principal.	
Condiciones de Salida			
Guarda las modificaciones realizadas.			
Caminos de Excepción /Flujos alternativos			
2.2 Usuario: No selecciona la opción Enviar. Sistema: No guarda la información modificada.			

Tabla 23. Descripción de Casos de Uso – Modificar Propiedad Intelectual

Título del Caso de Uso	Agregar Propiedad Intelectual	ID del Caso de Uso	CU 005
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo. Se inicia en el caso de uso CU 003 si el Operador selecciona la opción ‘Agregar Nueva Propiedad Intelectual’. Ofrece funcionalidad para agregar una nueva propiedad intelectual.			
Entidades/Actores Involucrados			
Operador Administrativo			

Condiciones de Entrada		
Se inicia en el caso de uso CU 003. Haber seleccionado la opción 'Agregar Nueva Propiedad Intelectual' para realizar la adición.		
Flujo normal de Eventos		
	Operador Administrativo	Sistema
1.	Llena el formulario que contendrá la información de la nueva Propiedad Intelectual.	Muestra los datos llenados.
2.	En la pestaña Resumen, selecciona la opción Enviar.	Muestra un mensaje de "Operación Exitosa" o de error y redirige a la página principal.
Condiciones de Salida		
Guarda el nuevo registro de la Propiedad Intelectual.		
Caminos de Excepción /Flujos alternativos		
X.1 Usuario: Selecciona opción 'Retroceder' Sistema: Muestra la tabla inicial de información resumida de las Propiedades Intelectuales.		
2.1 Usuario: No selecciona la opción Enviar. Sistema: No guarda la información modificada.		

Tabla 24. Descripción de Casos de Uso – Agregar Propiedad Intelectual

Título del Caso de Uso	Eliminar Propiedad Intelectual	ID del Caso de Uso	CU 006
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo. Se inicia en el caso de uso CU 003 si el Operador selecciona la opción 'Eliminar'. Ofrece funcionalidad para suprimir una nueva propiedad intelectual.			
Entidades/Actores Involucrados			
Operador Administrativo			
Condiciones de Entrada			
Se inicia en el caso de uso CU 003. Haber seleccionado la opción 'Eliminar' en determinada Propiedad Intelectual para realizar la eliminación.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.	Selecciona la opción 'Eliminar' en 'Gestionar Propiedad Intelectual'	Muestra dialogo de confirmación.	

2.	Confirma cuadro de dialogo.	Muestra un mensaje de “Operación Exitosa” o de error y redirige a la página principal.
Condiciones de Salida		
Guarda los cambios realizados.		
Caminos de Excepción /Flujos alternativos		
Ninguno.		

Tabla 25. Descripción de Casos de Uso – Eliminar Propiedad Intelectual

6.18.3 Administrar Parámetros

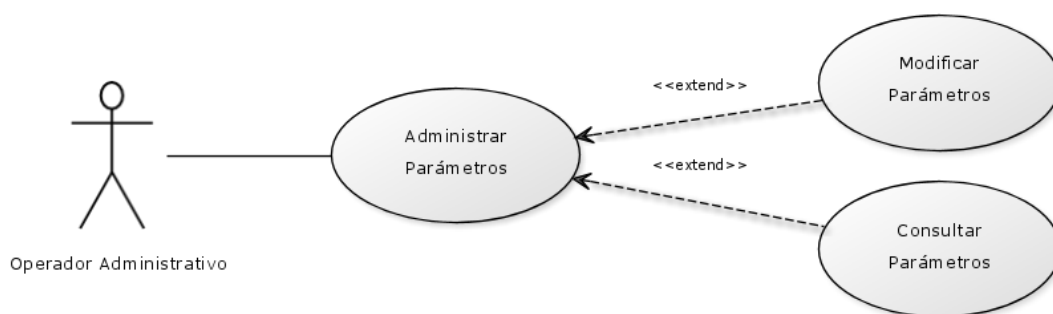


Figura 9. Diagrama de Casos de Uso – Administrar Parámetros

Título del Caso de Uso	Consultar Parámetros	ID del Caso de Uso	CU 007
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo cuando seleccione la opción que ofrece el componente web de ‘Administrar Parámetros’. Ofrece funcionalidad para consultar los parámetros del componente web.			
Entidades/Actores Involucrados			
Operador Administrativo			
Condiciones de Entrada			
Estar autenticado en el sistema. Tener permisos de acceso a la opción ‘Administrar Parámetros’.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.	Selecciona la opción ‘Administrar Parámetros’	Muestra un menú con las opciones parametrizadas del componente web. Las	

	opciones son: Título (donde se podrá cambiar los títulos de cada página del componente), Base de Datos (donde se podrá cambiar el nombre, puerto, ruta, entre otros aspectos de la Base de Datos) y Temas (donde se podrá cambiar todos los estilos del componente).
Condiciones de Salida	
Se ha desplegado un menú con todas las opciones parametrizadas que posee el sistema.	
Caminos de Excepción /Flujos alternativos	
Ninguno.	

Tabla 26. Descripción de Casos de Uso – Consultar Parámetros

Título del Caso de Uso	Modificar Parámetros	ID del Caso de Uso	CU 008
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Operador Administrativo. Se inicia en el caso de uso CU 007 si el Operador selecciona alguna opción del menú en ‘Administrar Parámetros’. Ofrece funcionalidad para modificar determinadas características del sistema.			
Entidades/Actores Involucrados			
Operador Administrativo			
Condiciones de Entrada			
Se inicia en el caso de uso CU 007. Haber seleccionado alguna opción del menú en ‘Administrar Parámetros’.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.1	Selecciona la opción ‘Título’ en el menú lateral.	Genera un formulario tipo acordeón, mostrando todos los títulos de las páginas del portal. Muestra el título actual, un campo para editarlo y un botón ‘Enviar’ para guardar cambios.	
1.2	Selecciona la opción ‘Base de Datos’ en el menú lateral.	Muestra un formulario con toda la información para el acceso a la Base de Datos del Sistema. Se muestran los campos: Nombre de la Base de Datos, Host, Puerto, Nombre de Usuario y Contraseña. Al	

		lado izquierdo, en rojo, se muestran los valores actuales de acceso. Y al final del formulario, el botón 'Enviar' para guardar cambios.
1.3	Selecciona la opción 'Color' en el menú lateral.	Se muestran seis tipos de Temas a escoger. Además, se ofrece la funcionalidad de pre visualizar los cambios generados a la hora de seleccionar cierto tema, mediante un formulario tipo acordeón se pre visualiza el ejemplo de una Tabla y el ejemplo de un Formulario. Por ultimo muestra una lista para que seleccione el tema y el botón 'Enviar' para guardar el tema seleccionado.
Condiciones de Salida		
Guarda las modificaciones realizadas.		
Caminos de Excepción /Flujos alternativos		
Ninguno.		

Tabla 27. Descripción de Casos de Uso – Modificar Parámetros

6.18.4 Ver y Generar Reportes

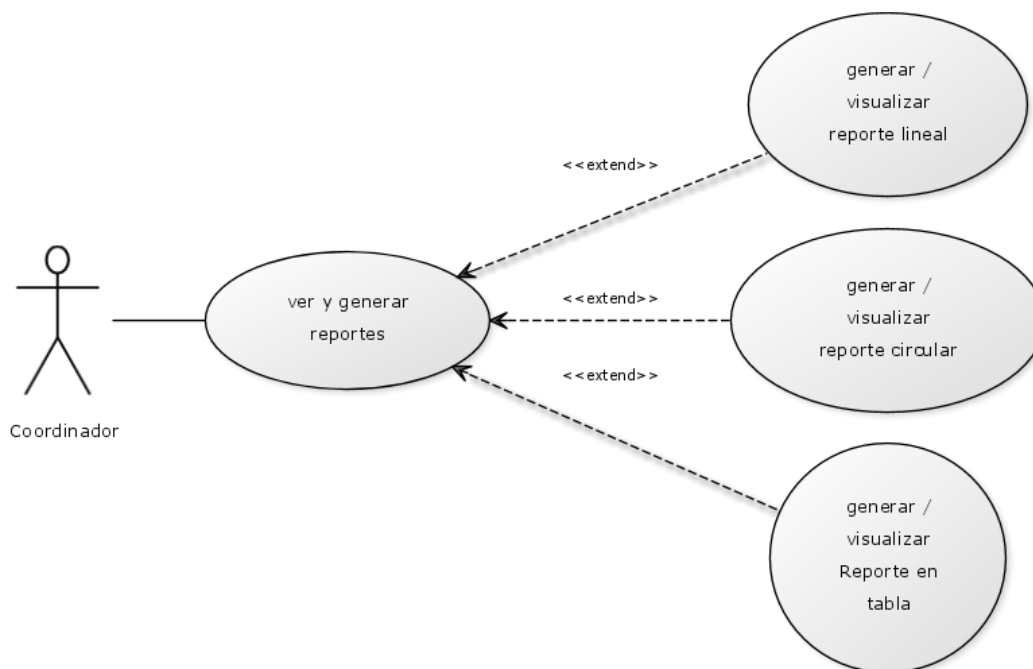


Figura 10. Diagrama de Casos de Uso – Ver y Generar Reportes

Título del Caso de Uso	Ver y Generar reportes	ID del Caso de Uso	CU 009
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado por el Coordinador y/o el Operador Administrativo cuando seleccione la opción que ofrece el componente web de 'Generar Reportes'. Ofrece funcionalidad para consultar y generar reportes.			
Entidades/Actores Involucrados			
Coordinador y Operador Administrativo			
Condiciones de Entrada			
Tener permisos de acceso a la opción 'Generar Reportes'.			
Flujo normal de Eventos			
	Operador Administrativo	Sistema	
1.	Selecciona la opción 'Generar Reportes'	Muestra un menú con los diferentes reportes que posee el componente web. Las opciones son: Idiomas, B. Distinciones, L. Distinciones y Géneros.	
2.1	Selecciona la opción 'Idiomas' en el menú lateral.	Se muestra una gráfica tipo Pie o Torta, la cual posee la distribución de porcentaje de estudiantes admitidos que cumplen con el nivel de dominio mínimo de una lengua extranjera establecido por el programa.	
2.2	Selecciona la opción 'B. Distinciones' en el menú lateral.	Se muestra una gráfica tipo Barra, la cual posee la cantidad de distinciones de profesores que han recibido en los últimos cinco años.	
2.3	Selecciona la opción 'L. Distinciones' en el menú lateral.	Se muestra una gráfica tipo Lineal, la cual posee la cantidad de distinciones de profesores que han recibido en los últimos cinco años.	
2.4	Selecciona la opción 'Géneros' en el menú lateral.	Se muestra una gráfica tipo Pie o Torta, la cual posee la distribución de porcentaje de los géneros de estudiantes que están matriculados en el programa de maestría.	
Condiciones de Salida			
Muestra en pantalla el reporte consultado o generado.			
Caminos de Excepción /Flujos alternativos			
Ninguno.			

Tabla 28. Descripción de Casos de Uso – Ver y Generar Reportes

6. DISEÑO DE OBJETOS, ESTRUCTURAS DE DATOS Y COMPONENTES

Una vez terminada la fase de análisis de requisitos se procede a establecer los patrones de diseño a utilizar, definir la estructura de datos, pantallas del sistema entre otros aspectos.

6.1 DISEÑO DE LA BASE DE DATOS

La elección de la tecnología de base de datos a utilizar y el diseño de la estructura de la base de datos fue de manera conjunta con los demás integrantes que participan en el proyecto de Autoevaluación de programas de posgrado. El equipo de trabajo consta de 12 estudiantes de pregrado en realización de proyecto de grado 1 y 2, así como estudiantes y egresados a nivel de posgrado.

Para la tecnología a usar se tomó la decisión de usar una base de datos no relacional o (No SQL) la cual presta ciertas ventajas comparándola con una base de datos relacional.

Las principales características que llamaron la atención al grupo sobre usar una base de datos no relacional fueron las siguientes:

- Las Bases de Datos No SQL facilitan la manera en que se pueda gestionar la información en temas como la posibilidad de escalamiento horizontal lo cual implica que al añadir diferentes nodos a la estructura mejorara notablemente el rendimiento y no requiere de procedimientos complicados.
- La velocidad en las consultas pequeñas se ve mejorada respecto a las bases de datos relacionales debido a que la información o datos relacionados se almacenan en un documento lo que implica que no se deben usar operaciones join.
- Esquema dinámico: con esta propiedad las bases de datos No SQL permiten la flexibilidad para evolucionar el esquema de datos.
- Estructura Dinámica: Los datos no tiene una definición de atributos rígida, cada registro puede contener una información con diferente forma pudiendo almacenar solo los atributos que sean de interés.

Para la creación de las diferentes entidades de la base de datos se reunió un grupo de diferentes actores dentro de los que se encontraban estudiantes de pregrado, posgrado, directora de proyecto y administrativos relacionados con la problemática de la autoevaluación en la escuela de ingeniería de sistemas e informática de la UIS.

A partir de eso se determinaron las siguientes entidades iniciales como: **PERSONAS** (que podrían ser administrativos, profesores o estudiantes), **GRUPOS_INVESTIGACIÓN** (las cuales conforman la maestría), **LINEAS_INVESTIGACIÓN**, **PROGRAMA_ACADÉMICO** y **PROPIEDAD_INTELLECTUAL**. Luego de determinar las entidades principales se hizo un proceso de revisión para determinar las principales características de cada una de ellas, donde se dio cuenta que no solamente existían estas entidades sino que existían además unas colecciones embebidas dentro de estas como por ejemplo: **DISTINCIONES**, **ESTUDIOS**, **EXPERIENCIA_PROFESIONAL** e **IDIOMAS**.

A continuación se presenta un esquema de la base de datos con sus respectivas entidades mencionadas anteriormente.

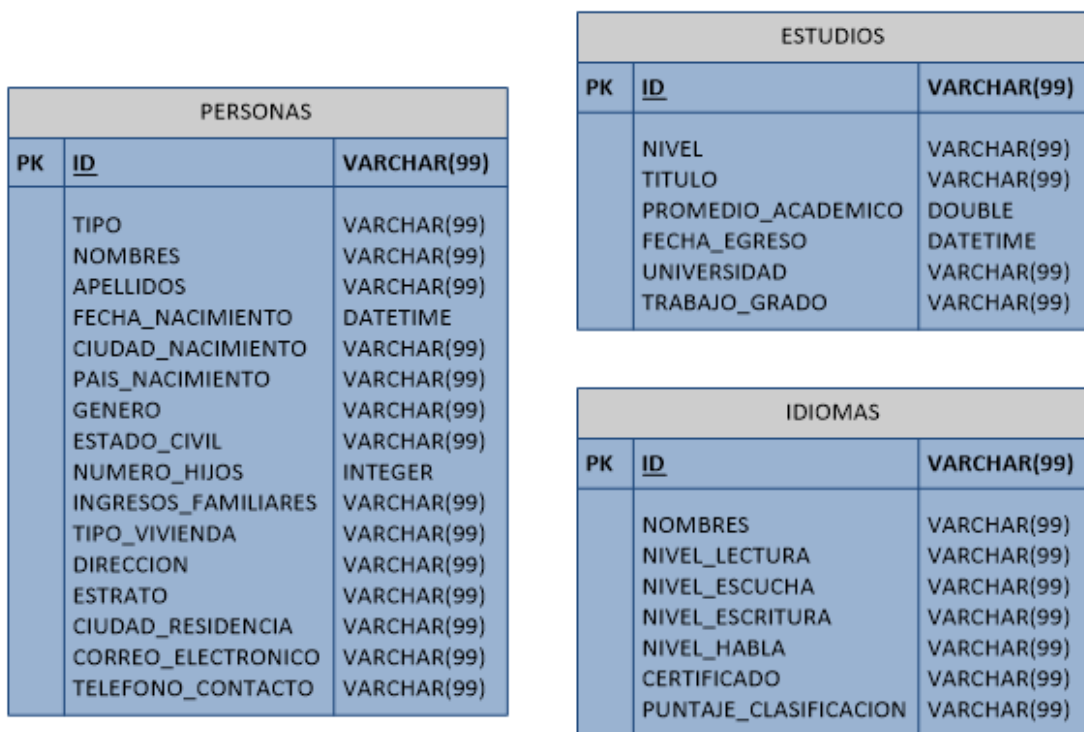


Figura 11. Esquema de la Base de Datos – Tablas Personas, Estudios e Idiomas

GRUPOS_INVESTIGACION		
PK	ID	VARCHAR(99)
	NOMBRE	VARCHAR(99)
	CATEGORIA	VARCHAR(99)
	INSTITUCION	VARCHAR(99)
	DIRECTOR	VARCHAR(99)
	PAGINA_WEB	VARCHAR(99)
	EMAIL	VARCHAR(99)
	AÑO_CREACION	DATETIME
	AREA_CONOCIMIENTO	VARCHAR(99)
	PROGRAMA_NACIONAL	VARCHAR(99)
	LINEAS_INVESTIGACION	VARCHAR(99)
	INTEGRANTES	VARCHAR(99)
	GRUP_LAC	VARCHAR(99)

PROGRAMA_ACADEMICO		
PK	ID	VARCHAR(99)
	NOMBRE	VARCHAR(99)
	OBJETIVO_GENERAL	VARCHAR(99)
	OBJETIVOS_ESPECIFICOS	VARCHAR(99)
	MISION	VARCHAR(99)
	VISION	VARCHAR(99)
	NIVEL_ACADEMICO	VARCHAR(99)
	UNIDAD_ACADEMICA	VARCHAR(99)
	COD_SNIES	INTEGER
	AÑO_INICIO	DATETIME
	TITULO	VARCHAR(99)
	DURACION	INTEGER
	PLAN_ESTUDIOS	VARCHAR(99)
	HORARIO_CLASE	VARCHAR(99)
	NUM_COHORTES	INTEGER
	NUM_EGRESADOS	INTEGER
	CUPOS_SEMESTRE	INTEGER
	VALOR_INSCRIPCION	DOUBLE
	VALOR_MATRICULA	DOUBLE
	PERFIL_ASPIRANTE	VARCHAR(99)
	PERFIL_EGRESADO	VARCHAR(99)
	DIRECCION	VARCHAR(99)
	TELEFONO	DOUBLE
	CORREO_ELECTRONICO	VARCHAR(99)
	HORARIO_ATENCION	VARCHAR(99)
	REGLAMENTO_POSGRADO	VARCHAR(99)
	REGLAMENTO_PROFESORES	VARCHAR(99)
	REGLAMENTO_INVESTIG	VARCHAR(99)
	FORMATOS	VARCHAR(99)

PROYECTOS_INVESTIGACION		
PK	ID	VARCHAR(99)
	AUTORES	VARCHAR(99)
	DIRECTOR	VARCHAR(99)
	CODIRECTOR	VARCHAR(99)
	TITULO	VARCHAR(99)
	PROBLEMA	VARCHAR(99)
	OBJETIVO_GENERAL	VARCHAR(99)
	OBJETIVOS_ESPECIFICOS	VARCHAR(99)
	PALABRAS_CLAVE	VARCHAR(99)
	TIPO_PROYECTO	VARCHAR(99)
	AÑO	DATETIME
	BIBLIOGRAFIA	VARCHAR(99)
	NOMBRE	VARCHAR(99)
	FECHA_ENTREGA	DATETIME
	FECHA_APROBACION	DATETIME
	EVALUADOR	VARCHAR(99)
	LINEAS_INVESTIGACION	VARCHAR(99)
	OBJETIVO_PROGRAMA	VARCHAR(99)

Figura 12. Esquema de la Base de Datos – Tablas Grupos Investigación, Programa Académico y Proyectos Investigación

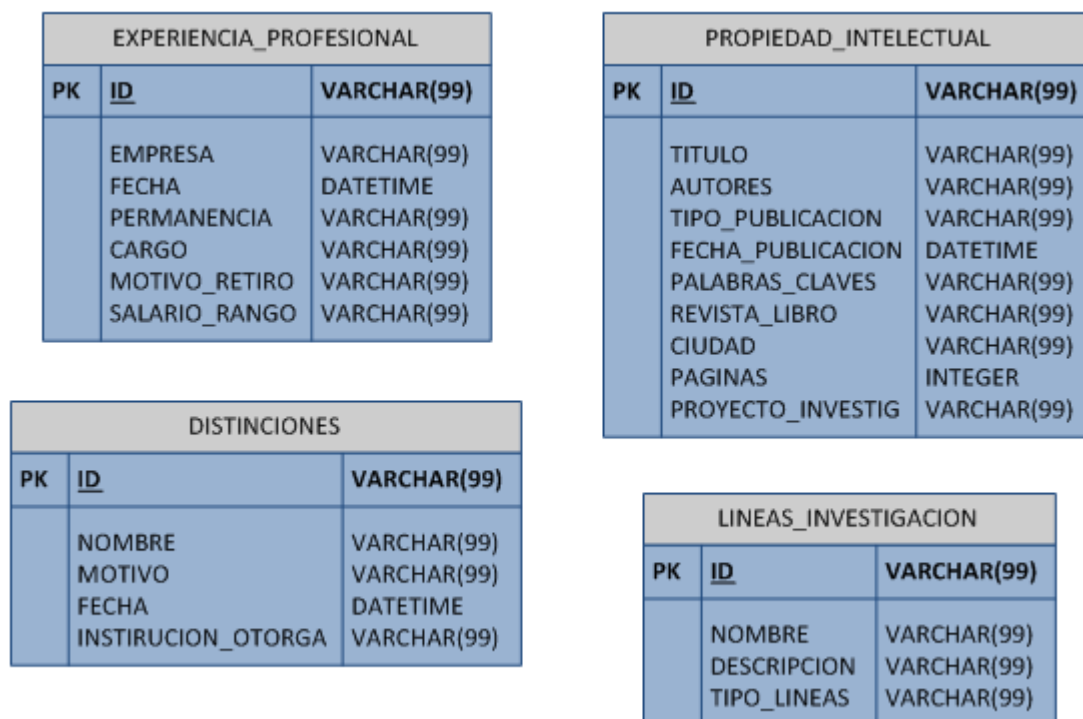


Figura 13. Esquema de la Base de Datos – Tablas Experiencia Profesional, Propiedad Intelectual, Distinciones y Líneas Investigación

Posteriormente se hizo el trabajo de observar las relaciones que tienen entre si y se planteó el siguiente modelo conceptual.

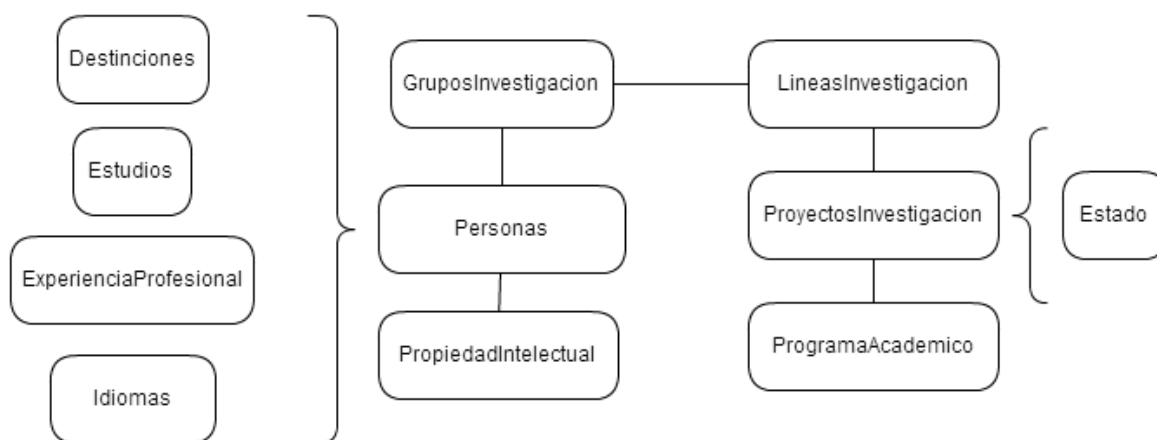


Figura 14. Modelo Conceptual de la Base de Datos

6.2 DISEÑO DE ENTIDADES UTILIZANDO MORPHIA FRAMEWORK

Para la creación de las diferentes clases de las entidades definidas anteriormente se utilizó el framework de Morphia el cual tomaba las entidades @Entity (personas, propiedad intelectual, etc.) y las convierte en una clase la cual contiene atributos que corresponden a los campos de la base de datos previamente definidos, dentro de estos atributos se agregan un id y una versión.

Para facilitar el acceso a diferentes propiedades de las entidades y la reducción del código se creó una clase en común la cual aloja los métodos comunes para todas las entidades llamada BaseEntity lo cual facilita usar las propiedades de la herencia.

```

package filoen.autoevaluacion.model;

import com.google.code.morphia.annotations.Id;
import com.google.code.morphia.annotations.Version;
import org.bson.types.ObjectId;

public class BaseEntity {

    @Id
    protected ObjectId id;

    @Version
    protected long version;

    public ObjectId getId() {
        return id;
    }

    public void setId(ObjectId id) {
        this.id = id;
    }

    public long getVersion() {
        return version;
    }

    public void setVersion(long version) {
        this.version = version;
    }
}

```

Figura 15. Diagrama de BaseEntity

A continuación se podrá observar parte del código utilizando el concepto de Morphia, donde se observan las propiedades de `@Embedded` y de `@Entity`, las cuales sirven para hacer referencias de datos que no necesariamente son obligatorias, ni hacen parte de un documento aparte de la colección debido a que están relacionadas unas a otras.

```

@Entity(value = "personas")
public class Personas extends BaseEntity {

    private String tipo;
    private String nombres;
    private String apellidos;
    private String nombreCompleto;
    private String tipoIdentificacion;
    private String numeroIdentificacion;
    private Date fechaNacimiento;
    private String paisNacimiento;
    private String genero;
    private String estadoCivil;
    private int numeroHijos;
    private String ingresosFamiliares;
    private String tipoVivienda;
    private String direccion;
    private int estrato;
    private String ciudadResidencia;
    private String correoElectronico;
    private String telefonoContacto;
    private ObjectId foto;
    private String url;
    private String tema;

    @Embedded
    private List<Estudios> estudios;
  
```

Figura 16. Diagrama de la @Entidad Personas

6.3 DIAGRAMA DE CLASES

Luego de definirse la estructura de datos se procede a crear la jeraquia de clases lo cual se plasmó en el siguiente diagrama.

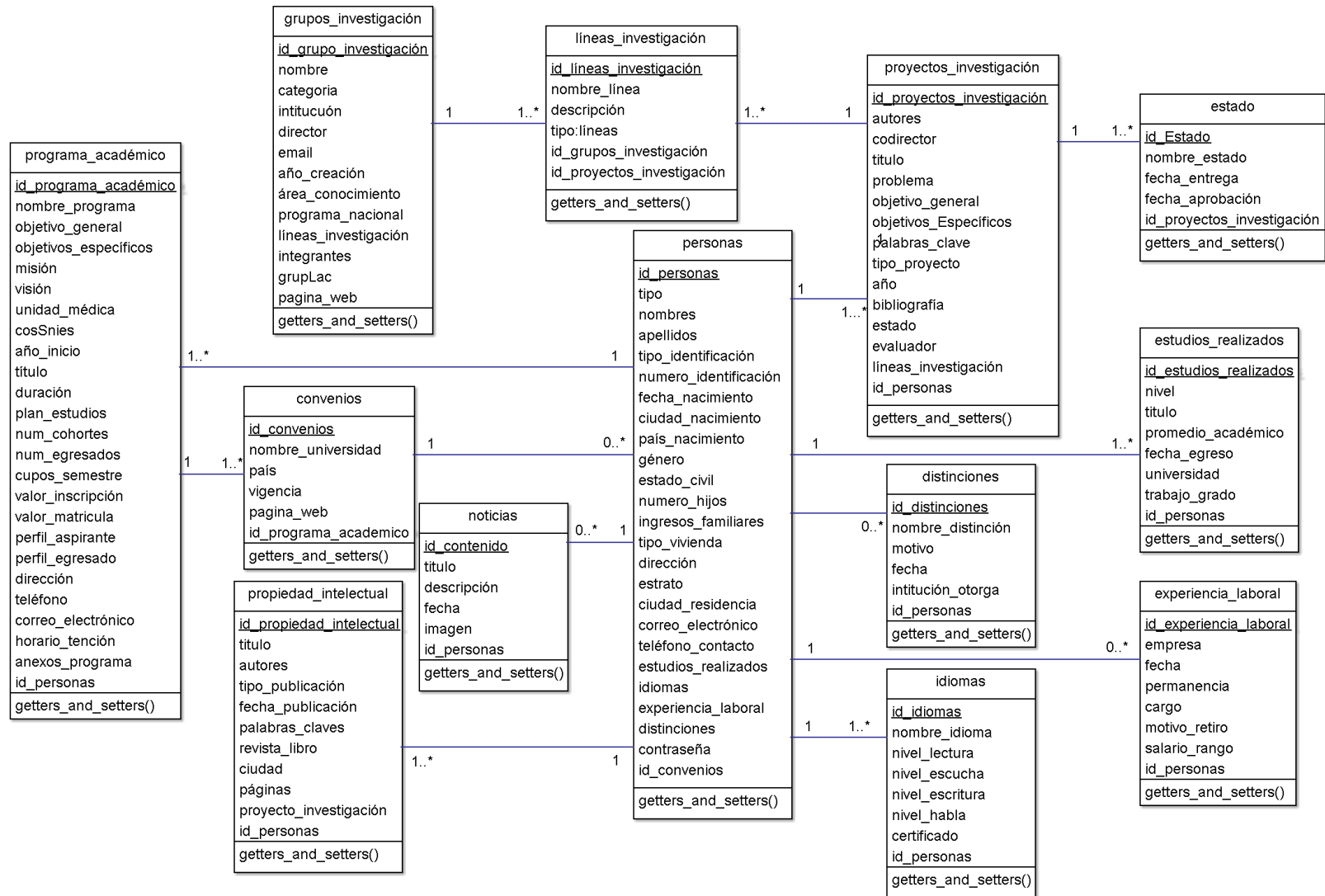


Figura 17. Diagrama de Clases

6.4 IMPLEMENTACIÓN DE DISEÑO DE CLASES

7.4.1 Organización de Paquetes

Una de las técnicas utilizadas para la implementación es la arquitectura EJB y el desarrollo de la herramienta hará parte de un framework que se desarrollará en diversos proyectos. De acuerdo con esto, los paquetes del proyecto quedan agrupados en el paquete *filoen.autoevaluacion*, que a su vez se encuentra organizado de la siguiente manera:

- Paquete model: contiene las colecciones (@Entity) a utilizar.
- Paquete dao: contiene los objetos de acceso a datos o DAO (@SessionBean).
- Paquete portal.controller: contiene todas las clases requeridas para la administración del *backend* de la herramienta.

7.4.2 Clases de Modelo de Negocio (Colecciones y Session BEAN)

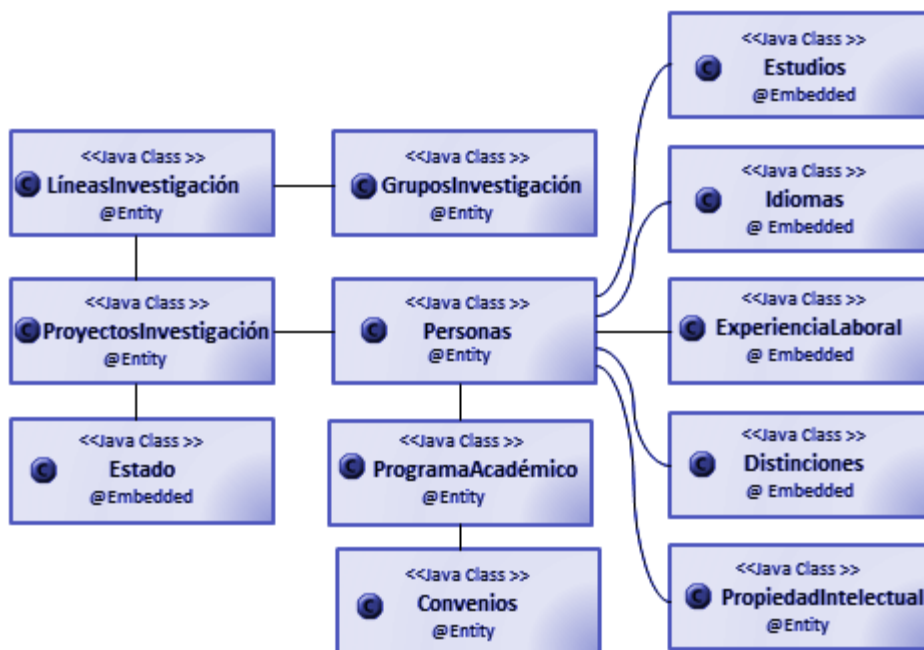


Figura 18. Clases de Modelo de Negocio

Conforme a lo planteado anteriormente, se generaron las clases de entidad (@Entity) las cuales representan las diferentes colecciones de la base de datos.

Las tablas 29 y 30 representan las clases asociadas al paquete modelo y al paquete dao; cuyo objetivo es encapsular los objetos del lado del servidor que almacena los datos, así como las clases asociadas al proceso de acceso a los datos de las bases de datos.

Filoen.autoevaluacion.model	Función
Distinciones.java	Clase almacenada como un atributo en la entidad Personas.java (@Embedded) y se guarda como una lista de la colección personas en MongoDB.
Estado.java	Clase almacenada como un atributo en la entidad ProyectoInvestigacion.java (@Embedded) y se guarda como una lista de la colección proyectosInvestigación en MongoDB.
Estudios.java	Clase almacenada como un atributo en la entidad Personas.java (@Embedded) y se guarda como una lista de la colección personas en MongoDB.
ExperienciaProfesional.java	Clase almacenada como un atributo en la entidad Personas.java (@Embedded) y se guarda como una lista de la colección personas en MongoDB.
GruposInvestigacion.java	Entidad asociada a la colección gruposInvestigación.
Idiomas.java	Clase almacenada como un atributo en la entidad Personas.java (@Embedded) y se guarda como una lista de la colección personas en MongoDB.
LineasInvestigacion.java	Entidad asociada a la colección líneasInvestigación.
Personas.java	Entidad asociada a la colección personas.
ProgramaAcademico.java	Entidad asociada a la colección programaAcadémico.
PropiedadIntelectual.java	Entidad asociada a la colección propiedadIntelectual.

ProyectosInvestigacion.java	Entidad asociada a la colección proyectosInvestigación.
-----------------------------	---

Tabla 29. Clases Asociadas al Paquete Modelo

Filoen.autoevaluacion.dao	Función
AbstractFacade.java	Clase abstracta con la definición de los métodos principales para el acceso a datos que utilizarán las demás clases Facade.
ArchivosFacade.java	Clase para el acceso a la estructura de almacenamiento de archivos GridFS de MongoDB.
GruposInvestigacionFacade.java	Clase de acceso a datos de la colección gruposInvestigación.
LineasInvestigacionFacade.java	Clase de acceso a datos de la colección líneasInvestigación.
PersonasFacade.java	Clase de acceso a datos de la colección personas.
ProgramaAcademicoFacade.java	Clase de acceso a datos de la colección programaAcadémico.
PropiedadIntelectualFacade.java	Clase de acceso a datos de la colección propiedadIntelectual.
ProyectosInvestigacionFacade.java	Clase de acceso a datos de la colección proyectosInvestigación.

Tabla 30. Clases Asociadas al Paquete Dao

En la siguiente figura se puede observar la conexión de los paquetes DAO, Facade y Controller para la clase ProgramaAcadémico.

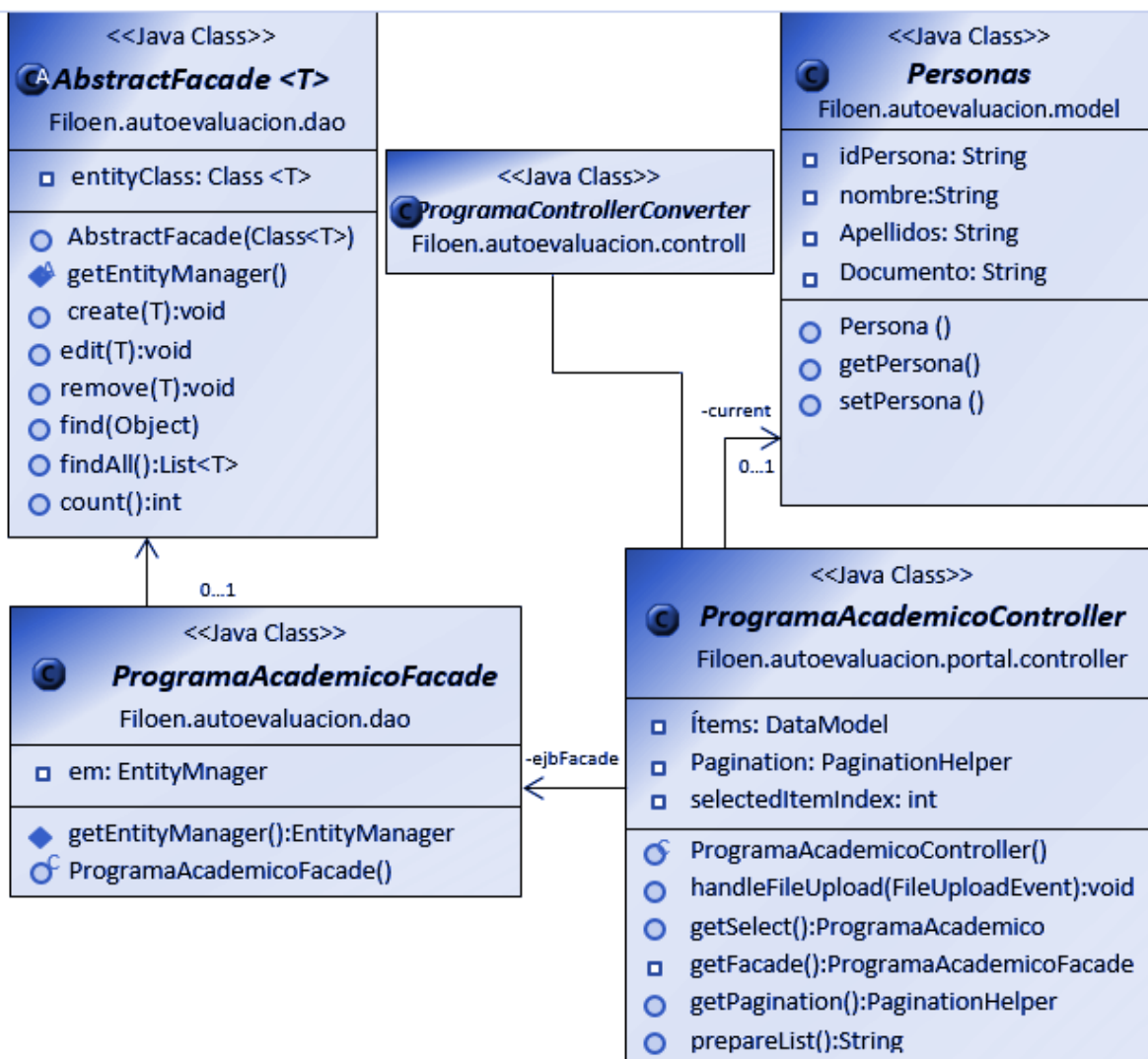


Figura 19. Conexión de los Paquetes Dao, Facade y Controller para la clase ProgramaAcadémico

7.4.3 Clases de Lógica de Negocio (ManagedBean)

El paquete portal.controller contiene todas las clases requeridas para la administración del *backend* y del *fronted* de la herramienta. Estas clases contienen las funciones más comunes asociadas a:

- La validación de datos de un componente.
- Manejo de eventos activados por un componente.
- Determinar el procesamiento a realizar y determinar la siguiente página a la que se debe navegar.

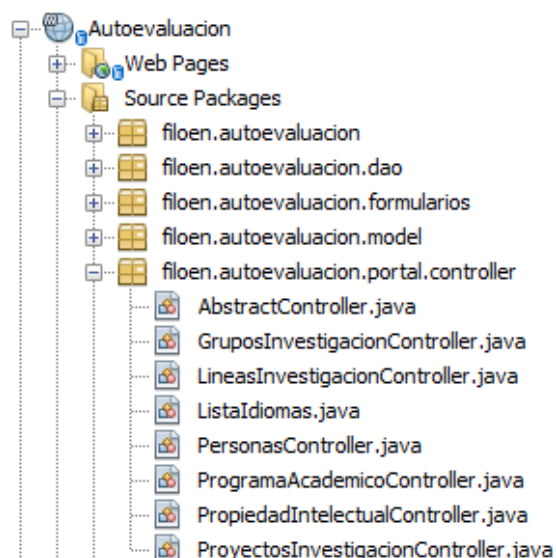


Figura 20. Vista del Paquete Controller en NetBeans

De acuerdo a lo anterior, la tabla 31 presenta la definición de las diferentes clases asociadas al paquete Controller, las cuales se encargan de interactuar con los campos que se han definido en la capa de vista mediante JSF.

Filoen.autoevaluacion. portal.controller	Función
AbstractController.java	Clase abstracta que encapsula las principales funcionalidades de control del sistema. En ella se basan todas las clases Controller del sistema
GruposInvestigacionController.java	Clase de la capa controlador del módulo de administración asociado a la gestión de los grupos de investigación
LineasInvestigacionController.java	Clase de la capa controlador del módulo de administración asociado a la gestión de las líneas de investigación
PersonasController.java	Clase de la capa controlador del módulo de administración asociado a la gestión de personas
ProgramaAcademicoController.java	Clase de la capa controlador del módulo de administración asociado a la gestión del programa académico

PropiedadIntelectualController.java	Clase de la capa controlador del módulo de administración asociado a la gestión de la propiedad intelectual
ProyectosInvestigacionController.java	Clase de la capa controlador del módulo de administración asociado a la gestión de los proyectos de investigación

Tabla 31. Clases Asociadas al Paquete Controller

7.4.4 Diagrama de Actividades

A continuación se describe mediante diagramas de actividades, la interacción entre el sistema y el operador administrativo, mostrando así, los tres procesos principales del componente: Administrar Parámetros, Gestionar Usuarios, y Generar Reportes.

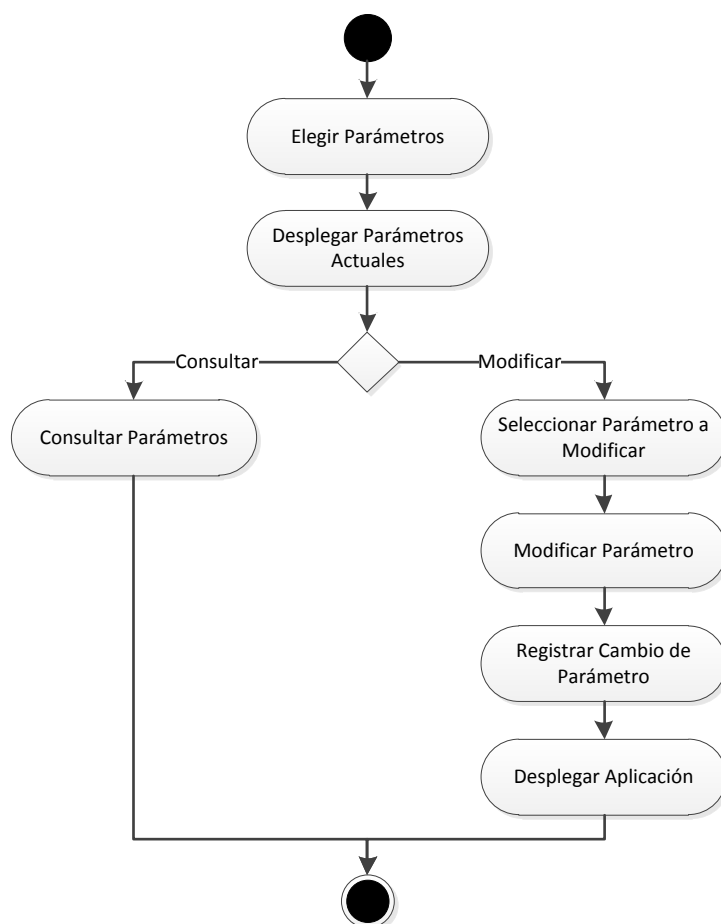


Figura 21. Diagrama de Actividades – Administrar Parámetros

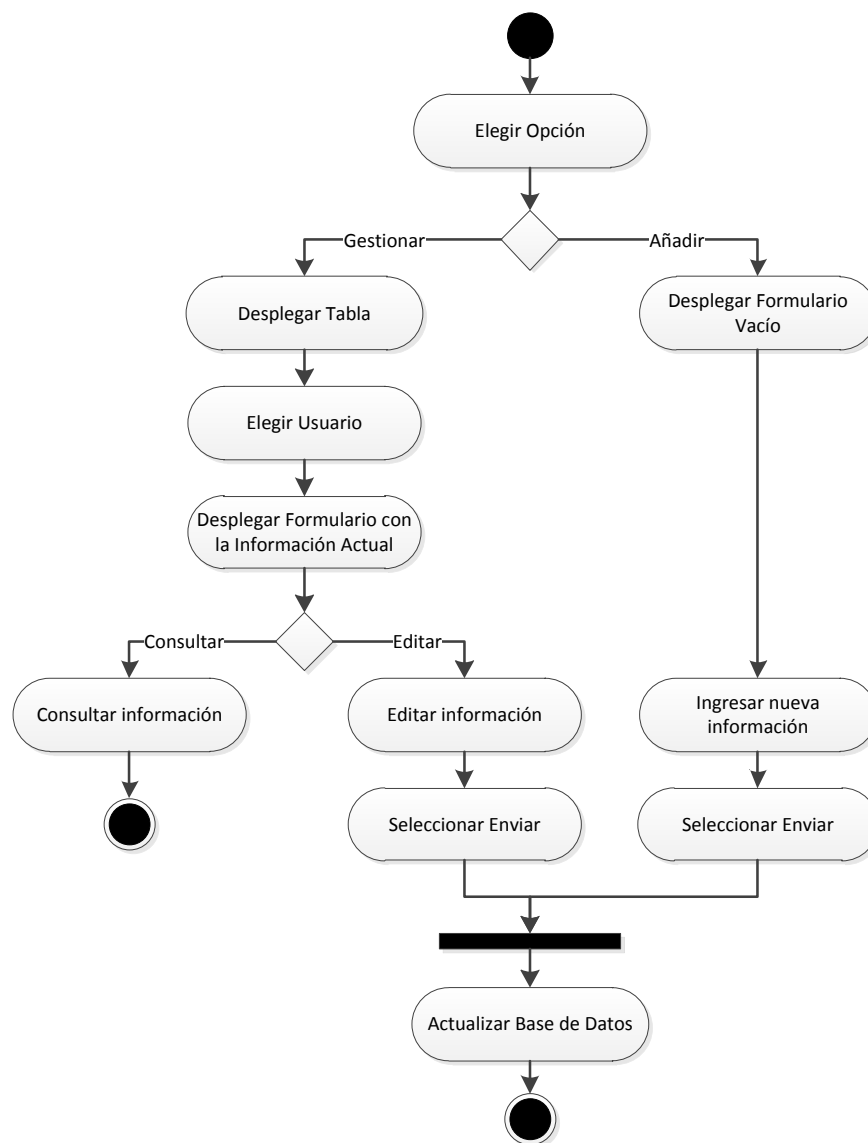


Figura 22. Diagrama de Actividades – Gestionar Usuarios



Figura 23. Diagrama de Actividades – Generar Reportes

7.4.5 Diagrama de Navegación

El siguiente modelo de navegación muestra las funcionalidades del componente tanto del usuario operador administrativo como el usuario coordinador, se puede observar cómo están enlazadas las páginas del componente mediante unidades de navegación (navigationLink) con su clase correspondiente así como su conexión a la administración del mismo (processClass).

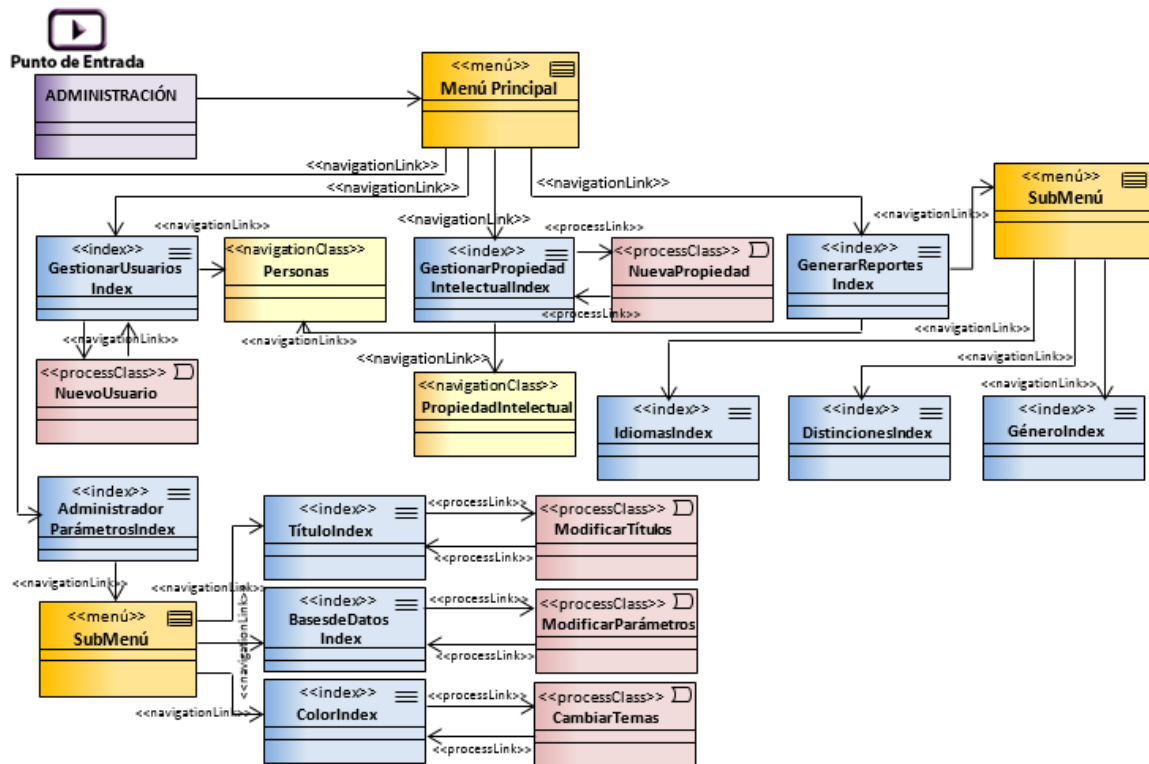


Figura 24. Diagrama de Navegación del Componente

7. DESARROLLO

Antes de iniciar con el proceso de desarrollo e implementación del componente se acordó usar ciertos patrones o arquitecturas, con el fin de organizar y estandarizar el proceso de programación para que en un futuro tenga ventajas para su mantenimiento a continuación se detallan los patrones utilizados en la construcción del componente.

7.1 MODELO VISTA CONTROLADOR

Para la realización del componente se creyó que era muy conveniente utilizar el patrón de diseño modelo vista controlador debido a que promueve la organización y estandarización de todo el proceso que realiza el componente.

Este patrón está definido como un patrón de arquitectura del software el cual se basa en separar los datos, la lógica y las interfaces de usuario. Para realizar esto el patrón propone crear tres tipos de componentes, el modelo, la vista y el controlador, es decir, por un lado se definen los componentes para la presentación de la información, y por otro lado la interacción con los usuarios. Este patrón de diseño se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

7.2 DESCRIPCIÓN DE LOS COMPONENTES

8.2.1 Modelo

En este componente es en donde se representa toda la información que se va a manejar en la aplicación y a su vez es quien gestiona todos los accesos (lectura / escritura) de dicha información, los respectivos privilegios impuestos para dichas acciones y facilita el envío de información requerida por los usuarios a la vista.

8.2.2 Vista

Este componente se relaciona con todo lo referente a la interfaz gráfica de usuario la cual permite mostrar/visualizar al usuario la información que envía el modelo.

8.2.3 Controlador

El controlador responde usualmente a peticiones hechas por el usuario de la aplicación y es quien se encarga de hacer las diferentes peticiones de información al modelo. En otras palabras es el componente que actúa de intermediario entre la vista y el modelo.

7.3 INTERACCIÓN DE COMPONENTES

A continuación se ilustrará con un diagrama de manera simplificada cómo interactúan los tres componentes que conforman el patrón MVC.

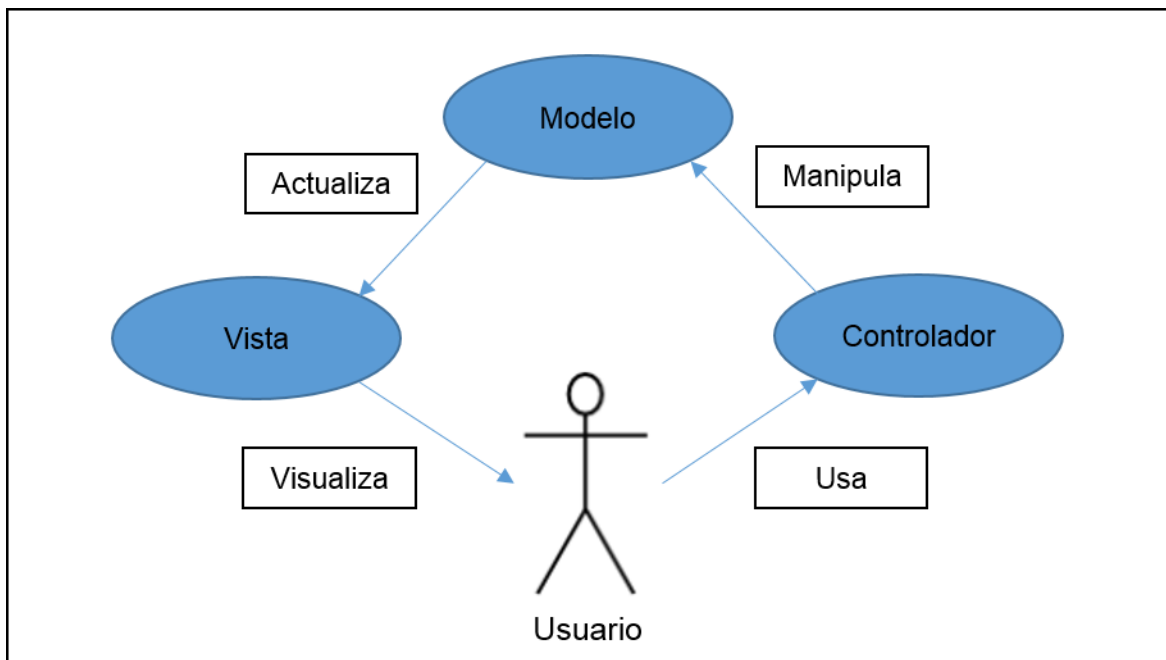


Figura 25. Interacción de Componentes

7.4 USO DE PATRONES DE DISEÑO: FACADE

El patrón de diseño Facade, o Fachada en español, es un tipo de patrón estructural cuya finalidad está en la de reducir la complejidad del código fuente llevando a cabo una división en subsistemas, logrando así minimizar las comunicaciones y dependencias de ciertos componentes.

Este patrón se aplica regularmente cuando se necesita proporcionar una interfaz simple para un subsistema complejo, o cuando se quiera estructurar varios subsistemas en capas, ya que las fachadas serían el punto de entrada a cada nivel.

Estructura

A continuación se representa un diagrama con la estructura del patrón de diseño. En el ejemplo se utiliza el patrón a través de la clase Facade la cual crea un método único que comparten clase 2 clase 2 y clase 3, este resume o facilita ya que la el cliente solo llama el método único.

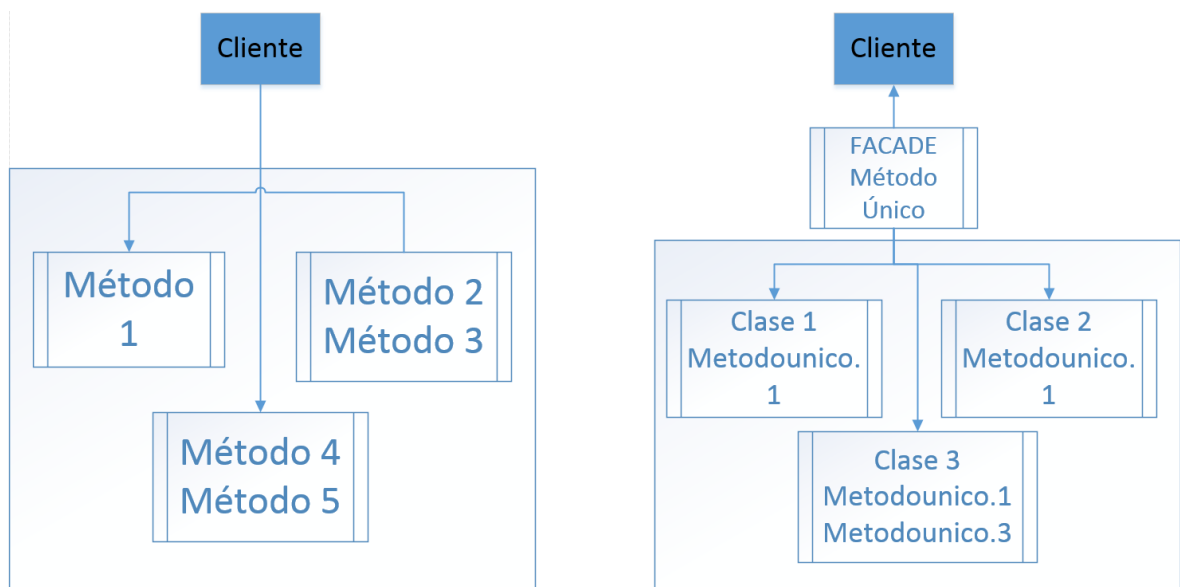


Figura 26. Estructura sin clase Facade vs. Estructura con clase Facade

A continuación se muestra la implementación de éste mismo patrón de diseño y su respectivo código aplicado a la Clase PersonasFacade donde se aprecia que PersonasFacade es una clase extendida de a clase AbstractFacade<T>

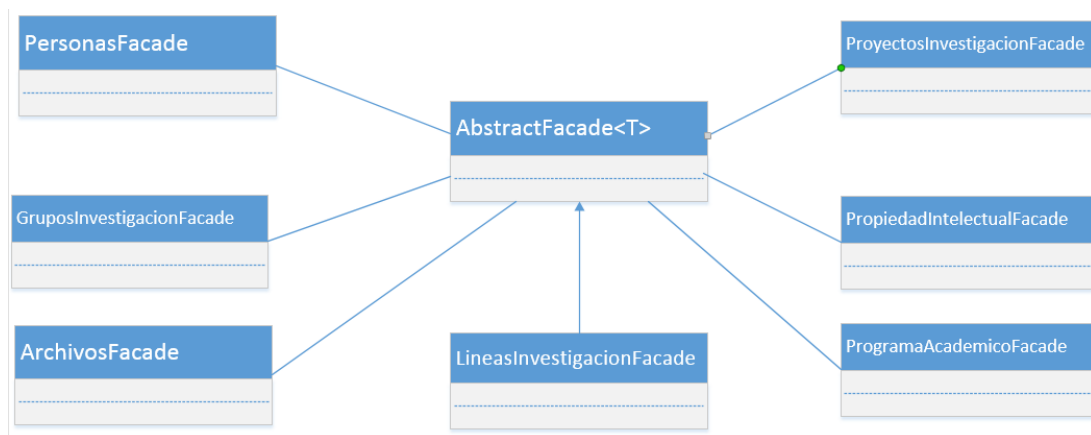


Figura 27. Estructura de la clase AbstractFacade<T>

7.5 USO DE PATRONES DE DISEÑO: MODELO VISTA CONTROLADOR (MVC)

Dando un bosquejo de cómo se implementó el patrón de diseño, MVC, en el proyecto general de autoevaluación de maestría, este se dividió en diferentes paquetes que contienen por separado los elementos de los componentes en este caso el modelo, la vista y el controlador. Para cada uno de los métodos de la aplicación se manejaron por controladores que en el proyecto se nombran como Controller, las vistas son representadas por los archivos XHTML y los objetos de acceso a los datos están representados por los Dao y Facade.

A continuación un diagrama de la representación de aplicación del modelo vista controlador en el componente.

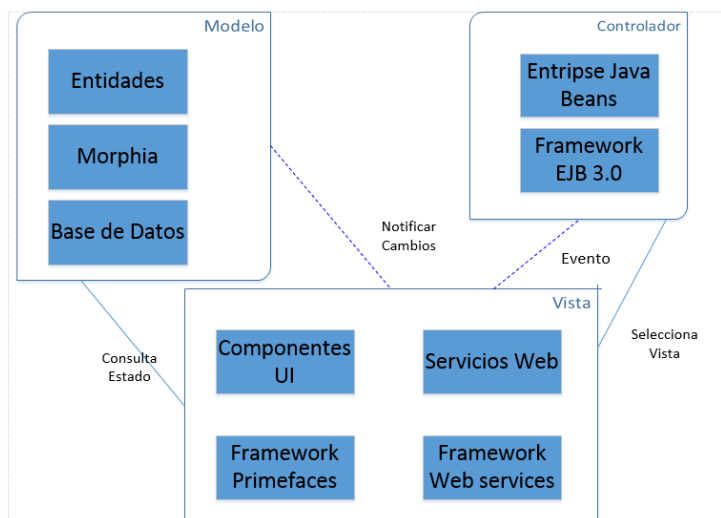


Figura 28. Representación de aplicación del patrón de diseño MVC

De acuerdo con el diseño y modelamiento de las diferentes interfaces, vistas y demás se logra tener un bosquejo final del desarrollo el cual se muestra a continuación la página principal de configuración.

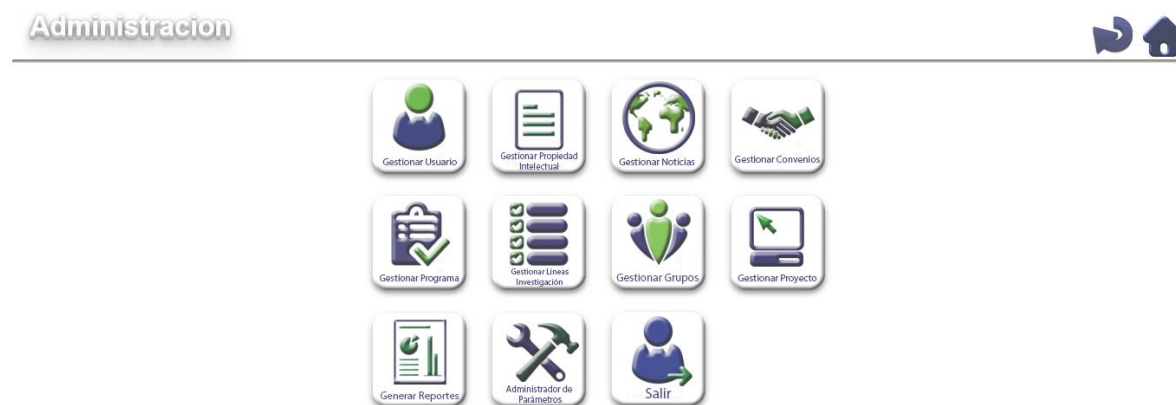


Figura 29. Bosquejo final de la página principal

Para ver más capturas de pantalla y demás explicaciones de funcionalidad del módulo, se recomienda revisar el anexo A que contiene el manual de usuario.

8. PRUEBAS DEL COMPONENTE

Para las pruebas de funcionalidad se abarco el método de la caja negra, el cual consiste en realizar pruebas de tal forma que se demuestre que cada función del sistema es operativa, cada prueba se denomina un caso de prueba. Se llama prueba de caja negra debido a que se evalúan las respuestas de una entrada en una funcionalidad del sistema sin conocer cómo funciona el código.

Metodología

- Hacer una revisión de las funcionalidades del sistema (componente).
- Encontrar de la revisión las funcionalidades más relevantes del sistema.
- Desarrollar los casos de pruebas según las funcionalidades encontradas.
- Realizar las pruebas basados en entradas
- Revisar las salidas y confirmar si cumplen con la funcionalidad esperada.

De las funcionalidades generales del componente se escogieron las más relevantes para el sistema que son: añadir/editar usuarios, añadir/editar propiedad intelectual, y generar reportes.

Los casos de prueba se mostraran en tablas especificando, el nombre del caso de prueba, la descripción, los valores de entrada, la respuesta a los valores y el resultado.

Caso de Prueba	Descripción	Valores de entrada	Respuesta	Resultado
Prueba1 Crear Persona	Se realiza el proceso de creación de un usuario con el formulario determinado solo con información básica	Tipo: Estudiante Nombre: David Apellido: García Documento: CC Género: Masculino Documento: 1098709379 Ciudad: Bucaramanga País: Colombia Correo: oscar@correo.com	Operación Exitosa, Bienvenido David.	Agregado

<p>Prueba2 Crear Persona</p>	<p>Se realiza el proceso de creación de un usuario con el formulario determinado formulario determinado con información completa.</p>	<p>Tipo: Estudiante Nombre: Oscar Apellido: Duitama Documento: CC Género: Masculino Documento: 109898379 Ciudad: Bucaramanga País: Colombia Correo: oscar@correo.com Estado Civil: Soltero Dirección: Calle 10 # 35 Estrato: 3 N° hijos: 0 Ingresos: 1 y 3 Salarios Vivienda: Propia Teléfono: 6345678</p> <hr/> <p>Idioma: Inglés Nivel Habla: Medio Nivel Escucha: Medio Nivel Lectura: Medio Nivel Escritura: Medio Certificado: Si</p> <hr/> <p>Estudio: Universitario Título: Ingeniero Promedio: 3.8 Institución: UIS Trabajo grado: GIB</p>	<p>Operación Exitosa, Bienvenido Oscar.</p>	<p>Agregado</p>
<p>Prueba3 Crear Persona</p>	<p>Se realiza el proceso de creación de un usuario con el formulario determinado con información requerida incompleta.</p>	<p>Tipo: Nombre: Apellido: Duitama Documento: CC Género: Masculino Documento: Ciudad: Bucaramanga País: Colombia Correo: oscar@correo.com</p>	<p>Estos campos son requeridos</p>	<p>No Agregado</p>

Tabla 32. Pruebas Unitarias – Crear Persona

Caso de Prueba	Descripción	Valores de entrada	Respuesta	Resultado
Prueba4 Crear Propiedad Intelectual	Se realiza el proceso de creación de una propiedad intelectual información completa.	Título: Interpretando Organizaciones Tipo: Libro Palabras Clave: Pensamiento Sistémico Revista/Libro: Propio Ciudad: Bucaramanga Formato: Digital Páginas: 200 Proyecto: Prueba Descripción: Breve	Operación Exitosa, se ha creado Interpretando Organizaciones	Agregado
Prueba5 Crear Propiedad Intelectual	Se realiza el proceso de creación de una propiedad intelectual con información incompleta.	Título: Tipo: Palabras Clave: Pensamiento Revista/libro: Propio Ciudad: Bucaramanga Formato: Digital Páginas: 200 Proyecto: Prueba Descripción: Breve	Estos campos son requeridos	No Agregado

Tabla 33. Pruebas Unitarias – Crear Propiedad Intelectual

Caso de Prueba	Descripción	Valores de entrada	Respuesta	Resultado
Prueba6 Generar Diagrama Circular	Generar Reporte de Idiomas	Datos necesarios extraídos de BD	Generado	Reporte Generado
Prueba7 Generar Diagrama Lineal	Generar Reporte de Distinciones	Datos necesarios extraídos de BD	Generado	Reporte Generado
Prueba8 Generar Diagrama de Barra	Generar Reporte de Distinciones	Datos necesarios extraídos de BD	Generado	Reporte Generado

Tabla 34. Pruebas Unitarias – Generar Reportes

8.1 PRUEBAS JUNIT

Después de haber seleccionado y creado los casos de prueba mencionados, se procede a realizar dichos Test por medio del framework JUnit el cual es muy utilizado en programación para hacer pruebas unitarias de aplicaciones Java.

Por cada caso de prueba, se creó un archivo .java llamados JUnitPruebaX, siendo X un número del 1 al 8. Estos archivos pueden ser encontrados en el paquete Test Package del proyecto.

A continuación, se muestran las capturas con los resultados de las pruebas ejecutadas y una breve descripción de cada prueba:

Las figuras 30, 31 y 32 corresponden a los resultados generados por el framework JUnit al ejecutar las pruebas diseñadas en la Tabla 32. Pruebas Unitarias – Crear Persona

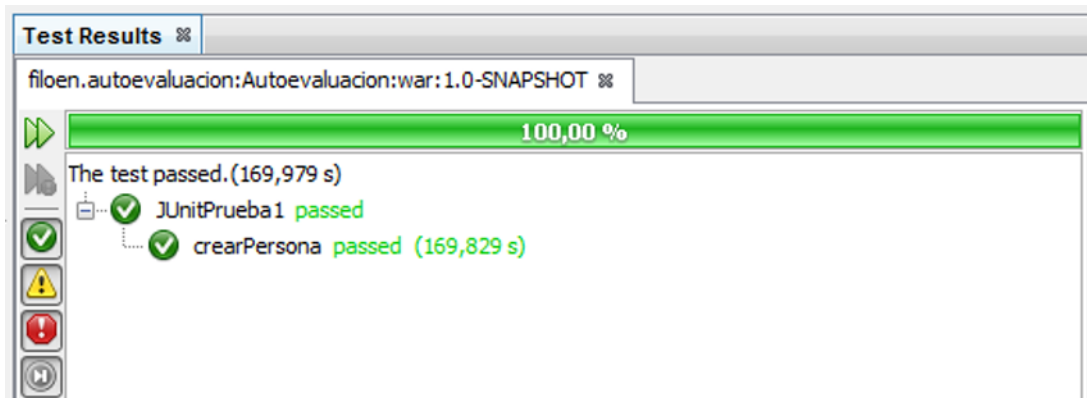


Figura 30. Pruebas JUnit – Prueba 1: crearPersona

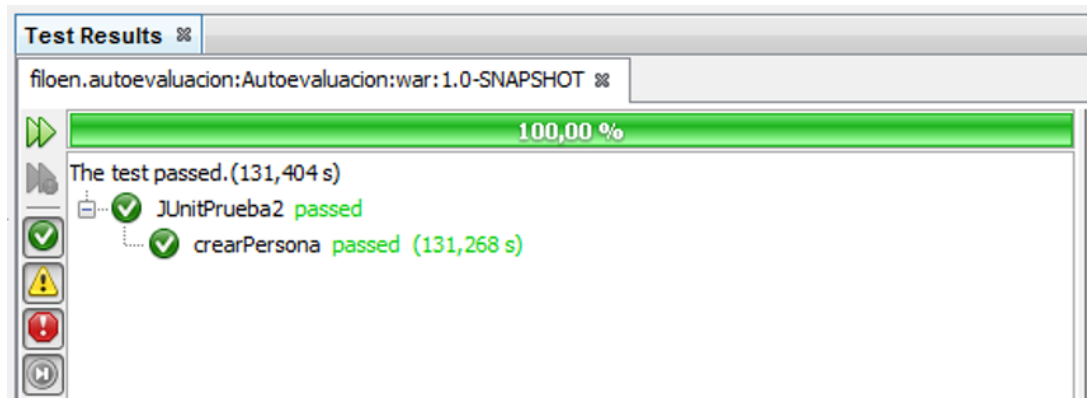


Figura 31. Pruebas JUnit – Prueba 2: crearPersona

Como se puede observar, en las figuras anteriores, el resultado de ambos test fue **passed**, el cual indica que la operación de crear el usuario fue exitosa.

Para la siguiente prueba, se ingresa un usuario con valores nulos en campos que son requeridos u obligatorios. Este fue el resultado arrojado al ejecutar la prueba:

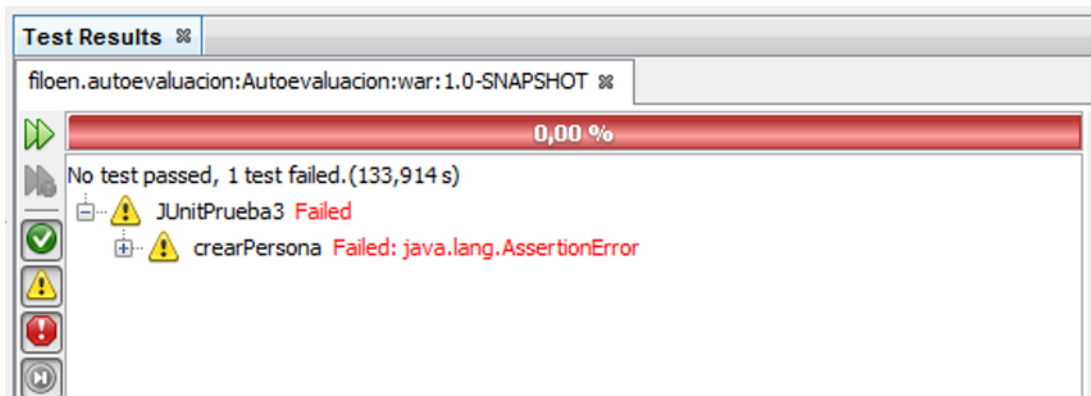


Figura 32. Pruebas JUnit – Prueba 3: crearPersona

Como se puede observar, el resultado generado fue **Failed**. Gracias a la ejecución de esta prueba se detecta este error, y al instante, se procede a hacer ciertos ajustes en desarrollo del proyecto para evitarlo.

Igualmente, se ejecutan las pruebas especificadas en la Tabla 33. Pruebas Unitarias – Crear Propiedad Intelectual, los resultados generados por JUnit al ejecutar estas pruebas, se muestran a continuación en las figuras 33 y 34:

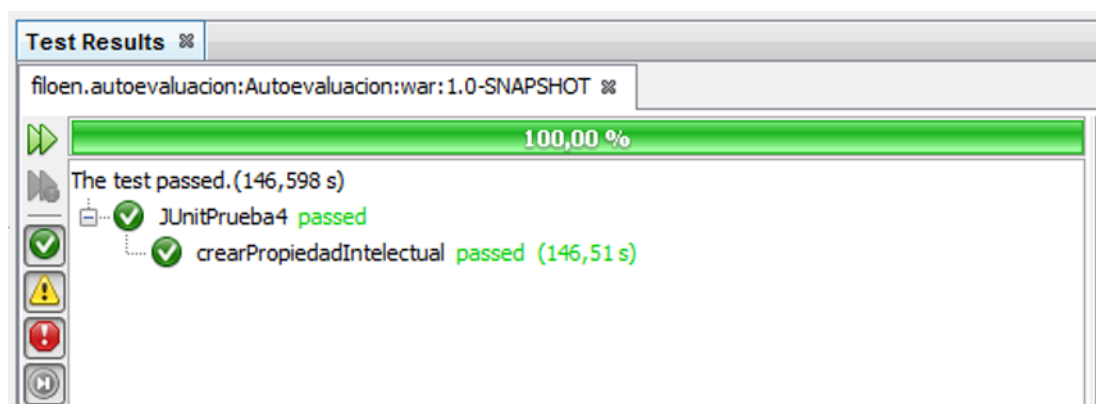


Figura 33. Pruebas JUnit – Prueba 4: crearPropiedadIntelectual

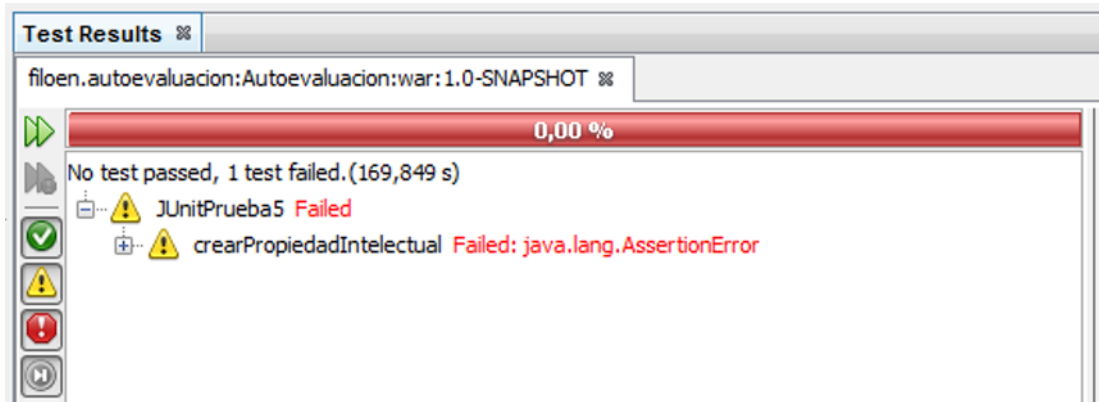


Figura 34. Pruebas JUnit – Prueba 5: crearPropiedadIntelectual

Y para finalizar, en las figuras 35, 36 y 37 se muestran los resultados al testear las pruebas que se encuentran en la Tabla 34. Pruebas Unitarias – Generar Reportes

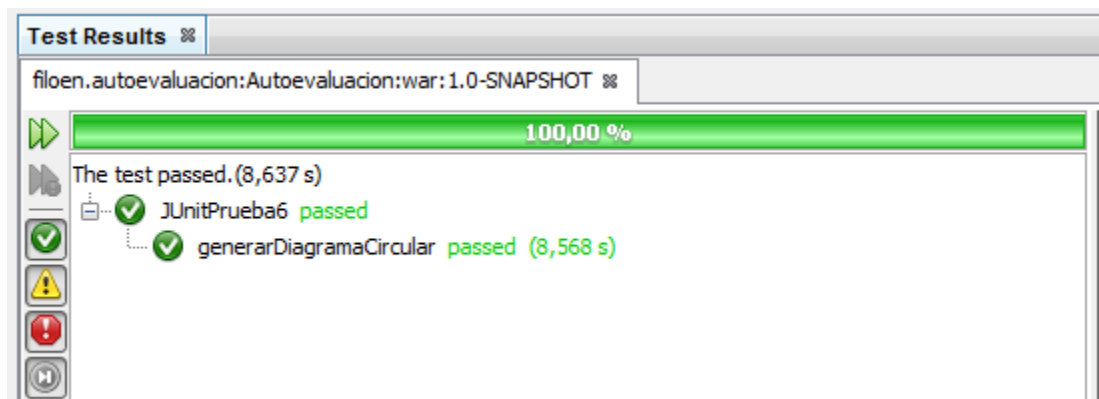


Figura 35. Pruebas JUnit – Prueba 6: generarDiagramaCircular

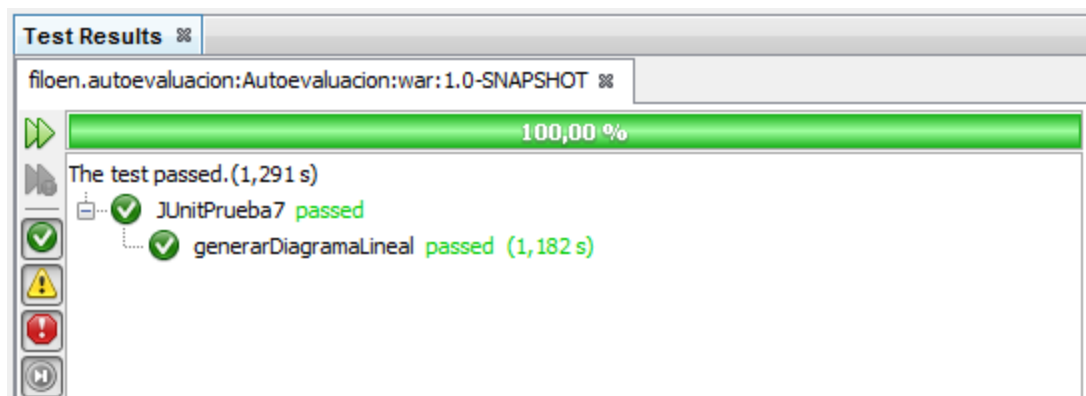


Figura 36. Pruebas JUnit – Prueba 7: generarDiagramaLineal

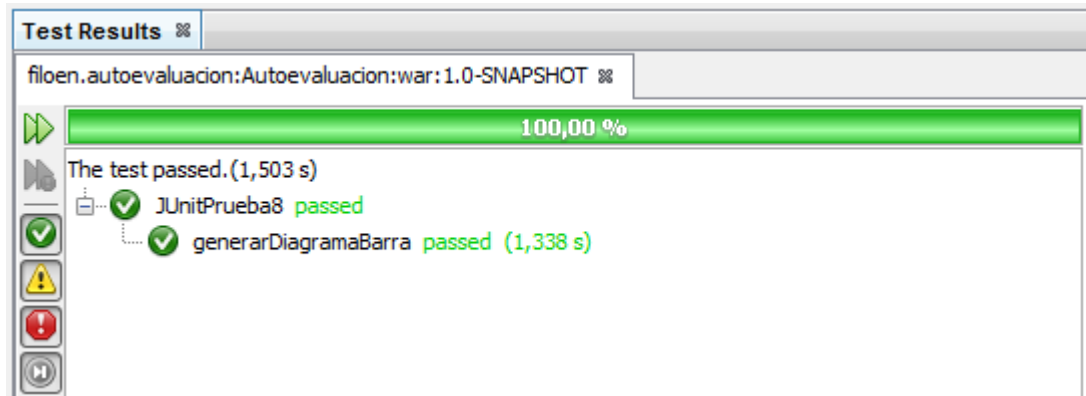


Figura 37. Pruebas JUnit – Prueba 8: generarDiagramaBarra

9. CONCLUSIONES

- En la etapa de análisis, se han especificado los requerimientos principales del sistema los cuales son Gestionar Usuarios, Gestionar Propiedades Intelectuales, Generar Reportes y Administrar Parámetros, usados por dos tipos de actores principales, el Operador Administrador y el Coordinador.
- De acuerdo a los requerimientos dados por el departamento de Postgrado en Ingeniería de Sistemas de la Universidad Industrial de Santander, se construyó una Base de Datos NO Relacional, usando MongoDB para el manejo de los datos.
- Se elaboró un componente web que permite a cualquier escuela de posgrado, registrar mediante formularios con campos dinámicos información sobre los usuarios de la escuela y analizar la información de estos mediante reportes estadísticos, con el fin de garantizar la calidad del proceso de autoevaluación. Además, se logró la comunicación con los otros componentes del sistema de información de autoevaluación mediante el diseño y el desarrollo de una serie de servicios web. Conjuntamente, el componente puede ser configurado, sin necesidad de ingresar directamente en el código fuente, desde la opción 'Administrar Parámetros'.
- Mediante el método de la caja negra, se ha diseñado un conjunto de pruebas unitarias las cuales se han aplicado usando el framework JUnit, y con los resultados generados, se ha podido corregir varios errores que no habían sido detectados en la etapa de análisis y diseño.

10. SUGERENCIAS Y RECOMENDACIONES

- Se sugiere generar las listas de los formularios del componente, por medio de una consulta a la Base de Datos para ofrecer un mantenimiento más fácil e intuitivo al sistema.
- Una vez integrados los demás proyectos del conjunto de autoevaluación, se recomienda administrar los parámetros de estos, como títulos de las nuevas páginas integradas, entre otros aspectos, que pueden servir para proyectos o sistemas futuros.
- Para mejorar el rendimiento en la generación de reportes y evitar problemas de recursos de servidor en consultas complejas futuras, se recomienda manejar la generación de dichos reportes, con manejo de colas y arquitectura de mensajería JMS la cual da un manejo asíncrono en los recursos que se destinan para dicha acción.
- Se recomienda implementar filtros dinámicos para la consulta de usuarios y propiedades dentro de sus respectivos gestores.
- Es viable implementar una versión móvil permitiendo extender el acceso a la plataforma.

BIBLIOGRAFÍA

- Çağatay Çivici, (2012), Data Table and properties. PrimeFaces User guide 4.0. US. Página 136.
- Çağatay Çivici, (2012), Theme switcher. PrimeFaces User guide 4.0. US. Página 447.
- Especialista Universitario Java Enterprise, (2006). ¿Qué es Java Enterprise? Universidad de Alicante. España. [en línea], [Consultado Marzo 2014], disponible en:
<http://www.jtech.ua.es/j2ee/2006-2007/jee.html>
- Institute for Informatics LMU. UWE-UML-Based Web Engineering. Research Unit of Programming and Software Engineering. Alemania. [en línea], [Consultado Octubre 2013], disponible en:
<http://uwe.pst.ifi.lmu.de/aboutUwe.html>
- Mkyong, Septiembre 2010, JSF 2.0 and Resource Bundles example. US. [en línea], [Consultado Abril 2014], disponible en:
<http://www.mkyong.com/jsf2/jsf-2-0-and-resource-bundles-example/>
- Mkyong, (2010), How to include cascading style sheets (CSS) in JSF. US. [en línea], [Consultado Abril 2014], disponible en:
<http://www.mkyong.com/jsf2/how-to-include-cascading-style-sheets-css-in-jsf/>
- Monge Rocero, P.; Ruiz Castrillo, D, (2005). Programación Distribuida: El Modelo EJB. Departamento de Informática y Automática. España.
- Mongo DB, (2011), Getting Started with Java Driver. US. [en línea], [Consultado Febrero 2014], disponible en:
<http://docs.mongodb.org/ecosystem/tutorial/getting-started-with-java-driver/>
- Mongo DB, (2011), Install MongoDB on Windows. US. [en línea], [Consultado Febrero 2014], disponible en:
<http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>

- Palermo Rolando, (2012). Introducción a los EJB. Science and Engineering blog. Peru. [en línea], [Consultado Marzo 2014], disponible en:
<http://blog.rolandopalermo.com/2012/03/ejb-30-netbeans-java.html>
- Silva Diego, (2009). Reportes en NetBeans con IReport. [en línea], [Consultado Mayo 2014], disponible en:
<http://wiki.netbeans.org/ReportesEnNetBeansConIReport>
- Víctor Gómez Adán, (2012), Pruebas de caja negra. España. [en línea], [Consultado Mayo 2014], disponible en:
<http://www.globetesting.com/2012/08/pruebas-de-caja-negra/>

ANEXOS

LISTADO DE IMÁGENES – ANEXOS

Imagen 1. Pantalla Inicial del Componente	81
Imagen 2. Vista de la opción Gestor de Usuarios	82
Imagen 3. Vista de la opción Añadir Usuario	82
Imagen 4. Vista de la pestaña Inf. Personal, mostrando validación en el campo Nombres. En la opción Añadir Usuario	83
Imagen 5. Vista de ventana emergente para añadir Idiomas	84
Imagen 6. Vista de la Pestaña Resumen	84
Imagen 7. Vista de la opción Gestor de Propiedad Intelectual	85
Imagen 8. Vista de la pestaña Propiedad Intelectual. En la opción Añadir Propiedad Intelectual	85
Imagen 9. Vista de la pestaña Descripción. En la opción Añadir Propiedad Intelectual	86
Imagen 10. Vista de la pestaña Resumen. En la opción Añadir Propiedad Intelectual	86
Imagen 11. Vista de la pantalla inicial de la opción Generar Reportes	87
Imagen 12. Vista del Reporte Estadístico de Idiomas - Circular	87
Imagen 13. Vista del Reporte Estadístico de Distinciones - Barra	88
Imagen 14. Vista del Reporte Estadístico de Distinciones - Lineal	88
Imagen 15. Vista del Reporte Estadístico por Género - Circular	89
Imagen 16. Vista de la pantalla inicial de la opción Parámetros	89
Imagen 17. Vista para modificar los Títulos de las páginas del Componente	90
Imagen 18. Vista para modificar el Acceso a la Base de Datos	90
Imagen 19. Vista para modificar los Estilos de las páginas del Componente	91
Imagen 20. Vista del Tema Sunny pre visualizado	91
Imagen 21. Vista para seleccionar el Tema	92

ANEXO A - MANUAL DE USUARIO

Una rápida visualización de como fue el resultado del componente se verá a continuación.

1. Administración del portal

En la siguiente captura se observa la vista inicial del componente que nos dirige a las diferentes opciones del administrador. Se localizan 10 botones que darán acceso a las diferentes tareas y 2 de redirección y cerrar sesión.



Imagen 1. Pantalla Inicial del Componente

2. Gestor de Usuarios

En el gestor de usuarios el administrador podrá acceder a una lista de cada uno de los usuarios registrados, dentro del cual podrá consultar / editar su información y pudiendo aplicar algunos filtros de búsqueda para facilitar la selección de algún usuario.

Gestor de usuarios



Lista de Personas				
Nombres	Apellidos	No. de Documento	E-mail	Ver/Editar
Carlos Humberto	Carreño Diaz	4455465466	cahucadi@gmail.com	
Sonia Cristina	Gamboa Sarmiento	234536464	scgamboa@uis.edu.co	
Oscar David	Duitama Garcia	1098709379	asdas@qq.com	
Sonia	Gamboa Sarmiento	234536464	scgamboa@uis.edu.co	
Cosme	Fulanito	1098567893	cosme@fulanito.com	
Carlos H	Carreño Diaz	4455465466	cahucadi@gmail.com	

Imagen 2. Vista de la opción Gestor de Usuarios

2.1 Crear Usuario

El botón Añadir Usuario redirige al formulario determinado para escribir toda la información pertinente acerca del nuevo usuario a registrar. En este formulario tendrá la opción de ingresar URL de CvLac para uso del proyecto de minería de datos.

Gestionar Persona

Tipo
Inf. Personal
Idiomas
Estudios
Distinciones
Exp. Prof.
Resumen

Rol de la persona

Tipo de usuario

Docente

Url CvLac (opcional)

Para nosotros es muy importante conocer la url de CvLac la cual facilitara la obtención/actualización de datos del usuario

Siguiente

Imagen 3. Vista de la opción Añadir Usuario

El formulario maneja diferentes campos dinámicos para el ingreso de cierta información de los usuarios como idiomas, estudios, distinciones, experiencia entre otros.

Los campos más importantes o requeridos de la información del usuario están previamente validados.

Gestionar Persona

The screenshot shows a web form titled "Gestionar Persona" with several tabs: "Tipo", "Inf. Personal", "Idiomas", "Estudios", "Distinciones", "Exp. Prof.", and "Resumen". The "Inf. Personal" tab is active, displaying a form with the following fields:

- Nombres:** A text input field with a red border and a tooltip that reads "Este campo es obligatorio." (This field is mandatory).
- Apellidos:** A text input field.
- Tipo de Documento:** A dropdown menu with "Seleccione" as the selected option.
- Género:** A dropdown menu with "Femenino" selected.
- Ciudad de Residencia:** A text input field.
- Dirección:** A text input field.
- Estado Civil:** A dropdown menu with "Seleccione" selected.
- Ingresos Familiares:** A dropdown menu with "Seleccione" selected.
- E-mail:** A text input field containing "su@correo.com".
- No. de Documento:** A text input field.
- Fecha de Nacimiento:** A date picker field with a calendar icon.
- País de Origen:** A text input field.
- Estrato:** A dropdown menu with "Seleccione" selected.
- No. de Hijos:** A text input field containing "0".
- Vivienda:** A dropdown menu with "Seleccione" selected.
- Telefono de Contacto:** A text input field.

At the bottom of the form, there are two navigation buttons: "Atras" (Back) and "Siguiente" (Next).

Imagen 4. Vista de la pestaña Inf. Personal, mostrando validación en el campo Nombres. En la opción Añadir Usuario

Para los campos de información adicional (opcional) se maneja con una ventana emergente que se despliega pidiendo los datos requeridos. Y luego de agregados, los despliega en una tabla dando la opción de editarlos o eliminarlos.

Gestionar Persona

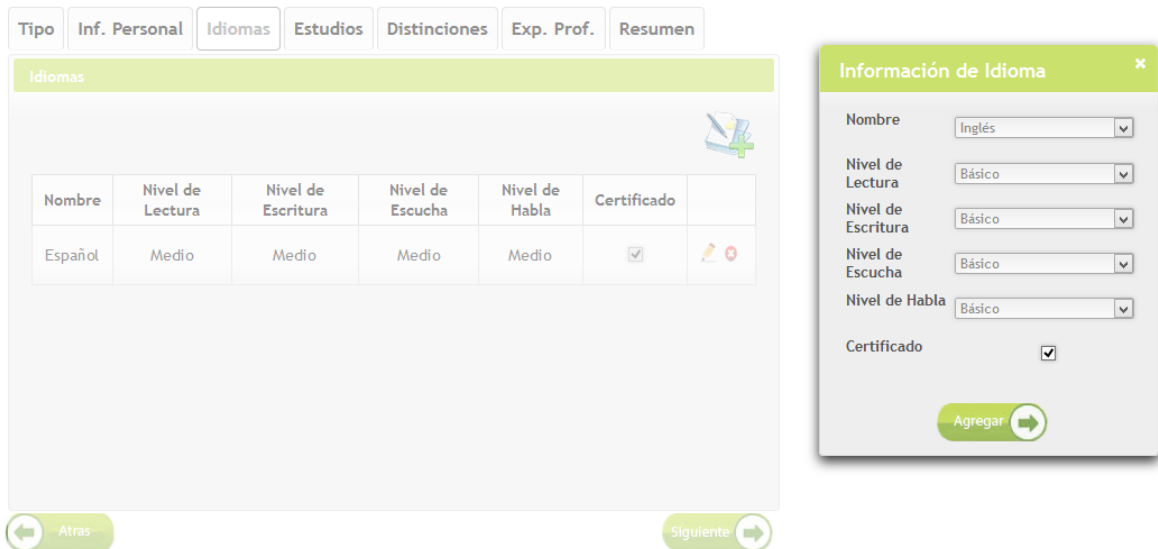


Imagen 5. Vista de ventana emergente para añadir Idiomas

En la pestaña final, antes de poder enviar la información a la Base de Datos, se muestra un breve resumen de la información ingresada en las pestañas previas.

Gestionar Persona



Imagen 6. Vista de la Pestaña Resumen

3. Gestor de Propiedad Intelectual

La gestión de la propiedad intelectual de los usuarios es manejada de manera similar a la de las personas, con un visor en una tabla de las diferentes propiedades existentes, como se muestra a continuación:

Gestor de Propiedad Intelectual

Lista de Propiedades Intelectuales					
Autor(es)	Tipo de Publicación	Lugar de Publicación	Publicado en	Ver/Editar	Eliminar
[Cosme, Sonia Cristina]	Capitulo de libro	ciudad	time		
[Oscar David, Oscar]	Correspondencia	Bucaramanga	una		
[Cosme, Oscar David]	Informe de Avance de Investigacion	Bucaramanga	Revista Ciencia		
[Oscar David]	Articulo inedito	Medellin	Cientifica		
[Oscar David, Sonia Cristina, Carlos Humberto, Sonia, Cosme, Carlos H, Oscar, Luis Carlo, Luis Carlos]	Tesis de Grado	bucaramanga	Senafa		

Imagen 7. Vista de la opción Gestor de Propiedad Intelectual

Los campos dinámicos también se manejaron para la asociación de nuevas propiedades intelectuales, de formas de emergente o con opción autocompletar.

Agregar Propiedad Intelectual

Propiedad Intelectual | Proyecto | Descripción | Resumen

Propiedad Intelectual

Título: Nueva Propiedad

Autor(es): Cosme Fulanito x Sonia Cristina Gamboa Sarmiento x

Tipo de Publicación: Capitulo de libro | Fecha de Publicación: 25/00/2014

Palabras Clave: nuevas palabras | Revista/Libro (En la que fue publicado): time

Ciudad: ciudad | Páginas: 1231

Formato: Seleccione

Siguiente

Imagen 8. Vista de la pestaña Propiedad Intelectual. En la opción Añadir Propiedad Intelectual

4. Reportes Estadísticos

El módulo de reportes estadísticos fue diseñado para ayudar el proceso de la autoevaluación del programa de maestría, en este módulo se darán a conocer algunos reportes estadísticos con la información obtenida previamente de los diferentes usuarios del sistema. Encontraremos diferentes formas interactivas de hacer los reportes por medio de gráficas, las cuales facilitarán la generación de reportes en futuras versiones.

Para la demostración del módulo de reportes damos la posibilidad de generar 4 tipos de gráficos estadísticos que están relacionadas con la preguntas requeridas por el CNA para la autoevaluación de programas de postgrado.



Imagen 11. Vista de la pantalla inicial de la opción Generar Reportes

4.1 Reporte de Idiomas

En este diagrama se plasma el porcentaje de estudiantes registrados que cumplen con el nivel requerido de segundo idioma.

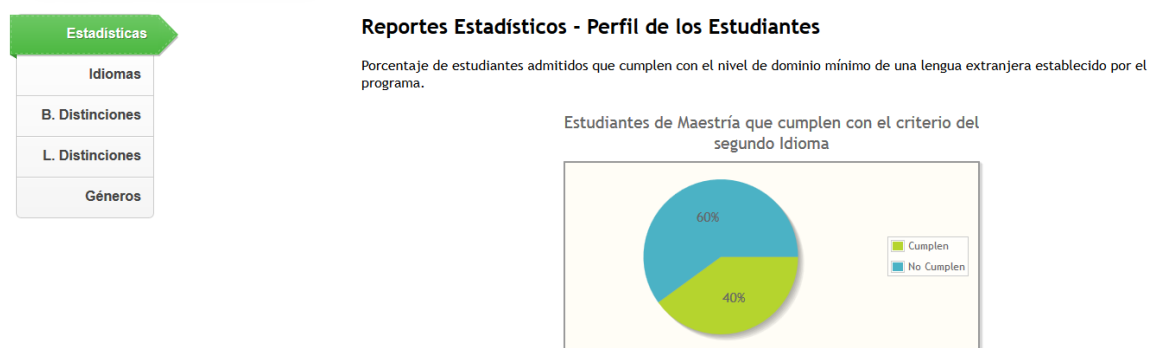


Imagen 12. Vista del Reporte Estadístico de Idiomas - Circular

4.2 Reporte de Distinciones

Se grafican el número de distinciones de profesores que han recibido en los últimos cinco años (Premios, Pertenencia por invitación a Academias Nacionales o Extranjeras o Invitación como Profesor Visitante).

Graficos Estadísticos



Estadísticas
Idiomas
B. Distinciones
L. Distinciones
Géneros

Reportes Estadísticos - Perfil de los Profesores

Número de distinciones de profesores que han recibido en los últimos cinco años (Premios, Pertenencia por invitación a Academias Nacionales o Extranjeras o Invitación como Profesor Visitante).



Imagen 13. Vista del Reporte Estadístico de Distinciones - Barra

En la siguiente imagen se representa el mismo reporte de Distinciones pero en diagrama lineal.

Graficos Estadísticos



Estadísticas
Idiomas
B. Distinciones
L. Distinciones
Géneros

Reportes Estadísticos - Perfil de los Profesores

Número de distinciones de profesores que han recibido en los últimos cinco años (Premios, Pertenencia por invitación a Academias Nacionales o Extranjeras o Invitación como Profesor Visitante).



Imagen 14. Vista del Reporte Estadístico de Distinciones - Lineal

4.3 Reporte de Género

Se muestra el porcentaje de estudiantes que están matriculados en el programa de maestría filtrado por género.

Graficos Estadísticos



Estadísticas
Idiomas
B. Distinciones
L. Distinciones
Géneros

Reportes Estadísticos de Género

En este campo se muestra la información estadística de los géneros que están matriculados en el programa de maestría.

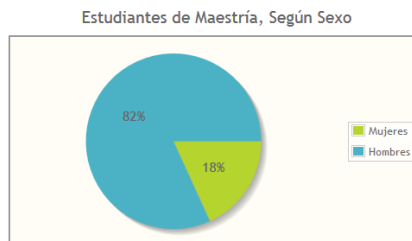


Imagen 15. Vista del Reporte Estadístico por Género - Circular

5. Parámetros

Con el apartado de parámetros se busca facilitar la administración del sitio logrando hacer cambios de ciertos aspectos sin ingresar directamente en el código fuente. Para esta versión se contará con la parametrización de 3 aspectos del portal los cuales incluyen: Títulos de Páginas, Acceso a la Base de Datos y Visualización del Portal.

Administrar Parámetros



Parámetros
Título
Base de Datos
Color



Parametrización

Una de las formas para facilitar la administración del sitio es la de lograr hacer cambios de ciertos aspectos sin ingresar directamente en el código fuente. Para esta versión se contará con la parametrización de 3 aspectos del portal los cuales incluyen: Títulos de Páginas, Acceso a la Base de Datos y Visualización del Portal

Imagen 16. Vista de la pantalla inicial de la opción Parámetros

5.1 Títulos

El componente tiene la posibilidad de modificar 4 de los más importantes títulos de las páginas web, dentro de los que se encuentran gestor de usuarios, gestor de propiedad intelectual, administración y reportes.



Parámetros

- Título
- Base de Datos
- Color

Modificación de Títulos del Portal

Título de Gestor de Usuarios

El Título actual de Gestor de Usuarios es **Gestor de Usuarios2**

Ingrese el Nuevo Título

Enviar →

- Título de Gestor de Propiedad Intelectual
- Título de Administración
- Título de Generar Reportes

Imagen 17. Vista para modificar los Títulos de las páginas del Componente

5.2 Acceso a Base de Datos

Acá se encuentran las opciones para poder modificar las características de la conexión a la base de datos tales como nombre de la base de datos, host, puerto, nombre de usuario y contraseña. Dada la importancia y delicadeza de la modificación de estos datos los campos están previamente validados para ser requeridos y validados según el tipo.



Parámetros

- Título
- Base de Datos
- Color

Parametrización para el acceso a la Base de Datos

Modificar Parámetros de Base de Datos

Las propiedades actuales de la conexión de la base de datos son:

Nombre de la Base de Datos: **proyecto**

Host: **localhost**

Puerto: **27017**

Nombre de Usuario:

Contraseña:

Enviar →

Imagen 18. Vista para modificar el Acceso a la Base de Datos

5.3 Temas

Para el apartado de temas se tiene un panel dinámico con el cual podremos pre visualizar los 6 tipos de temas que se establecieron, aplicando la visualización

temporal a los ejemplos mostrados (tabla y formulario) dentro de los paneles inferiores, para aplicar el tema definitivamente al sitio se utiliza el selector de la parte final de la página.

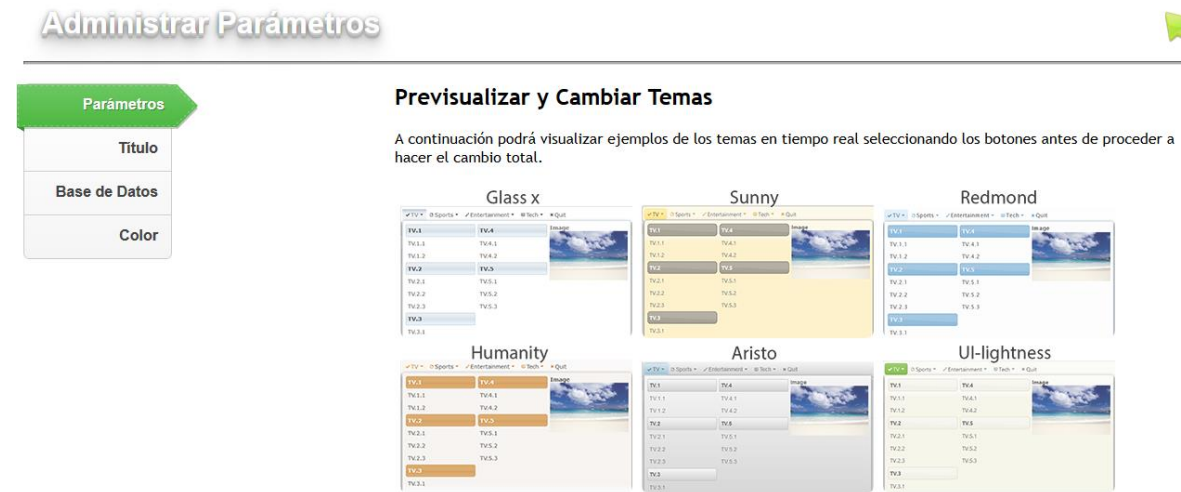


Imagen 19. Vista para modificar los Estilos de las páginas del Componente

Se pre visualiza el cambio en una tabla y en un formulario:



Imagen 20. Vista del Tema Sunny pre visualizado

Selector de tema definitivo:

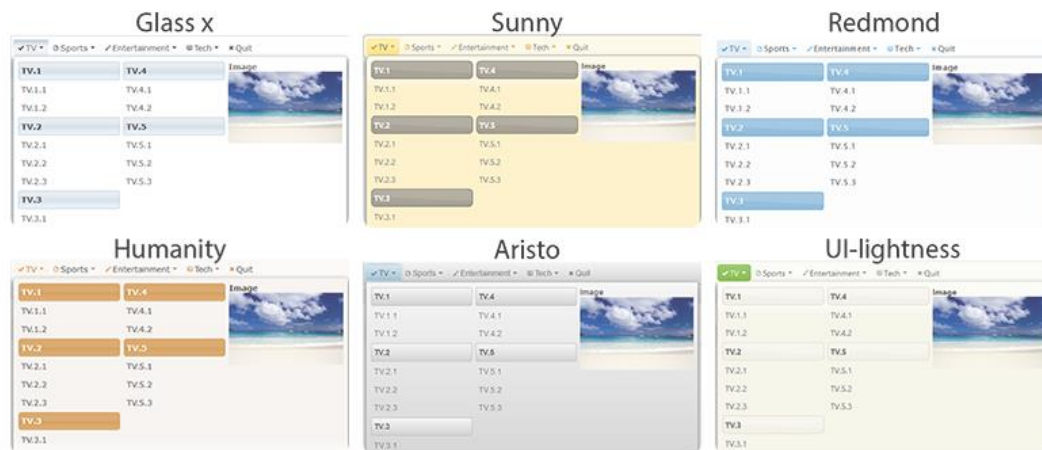


Tabla (Ejemplo)

Formulario (Ejemplo)

Cambiar Tema


Tema 

Imagen 21. Vista para seleccionar el Tema