

**APLICACIÓN PARA CREAR CONSULTAS A LAS BASES DE DATOS DE UN  
SISTEMA DE INFORMACIÓN DE PROGRAMAS DE POSGRADO**

**CRISTIAN ANDRÉS HERNÁNDEZ GARCÍA  
NADIA MARINELLY RUIZ MONTERO**

**UNIVERSIDAD INSUSTRIAL DE SANTANDER  
FACULTAD DE INGENERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2014**

**APLICACIÓN PARA CREAR CONSULTAS A LAS BASES DE DATOS DE UN  
SISTEMA DE INFORMACIÓN DE PROGRAMAS DE POSGRADO**

**CRISTIAN ANDRÉS HERNÁNDEZ GARCÍA  
NADIA MARINELLY RUIZ MONTERO**

**PROYECTO DE GRADO PARA OPTAR AL TÍTULO DE  
INGENIERO DE SISTEMAS**

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

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

**UNIVERSIDAD INSUSTRIAL DE SANTANDER  
FACULTAD DE INGENERÍAS FÍSICO MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA**

**2014**

## DEDICATORIA

*El autor Cristian Andrés Hernández García dedica este logro a:*

*A Dios por brindarme la energía, el entusiasmo y sabiduría en cada paso  
para lograr esta meta.*

*A mis Padres, que desde el cielo me guían y me dan luz en cada decisión  
de mi vida.*

*A mi Familia y Amigos, porque han sido parte fundamental en este proceso.*

## DEDICATORIA

*La autora Nadia Marinelly Ruiz Montero dedica este logro a:*

*A Dios por estar presente en todas mis decisiones, por darme la fortaleza y  
por guiarme en este camino.*

*A mis padres Alberto Ruiz y María Montero, por su apoyo incondicional, por  
su esfuerzo, amor y confianza. Son el motor de mi vida.*

*A mis hermanos, porque siempre me han brindado su apoyo y respeto.*

## **AGRADECIMIENTOS**

Los autores expresan su agradecimiento a:

A nuestros Familiares y amigos, por confiar en nosotros, por el apoyo incondicional y por todos los momentos vividos.

A la escuela de Ingeniería de Sistemas e Informática y a cada uno de los docentes porque sin su ayuda no estaríamos en este punto.

A la profesora Sonia Cristina Gamboa, por confiar en nosotros y abrirnos las puertas del grupo de investigación.

A todas las personas que hicieron parte de nuestra formación en la Universidad Industrial de Santander.

Cristian Hernández y Nadia Ruiz.

## CONTENIDO

INTRODUCCIÓN .....	17
1. JUSTIFICACIÓN.....	18
2. OBJETIVOS.....	20
2.1 OBJETIVO GENERAL .....	20
2.2 OBJETIVOS ESPECÍFICOS.....	20
3. MARCO DE REFERENCIA.....	22
3.1 AUTOEVALUACIÓN.....	22
3.2 LINEAMIENTOS PARA LA ACREDITACIÓN DE ALTA CALIDAD.....	22
3.3 CONSULTAS INTELIGENTES .....	25
3.3.1 Datos.....	26
3.3.2 Información .....	26
3.3.4 Conocimiento .....	27
3.4 ARQUITECTURAS EJB (ENTERPRISE JAVA BEAN).....	27
3.5 MONGODB.....	29
3.6 LIBRERIAS UTILIZADAS.....	30
3.6.1Primefaces .....	30
3.6.2 Morphia .....	30
3.6.3 Java Server Faces (JSF) .....	30
3.6.4 GITLAB .....	31
3.6.5 BOOTSTRAP 3.2.0.....	31
4. METODOLOGÍA .....	33
4.1 ANÁLISIS DE REQUISITOS.....	34
4.2 DISEÑO DEL SISTEMA .....	34
4.3 IMPLEMENTACIÓN.....	34
4.4 PRUEBAS.....	34
5. ANÁLISIS, REQUERIMIENTOS Y CASOS DE USO.....	35
5.1 ESPECIFICACIÓN DE REQUISITOS.....	35

5.2 PROPÓSITO.....	35
5.3 ALCANCE DEL SISTEMA .....	35
5.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS .....	36
5.5 PERSPECTIVA DEL PRODUCTO .....	37
5.6 FUNCIONES DEL PRODUCTO .....	37
5.6.1 Creación de consultas.....	37
5.6.2 Administrar las consultas para su previa modificación.....	37
5.6.3 Generar un reporte tabular de la consulta.....	37
5.6.4 Producir una gráfica.....	37
5.7 CARACTERÍSTICAS DEL USUARIO .....	38
5.8 RESTRICCIONES .....	38
5.9 SUPOSICIONES Y DEPENDENCIAS.....	38
5.10 REQUISITOS FUTUROS.....	39
5.11 REQUERIMIENTOS FUNCIONALES.....	39
5.12 REQUERIMIENTOS NO FUNCIONALES.....	40
5.13 ACTORES DEL SISTEMA.....	42
5.14 DESCRIPCIÓN DE ACTORES.....	42
5.15 DIAGRAMA GENERAL DE CASOS DE USO Y ACTORES .....	43
5.16 DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO.....	43
6. DISEÑO DE LA APLICACIÓN Y ESTRUCTURA DE DATOS .....	48
6.1 DISEÑO DE LA BASE DE DATOS.....	48
6.2 ESQUEMA DE LA BASE DE DATOS.....	50
6.3 DISEÑO DE ENTIDADES UTILIZANDO <i>MORPHIA FRAMEWORK</i> .....	55
6.4 ORGANIZACIÓN DE LOS PAQUETES.....	57
6.5 CLASES DE LÓGICA DE NEGOCIO (CONTROLLER).....	58
7. DESARROLLO .....	60
7.1 MODELO VISTA CONTROLADOR .....	60
7.2 DESCRIPCIÓN DE LOS COMPONENTES .....	60
7.2.1 Modelo .....	60
7.2.2 Vista.....	61

7.2.3 Controlador .....	61
7.3 USO DE PATRONES DE DISEÑO: FACADE .....	61
7.4 INTERACCIÓN DE LOS COMPONENTES .....	62
8. PRUEBAS.....	64
9. INTEGRACIÓN DE LA APLICACIÓN .....	68
10. CONCLUSIONES .....	69
11. SUGERENCIAS Y RECOMENDACIONES.....	70
BIBLIOGRAFIA.....	71

## LISTA DE TABLAS

Tabla 1.	Requerimiento Funcional 01 – Crear Consulta.....	39
Tabla 2.	Requerimiento Funcional 02 – Guardar Consulta.....	39
Tabla 3.	Requerimiento Funcional 03 – Cargar Consulta.....	39
Tabla 4.	Requerimiento Funcional 04 – Eliminar Consulta.....	39
Tabla 5.	Requerimiento Funcional 05 – Aplicar Consulta.....	40
Tabla 6.	Requerimiento Funcional 06 – Generar Gráfica .....	40
Tabla 7.	Requerimiento Funcional 07 – Exportar Datos .....	40
Tabla 8.	Requerimiento No Funcional 01 – Interfaz del Sistema.....	40
Tabla 9.	Requerimiento No Funcional 02 – Confiabilidad del Sistema.....	41
Tabla 10.	Requerimiento No Funcional 03 – Seguridad de la información .....	41
Tabla 11.	Requerimiento No Funcional 04 – Arquitectura.....	41
Tabla 12.	Requerimiento No Funcional 05 – Ventanas de Confirmación.....	41
Tabla 13.	Requerimiento No Funcional 06 – Permitir Pruebas .....	41
Tabla 14.	Descripción de Actores - Administrador .....	42
Tabla 15.	Descripción de Casos de Uso – Crear Consulta .....	43
Tabla 16.	Descripción de Casos de Uso – Guardar Consulta.....	44
Tabla 17.	Descripción de Casos de Uso – Cargar Consulta .....	44
Tabla 18.	Descripción de Casos de Uso – Eliminar Consulta .....	46
Tabla 19.	Descripción de Casos de Uso – Aplicar Consulta .....	46
Tabla 20.	Descripción de Casos de Uso – Generar Gráfica .....	47
Tabla 21.	Descripción de Casos de Uso – Exportar Datos .....	47
Tabla 22.	Pruebas Funcionalidad – Existencias de Consultas .....	64
Tabla 23.	Pruebas Funcionalidad – Filtros.....	65
Tabla 24.	Pruebas Funcionalidad – Resultados.....	65
Tabla 25.	Pruebas Funcionalidad – Resultados Finales .....	66
Tabla 26.	Prueba de Usabilidad .....	66

## LISTA DE FIGURAS

Figura 1. Inteligencia de negocios .....	26
Figura 2. Arquitectura EJB .....	29
Figura 3. Metodología en Cascada .....	33
Figura 4. Actores del Sistema.....	42
Figura 5. Diagrama general de Casos de Uso y Actores.....	43
Figura 6. Esquema de la base de datos – Tablas Personas, Estudios y Experiencia_Profesional.....	50
Figura 7. Esquema de la base de datos–Programa_Academico, Distinciones y Propiedad_Intelectual.....	51
Figura 8. Esquema de la base de datos – Grupos_Investigacion y Proyectos_Investigacion.....	52
Figura 9. Esquema de la base de datos – Lineas_Investigacion e Idiomas .....	52
Figura 10. Esquema de la base de datos – Filtros y Búsqueda .....	53
Figura 11. Modelo conceptual de la base de datos .....	55
Figura 12. Diagrama de BaseEntity.....	56
Figura 13. Diagrama de la @Entidad Personas .....	57
Figura 14. Estructura de Paquetes en NetBeans .....	58
Figura 15. Vista del paquete controller en NetBeans .....	59
Figura 16. Estructura sin clase Facade vs. Estructura con clase Facade .....	62
Figura 17. Interacción de los componentes.....	62
Figura 18. Bosquejo Final de la página principal.....	63

## RESUMEN

TÍTULO: APLICACIÓN PARA CREAR CONSULTAS A LAS BASES DE DATOS DE UN SISTEMA DE INFORMACIÓN DE PROGRAMAS DE POSGRADO\*

AUTORES:

Cristian Andrés Hernández García  
Nadia Marinelly Ruiz Montero\*\*

PALABRAS CLAVE:

Consultas inteligentes, sistema de información, MongoDB, morphia.

DESCRIPCIÓN:

Los programas académicos de posgrado en Colombia tienen que ajustarse a los requerimientos del Ministerio de Educación Nacional y a los lineamientos para la autoevaluación ofrecidos por el Consejo Nacional de Acreditación. Según estos lineamientos, los programas de posgrado deberían llevar a cabo procesos de autoevaluación a partir de un sistema en el que interactúan características, factores, variables e indicadores; este sistema se determina siguiendo los requerimientos y expectativas del programa. Estos procesos deberían ser permanentes, de manera que permitan el control de tales variables.

La Universidad Industrial de Santander en los últimos años ha notado que los programas académicos de posgrado requieren tener visibilidad en la web para facilitar la interacción con información actualizada acerca de sus programas, esta información es un preciado recurso para poder identificar posibles patrones, tendencias u otros conocimientos que favorezcan los procesos de toma de decisiones, planificación y autoevaluación; estos factores son de suma importancia en el momento de llevar a cabo un correcto proceso de autoevaluación y así obtener la acreditación, esto garantiza que los programas de posgrados tengan la mejor calidad para ofrecerle a sus estudiantes.

Este proyecto tiene por objeto diseñar una aplicación web que permita crear consultas específicas requeridas a las bases de datos del sistema de información de un programa de posgrado, de manera que sea posible calcular automáticamente los factores y variables de un sistema de autoevaluación para tal programa.

---

\* Trabajo de grado

\*\* Facultad de ingenierías fisicomecánicas. Escuela de ingeniería de sistemas e informática. Directora PhD Sonia Cristina Gamboa Sarmiento

## ABSTRACT

TITLE: APPLICATION TO CREATE ENQUIRIES TO INFORMATION SYSTEM DATA BASES OF GRADUATE PROGRAMS\*

AUTHORS:

Cristian Andrés Hernández García

Nadia Marinelly Ruiz Montero\*\*

KEYWORDS:

Smart enquiries, information system, MongoDB, morphia.

DESCRIPTION:

Graduate programs in Colombia have to fulfill the requirements of the National Education Ministry, and the self-assessment guidelines provided by the National Accreditation Council. According to these guidelines, graduate programs should carry out assessment processes from a system where characteristics, factors, variables and indicators interact. This system is designed by following the requirements and expectations of the program. These processes should be permanent, in such a way that the aforementioned variables are controlled.

Industrial University of Santander in recent years have noticed that academic graduate programs require you to have visibility on the web to facilitate interaction with updated information about their programs, this information is a valuable resource to identify possible patterns, trends or other knowledge to support the making desitions processes, planning and self-assessment; These factors are very important at the time to conduct a proper assessment process and obtain accreditation, this ensures that graduate programs have the best quality to offer its students.

This project aims at the design of a web application that allows the creation of specific enquiries required by the data bases of the information system of a graduate program, allowing the automatic calculation of the factors and variables involved in a self-assessment system for such program.

---

\* Final year Project

\*\*Physical-Mechanical Engineering Faculty. Systems and Informatics Engineering School. Project director PhD Sonia Cristina Gamboa Sarmiento

## INTRODUCCIÓN

Los programas académicos tienen que hacer constante verificación, coordinación y análisis de sus procesos educativos para identificar fortalezas, debilidades, amenazas y oportunidades que ayuden al mejoramiento continuo de la calidad de enseñanza. Este proceso es conocido como autoevaluación.

En Colombia los programas académicos de posgrado tienen que ajustarse a los requerimientos del Ministerio de Educación Nacional y a los lineamientos para la autoevaluación ofrecidos por el Consejo Nacional de Acreditación. Según estos lineamientos, los programas de posgrado deberían llevar a cabo procesos de autoevaluación a partir de un sistema en el que interactúan características, factores, variables e indicadores; este sistema se determina siguiendo los requerimientos y expectativas del programa.

La escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander propone la creación de una aplicación para crear consultas de manera personalizada a la base de datos de sus programas de posgrado. Además de la creación de consultas inteligentes, brindar la posibilidad de generar reportes gráficos y de exportar datos.

Durante la presentación de este proyecto se presentará el marco de referencia con conceptos claves que ayudarán al entendimiento de proceso de autoevaluación en el contexto académico, y cómo fue desarrollado un componente software que ayuda 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

En los últimos años la Universidad Industrial de Santander ha notado que los programas académicos de posgrado requieren tener visibilidad en la web para facilitar la interacción con información actualizada acerca de sus propios programas. Esta información es un preciado recurso en los programas académicos, a partir de esta es posible encontrar patrones, tendencias u otros conocimientos que favorezcan los procesos de toma de decisiones, planificación y autoevaluación.

Los programas académicos de posgrado en Colombia tienen que ajustarse a los requerimientos del Ministerio de Educación Nacional y a los lineamientos para la autoevaluación ofrecidos por el Consejo Nacional de Acreditación –CNA<sup>1</sup>–. Según estos lineamientos, los programas de posgrado deberían llevar a cabo procesos de autoevaluación a partir de un sistema en el que interactúan características, factores, variables e indicadores; este sistema se determina siguiendo los requerimientos y expectativas del programa.

Estos procesos deberían ser permanentes, de manera que permitan el control de tales variables. Este proyecto tiene por objeto diseñar una aplicación web que permita crear consultas específicas requeridas a las bases de datos del sistema de información de un programa de posgrado, de manera que se sea posible calcular automáticamente los factores y variables de un sistema de autoevaluación para tal programa.

---

<sup>1</sup> CNA: Consejo de Acreditación Nacional

Este proyecto llega a integrarse a la primera fase del proyecto Sistema de Información de Autoevaluación, que se está desarrollando en la UIS y está por culminar, bajo la dirección de la PhD Sonia Cristina Gamboa Sarmiento.

## **2. OBJETIVOS**

### **2.1 OBJETIVO GENERAL**

Diseñar e implementar un componente software para la generación de consultas personalizadas en bases de datos de un sistema de información de programas de posgrado.

### **2.2 OBJETIVOS ESPECÍFICOS**

Análisis de la estructura

- Especificar una estructura de datos no relacional que permita el almacenamiento histórico de la información registrada en el Sistema de Autoevaluación de Posgrado.

Construcción del modelo

- Diseñar una interfaz que ofrezca a los usuarios la creación de consultas inteligentes de selección en la base de datos utilizando operadores lógicos y aritméticos, así como la presentación tabular y gráfica de resultados.
- Integrar el componente desarrollado al Sistema de Información de Autoevaluación de programas de posgrado.

Pruebas de funcionalidad

- Validar la funcionalidad y usabilidad del componente implantado a partir de un conjunto de pruebas unitarias.

## Documentación

- Elaborar la documentación del componente software desarrollado, de forma que facilite el mantenimiento o desarrollo de nuevas versiones, utilizando una herramienta para el control de versiones.

### **3. MARCO DE REFERENCIA**

En este capítulo se presentan las bases teóricas en las cuales se fundamenta el desarrollo de la aplicación.

#### **3.1 AUTOEVALUACIÓN**

Es un proceso que los diferentes programas de posgrado realizan libremente para generar un autodiagnóstico de sus funciones sustantivas de la educación y las dinámicas propias de su desarrollo a partir de la autonomía universitaria, debe ser claro, suficientemente argumentado y explicativo, pero concreto. Se puede expresar de manera cualitativa o cuantitativa teniendo en cuenta que sus argumentos puedan ser medibles y comprobables.

El punto de partida para analizar la institución o los programas académicos y avanzar así en el proceso es por medio de la autoevaluación; en esta se realiza un ejercicio autocrítico que de forma permanente, debe realizarse para alcanzar alta calidad. Este proceso puede arrojar diferentes resultados los cuales reflejan la calidad en sus procesos, en base a esto se construye un plan de mejoramiento que tiene como objetivo principal corregir debilidades encontradas y mejorar las operaciones en sus programas académicos.

#### **3.2 LINEAMIENTOS PARA LA ACREDITACIÓN DE ALTA CALIDAD**

En la actualidad los programas de posgrado y doctorado juegan un papel especialmente importante en el desarrollo de habilidades para la investigación y por lo tanto, de generación de conocimiento. La política que rige los posgrados centra su atención en el desarrollo de las maestrías tanto de investigación como de profundización y en los doctorados, más que en las especializaciones. Cabe

resaltar que en ciertos contextos y para ciertos fines, las maestrías de profundización desempeñan un papel de gran importancia en la cualificación del tipo de profesional que el entorno requiere.

Para poder lograr la acreditación un programa de posgrado debe cumplir con ciertos factores, características e indicadores. El consejo nacional de acreditación ha definido en la “GUIA DE PROCEDIMIENTO” en el año 2010 los siguientes factores y características de alta calidad que permiten apreciar el desarrollo de las funciones sustantivas de cada programa académico e institución. Para comprender y dimensionar el alcance de los factores y características en cada uno de los procesos (programas de pregrado, posgrado e institucional), los pares deben conocer el documento de Lineamientos correspondiente. Los factores anteriormente mencionados se presentan a continuación:

Factor 1: Cumplimiento de los objetivos del programa y coherencia con la Misión y Visión de la Universidad.

Factor 2: Estudiantes:

Características:

- 1) Características de los estudiantes en el momento de su ingreso.
- 2) Permanencia y desempeño de los estudiantes durante el posgrado.
- 3) Características de los graduados del programa.

Factor 3: Profesores

Características:

- 4) Perfil de los profesores.
- 5) Producción científica de los profesores.
- 6) Relación estudiante/tutor.
- 7) Política sobre profesores.

Factor 4: Procesos académicos y lineamientos curriculares

Características:

- 8) Formación, aprendizaje y desarrollo de investigadores.
- 9) Formación del investigador en términos de su capacidad para comprender el entorno social y geopolítico de la ciencia.
- 10) Flexibilidad del currículo.
- 11) Aseguramiento de la calidad y mejora continúa.

Factor 5: Investigación, generación de conocimiento y producción artística

Características:

- 12) Articulación de la investigación al programa.
- 13) Los grupos de investigación y sus líneas.
- 14) Productos de la investigación y su impacto.

Factor 6: Articulación con el entorno y capacidad para generar procesos de innovación

Características:

- 15) Posibilidad de trabajo inter y transdisciplinario.
- 16) Relevancia de las líneas de investigación y de las tesis de grado para el desarrollo del país o de la región.
- 17) Experiencia de interacción con el entorno.

Factor 7: Internacionalización, alianzas estratégicas e inserción en redes científicas globales

Características:

- 18) Internacionalización del currículo y bilingüismo.
- 19) Internacionalización de estudiantes y profesores.
- 20) Internacionalización de la investigación.

Factor 8: Bienestar y ambiente institucional

Factor 9: Graduados y análisis de impacto

Características:

21) La producción científica de los graduados.

22) Análisis de impacto del programa.

Factor 10: Recursos físicos y gestión administrativa y financiera

Características:

23) Infraestructura física adecuada.

24) Recursos bibliográficos, informáticos y de comunicación.

25) Adecuado apoyo administrativo a las actividades de docencia, investigación y extensión.<sup>22</sup>

### 3.3 CONSULTAS INTELIGENTES

Por medio de inteligencia de negocios que es considerado como factor indispensable para la toma de decisiones, se busca brindarle a los programas de posgrado la oportunidad de ser más eficientes en el momento de afrontar el proceso de autoevaluación.

Con inteligencia de negocios se puede transformar los datos en información y la información en conocimiento, lo que facilita la toma de decisiones en una organización, en nuestro caso la universidad industrial de Santander.

---

<sup>2</sup> Conozca más en: [http://www.cna.gov.co/1741/articles-186376\\_manual\\_pares.pdf](http://www.cna.gov.co/1741/articles-186376_manual_pares.pdf)

Figura 1. Inteligencia de negocios



Fuente: Sinnexus [En línea]. Disponible en: [http://www.sinnexus.com/business\\_intelligence/](http://www.sinnexus.com/business_intelligence/)

La inteligencia de negocios se centra en estos tres conceptos:

### 3.3.1 Datos

Los datos son la mínima expresión semántica y son considerados como elementos primarios de información que por sí solos son irrelevantes en la toma de decisiones. Se pueden ver como un conjunto de valores, que no dicen nada del porqué de las cosas y no sirven para la acción.

### 3.3.2 Información

Se puede considerar como un conjunto de datos procesados que tienen un significado, por lo tanto son indispensables para la toma de decisiones. Al contextualizar, categorizar, calcular, corregir y condensar los datos estos se pueden transformar en información.

“Por tanto, la información es la comunicación de conocimientos o inteligencia, y es capaz de cambiar la forma en que el receptor percibe algo, impactando sobre sus juicios de valor y sus comportamientos.”<sup>3</sup>

---

<sup>3</sup> Conozca más en: [http://www.sinnexus.com/business\\_intelligence/piramide\\_negocio.aspx](http://www.sinnexus.com/business_intelligence/piramide_negocio.aspx)

**3.3.4 Conocimiento.** El conocimiento es derivado de la información, así como la información de los datos. Es una mezcla de experiencia y valores que sirven como base para la integración de nueva información y experiencias, por lo tanto es útil para la acción.

Es necesario realizar acciones como: comparaciones con otros elementos, buscar conexiones, poder realizar una predicción de consecuencias y conversar con otros portadores de conocimiento, para que la información pueda convertirse en conocimiento.

### **3.4 JAVA**

Es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Es un lenguaje multiplataforma, su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Es considerado lenguaje de alto nivel y equipado con las herramientas necesarias para el usuario, es altamente seguro, sólido y provee a los desarrolladores de un conjunto de clases y paquetes predefinidos que facilitan el desarrollo de las aplicaciones.

### **3.5 ARQUITECTURAS EJB (ENTERPRISE JAVA BEAN)**

Cuando se asume el desarrollo de una aplicación, se deben tener ciertas responsabilidades, como lo son la seguridad, el control de concurrencia, las

transacciones, entre otras. EJB permite que el desarrollador se centre en la lógica de negocio del problema que se desea solucionar y que se deje de lado las responsabilidades al contenedor de aplicaciones donde se ejecutará la aplicación.

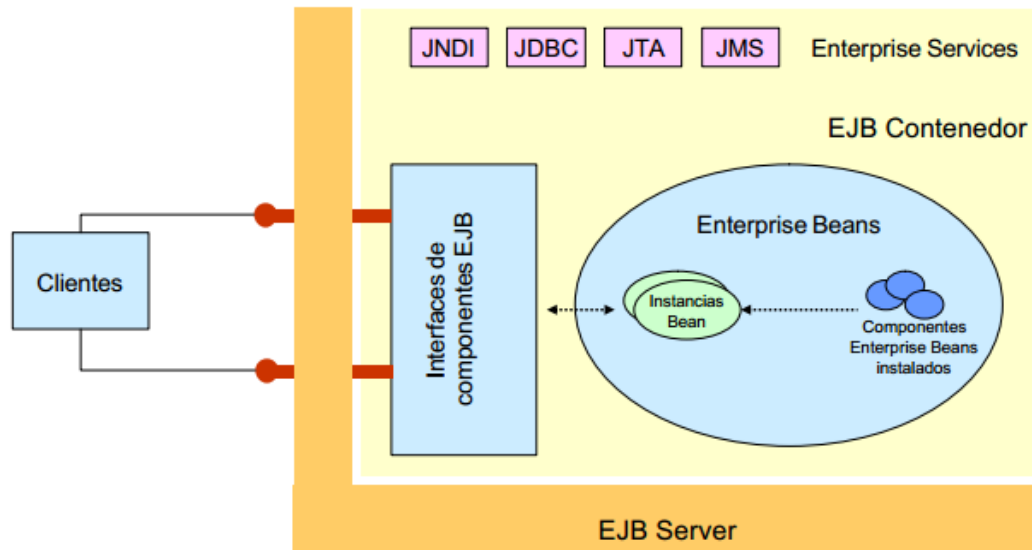
Al estar desarrollado en componentes, permite reutilizar un mayor nivel de funcionalidades e incluso es posible modificar estas funcionalidades y adaptarlas a cada entorno de trabajo particular sin tocar el código del componente desarrollado. La arquitectura EJB ofrece un servidor EJB que facilita a los contenedores la conexión con la infraestructura de comunicaciones, alta disponibilidad y balanceo de carga, proporciona funciones de administración como lo son: gestión de errores, gestión de conexiones entre otras.

Para la ejecución de los componentes instalados, se cuenta con un contenedor EJB en donde se lleva a cabo el manejo de las instancias-Bean, control del ciclo de vida y el acceso a los servicios Enterprise estándar.

La lógica del negocio se encuentra almacenada en los componentes de servicio conocidos como Enterprise beans, de estos hay tres tipos: Sesión Beans en donde está el proceso del negocio, Entity Beans que son los datos de negocio y Message-Driven Beans donde se procesan mensajes asincrónicos.

A EJB pueden acceder clientes Java, clientes basados en browser, clientes corba o simplemente otros Enterprise Beans.

Figura 2. Arquitectura EJB



Fuente: it.uc3m [En línea] Disponible en: [http://www.it.uc3m.es/~celestec/docencia/ittt/swc0405/traspas/2\\_Enterprise\\_JavaBeans.pdf](http://www.it.uc3m.es/~celestec/docencia/ittt/swc0405/traspas/2_Enterprise_JavaBeans.pdf)

### 3.6 MONGODB

Las bases de datos son programas que facilitan que la información se maneje de manera ordenada, aparte de hacerla fácil de buscar. Sus características pueden ser ventajosas o desventajosas: pueden ayudarnos para almacenar, organizar, recuperar, comunicar y manejar información en formas que serían imposibles sin los computadores, pero también nos afecta de alguna manera ya que existen enormes cantidades de información en bases de datos de las que no se tiene control del acceso.

MongoDB es una base de datos orientada a documentos, esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos, para luego ser almacenados. 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.

Es especialmente útil en entornos que requieran escalabilidad. Con la opción de replicación que es muy sencilla de configurar, se puede conseguir un sistema que escale horizontalmente sin demasiados problemas.

### 3.7 LIBRERIAS UTILIZADAS

3.7.1 Primefaces. Es una librería de componentes que sirve para agilizar los desarrollos con Java Server Faces, el cual nos permite crear tablas más dinámicas, campos de texto, gráficas y otros tipos de controles basados en tecnología asíncrona, todo esto separando de manera adecuada la vista u otras extensiones y los controladores. Una de las ventajas al utilizar PrimeFaces se centra en la experiencia que presentan sus usuarios finales, ya que los componentes son amigables al usuario además de contar con un diseño innovador.

3.7.2 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).

3.7.3 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, proporciona un modelo de componentes completos con el control de eventos y representación de componentes. Utiliza la tecnología Java Server Pages (JSP) la cual permite hacer el despliegue de las páginas.

En pocas palabras, Java Server Faces facilita el desarrollo de aplicaciones basadas en Web, ya que:

- Permite crear interfaces de usuario a partir de un conjunto de componentes de servidor estándar y reutilizables.
- Proporciona un conjunto de etiquetas JSP para acceder a los componentes.
- Encapsula el control de eventos y el componente de representación para que pueda usar componentes JSF estándar o componentes personalizados para apoyar a los lenguajes de marcas que no sean HTML.

3.7.4 GITLAB. Es una aplicación gratuita para la administración de repositorios. Es ideal para los desarrolladores que quieren mantener su código privado, ofrece a los usuarios la posibilidad de administrar: usuarios, repositorios, gestionar ramas, permisos, 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 posgrado y mantener copias de seguridad alojadas en la nube en caso de presentarse cualquier eventualidad.

3.7.5 BOOTSTRAP 3.2.0. Conjunto de herramientas de software libre que sirve para la creación de sitios y aplicaciones web. Está dotado de un sin número de plantillas de diseño como formularios, botones, cuadros, menús de navegación entre otros elementos de diseño basados en HTML y hojas de estilo, también

posee extensiones de JavaScript opcionales. Una de las ventajas al utilizar este framework es que dependiendo del tamaño del dispositivo en el que se visualice automáticamente se adapta al tamaño del ordenador sin que el usuario tenga que hacer nada, esto se denomina diseño adaptativo.

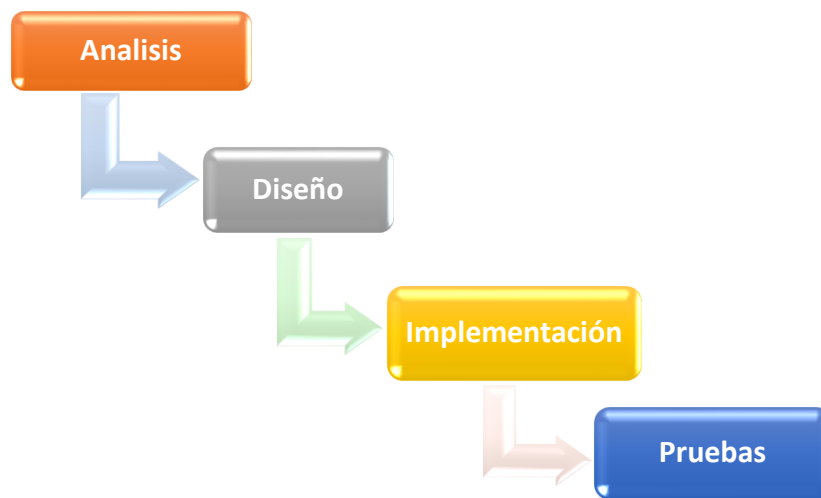
## 4. METODOLOGÍA

Al utilizar una metodología, se desea encontrar una correcta planificación, tener una idea de los costos y el debido aseguramiento de la calidad en el desarrollo del software en cada etapa del desarrollo de un programa. Permitiendo así, una forma sistemática para poder obtener un producto excelente y libre de errores. Para la realización de este proyecto se utilizó la metodología en cascada.

La metodología en cascada es considerada como un método puro que sigue rigurosamente las etapas del proceso de desarrollo del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.

Las etapas que conforman esta metodología son:

Figura 3. Metodología en Cascada



## **4.1 ANÁLISIS DE REQUISITOS**

En esta etapa se analizan las necesidades de los usuarios finales del software para determinar el alcance del proyecto. Se debe plantear todo lo que se requiere del sistema y será lo que se va a seguir en las siguientes etapas, quedando claro que a mitad del proceso de elaboración del software no se podrán hacer cambios. Como resultado de esta etapa surge un documento de especificación de requisitos que contiene lo que debe hacer el sistema sin entrar en detalles internos.

## **4.2 DISEÑO DEL SISTEMA**

Se descompone y organiza el sistema en módulos que puedan elaborarse por separado, esto se puede hacer porque es un trabajo en equipo. Lo que se desea de esta etapa es que quede clara la estructura que va a tener el software, para esto se puede recurrir a diagramas y texto.

## **4.3 IMPLEMENTACIÓN**

En esta etapa se centra toda la parte de programación. Como producto final es el código en cualquier nivel, incluyendo las bibliotecas y componentes que se crean automáticamente dentro del mismo proyecto, esto hace que la programación sea un proceso mucho más rápido.

## **4.4 PRUEBAS**

Cuando se tiene el software terminado, tenemos que cerciorarnos de que los elementos programados se integran y funcionan correctamente, aparte de esto que cumpla con los requisitos planteados al inicio del proyecto. Para esto se realizan una serie de pruebas antes de ser entregado al usuario final.

## **5. ANÁLISIS, REQUERIMIENTOS Y CASOS DE USO**

### **5.1 ESPECIFICACIÓN DE REQUISITOS**

Especificación de Requisitos Software (ERS) de la aplicación para crear consultas a las bases de datos de un sistema de información de programas de posgrado. Este documento 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**

La razón de la especificación de requisitos software es definir de manera clara y precisa las funcionalidades y restricciones de la aplicación que se va a desarrollar. Este 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**

En la actualidad, el departamento de posgrado de ingeniería de sistemas de la Universidad Industrial de Santander no posee un sistema de información propio, lo que impide tener visibilidad en la web para facilitar la interacción con información actualizada acerca de sus propios programas. Así mismo, estos programas deben llevar a cabo procesos de autoevaluación a partir de un sistema en el que interactúan características, factores, variables e indicadores.

La aplicación para crear consultas a las bases de datos de un sistema de información de programas de posgrado, en adelante Consultas Inteligentes, tiene como objetivo principal facilitar la realización de ciertas tareas para apoyar el proceso de autoevaluación, en concreto las siguientes:

- La creación de consultas a la base de datos de programas de posgrado.
- Administrar las consultas para su previa modificación.
- Generar un reporte tabular de la consulta.
- Si el usuario lo desea puede producir una gráfica.

#### **5.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS**

- **CONSULTAS INTELIGENTES:** nombre clave que se le da a la aplicación para crear consultas a las bases de datos de un sistema de información de programas de posgrado.
- **JSF:** Java Server Faces
- **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.
- **JSP:** Java Server Pages
- **HTML:** HyperText Markup Language
- **RF:** Requisitos Funcionales.
- **RNF:** Requisitos No Funcionales.
- **MongoDB:** base de datos no relacional que permite realizar desarrollos de forma muy ágil y dinámica, ya que a diferencia de las bases de datos relacionales no toca definir un modelo Entidad - Relación para poder almacenar los datos.
- **CASOS DE USO:** Descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

## 5.5 PERSPECTIVA DEL PRODUCTO

La aplicación Consultas Inteligentes va a interactuar con ciertos productos software, estos serán adquiridos y otros desarrollados. Estos productos serán:

- Un servidor de web, que permitirá el acceso de la aplicación Consultas Inteligentes 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.

## 5.6 FUNCIONES DEL PRODUCTO

**5.6.1 Creación de consultas.** Para el proceso de autoevaluación es necesario que el usuario realice ciertas consultas a la base de datos. Consultas Inteligentes le permite crear las consultas que necesite, sin limitar su búsqueda.

**5.6.2 Administrar las consultas para su previa modificación.** Dentro del complemento se permitirá guardar, cargar y eliminar una consulta, esto se hace de manera fácil y de forma amigable buscando que el usuario no recurra al experto ni entrar en el código fuente del componente.

**5.6.3 Generar un reporte tabular de la consulta.** Cuando se tiene la consulta definida, se le da la opción al usuario de aplicar consulta, al hacer esto se genera la solución a dicha consulta, esta solución se da en tablas.

**5.6.4 Producir una gráfica.** Ya teniendo las tablas con las consultas deseadas, consultas inteligentes brinda al usuario la posibilidad de escoger que atributos de las tablas quiere utilizar para generar la gráfica.

## **5.7 CARACTERÍSTICAS DEL USUARIO**

Esta aplicación va dirigida a un tipo de usuario, el administrador, este es el encargado de realizar las consultas a la base de datos cuando sea necesario y para el proceso de autoevaluación. Por ello, Consultas Inteligentes deberá ofrecer una interfaz de usuario intuitivo, fácil de aprender y sencillo de manejar.

## **5.8 RESTRICCIONES**

- Proporcionará una interfaz Web, lo cual facilitará su utilización en múltiples plataformas, a menor coste.
- La disponibilidad de la aplicación 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.
- Usará una base de datos no relacional: MongoDB.
- Lenguaje de marcado XHTML.
- Framework web bootstrap y PrimeFaces.

## **5.9 SUPOSICIONES Y DEPENDENCIAS**

- Solo el usuario mencionado anteriormente tendrá acceso al componente por medio del portal de autoevaluación de posgrados.
- Se asume que existirá una conexión a Internet en los equipos donde se desee acceder al sistema.
- Está sujeta al correcto funcionamiento del servidor suministrado por la Universidad Industrial de Santander.

## 5.10 REQUISITOS FUTUROS

Como posibles mejoras para el sistema a implementar serían:

- La versión móvil para Smartphone y Tablet.

## 5.11 REQUERIMIENTOS FUNCIONALES

Tabla 1. Requerimiento Funcional 01 – Crear Consulta

Identificación	RF01
Nombre	Crear Consulta
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad que tiene el administrador de crear una o varias consultas con el nombre deseado. Cada consulta está conformada por uno o varios filtros. Para crear el filtro se debe especificar: la colección a consultar, el campo, un operador lógico y por ultimo asignarle un valor.

Tabla 2. Requerimiento Funcional 02 – Guardar Consulta

Identificación	RF02
Nombre	Guardar Consulta
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad que tiene el administrador de guardar la consulta que crea que va a ser útil para el proceso de autoevaluación.

Tabla 3. Requerimiento Funcional 03 – Cargar Consulta

Identificación	RF03
Nombre	Cargar Consulta
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad que tiene el administrador de cargar las consultas guardadas en la base de datos.

Tabla 4. Requerimiento Funcional 04 – Eliminar Consulta

Identificación	RF04
Nombre	Eliminar Consulta
Prioridad	Alta
	Este requisito hace referencia a la posibilidad que tiene el administrador de

Descripción	eliminar una consulta cuando decide que no va a ser de importancia para el proceso de autoevaluación.
-------------	---

Tabla 5. Requerimiento Funcional 05 – Aplicar Consulta

Identificación	RF05
Nombre	Aplicar Consulta
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad que tiene el administrador de aplicar la consulta, al hacer esto puede obtener el resultado de la consulta ordenado en tablas. Si se consultan varias colecciones, al aplicar la consulta se van a reflejar los resultados de cada colección.

Tabla 6. Requerimiento Funcional 06 – Generar Gráfica

Identificación	RF06
Nombre	Generar Gráfica
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad que tiene el administrador de generar una gráfica con los datos que desee de las colecciones consultadas. Para cada colección la aplicación permite que se produzca su respectiva gráfica con los atributos deseados.

Tabla 7. Requerimiento Funcional 07 – Exportar Datos

Identificación	RF07
Nombre	Exportar Datos
Prioridad	Alta
Descripción	Este requisito hace referencia a la posibilidad que tiene el administrador de exportar los datos en formato Excel. Esto es posible una vez haya aplicado la consulta.

## 5.12 REQUERIMIENTOS NO FUNCIONALES

Tabla 8. Requerimiento No Funcional 01 – Interfaz del Sistema

Identificación	RNF01
Nombre	Interfaz del Sistema
Características	Se presenta una interfaz sencilla para que sea fácil de manejar para cualquier tipo de usuario.
Prioridad	Alta

Descripción	La aplicación contaría con una interfaz de uso sencillo e intuitivo.
-------------	--

Tabla 9. Requerimiento No Funcional 02 – Confiabilidad del Sistema

Identificación	RNF02
Nombre	Confiabilidad del Sistema
Características	El sistema debe estar siempre en correcto 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 requerido por el usuario, garantizando el mejor funcionamiento ante cualquier falla en alguno de sus componentes.

Tabla 10. Requerimiento No Funcional 03 – Seguridad de la información

Identificación	RNF03
Nombre	Seguridad de la información
Características	Garantizar la seguridad de la información a los usuarios.
Prioridad	Alta
Descripción	Proporcionar confianza al usuario respecto al sistema protegiendo la información y los datos que se manejen mediante protocolos de seguridad.

Tabla 11. Requerimiento No Funcional 04 – Arquitectura

Identificación	RNF04
Nombre	Arquitectura
Características	La aplicación debe ser desarrollada para ser utilizada en la web y toda la administración debe realizarse desde un navegador.
Prioridad	Alta
Descripción	La aplicación debe contar con interfaces gráficas de administración y de operación en ambiente web, para que su utilización sea a través de exploradores o navegadores web.

Tabla 12. Requerimiento No Funcional 05 – Ventanas de Confirmación

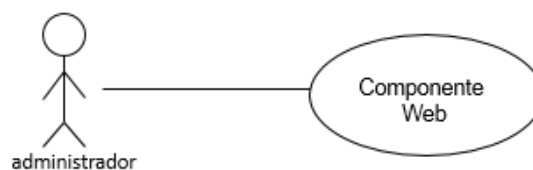
Identificación	RNF05
Nombre	Ventanas de Confirmación
Características	La aplicación debe cerciorarse de que la creación de las consultas y su administración sean clara para el usuario.
Prioridad	Alta
Descripción	Cada vez que se cree una consulta o se modifique alguna, la aplicación mostrara, ventanas de confirmación, estas confirman que los cambios realizados o la creación de consultas se realizaron exitosamente.

Tabla 13. Requerimiento No Funcional 06 – Permitir Pruebas

Identificación	RNF06
Nombre	Permitir Pruebas
Características	El sistema debe contar con facilidad para la identificación de 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.

### 5.13 ACTORES DEL SISTEMA

Figura 4. Actores del Sistema



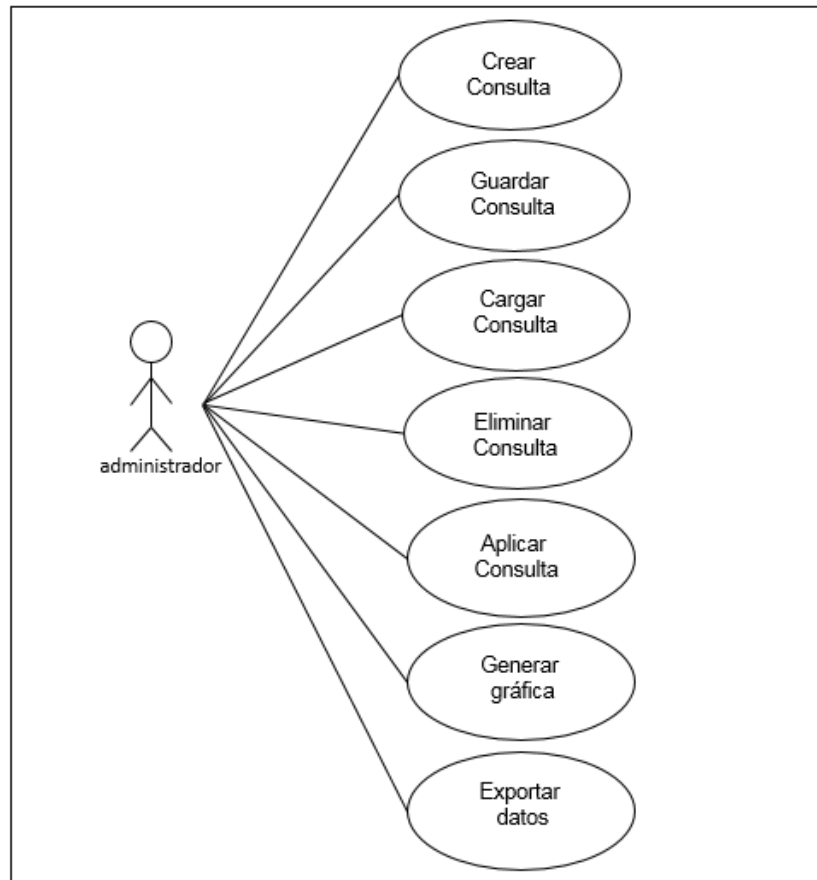
### 5.14 DESCRIPCIÓN DE ACTORES

Tabla 14. Descripción de Actores - Administrador

Actor	Administrador
Caso de Uso	Crear consulta, administrar consultas, aplicar consulta, generar gráfica y exportar datos.
Descripción	Este usuario tiene control para poder acceder a todo el componente web pudiendo realizar todas las operaciones como: crear, administrar y aplicar consultas. También generar gráficas y exportar datos.
Tipo	Principal

## 5.15 DIAGRAMA GENERAL DE CASOS DE USO Y ACTORES

Figura 5. Diagrama general de Casos de Uso y Actores



## 5.16 DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO

Tabla 15. Descripción de Casos de Uso – Crear Consulta

Título del Caso de Uso	Crear Consulta	ID del Caso de Uso	CU 001
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web 'crear consulta'.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			

Ingresar con su respectivo usuario y contraseña al sistema		
Flujo normal de Eventos		
	Administrador	Sistema
1.	Selecciona la opción 'crear consulta'.	Muestra una ventana en la cual está la opción de crear la consulta con un nombre específico.
Condiciones de Salida		
Cerrar la búsqueda o seleccionar la opción 'limpiar'		
Caminos de Excepción/Flujos alternativos		
Ninguno.		

Tabla 16. Descripción de Casos de Uso – Guardar Consulta

Título del Caso de Uso	Guardar Consulta	ID del Caso de Uso	CU 002
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web 'guardar consulta'. Esto ocurre cuando el administrador decide que la consulta creada le va a ser útil en el proceso de autoevaluación.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Ingresar con su respectivo usuario y contraseña al sistema			
Flujo normal de Eventos			
	Administrador	Sistema	
1.	Selecciona la opción 'guardar consulta'.	Muestra una ventana en donde está la opción de 'guardar consulta', al escoger esta opción la consulta será guardada de inmediato en la base de datos. Esta opción se encuentra en la misma ventana de crear consulta.	
Condiciones de Salida			
Cerrar la consulta.			
Caminos de Excepción/Flujos alternativos			
Ninguno.			

Tabla 17. Descripción de Casos de Uso – Cargar Consulta

Título del Caso de Uso	Cargar Consulta	ID del Caso de Uso	CU 003
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web 'cargar consulta', ofrece la posibilidad de ver cuáles son las consultadas que están guardadas en la			

base de datos, evitando volver a crear una consulta existente.		
Entidades/Actores Involucrados		
Administrador		
Condiciones de Entrada		
Ingresar con su respectivo usuario y contraseña al sistema		
Flujo normal de Eventos		
	Administrador	Sistema
1.	Selecciona la opción 'cargar consulta'.	Muestra una lista de todas las consultas guardadas en la base de datos, también ofrece un autocompletado de palabras para ayudar a buscar la consulta que se desea cargar.
Condiciones de Salida		
Cerrar la búsqueda o seleccionar la opción 'limpiar'		
Caminos de Excepción/Flujos alternativos		
Ninguno.		

Tabla 18. Descripción de Casos de Uso – Eliminar Consulta

Título del Caso de Uso	Eliminar Consulta	ID del Caso de Uso	CU 004
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web 'eliminar consulta', al elegir esta opción la consulta será eliminada a su vez de la base de datos.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Ingresar con su respectivo usuario y contraseña al sistema			
Flujo normal de Eventos			
	Administrador	Sistema	
1.	Selecciona la opción 'eliminar consulta'.	Elimina la consulta de la base de datos y refresca la lista de las consultas guardadas.	
Condiciones de Salida			
Cerrar la búsqueda			
Caminos de Excepción/Flujos alternativos			
Ninguno.			

Tabla 19. Descripción de Casos de Uso – Aplicar Consulta

Título del Caso de Uso	Aplicar Consulta	ID del Caso de Uso	CU 005
Descripción General del Caso de Uso o Requerimiento			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web 'Aplicar consulta, al elegir esta opción se da solución a los filtros que conforman la consulta, se consulta en la base de datos quienes cumplen con la condición y se refleja su resultado.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Ingresar con su respectivo usuario y contraseña al sistema			
Flujo normal de Eventos			
	Administrador	Sistema	
1.	Selecciona la opción 'Aplicar consulta'.	En una nueva interfaz, muestra las respectivas tablas de las colecciones consultadas, ofrece la posibilidad de eliminar campos no requeridos en las tablas.	
Condiciones de Salida			
Cerrar la búsqueda			
Caminos de Excepción/Flujos alternativos			
Ninguno.			

Tabla 20. Descripción de Casos de Uso – Generar Gráfica

Título del Caso de Uso	Generar Gráfica	ID del Caso de Uso	CU 006
<b>Descripción General del Caso de Uso o Requerimiento</b>			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web 'generar grafica'. La aplicación permite crear una gráfica por colección, esto quiere decir que si en el momento de aplicar dos colecciones diferentes, el administrador debe generar dos gráficas. La creación de las gráficas no es obligatoria, solo se generan si el administrador lo desea y cree que le va a ser útil para el proceso de autoevaluación.			
<b>Entidades/Actores Involucrados</b>			
Administrador			
<b>Condiciones de Entrada</b>			
Ingresar con su respectivo usuario y contraseña al sistema			
<b>Flujo normal de Eventos</b>			
	Administrador	Sistema	
1.	Selecciona la opción 'grupo'	Muestra una lista con todos los atributos de la colección que se aplicó en la consulta.	
1.1	Selecciona la opción 'generar grafica'	Muestra una ventana en la cual el administrador puede escoger entre tres tipos de gráficas: circular, barras o líneas.	
<b>Condiciones de Salida</b>			
Cerrar la búsqueda			
<b>Caminos de Excepción/Flujos alternativos</b>			
Ninguno.			

Tabla 21. Descripción de Casos de Uso – Exportar Datos

Título del Caso de Uso	Exportar Datos	ID del Caso de Uso	CU 007
<b>Descripción General del Caso de Uso o Requerimiento</b>			
Este caso de uso es iniciado cuando el administrador seleccione la opción que ofrece la aplicación web de exportar datos, esta exportación se hará en formato Excel.			
<b>Entidades/Actores Involucrados</b>			
Administrador			
<b>Condiciones de Entrada</b>			
Ingresar con su respectivo usuario y contraseña al sistema			
<b>Flujo normal de Eventos</b>			
	Administrador	Sistema	
1.	Selecciona 'icono de Excel'.	Muestra un icono de Excel, este icono aparece en la parte izquierda debajo de cada tabla, esto ocurre cada vez que el administrador decide aplicar una consulta.	
<b>Condiciones de Salida</b>			
Cerrar la búsqueda			
<b>Caminos de Excepción/Flujos alternativos</b>			
Ninguno.			

## **6. DISEÑO DE LA APLICACIÓN Y ESTRUCTURA DE DATOS**

Una vez finalizada la fase de análisis de requerimientos se procede a definir los patrones de diseño a utilizar, la estructura de datos, pantallas del sistema entre otros.

### **6.1 DISEÑO DE LA BASE DE DATOS**

Esta aplicación hace parte de la fase dos del proyecto de autoevaluación, por tal manera, se adoptan los criterios de diseño utilizados en la fase uno, los cuales son, la tecnología de base de datos a utilizar y el diseño de la estructura de la base de datos. Estos criterios de diseño son tomados de manera conjunta entre los integrantes de las dos fases del proyecto de autoevaluación, así como estudiantes y egresados a nivel de posgrado.

Se toma la decisión de utilizar una base de datos no relacional (No SQL) la cual presta ciertas ventajas comparándola con una base de datos relacional.

Las principales ventajas o características por las cuales el grupo decidió usar una base de datos no relacional fueron:

- 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.
- Las bases de datos no relacionales (No SQL) facilitan la manera de 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.

- Incluye un esquema dinámico, con esta propiedad las bases de datos No SQL permiten la flexibilidad para evolucionar el esquema de datos.
- Con una estructura dinámica, los datos no tiene una definición de atributos rígida, cada registro puede contener una información con diferente forma brindando la posibilidad almacenar solo los atributos que sean de interés.

Para la creación de las entidades que hacen parte de la base de datos se hizo una reunión con los diferentes actores involucrados en el proyecto de autoevaluación, en los que se encontraban, estudiantes de pregrado, posgrado, la directora del proyecto y administrativos relacionados con el proceso de autoevaluación en la escuela de ingeniería de sistemas de la Universidad Industrial de Santander.

Después de llegar a un consenso entre las personas involucradas en el proceso de autoevaluación, se determinaron las siguientes colecciones iniciales como: PERSONAS (podrían ser administrativos, profesores o estudiantes), GRUPOS\_INVESTIGACIÓN (todos los grupos que conforman la maestría), LINEAS\_INVESTIGACIÓN, PROPIEDAD\_INTELECTUAL Y PRORAMA\_ACADEMICO. Después de haber determinado las principales colecciones, se hizo una revisión para definir las principales características, en este proceso se dio cuenta que no solamente existían estas colecciones sino que existían colecciones embebidas dentro de estas como: ESTUDIOS, EXPERIENCIA\_PROFESIONAL, DISTINCIONES e IDIOMAS.

## 6.2 ESQUEMA DE LA BASE DE DATOS

Figura 6. Esquema de la base de datos – Tablas Personas, Estudios y Experiencia\_Profesional

PERSONAS		
PK	ID	VARCHAR(99)
	TIPO	VARCHAR(99)
	NOMBRES	VARCHAR(99)
	APELLIDOS	VARCHAR(99)
	FECHA_NACIMIENTO	DATETIME
	CIUDAD_NACIMIENTO	VARCHAR(99)
	PAIS_NACIMIENTO	VARCHAR(99)
	GENERO	VARCHAR(99)
	ESTADO_CIVIL	VARCHAR(99)
	NUMERO_HUOS	INTEGER
	INGRESOS_FAMILIARES	VARCHAR(99)
	TIPO_VIVIENDA	VARCHAR(99)
	DIRECCION	VARCHAR(99)
	ESTRATO	VARCHAR(99)
	CIUDAD_RESIDENCIA	VARCHAR(99)
	CORREO_ELECTRONICO	VARCHAR(99)
	TELEFONO_CONTACTO	VARCHAR(99)

ESTUDIOS		
PK	ID	VARCHAR(99)
	NIVEL	VARCHAR(99)
	TITULO	VARCHAR(99)
	PROMEDIO_ACADEMICO	DOUBLE
	FECHA_EGRESO	DATETIME
	UNIVERSIDAD	VARCHAR(99)
	TRABAJO_GRADO	VARCHAR(99)

EXPERIENCIA_PROFESIONAL		
PK	ID	VARCHAR(99)
	EMPRESA	VARCHAR(99)
	FECHA_EGRESO	VARCHAR(99)
	PERMANENCIA	VARCHAR(99)
	CARGO	VARCHAR(99)
	MOTIVO_RETIRO	VARCHAR(99)
	SALARIO_RANGO	VARCHAR(99)

Figura 7. Esquema de la base de datos–Programa\_Academico, Distinciones y Propiedad\_Intelectual

PROGRAMA_ACADEMICO		
PK	ID	VARCHAR(39)
	NOMBRE	VARCHAR(39)
	OBJETIVO_GENERAL	VARCHAR(39)
	OBJETIVOS_ESPECIFICOS	VARCHAR(39)
	MISION	VARCHAR(39)
	VISION	VARCHAR(39)
	NIVEL_ACADEMICO	VARCHAR(39)
	UNIDAD_ACADEMICA	VARCHAR(39)
	COD_SNIES	INTEGER
	AÑO_INICIO	DATETIME
	TITULO	VARCHAR(39)
	DURACION	INTEGER
	PLAN_ESTUDIOS	VARCHAR(39)
	HORARIO_CLASE	VARCHAR(39)
	NUM_COHORTES	INTEGER
	NUM_EGRESADOS	INTEGER
	CUPOS_SEMESTRE	INTEGER
	VALOR_INSCRIPCION	DOUBLE
	VALOR_MATRICULA	DOUBLE
	PERFIL_ASPIRANTE	VARCHAR(39)
	PERFIL_EGRESADO	VARCHAR(39)
	DIRECCION	VARCHAR(39)
	TELEFONO	DOUBLE
	CORREO_ELECTRONICO	VARCHAR(39)
	HARARIO_ATENCION	VARCHAR(39)
	REGLAMENTO_POSGRADO	VARCHAR(39)
	REGLAMENTO_INVESTIG	VARCHAR(39)
	FORMATOS	VARCHAR(39)

DISTINCIONES		
PK	ID	VARCHAR(99)
	NOMBRE	VARCHAR(99)
	MOTIVO	VARCHAR(99)
	FECHA	DATETIME
	INSTITUCION_OTORGA	VARCHAR(99)

PROPIEDAD_INTELCTUAL		
PK	ID	VARCHAR(99)
	TITULO	VARCHAR(99)
	AUTORES	VARCHAR(99)
	TIPO_PUBLICACION	VARCHAR(99)
	FECHA_PUBLICACION	DATETIME
	PALABRAS_CLAVES	VARCHAR(99)
	REVISTA_LIBRO	VARCHAR(99)
	CIUDAD	VARCHAR(99)
	PAGINAS	INTEGER
	PROYECTO_INVESTIG	VARCHAR(99)

Figura 8. Esquema de la base de datos – Grupos\_Investigacion y Proyectos\_Investigacion

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)

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_INVESTIVACION	VARCHAR(99)
	OBJETIVO_PROGRAMA	VARCHAR(99)

Figura 9. Esquema de la base de datos – Lineas\_Investigacion e Idiomas

LINEAS_INVESTIGACION		
PK	ID	VARCHAR(99)
	NOMBRE	VARCHAR(99)
	DESCRIPCION	VARCHAR(99)
	TIPO_LINEAS	VARCHAR(99)

IDIOMAS		
PK	ID	VARCHAR(99)
	NOMBRES	VARCHAR(99)
	NIVEL_LECTURA	VARCHAR(99)
	NIVEL_ESCUCHA	VARCHAR(99)
	NIVEL_ESCRITURA	VARCHAR(99)
	NIVEL_HABLA	VARCHAR(99)
	CERTIFICADO	VARCHAR(99)
	PUNTAJE_CLASIFICACION	VARCHAR(99)

Figura 10. Esquema de la base de datos – Filtros y Busqueda

FILTROS		
PK	ID	VARCHAR(99)
	NOMBRE	VARCHAR(99)
	COLLECTION	VARCHAR(99)
	PROPERTY	VARCHAR(99)
	OPERATOR	VARCHAR(99)
	VALUE	VARCHAR(99)

BUSQUEDA		
PK	ID	VARCHAR(99)
	NOMBRE	VARCHAR(99)

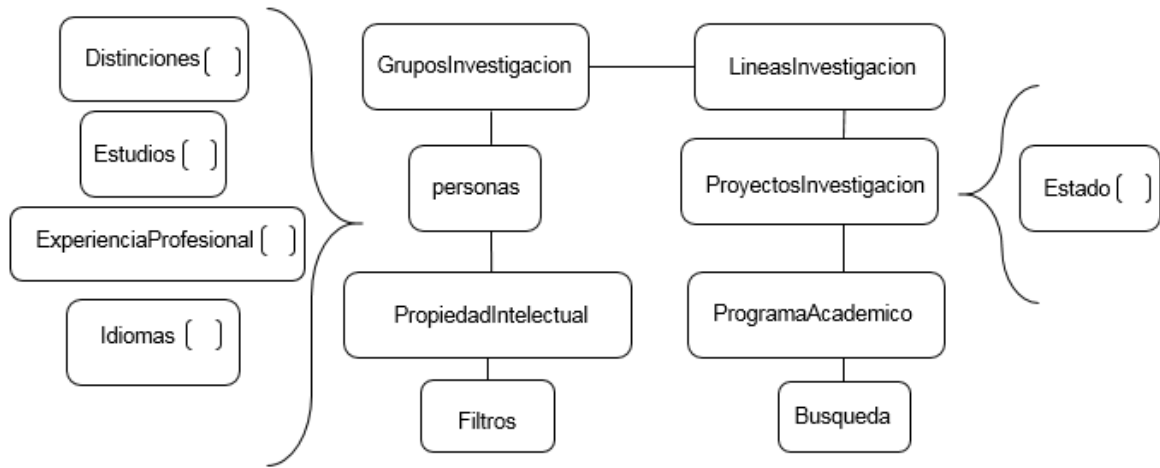
La estructura de datos utilizada, cuenta con las siguientes colecciones de la base de datos no relacional MongoDB las cuales representan las diferentes entidades relacionadas en el proyecto.

- **BUSQUEDA:** Entidad asociada a la colección Busqueda
- **CONVENIOS:** Entidad asociada a la colección Convenios
- **DISTINCIONES:** Entidad almacenada como un atributo en la entidad Personas (@Embedded) y se guarda como un array de la colección personas en Mongodb
- **ESTADO:** Entidad almacenada como un atributo en la entidad ProyectoInvestigacion (@Embedded) y se guarda como un array de la colección proyectosInvestigación en Mongodb
- **ESTUDIOS:** Entidad almacenada como un atributo en la entidad Personas (@Embedded) y se guarda como un array de la colección personas en Mongodb
- **EXPERIENCIAPROFESIONAL:** Entidad almacenada como un atributo en la entidad Personas (@Embedded) y se guarda como un array de la colección personas en Mongodb
- **FILTROS:** Entidad asociada a la colección Filtros
- **GRUPOSINVESTIGACION:** Entidad asociada a la colección gruposInvestigación

- IDIOMAS: Entidad almacenada como un atributo en la entidad Personas (@Embedded) y se guarda como un array de la colección personas en MongoDB
- LINEASINVESTIGACION: Entidad asociada a la colección líneasInvestigación.
- NOTICIAS: Entidad asociada a la colección noticias
- PERSONAS: Entidad asociada a la colección personas
- PROGRAMAACADEMICO: Entidad asociada a la información del programa académico
- PROPIEDADINTELECTUAL: Entidad asociada a la colección propiedadIntelectual
- PROYECTOSINVESTIGACION: Entidad asociada a la colección proyectosInvestigación

Teniendo en cuenta que las bases de datos no relaciones no poseen el concepto de integridad referencial, las relaciones que se presentan en el siguiente modelo conceptual representan referencias existentes entre relaciones. Las llave abierta o “{” representa una colección que agrupa otras colecciones de objetos, la cuales se denominan colecciones embebidas, y las llaves cerradas “}” representan que dichas colecciones se comportan como arreglos que hacen parte de la clase en la cual están embebidas.

Figura 11. Modelo conceptual de la base de datos



### 6.3 DISEÑO DE ENTIDADES UTILIZANDO MORPHIA FRAMEWORK

Se utiliza este *framework*, para la creación de las diferentes clases de las entidades definidas anteriormente, ya que toma las entidades @Entity (personas, grupos de investigación, etc.) y las convierte en una clase, cada clase contiene los atributos que corresponden a los campos alojados en la base de datos previamente definidos, *morphia* agrega a estos atributos un id y una versión.

Con el fin de facilitar el acceso a las propiedades de las entidades y la reducción del código, se creó una clase en común para alojar los métodos que son comunes para todas las entidades, llamada BaseEntity, lo cual facilita usar las propiedades de la herencia.

Figura 12. Diagrama de BaseEntity

```
package filoen.autoevaluacion.model;

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

public class BaseEntity {

    @Id
    protected ObjectId id;

    @Version
    protected long version;

    @Transient
    protected String className;

    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;
    }

    public String getClassName() {
        return className;
    }
}
```

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

Figura 13. Diagrama de la @Entidad Personas

```
@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 String codigoUsuario;
    private int codigoEstudiante;

    private Date fechaNacimiento;
    private String paisNacimiento;
    private String genero;
    private String estadoCivil;
    private int numeroHijos;

    @Embedded
    private List<Estudios> estudios;

    @Embedded
    private List<Idiomas> idiomas;

    @Embedded
    private List<ExperienciaProfesional> experienciaLaboral;

    @Embedded
    private List<Distinciones> distinciones;
```

## 6.4 ORGANIZACIÓN DE LOS PAQUETES

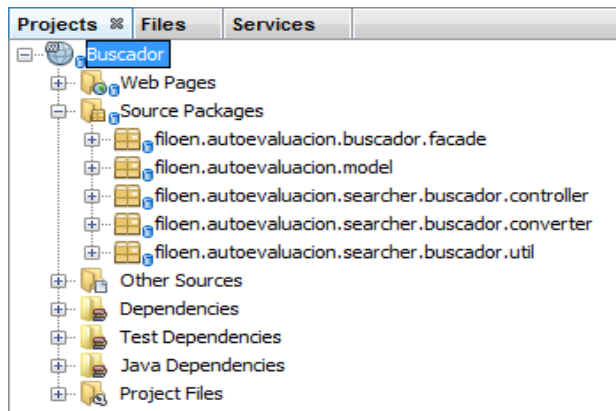
Una de las técnicas utilizadas para la implementación es la arquitectura EJB, el desarrollo de la aplicación hará parte de un framework que se desarrollará en diversos proyectos. De acuerdo con esto, los paquetes del proyecto quedan agrupados en el paquete *filoen.autoevaluacion*, que está organizado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

- Paquete autoevaluación.model: contiene las colecciones (@Entity) a utilizar.

- Paquete buscador.facade: contiene los objetos de acceso a datos. (@SessionBean).
- Paquete buscador.controller: contiene todas las clases requeridas para la administración del *backend* de la herramienta.

A continuación se mostrara parte del código en donde se pueden apreciar los paquetes anteriormente mencionados

Figura 14. Estructura de Paquetes en NetBeans

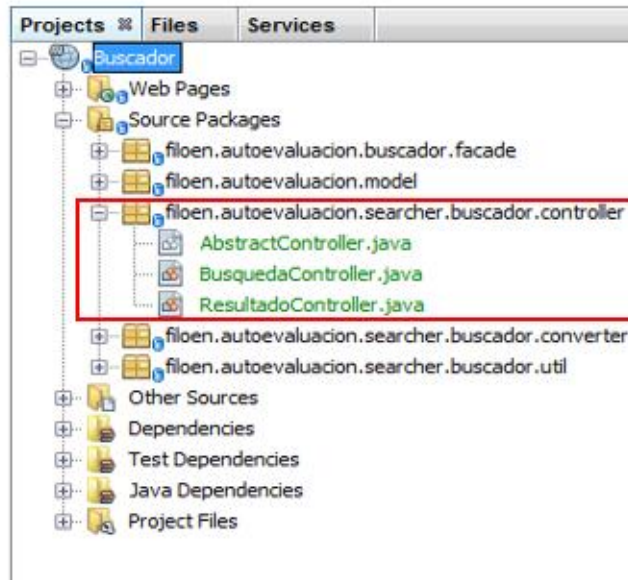


## 6.5 CLASES DE LÓGICA DE NEGOCIO (CONTROLLER)

El paquete `filoen.autoevaluacion.searcher.buscador.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 15. Vista del paquete controller en NetBeans



## 7. DESARROLLO

Para dar comienzo al proceso de desarrollo e implementación de la aplicación se acordó utilizar ciertos patrones o arquitecturas con el fin del que el proyecto fuese más organizado y estandarizar el proceso de programación para que en el futuro sea más fácil su mantenimiento, a continuación se detallan los patrones usados en la construcción del componente.

### 7.1 MODELO VISTA CONTROLADOR

Está definido como un patrón de arquitectura del software, se basa en separar los datos, la lógica y las interfaces de usuario, por esta razón se determina conveniente la utilización de este patrón de diseño debido a que ayuda en la organización y estandarización de todo el proceso que realiza el componente.

Este 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, con estas características lo que se busca es facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

### 7.2 DESCRIPCIÓN DE LOS COMPONENTES

**7.2.1 Modelo.** En este componente 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.

**7.2.2 Vista.** Es todo lo referente a la interfaz gráfica de usuario la cual permite mostrar/visualizar al usuario la información que envía el modelo.

**7.2.3 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. Se podría decir que es el componente que actúa de intermediario entre la vista y el modelo.

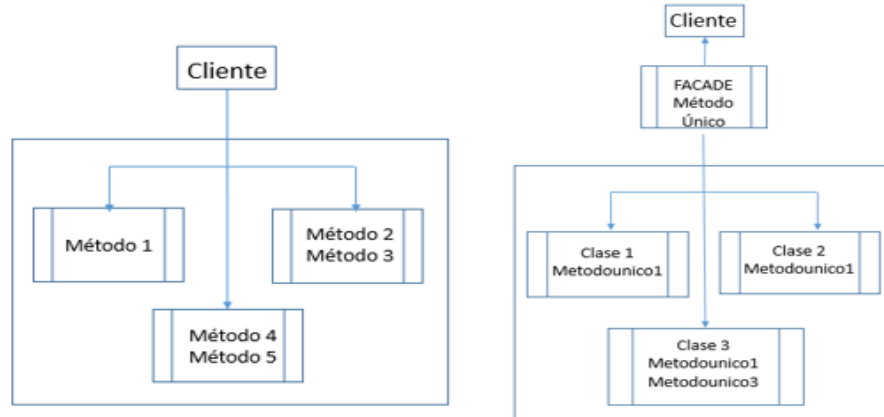
### 7.3 USO DE PATRONES DE DISEÑO: FACADE

Es un tipo de patrón estructural cuya finalidad es la de reducir la complejidad del código fuente llevando a cabo una división en subsistemas, así pues logrando 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.

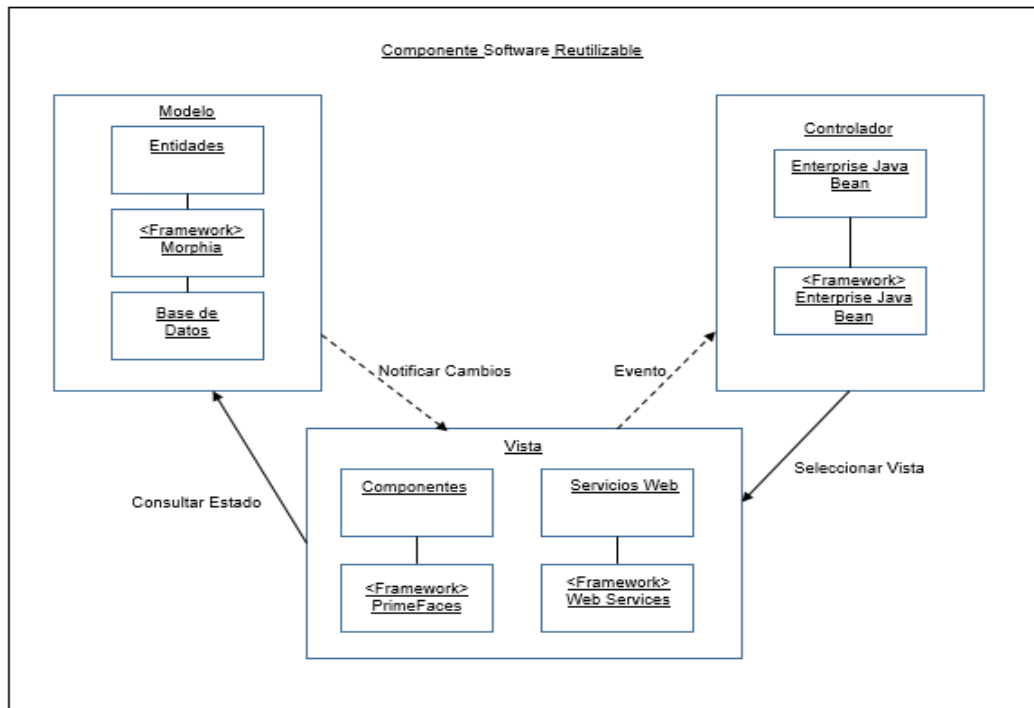
A continuación se representa un diagrama con la estructura del patrón de diseño. En el ejemplo se utiliza el patrón a través de la clase Facade la cual crea un método único que comparten la clase 1, clase 2 y clase 3, este resume o facilita ya que el cliente solo llama el método único.

Figura 16. Estructura sin clase Facade vs. Estructura con clase Facade



## 7.4 INTERACCIÓN DE LOS COMPONENTES

Figura 17. Interacción de los componentes



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.

Figura 18. Bosquejo Final de la página principal

Búsqueda

## Consulta Crear consultas dinámicas

[Crear Consulta](#)

Consultas

Consulta:

Filtros

Colección	Campo	Operador	Valor	Opciones
No hay filtros.				

## 8. PRUEBAS

Las pruebas unitarias enfocan sus esfuerzos a la verificación de las unidades más pequeñas del software diseñado – un componente software o un módulo <sup>4</sup>

Para la realización de las pruebas unitarias fueron seleccionados los métodos `BusquedaController` y `Resultado Controller` de la clase `com.searcher.buscador.facade.FiltroFacade`

Las pruebas se realizaron haciendo uso de una base de datos no relacional, MongoDB, con unos valores de referencia.

Para el método `BusquedaController` se definieron las siguientes pruebas:

Tabla 22. Pruebas Funcionalidad – Existencias de Consultas

IDENTIFICADOR	TITULO	DESCRIPCION	VALORES DE ENTRADA	RESULTADOS ESPERADOS
Prueba1	Busqueda de consultas en la base de datos vacía	En ausencia de consultas guardadas en la base de datos, la búsqueda no debería retornar ningún valor	Prueba	void
Prueba2	Busqueda de una consulta existente en la base de datos	Se realizara la búsqueda de una consulta existente en la base de datos, la cual fue previamente guardada.	Prueba1	Arraylist
Prueba3	Busqueda de una consulta que no existe en la base de datos	Se realizara la búsqueda de una consulta que no existe en la base de datos, es decir que no haya sido previamente guardada.	Prueba0	null

<sup>4</sup> Libro Software Engineering A pracioner's Approach de Pressman, Roger S página 456

Para el método ResultadoControler se definieron las siguientes pruebas:

Tabla 23. Pruebas Funcionalidad – Filtros

IDENTIFICADOR	TITULO	DESCRIPCION	VALORES DE ENTRADA	RESULTADOS ESPERADOS
Prueba4	Aplicación de filtros de una consulta que genera resultados	Para la consulta 'Prueba1' se crearon unos filtros y se aplicó la consulta	Colección: personas Campo: Genero Operador: igual Valor: Masculino Y Colección: gruposInvestigacion Campo: categoría Operador: igual Valor: A	Cadena de Caracteres (String)
Prueba5	Aplicación de filtros de una consulta que no genera resultados	Se creó una consulta con el nombre de 'Prueba2'	Colección: personas Campo: Genero Operador: igual Valor: Masculino Y Colección: personas Campo: genero Operador: igual Valor: Femenino	null

Los resultados obtenidos al ejecutar las pruebas se listan a continuación:

Fecha de Realización: 09/09/2014

Tabla 24. Pruebas Funcionalidad – Resultados

IDENTIFICADOR	TITULO	RESULTADOS
Prueba1	Busqueda de consultas en la base de datos vacía	Exitosa
Prueba2	Busqueda de una consulta existente en la base de datos	Fallida
Prueba3	Busqueda de una consulta que no existe en la base de datos	Exitosa
Prueba4	Aplicación de filtros de una consulta que genera resultados	Fallida
Prueba5	Aplicación de filtros de una consulta que no genera resultados	Exitosa

Se realizaron las acciones correctivas correspondientes a las pruebas fallidas.

Fecha de Realización: 06/10/2014

Tabla 25. Pruebas Funcionalidad – Resultados Finales

IDENTIFICADOR	TITULO	RESULTADOS
Prueba01	Busqueda de consultas en la base de datos vacía	Exitosa
Prueba02	Busqueda de una consulta existente en la base de datos	Exitosa
Prueba03	Busqueda de una consulta que no existe en la base de datos	Exitosa
Prueba04	Aplicación de filtros de una consulta que genera resultados	Exitosa
Prueba05	Aplicación de filtros de una consulta que no genera resultados	Exitosa

Todas las pruebas se ejecutaron correctamente.

Para realizar las pruebas de usabilidad del componente se realizó una entrevista con la persona encargada del proceso de autoevaluación y por ende del sitio web.

Las preguntas se califican con un valor numérico de 1 a 5, siendo 1 el valor más bajo y 5 el valor más

Fecha de Prueba: 17 de Octubre 2014

Tabla 26. Prueba de Usabilidad

PRUEBA DE USABILIDAD					
	1	2	3	4	5
1. Facilidad al usar la aplicación					x
2. Utilidad de la aplicación				x	
3. Datos arrojados son confiables					x

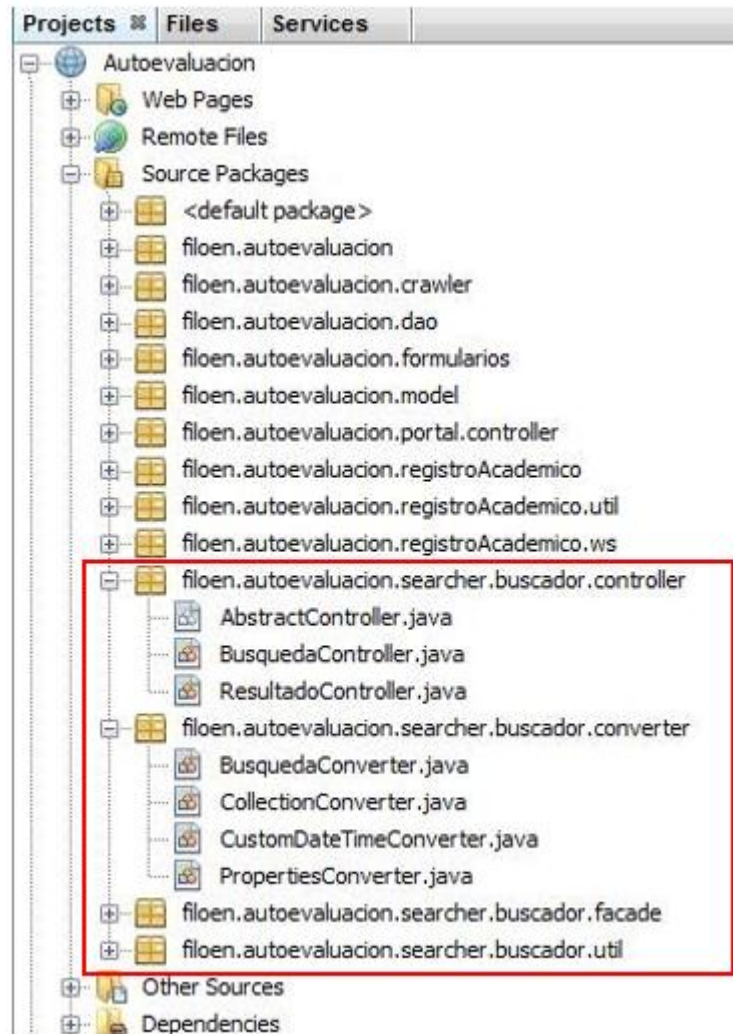
Como posibles mejoras indica que:

- La base de datos utilizada en el proyecto de autoevaluación este enlazada a la base de Datos del departamento de admisiones de la Universidad Industrial de Santander, para poder obtener datos como: promedio, materias cursadas, etc.
- La creación de una colección de egresados en la base de datos.

## 9. INTEGRACIÓN DE LA APLICACIÓN

En la figura 19. Se pueden observar los paquetes que pertenecen a la aplicación web – Buscador – integrada al proyecto de Autoevaluación realizado en la primera fase.

Figura 19. Integración de la aplicación web al proyecto Autoevaluación



## 10. CONCLUSIONES

En la etapa de análisis, se han especificado los requerimientos principales del sistema, los cuales consisten en la creación de consultas, la administración de las mismas, para su previa modificación, generando un reporte tabular y si es necesario producir una gráfica, usados por el actor principal, el Administrador

Esta aplicación hace parte de la fase dos del proyecto de autoevaluación, por tal manera, se adoptan los criterios de diseño utilizados en la fase uno, los cuales son, la tecnología de base de datos a utilizar y el diseño de la estructura de la base de datos.

Se utiliza GitLab como administrador de repositorios, permitiendo la integración de los demás proyectos que hacen parte del conjunto de autoevaluación de programas de posgrado y mantener copias de seguridad alojadas en la nube en caso de presentarse cualquier eventualidad.

Se verificó el correcto funcionamiento de la aplicación, donde se realizaron pruebas de funcionalidad y usabilidad, estas se realizaron con los datos que se encuentran guardados en la base de datos utilizada en la primera fase del proyecto de autoevaluación e incorporada a este proyecto, cumpliendo con las necesidades establecidas en los requerimientos del producto.

## **11. SUGERENCIAS Y RECOMENDACIONES**

Para que la aplicación tenga una completa utilidad, es recomendable que la base de datos del programa de posgrado utilizada sea enlazada con la base de datos del departamento de admisiones de la Universidad Industrial de Santander, esto con el fin de tener más datos en el momento de la creación de las consultas.

Se sugiere definir unas políticas que permitan controlar los cambios realizados en los componentes y que pueden afectar el funcionamiento del sistema que actualmente los implementa, esta política debería también adaptarse a los cambios en las estructuras de datos y entidades de uso general.

Partiendo del hecho que hoy en día las Aplicaciones Móviles hacen parte de la cotidianidad de las personas, se hace necesario explorar y desarrollar una aplicación que pueda brindar nuevos servicios al proceso de autoevaluación.

## BIBLIOGRAFIA

ÇAĞATAY, Çivici. Data Table and properties. Primefaces User guide 4.0. US. 2012.

\_\_\_\_\_. Theme switcher. PrimeFaces User guide 4.0. US. 2012.

GAUCHET, T. SQL Server 2008 R2 Implementación y Despliegue de una solución de Business Intelligence. Cornellà de Llobregat. Barcelona: ENI. 2011

MKYOUNG. JSF 2.0 and Resource Bundles example. US. [en línea]. 2010. [Consultado Junio 2014]. Disponible en: <http://www.mkyong.com/jsf2/jsf-2-0-and-resource-bundles-example/>

MONGE ROCERO, P. y RUIZ CASTRILLO, D. Programación Distribuida: El Modelo EJB. Departamento de Informática y Automática. España. 2005

MONGO, DB. Getting Started with Java Driver. US. [en línea]. 2011 [Consultado Abril 2014], disponible en: <http://docs.mongodb.org/ecosystem/tutorial/getting-started-with-java-driver/>

\_\_\_\_\_. Install MongoDB on Windows. US. [en línea]. 2011. [Consultado Abril 2014]. Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>

PALERMO, Rolando. Introducción a los EJB [en línea]. Science and Engineering blog. Peru. 2012 [Consultado Mayo 2014], disponible en: <http://blog.rolandopalermo.com/2012/03/ejb-30-netbeans-java.html>

PRESSMAN, Roger y MAXIM, Bruce. Software Engineering a practitioner's Approach. 7 ed. New York: McGraw-Hill: Inc. 2009.

UNIVERSIDAD DE ALICANTE. Especialista Universitario Java Enterprise . ¿Qué es Java Enterprise? [en línea]. España. 2006 [Consultado Mayo 2014]. Disponible en: <http://www.jtech.ua.es/j2ee/2006-2007/jee.html>