

**APLICACIÓN WEB PARA CREAR ENCUESTAS PARA AUTOEVALUACIÓN DE
PROGRAMAS DE POSGRADO**

DAVID JESÚS VÁSQUEZ RIOS

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

2015

**APLICACIÓN WEB PARA CREAR ENCUESTAS PARA AUTOEVALUACIÓN DE
PROGRAMAS DE POSGRADO**

DAVID JESÚS VÁSQUEZ RIOS

**Trabajo de Grado para optar al título de
Ingeniero de Sistemas**

Directora

PhD. SONIA CRISTINA GAMBOA SARMIENTO

Doctora en Educación

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

2015

DEDICATORIA

Este logro lo dedico principalmente a Dios, quien me ha dado la vida y la sabiduría para hacerlo posible, Él me ha dotado de cualidades para lograr grandes cosas.

A mis padres José de Jesús Vásquez Mendez y Nubia Cecilia Rios Dueñas, quienes con amor y gran esfuerzo me han ofrecido más de los elementos necesarios para cumplir mis metas.

A mis hermanos y mi novia, su apoyo incondicional, compañía y motivación fueron piezas clave en este logro.

AGRADECIMIENTOS

Agradezco a mi familia y mi novia, que siempre estuvieron presentes cuando necesité de su ayuda. Su apoyo voluntario, compañía, confianza y motivación me impulsaron a lograr esta meta.

A la profesora Sonia Cristina Gamboa Sarmiento, por la orientación aportada en mis últimos semestres de carrera, los conocimientos compartidos, la disposición y dedicación para apoyar el cumplimiento de los objetivos propuestos. Por su confianza en las oportunidades dadas.

A la Escuela de Ingeniería de Sistemas e Informática y las personas que trabajan para mantener sus funciones y para ofrecer los servicios que contribuyen a la formación de más profesionales.

Al conjunto de profesores de la escuela por su excelente formación, exigencia y conocimientos compartidos.

A mis amigos y compañeros que compartieron este proceso conmigo, quienes se alegraban en mis buenos momentos y en los malos me apoyaban.

CONTENIDO

INTRODUCCIÓN	16
1. JUSTIFICACIÓN.....	18
2. OBJETIVOS.....	19
2.1 OBJETIVO GENERAL.....	19
2.2 OBJETIVOS ESPECÍFICOS	19
3. MARCO DE REFERENCIA.....	21
3.1 ACREDITACIÓN DE ALTA CALIDAD	21
3.2 CONSEJO NACIONAL DE EDUCACIÓN SUPERIOR	22
3.3 SISTEMA NACIONAL DE ACREDITACIÓN (SNA)	23
3.4 CONSEJO NACIONAL DE ACREDITACIÓN (CNA)	23
3.5 JAVA EE 6.....	25
3.6 FRAMEWORKS Y HERRAMIENTAS A UTILIZAR.....	26
3.6.1 JavaServer Faces (JSF 2.2)	27
3.6.2 Primefaces 5.0.....	27
3.6.3 Morphia 0.108.....	28
3.6.4 Bootstrap v3.3.1.....	28
3.7 Enterprise JavaBeans (EJB).....	28
3.8 MongoDB 2.4.12.....	29
4. METODOLOGÍA	32
4.1 PLAN DE TRABAJO	33
4.1.1 Etapa de análisis	34
4.1.2 Etapa de diseño	34
4.1.3 Etapa de implementación	35
4.1.4 Etapa de pruebas	35
5. ANÁLISIS, REQUISITOS Y CASOS DE USO	37
5.1 ESPECIFICACIÓN DE REQUISITOS	37

5.2 PROPÓSITO	37
5.3 ALCANCE DEL SISTEMA	37
5.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	38
5.5 PERSPECTIVA DEL PRODUCTO	39
5.6 FUNCIONES DEL PRODUCTO	39
5.6.1 Crear preguntas.	39
5.6.2 Generar encuestas.	39
5.6.3 Enviar encuestas por correo.	39
5.6.4 Almacenar preguntas resueltas.	39
5.6.5 Mostrar resultados.	39
5.7 CARACTERÍSTICAS DEL USUARIO	40
5.8 RESTRICCIONES	40
5.9 SUPOSICIONES Y DEPENDENCIAS	40
5.10 REQUISITOS FUTUROS	41
5.11 REQUISITOS FUNCIONALES	41
5.12 REQUISITOS NO FUNCIONALES	44
5.13 ACTORES DEL SISTEMA	46
5.14 DESCRIPCIÓN DE ACTORES	46
5.15 DIAGRAMA DE CASOS DE USO	47
5.16 DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO	48
6. DISEÑO DE LA APLICACIÓN Y ESTRUCTURA DE DATOS	53
6.1 DISEÑO DE LA BASE DE DATOS	53
6.2 ESQUEMA DE LA BASE DE DATOS	54
6.3 DISEÑO DE LAS ENTIDADES	58
6.4 ORGANIZACIÓN DE LOS PAQUETES	59
6.5 CLASES DE LÓGICA DE NEGOCIO	61
7. DESARROLLO	62
7.1 MODELO VISTA CONTROLADOR (MVC)	62
7.2 DESCRIPCIÓN DE COMPONENTES	62
7.2.1 Modelo	62

7.2.2 Vista.....	63
7.2.3 Controlador.....	63
7.3 USO DE PATRONES DE DISEÑO: SINGLETON.....	63
7.4 INTERACCIÓN DE LOS COMPONENTES.....	65
8. PRUEBAS.....	66
9. INTEGRACIÓN DE LA APLICACIÓN.....	70
10. CONCLUSIONES.....	71
11. SUGERENCIAS Y RECOMENDACIONES.....	72
BIBLIOGRAFÍA.....	73
ANEXOS.....	75

LISTA DE TABLAS

Tabla 1. Requisito Funcional 01 – Crear Pregunta.....	41
Tabla 2. Requisito Funcional 02 – Modificar Pregunta	41
Tabla 3. Requisito Funcional 03 – Eliminar Pregunta.....	42
Tabla 4. Requisito Funcional 04 - Asociar Pregunta.....	42
Tabla 5. Requisito Funcional 05 – Consultar Preguntas.....	42
Tabla 6. Requisito Funcional 06 – Generar Encuestas.....	43
Tabla 7. Requisito Funcional 07 – Enviar Encuestas	43
Tabla 8. Requisito Funcional 08 – Registrar Respuestas	43
Tabla 9. Requisito Funcional 09 – Consultar Resultados	44
Tabla 10. Requisito No Funcional 01 – Veracidad de la Información	44
Tabla 11. Requisito No Funcional 02 – Interfaces del Sistema	44
Tabla 12. Requisito No Funcional 03 – Análisis de Respuestas.....	45
Tabla 13. Requisito No Funcional 04 – Desempeño del Sistema.....	45
Tabla 14. Requisito No Funcional 05 – Confiabilidad del Sistema	45
Tabla 15. Requisito No Funcional 06 – Seguridad de la Información	46
Tabla 16. Descripción de Actores - Administrador.....	46
Tabla 17. Descripción de Actores - Participante.....	47
Tabla 18. Descripción de Casos de Uso – Crear Pregunta	48
Tabla 19. Descripción de Casos de Uso – Eliminar Pregunta	48
Tabla 20. Descripción de Casos de Uso – Editar Pregunta.....	49
Tabla 21. Descripción de Casos de Uso – Generar Encuestas.....	49
Tabla 22. Descripción de Casos de Uso – Enviar Encuestas.....	50
Tabla 23. Descripción de Casos de Uso – Resolver Encuestas.....	51
Tabla 24. Descripción de Casos de Uso – Ver Resultados	51
Tabla 25. Pruebas de funcionalidad – Módulo Preguntas	66
Tabla 26. Pruebas de funcionalidad – Módulo Encuestas.....	67

Tabla 27. Pruebas de funcionalidad – Módulo Resultados.....	68
Tabla 28. Pruebas de funcionalidad – Resultados finales	68
Tabla 29. Prueba de usabilidad.....	69

LISTA DE FIGURAS

Figura 1. Uso de referencias a enlaces de documentos.....	30
Figura 2. Campos embebidos que contienen información relacionada.	31
Figura 3. Modelo en Cascada.....	33
Figura 4. Actores del Sistema.....	46
Figura 5. Diagrama de Casos de Uso.....	47
Figura 6. Esquema de la base de datos – Colección Personas	55
Figura 7. Esquema de la base de datos – Colección Elemento	56
Figura 8. Esquema de la base de datos – Colección Encuesta.....	56
Figura 9. Esquema de la base de datos – Colección Respuesta	57
Figura 10. Segmento de la entidad Encuesta.....	58
Figura 11. Documento de una encuesta almacenada en MongoDB.	59
Figura 12. Organización de paquetes JAVA en NetBeans	60
Figura 13. Paquete filoen.autoevaluacion.controller	61
Figura 14. Diagrama de la clase MongoResource con Singleton	64
Figura 15. Interacción de los Componentes	65
Figura 16. Integración de la aplicación al proyecto Autoevaluación	70

LISTA DE ANEXOS

Anexo A: Manual de usuario de la aplicación	75
Anexo B: Diagrama de clases UML del modelo de datos	101

RESUMEN

TÍTULO: APLICACIÓN WEB PARA CREAR ENCUESTAS PARA AUTOEVALUACIÓN DE PROGRAMAS DE POSGRADO*

AUTOR:

David Jesús Vásquez Rios**

PALABRAS CLAVE:

Crear encuestas, autoevaluación, aplicación web, MongoDB, encuesta personalizada.

DESCRIPCIÓN:

Los programas académicos de postgrado 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 factores, características, 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.

En la Universidad Industrial de Santander, los programas de posgrado buscan constantemente mejorar sus procesos de autoevaluación para que sean más eficaces y efectivos al momento de optar por la acreditación de alta calidad. Los procesos de autoevaluación son soportados por diferentes instrumentos que incentivan la participación de las personas que hacen parte del programa. Generalmente, el instrumento utilizado para este proceso son las encuestas, ya que permiten recolectar información proporcionada por los mismos participantes del proceso y a su vez generan resultados de fácil interpretación para la persona encargada de este proceso.

Este proyecto tiene por objeto diseñar una aplicación web que permita generar y automatizar encuestas dirigidas al perfil de cada persona dentro de la comunidad académica de un programa de posgrado, que afecten directamente los factores y variables de un sistema de autoevaluación automatizado.

* Trabajo de grado

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

ABSTRACT

TITLE: WEB APPLICATION TO CREATE SURVEYS TO SELF-ASSESSMENT FOR GRADUATE PROGRAMS*

AUTHOR:

David Jesús Vásquez Ríos**

KEYWORDS:

Create Surveys, self-assessment, web application, MongoDB, custom survey.

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 factors, characteristics 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 aforementioned variables are controlled.

At Industrial University of Santander, graduate programs constantly try to improve self-assessment processes to be more efficient and effective at the time of opting for high quality accreditation. Self-assessment processes are supported by different instruments that encourage the participation of people who are part of the program. Generally, the instrument used for this process are surveys, because they allow collect information provided by the participants of the process and generates results easy to interpret for the person responsible for this process.

This project aims to design a web application to generate and automate surveys targeted to the profile of each person in the academic community of a graduate program, which directly affect the factors and variables of an automated system of self-assessment surveys.

* Undergraduate Final Project

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

INTRODUCCIÓN

Actualmente las instituciones de educación superior realizan un desarrollo constante en sus funciones de docencia, investigación y extensión, este desarrollo constante se basa en una permanente búsqueda por los más altos índices de calidad. La Acreditación es un mecanismo que garantiza la calidad de un programa académico o de una institución de educación superior y hace reconocimiento público del nivel de calidad de aquellas entidades que la logren por medio de un proceso que incluye la Autoevaluación de las mismas.

Para lograr la Acreditación de alta calidad, los programas de posgrado en Colombia deben cumplir con los lineamientos promulgados por el Consejo Nacional de Acreditación (CNA), lineamientos que siguen el modelo de acreditación que el CNA ha utilizado desde su creación. Algunas instituciones de educación superior están diseñando sus propias guías de Autoevaluación para lograr la Acreditación de alta calidad, estas guías son supervisadas por el CNA para asegurar la compatibilidad y su armonía con los lineamientos del CNA.

Es importante que los órganos de dirección de las entidades que pretendan lograr la Acreditación diseñen un sistema o modelo de Autoevaluación bajo una política de mejoramiento de la calidad en el que se incluya a toda la comunidad perteneciente a éstas entidades con el fin de recolectar información pertinente con la cual se pueden generar juicios de calidad, soportados y argumentados por los mismos participantes. Para el proceso de recolección de información de los participantes es necesario definir instrumentos o métodos que realicen esta labor de forma directa y precisa. Las encuestas son un ejemplo de instrumento de recolección de información necesaria para el proceso de autoevaluación de un programa de posgrado o de una institución de educación superior.

A lo largo de la presentación de este documento se describe el proceso de creación de un software que se encarga de dar soporte al proceso de autoevaluación funcionando como un instrumento de recolección de información por medio de encuestas. Se presentará el marco de referencia necesario para la creación y el entendimiento del software, la metodología utilizada y la ingeniería del software aplicada en cada fase del desarrollo.

1. JUSTIFICACIÓN

Los programas académicos de postgrado 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 factores, características, 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.

Actualmente la universidad Industrial de Santander realiza los procesos de autoevaluación utilizando las encuestas como medio de recolección de datos de los participantes para generar juicios de calidad. Para crear las encuestas y analizar sus resultados se utiliza un software que pertenece a una entidad ajena a la Universidad Industrial de Santander que además restringe el uso de determinadas acciones.

Este proyecto tiene por objeto diseñar una aplicación web que permita generar y automatizar encuestas dirigidas al perfil de cada persona dentro de la comunidad académica de un programa de posgrado, que afecten directamente los factores y variables de un sistema de autoevaluación automatizado.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Diseñar e implementar una aplicación web para crear encuestas personalizadas para autoevaluación de programas de posgrado.

2.2 OBJETIVOS ESPECÍFICOS

- Especificar los principales requisitos del sistema de autoevaluación, identificando los diferentes actores y escenarios que son necesarios para el diseño de las encuestas.
- Diseñar un modelo de datos que permita generar, almacenar y consultar las encuestas dirigidas a cada uno de los participantes.
- Diseñar el prototipo de la aplicación web que cumpla con las siguientes funcionalidades:
 - Identificar los perfiles determinados (de los actores del proceso de autoevaluación) y enviar encuestas a cada perfil con preguntas conformes a cada uno de estos.
 - Recolectar y procesar las encuestas realizadas por los actores del proceso.
 - Presentar los resultados en una interfaz que permita una fácil interpretación por parte del usuario de la aplicación.

- Implementar el prototipo de la aplicación e integrarlo con los demás componentes del Sistema de Información de Autoevaluación.
- Realizar un conjunto de pruebas que permitan validar la funcionalidad y usabilidad de la aplicación web después de integrada.
- Elaborar la documentación detallada de la aplicación web que facilite, en un instante dado, el mantenimiento o desarrollo de nuevas versiones.

3. MARCO DE REFERENCIA

3.1 ACREDITACIÓN DE ALTA CALIDAD

Formalmente la acreditación de alta calidad para las maestrías y los doctorados inició en Colombia a partir del mes de julio de 2010, esta modalidad fue aprobada por el Consejo Nacional de Educación superior (CESU).

La acreditación de alta calidad es un proceso principalmente académico que busca continuamente mayores niveles de calidad. Optar por la acreditación de los programas de maestría y de doctorado es una decisión voluntaria que inicia un proceso en la institución o el programa en favor del desarrollo de una cultura de evaluación y mejoramiento continuo, además de su capacidad de autorregulación.

El proceso de acreditación de los programas de maestría y de doctorado está compuesto de forma similar a los procesos de acreditación de instituciones de educación superior y programas de pregrado:

- Una autoevaluación realizada por la propia institución haciendo uso de guías coherentes con los criterios y características de calidad definidas por el CNA.
- Una evaluación externa realizada por los Pares Académicos designados por el CNA. Los Pares Académicos debe visitar a la institución y al programa en condiciones normales de funcionamiento.
- La respuesta de la institución que ha sometido su programa al proceso de acreditación, frente al informe de la evaluación externa y a las recomendaciones en él formuladas.
- La recomendación final: “el concepto técnico emitido por el CNA al Ministerio de Educación Nacional, para la expedición del acto de

acreditación”, (CNA lineamientos, 1996) tomando en consideración los tres anteriores aportes.

- El acto de acreditación que le corresponde al Ministerio de Educación Nacional.

3.2 CONSEJO NACIONAL DE EDUCACIÓN SUPERIOR

Es un organismo encargado de ejercer funciones de planificación, asesoría, coordinación y recomendación en el nivel de educación superior que apoya al Ministerio de Educación Nacional en la consecución de los fines y propósitos del Sistema de Aseguramiento de la Calidad.

El CESU está integrado por representantes de las comunidades científicas y académicas y por algunos representantes de las instituciones oficiales que están relacionadas con el desarrollo del sector educativo en el nivel superior.

El CESU está integrado de la siguiente forma:

- El Ministro de Educación Nacional, quien lo preside.
- El Jefe del Departamento Nacional de Planeación.
- El Rector de la Universidad Nacional de Colombia.
- El Director del Fondo Colombiano de Investigaciones Científicas y Proyectos especiales “Francisco José de Caldas”, Colciencias.
- Un rector de la universidad estatal u oficial.
- Dos rectores de universidades privadas.
- Un rector de universidad de economía solidaria.
- Un rector de una institución universitaria o escuela tecnológica, estatal u oficial.
- Un rector de institución técnica profesional técnica estatal u oficial.
- Dos representantes del sector productivo.

- Un representante de la comunidad académica de universidad estatal u oficial.
- Un profesor universitario.
- Un estudiante de los últimos años de universidad.
- El Director del Instituto Colombiano para el Fomento de la Educación Superior (Icfes), con voz pero sin voto.

3.3 SISTEMA NACIONAL DE ACREDITACIÓN (SNA)

El sistema de acreditación nacional es el conjunto que contiene diferentes políticas, estrategias, procesos y organismos con los cuales busca garantizar a la sociedad que aquellas instituciones que pertenecen al sistema cumplen con los más altos requisitos de calidad realizando sus propósitos y objetivos.

Para ser parte del Sistema de Acreditación Nacional y obtener la respectiva acreditación las instituciones deberán expresar de formalmente y por escrito el deseo voluntario de iniciar el debido proceso para aquellos programas que cumplan con los lineamientos establecidos por el Consejo Nacional de Acreditación (CNA).

3.4 CONSEJO NACIONAL DE ACREDITACIÓN (CNA)

El Consejo Nacional de Acreditación ha sido creado como un organismo académico por la Ley 30 de 1992 y se compone de siete académicos. El CNA es de naturaleza académica y depende del Consejo Nacional de Educación Superior (CESU) el cual es integrado por personas de las más altas calidades científicas y profesionales.

El CNA es el organismo encargado de la revisión del proceso de acreditación, su organización, fiscalización, certificación de su calidad y recomienda al Ministro de

Educación Nacional acreditar a aquellos programas e instituciones que lo merezcan.

Para obtener la acreditación de alta calidad de los programas de Maestría y de Doctorado, el CNA ha establecido los lineamientos¹ necesarios para tal fin. Estos lineamientos deben cumplirse mediante un proceso que incluye diferentes etapas:

- a) Apreciación de condiciones iniciales, si las instituciones no se encuentran vinculadas al Sistema de Acreditación Nacional.
- b) Autoevaluación desarrollada por el programa y presentación del informe correspondiente al CNA.
- c) Evaluación externa realizada por los Pares Académicos que sean designados para tal fin.
- d) Evaluación final por parte del CNA en la cual se emite el juicio de calidad.
- e) Resolución de Acreditación que es emitida por el Ministerio de Educación Nacional.

El CNA también ha establecido un mínimo de requisitos que deben ser cumplidos para obtener la acreditación de alta calidad como los siguientes:

- Contar con el registro calificado.
- Demostrar una trayectoria mínima de 8 años a partir del ingreso de los primeros estudiantes.
- Tener como mínimo nueve (9) graduados de doctorado o veinte (20) graduados de maestría.

¹ Consejo Nacional de Acreditación. “*Lineamientos para la acreditación de alta calidad de los programas de Maestrías y Doctorados*”; Bogotá, 2010.

3.5 JAVA EE 6

Actualmente JAVA es uno de los lenguajes de programación más utilizados en el mundo². Este lenguaje de programación cuenta con una popularidad muy grande y una inmensa fuente de recursos de apoyo, lo que favorece a los desarrolladores al momento de producir software y de esta forma JAVA expande su dominio en el mercado de los lenguajes de programación.

JAVA es un lenguaje basado en clases, de esta forma en JAVA cada clase representa una entidad, la cual contiene un conjunto de características que son compartidas por un conjunto de objetos, permitiendo crear instancias u objetos, que son la particularización de una clase determinada. Por último, JAVA es un lenguaje de programación orientado a objetos que usa técnicas tales como: Encapsulamiento, herencia, polimorfismo, abstracción, cohesión y acoplamiento.

JAVA es un lenguaje de alto nivel y fuertemente tipado, tiene un alto nivel de seguridad, es robusto y provee a los desarrolladores de un conjunto de clases y paquetes predefinidos que facilitan el desarrollo de las aplicaciones. Además, JAVA admite las familias de protocolos TCP/IP lo que hace que sea un lenguaje ideal para el desarrollo de aplicaciones web.

La principal ventaja de JAVA es que es un lenguaje multiplataforma, esto quiere decir que las aplicaciones programadas en JAVA se pueden ejecutar en diferentes sistemas operativos incluyendo sistemas operativos móviles. Esta ventaja se debe a que las aplicaciones JAVA son compiladas en bytecode para finalmente ser interpretadas por la máquina virtual de JAVA alojada en el sistema operativo.

² Programming Language Popularity [<http://www.langpop.com/>]

Java Platform, Enterprise Edition o Java EE es una plataforma de programación usada para desarrollar y ejecutar software de todo tipo de aplicaciones basadas en el lenguaje JAVA. Una de sus características es que permite utilizar arquitecturas de N capas distribuidas haciendo uso de los componentes modulares del software ejecutándose en un servidor de aplicaciones. Java EE contiene especificaciones únicas para sus componentes, permitiendo a los desarrolladores crear aplicaciones portables entre plataformas, escalables y que a su vez se integran con tecnologías anteriores.

3.6 FRAMEWORKS Y HERRAMIENTAS A UTILIZAR

Dentro del contexto del desarrollo de software, un Framework es una estructura predefinida que da soporte y facilita el desarrollo de software. Los Frameworks son estructuras conceptuales que proveen de módulos concretos de software a los desarrolladores, evitándoles el trabajo de implementar funciones del software que son genéricas y permiten dedicar más tiempo en las funciones que son específicas del software en construcción. Incluyen muchas herramientas como soporte de programas, bibliotecas y un lenguaje interpretado entre otros.

El uso de Frameworks en el desarrollo de software aporta grandes ventajas que ayudan a reducir el tiempo y la complejidad de la implementación, como por ejemplo: la reutilización de componentes de software para diferentes proyectos, funciones probadas y libres de errores en casi su totalidad y, muy importante, establecen un lenguaje común entre los desarrolladores debido a que la mayoría de los Frameworks poseen su propia documentación, de esta forma ayudan a agilizar el mantenimiento del código de una aplicación o la reparación de la misma.

Aunque el principal propósito de los Frameworks es facilitar el desarrollo de software, existen ciertas desventajas que se crean o se descubren con el tiempo, como por ejemplo el código innecesario que contienen, la incertidumbre de cómo

están implementadas las funciones, dependencia del código fuente de una aplicación con respecto al código fuente del Framework y que el aprendizaje de algún Framework aumenta su dificultad a medida que los crean más robustos y más complejos.

3.6.1 JavaServer Faces (JSF 2.2). JavaServer Faces (JSF) es un framework que facilita la construcción de interfaces de usuario en aplicaciones JAVA del lado del servidor. JSF hace uso de JSP3 para manejar las diferentes páginas HTML4 por medio de una biblioteca de etiquetas que se encargan de crear los elementos de las vistas como por ejemplo los elementos de un formulario.

Cuando se desarrollan aplicaciones web por medio del patrón MVC⁵ las aplicaciones contienen generalmente un conjunto de vistas, estas vistas son las páginas HTML que contienen textos, cajas, botones, tablas y elementos de selección con los cuales los usuarios interactúan. Todo ese conjunto de elementos están contenidos principalmente en formularios HTML y de esta forma las aplicaciones web reciben los datos otorgados por el usuario y son enviados o guardados posteriormente a un servidor.

El objetivo o la función principal de JSF es asociar clases JAVA a las pantallas o vistas, las cuales recolectan la información que es introducida por los usuarios. Las clases también contienen métodos cuyas funciones son de responder a determinadas acciones o peticiones de los usuarios.

3.6.2 Primefaces 5.0. Primefaces es un componente software de código abierto que contiene un conjunto de componentes diseñados para facilitar la creación y el desarrollo de aplicaciones web. Es un Framework desarrollado para soportar

³ JSP: Java Server Pages [<http://www.oracle.com/technetwork/java/jsp-138432.html>]

⁴ HTML: HyperText Markup Language

⁵ MVC: Modelo Vista Controlador

JavaServer Faces y contiene gran cantidad de características que facilitan la implementación de las aplicaciones web, como por ejemplo: más de 30 temas diferentes para las interfaces de las aplicaciones, soporte de Ajax, componentes para páginas web (generación de gráficas, tablas que muestran registros de las bases de datos, cajas de texto de autocompletado, editores de HTML, entre otros).

3.6.3 Morphia 0.108. Morphia es una librería de Java que funciona como motor de mapeo de objetos a/desde MongoDB⁶. Es una librería ligera que permite fluidez en las consultas con validación en tiempo de ejecución. Morphia utiliza anotaciones para que no exista la necesidad de crear archivos XML para gestionar o actualizar. Esta librería es recomendada para los desarrolladores con experiencia en JPA.

3.6.4 Bootstrap v3.3.1. Bootstrap es un Framework muy popular que contiene HTML, CSS y JS para facilitar el diseño adaptativo, móvil y con una interfaz gráfica amigable para los usuarios en aplicaciones web. Bootstrap hace que el desarrollo web del lado del cliente sea más rápido y más fácil, además permite que las aplicaciones desarrolladas bajo este Framework se puedan visualizar sin problema en diferentes dispositivos (celular, computador, Tablet, etc.) y en diferentes navegadores. Bootstrap es de código abierto, está organizado, desarrollado y mantenido en GitHub⁷.

3.7 Enterprise JavaBeans (EJB). La tecnología EJB es una arquitectura de componentes del lado del servidor desarrollada para la plataforma JAVA, especialmente para JAVA EE⁸. Es una API⁹ que hace parte del estándar de desarrollo de aplicaciones empresariales bajo la plataforma de programación JAVA EE.

⁶ MongoDB: Sistema de base de datos NoSQL

⁷ GitHub: [<https://github.com/>]

⁸ Java Platform, Enterprise Edition: [<http://docs.oracle.com/javasee/>]

⁹ Application Programming Interface:
[http://en.wikipedia.org/wiki/Application_programming_interface]

Cada EJB de una aplicación, es una clase JAVA que implementa una interfaz determinada. Los EJB permiten que un contenedor de aplicaciones sea capaz de crear instancias y utilizar las clases JAVA de los EJB como lo necesite.

EJB facilita el proceso de desarrollo de aplicaciones de componentes empresariales y por medio de su uso, las aplicaciones construidas son fácilmente escalables, fiables y seguras sin la necesidad de escribir código de la infraestructura. Las aplicaciones construidas con esta tecnología tienen la característica de ser portables entre diferentes plataformas y servidores de aplicaciones.

3.8 MongoDB 2.4.12. MongoDB es un sistema de base de datos orientada a documentos, esto significa que no almacena los datos en tablas como lo hacen las bases de datos relacionales sino que los guarda de forma estructurada en documentos tipo JSON¹⁰ de codificación binaria, a estos documentos MongoDB los llama BSON.

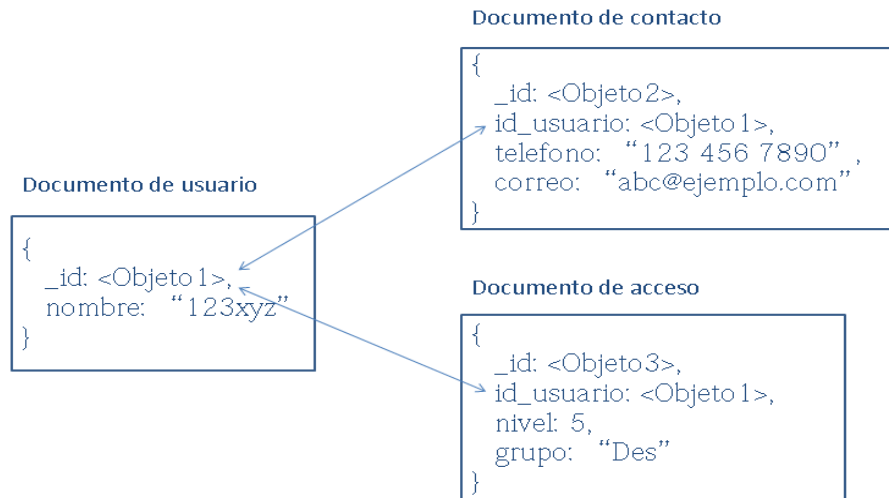
En MongoDB cada documento en una misma colección no necesita tener el mismo número de campos o estructura de los otros documentos, y los campos comunes en una colección pueden almacenar diferentes tipos de datos.

La estructura de un documento en MongoDB está dada principalmente por dos herramientas que representan la relación entre los datos de diferentes documentos, estas herramientas son: las referencias y los documentos embebidos.

¹⁰ JavaScript Object Notation [<http://json.org/>]

Las referencias almacenan las relaciones entre los datos incluyendo enlaces a otros documentos. Las aplicaciones pueden resolver estas referencias y acceder a los datos contenidos en los documentos.

Figura 1. Uso de referencias a enlaces de documentos.



Fuente: Autor.

Los documentos embebidos son sub-documentos que están contenidos dentro de los campos de otros documentos. Esto es debido a que los documentos obtienen las referencias entre los datos de otros documentos y las usan para almacenar información como una estructura simple de un documento.

Figura 2. Campos embebidos que contienen información relacionada.



Fuente: Autor.

Debido a que MongoDB está escrito en C++ adquiere una relación directa a los recursos hardware de la máquina donde se ejecute, permitiéndole realizar sus tareas de gestión de bases de datos con un alto rendimiento y velocidad. Las bases de datos en MongoDB son dinámicas, quiere decir que la estructura de los documentos puede variar a medida que los requisitos de la aplicación cambien y los resultados serán reflejados inmediatamente sin problemas de incompatibilidad.

4. METODOLOGÍA

Existe una gran cantidad de metodologías que se usan para dirigir proyectos de desarrollo de software, éstas metodologías se enfocan en la solución de varios problemas o situaciones que se presentan a menudo en este tipo de proyectos, la planificación del proyecto, la variación de los requisitos durante el proyecto, el aseguramiento de la calidad del software, la sincronización de los equipos de trabajo, crear la documentación necesaria que permite su posterior escalabilidad, son algunos ejemplos de éstos problemas o situaciones.

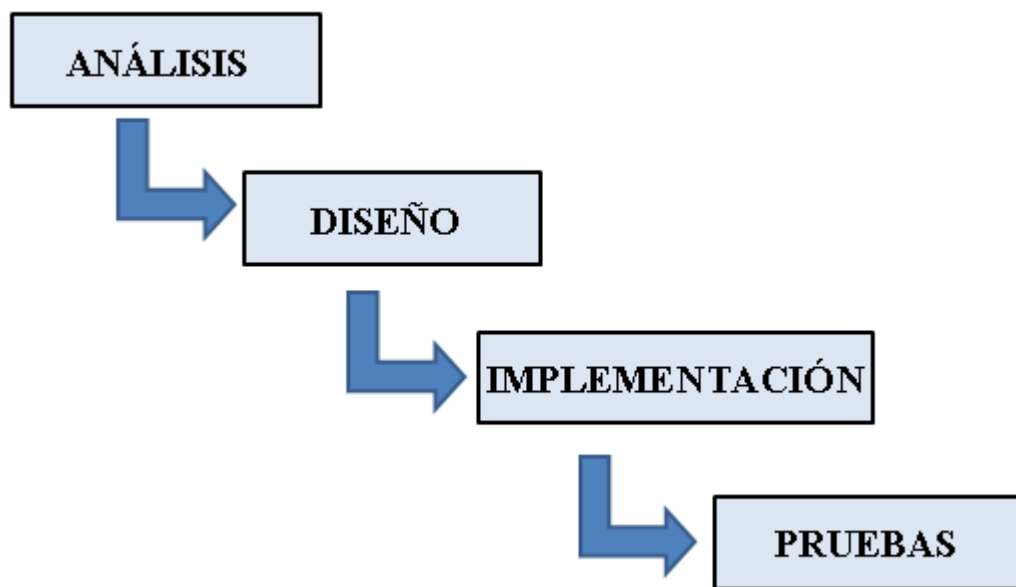
A cada proyecto de desarrollo de software le pertenecen una serie de características o condiciones bajo las que se debe trabajar, requisitos fijos o variables, un grupo de desarrollo grande o pequeño, productos entregables a corto o a largo plazo, entre otros. El éxito de un proyecto de desarrollo de software depende, en gran medida, de la metodología escogida para lidiar con los problemas que se puedan presentar durante el desarrollo del software bajo las características o condiciones del proyecto.

La aplicación planteada en el presente proyecto debe cumplir con los requisitos establecidos por el proceso de autoevaluación del CNA, el conjunto de métodos que definen la generación de las encuestas para perfiles particulares y los procesos de envío y recolección de las encuestas realizadas para su posterior análisis de resultados. Conocidas las condiciones bajo las cuales se debe desarrollar el software del presente proyecto se trabajó con la metodología de desarrollo de software llamada Modelo en Cascada.

El Modelo en Cascada se caracteriza por cumplir estrictamente un conjunto de etapas o pasos de desarrollo de software definidos y secuenciales, de esta forma, el inicio de una etapa debe esperar a que culmine su etapa inmediatamente

anterior. Se debe resaltar que el Modelo en Cascada es una metodología predictiva que se acopla perfectamente a los requisitos del software presentado en este presente documento debido a que estos requisitos no van a variar durante el proceso de desarrollo del software. Las etapas del Modelo en Cascada usadas en el desarrollo del software fueron:

Figura 3. Modelo en Cascada



Fuente: Autor.

4.1 PLAN DE TRABAJO

El presente proyecto se desarrolló siguiendo las etapas de trabajo explicadas a continuación siguiendo la metodología elegida.

4.1.1 Etapa de análisis. En esta etapa se profundizó el conocimiento en cuanto al proceso actual de autoevaluación de los programas de posgrado haciendo un análisis completo de todas las variables que intervienen y que pueden influenciar en dicho proceso. La creación de los escenarios y casos de uso en esta etapa es fundamental para representar de forma clara el comportamiento del proceso mediante su interacción con los usuarios. Este análisis permitió la planificación del desarrollo del software en etapas posteriores, haciendo que el desarrollo dependiera en su mayor parte de la lógica del proceso de autoevaluación.

Terminada esta etapa se cumplieron los siguientes ítems:

- Estudio de los documentos elaborados por el CNA referentes al proceso de autoevaluación.
- Descripción del proceso actual de elaboración de encuestas para la autoevaluación.
- Investigación de las tecnologías a usar y el estado del arte de los sistemas de información actuales relacionados a este tipo de desarrollo.
- Elaboración de los escenarios y casos de uso.
- Elaboración de la especificación de los requisitos.

4.1.2 Etapa de diseño. En esta etapa se realizó el diseño completo del modelo de datos y los prototipos de las interfaces de usuario que permiten a los usuarios del sistema comprender fácilmente su uso correcto. El diseño de la base de datos se realizó a partir de la descripción del proceso actual de elaboración de las encuestas tomando las entidades y sus relaciones y se construyó un diagrama de clases. La base de datos en esta etapa fue instalada y probada para comprobar su correcto funcionamiento.

Terminada esta etapa se cumplieron los siguientes ítems:

- Diseño de tablas con entidades y sus referencias.
- Diseño de las interfaces de usuario.
- Especificación de los requisitos funcionales.

4.1.3 Etapa de implementación. Después de realizar un proceso detallado en las etapas anteriores, en esta etapa se procede a programar los procesos, las interfaces de usuario, y si es necesario, realizar cambios para la implementación definitiva de la base de datos. Gracias a la metodología escogida, antes de comenzar esta etapa se pueden manejar algunos cambios de los requisitos de la aplicación.

Terminada esta etapa se cumplieron los siguientes ítems:

- Aprendizaje del lenguaje de programación a utilizar.
- Implementación de la base de datos.
- Implementación de las interfaces de usuario o las vistas del modelo MVC.
- Conexión de la base de datos a las vistas por medio de los controladores.
- Validación por medio de pruebas a la aplicación.

4.1.4 Etapa de pruebas. En esta etapa la aplicación fue sometida a una serie de pruebas para garantizar su correcto funcionamiento. Además, por medio de un uso general y de las diferentes funciones de la aplicación, se rectificó que se cumplieran los requisitos planteados en las etapas anteriores y que sean coherentes con los requisitos que define el CNA para el correcto uso de los instrumentos de recolección de datos por parte de los participantes del proceso de autoevaluación.

Terminada esta etapa se cumplieron los siguientes ítems:

- Conjunto de pruebas realizadas.

- Corrección y ajustes leves de fallas encontradas en la aplicación.
- Prueba de usabilidad y funcionalidad por parte del usuario final.
- Manual de usuario para facilitar el uso de la aplicación.

5. ANÁLISIS, REQUISITOS Y CASOS DE USO

5.1 ESPECIFICACIÓN DE REQUISITOS

La Especificación de Requisitos Software (ERS) de la aplicación web para crear encuestas para la autoevaluación de programas de posgrado ha sido elaborada bajo la dirección y colaboración de la Doctora Sonia Cristina Gamboa Sarmiento PhD en Educación, Filosofía y Enseñanza de la Filosofía. Esta Especificación de Requisitos Software fue diseñada en base al estándar “IEEE Recommended Practice for Software Requirement Specifications ANSI/IEEE 830 1998”.

5.2 PROPÓSITO

En este segmento se presentarán aspectos generales y específicos que serán clave para el entendimiento del desarrollo y funcionamiento de la aplicación descrita en este documento. Este segmento se presenta principalmente para diseñadores y desarrolladores de software involucrados en la sistematización e implementación del Sistema de Información de Autoevaluación de Programas de Posgrado con el fin de coordinar los requisitos generales y específicos dados por el CNA.

5.3 ALCANCE DEL SISTEMA

Para el sistema que debe generar encuestas para la autoevaluación de programas de posgrado se ha determinado desarrollar una aplicación web que cumpla con los requisitos de creación, envío y recolección de encuestas de los participantes del proceso de autoevaluación para su debido procesamiento e interpretación.

Las funciones principales que la aplicación web debe cumplir son:

- Permitir que el administrador de la aplicación gestione un banco de preguntas donde pueda crear, editar y eliminar las preguntas que serán asignadas a los participantes del proceso.
- Generar automáticamente las encuestas para cada participante realizando la comparación correspondiente entre la relación de la pregunta y las características de la persona.
- Enviar cada encuesta al correo de la persona correspondiente.
- Mostrar las respuestas dadas por los participantes del proceso de autoevaluación y permitir visualizarlas por el tipo del participante.

5.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

- Encuesta personalizada: Encuesta generada para cada persona con preguntas relacionadas a ella misma.
- JSF (JavaServer Faces): Tecnología que facilita el desarrollo de aplicaciones web basadas en Java.
- Autoevaluación: Proceso interno realizado por las instituciones de educación superior que desean lograr la acreditación de alta calidad, en este proceso se hace una evaluación a la entidad por parte de las personas que trabajan o están directamente relacionadas con dicha entidad.
- MongoDB: Mongo Data Base, es una base de datos no relacional que almacena sus datos en documentos permitiendo realizar grandes cambios en la estructura de los datos sin afectar la información ni el sistema actual.
- RF: Requisitos Funcionales.
- RNF: Requisitos No Funcionales.

5.5 PERSPECTIVA DEL PRODUCTO

Este proyecto cumplirá con la implementación de una aplicación web para la autoevaluación de los programas de posgrado de la Escuela de Ingeniería de Sistemas e Informática, donde se propone realizar y administrar encuestas como medio de recolección de datos de los participantes del proceso de Autoevaluación. Además la aplicación debe tener un entorno amigable donde el administrador de la aplicación pueda diseñar las encuestas y el usuario final, quien es el encargado de completar las encuestas, pueda desarrollarlas sin problema alguno.

5.6 FUNCIONES DEL PRODUCTO

5.6.1 Crear preguntas. La aplicación contiene un banco de preguntas donde son almacenadas las preguntas que son creadas por el administrador. Cada pregunta debe ser creada con su respectiva relación para que pueda ser asignada a la persona que le corresponda.

5.6.2 Generar encuestas. Cuando el banco de preguntas está completamente revisado y cada pregunta está en orden, se le indica a la aplicación que genere la encuesta de cada participante, esta acción se realiza en un solo clic y el administrador tiene la posibilidad de revisar las encuestas antes de enviarlas.

5.6.3 Enviar encuestas por correo. Cada participante tiene información de su correo electrónico en la base de datos del sistema de información de Autoevaluación, a este correo electrónico se envía un enlace que dirige al participante a realizar su propia encuesta.

5.6.4 Almacenar preguntas resueltas. A medida que el participante del proceso de autoevaluación resuelve la encuesta las preguntas son almacenadas en la base de datos con la respuesta dada por el participante.

5.6.5 Mostrar resultados. Durante el transcurso de tiempo en que son realizadas las encuestas o después de que todas han sido resueltas, los resultados pueden ser visualizados por el administrador de la aplicación para su respectivo análisis.

5.7 CARACTERISTICAS DEL USUARIO

La aplicación web permitirá su uso completo a un solo usuario, el administrador. El administrador será el encargado de la creación y modificación de las preguntas que van a contener las encuestas, como también de la visualización e interpretación de los datos recogidos por medio de la aplicación.

5.8 RESTRICCIONES

- Lenguaje y tecnologías en JAVA.
- Desarrollado en NetBeans IDE.
- Uso de base de datos No Relacional (MongoDB).
- Lenguaje de marcado XHTML.
- Uso de Frameworks y librerías: Bootstrap, Primefaces, Morphia y JavaServer Faces.
- Es necesario un servidor donde se instanciará la base de datos.
- La conexión al servidor donde se instancia la base de datos debe ser estable y segura.

5.9 SUPOSICIONES Y DEPENDENCIAS

- Solo el administrador puede hacer modificaciones a los elementos de la aplicación y acceder a la información almacenada.
- La aplicación funciona correctamente en cualquier computador y en cualquier navegador web, siempre que exista una conexión a internet.
- La aplicación depende del funcionamiento del servidor donde se aloja la base de datos.

5.10 REQUISITOS FUTUROS

Una posible mejora a la aplicación web sería:

- Permitir que el administrador de la aplicación pueda diseñar la presentación de las encuestas enviadas.

5.11 REQUISITOS FUNCIONALES

Tabla 1. Requisito Funcional 01 – Crear Pregunta

Identificación	RF01
Nombre	CREAR PREGUNTA
Características	Las preguntas deben estar en una base de datos.
Prioridad	ALTA
Descripción	Todas las preguntas se deben crear e insertar en la base de datos para poder generar las encuestas. Deben almacenarse con su tipo, número de opciones, texto de cada opción, enunciado y relación. Las preguntas son creadas por el administrador de la aplicación.

Fuente: Autor

Tabla 2. Requisito Funcional 02 – Modificar Pregunta

Identificación	RF02
Nombre	MODIFICAR PREGUNTA
Características	Las preguntas pueden variar para siguientes autoevaluaciones.
Prioridad	ALTA
Descripción	El administrador podrá realizar cambios en el enunciado de las preguntas, en sus opciones de respuesta y demás características.

Fuente: Autor

Tabla 3. Requisito Funcional 03 – Eliminar Pregunta

Identificación	RF03
Nombre	ELIMINAR PREGUNTA
Características	Las preguntas obsoletas se deben eliminar.
Prioridad	ALTA
Descripción	Las preguntas que recolectan información que no es necesaria para la autoevaluación deben ser eliminadas por el administrador de la aplicación.

Fuente: Autor.

Tabla 4. Requisito Funcional 04 - Asociar Pregunta

Identificación	RF04
Nombre	ASOCIAR PREGUNTA
Características	Preguntas concernientes para cada usuario.
Prioridad	ALTA
Descripción	Cada usuario debe responder solo las preguntas que le conciernen. Para esto, la aplicación se encarga de asignar un conjunto de preguntas a cada persona y se las envía en forma de encuesta.

Fuente: Autor.

Tabla 5. Requisito Funcional 05 – Consultar Preguntas

Identificación	RF05
Nombre	CONSULTAR PREGUNTAS
Características	El administrador puede ver las preguntas actuales en el banco de preguntas.
Prioridad	ALTA
Descripción	Cuando el administrador intente insertar una nueva pregunta, tiene la opción de listar las preguntas actuales y buscar si existe una pregunta similar para decidir si es necesaria la acción de crearla.

Fuente: Autor.

Tabla 6. Requisito Funcional 06 – Generar Encuestas

Identificación	RF06
Nombre	GENERAR ENCUESTAS
Características	Un conjunto de preguntas se envían a cada persona en una encuesta.
Prioridad	ALTA
Descripción	Cada persona será responsable de responder una encuesta que contiene preguntas solamente de su interés o incumbencia.

Fuente: Autor.

Tabla 7. Requisito Funcional 07 – Enviar Encuestas

Identificación	RF07
Nombre	ENVIAR ENCUESTAS
Características	Se enviará a cada persona la encuesta correspondiente.
Prioridad	ALTA
Descripción	Cada persona recibirá en su correo electrónico una dirección URL que contiene su respectiva encuesta en la aplicación web.

Fuente: Autor.

Tabla 8. Requisito Funcional 08 – Registrar Respuestas

Identificación	RF08
Nombre	REGISTRAR RESPUESTAS
Características	Al momento de resolver la encuesta se deben guardar las respuestas en la base de datos.
Prioridad	ALTA
Descripción	Las respuestas deben quedar guardadas en la base de datos para permitir hacer diferentes tipos de análisis de los resultados.

Fuente: Autor.

Tabla 9. Requisito Funcional 09 – Consultar Resultados

Identificación	RF09
Nombre	CONSULTAR RESULTADOS
Características	La aplicación permitirá visualizar los resultados de las encuestas.
Prioridad	ALTA
Descripción	Los resultados se podrán visualizar clasificándolos por el tipo (Egresado, Estudiante o Profesor) de las personas que hacen parte del proceso de autoevaluación.

Fuente: Autor.

5.12 REQUISITOS NO FUNCIONALES

Tabla 10. Requisito No Funcional 01 – Veracidad de la Información

Identificación	RNF01
Nombre	VERACIDAD DE LA INFORMACIÓN
Características	La información solicitada debe ser veraz.
prioridad	ALTA
Descripción	La información recolectada de las encuestas debe ser concerniente al proceso de autoevaluación.

Fuente: Autor.

Tabla 11. Requisito No Funcional 02 – Interfaces del Sistema

Identificación	RNF02
Nombre	INTERFACES DEL SISTEMA
Características	La aplicación presentará todas sus funciones al administrador.
prioridad	ALTA
Descripción	Todas las funciones de gestión de preguntas, gestión de encuestas y análisis de resultados serán ofrecidas por medio de interfaces intuitivas al administrador.

Fuente: Autor.

Tabla 12. Requisito No Funcional 03 – Análisis de Respuestas

Identificación	RNF03
Nombre	ANÁLISIS DE RESPUESTAS
Características	Las respuestas se presentarán de diferentes formas para su análisis.
prioridad	ALTA
Descripción	Las respuestas se podrán visualizar por la clasificación del tipo de las personas y con los respectivos porcentajes de respuesta que facilitan su interpretación.

Fuente: Autor.

Tabla 13. Requisito No Funcional 04 – Desempeño del Sistema

Identificación	RNF04
Nombre	DESEMPEÑO DEL SISTEMA
Características	La aplicación tendrá un buen desempeño en sus acciones de recolección y manejo de datos.
prioridad	ALTA
Descripción	Para lograr un buen desempeño, la aplicación debe ser programada para evitar problemas de sobrecarga.

Fuente: Autor.

Tabla 14. Requisito No Funcional 05 – Confiabilidad del Sistema

Identificación	RNF05
Nombre	CONFIABILIDAD DEL SISTEMA
Características	La disponibilidad tiene que ser continua para el servicio de los usuarios.
prioridad	ALTA
Descripción	La aplicación deberá estar siempre en funcionamiento, ya que está diseñada para la carga de datos y consulta de los mismos.

Fuente: Autor.

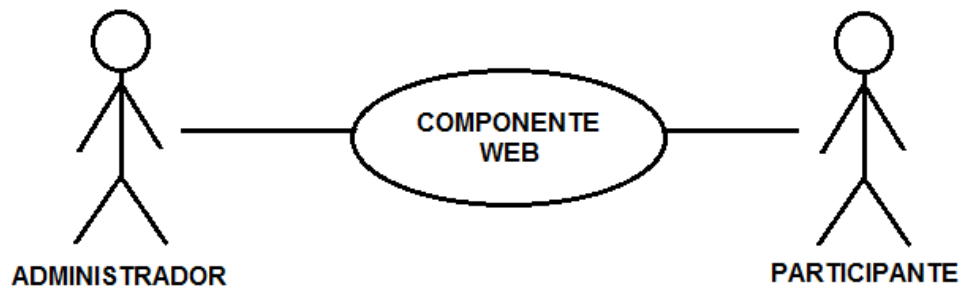
Tabla 15. Requisito No Funcional 06 – Seguridad de la Información

Identificación	RNF06
Nombre	SEGURIDAD DE LA INFORMACIÓN
Características	Garantizar la seguridad de los datos suministrados por los usuarios.
prioridad	ALTA
Descripción	Proporcionar confianza al usuario respecto a la aplicación protegiendo la información y datos que se manejen mediante protocolos de seguridad.

Fuente: Autor.

5.13 ACTORES DEL SISTEMA

Figura 4. Actores del Sistema



Fuente: Autor

5.14 DESCRIPCION DE ACTORES

Tabla 16. Descripción de Actores - Administrador

Actor	Administrador
Caso de Uso	Crear pregunta, modificar pregunta, eliminar pregunta, generar encuestas, enviar encuestas y ver resultados.
Descripción	Este usuario no tiene restricciones en las

	operaciones que realice en la aplicación, tiene acceso completo a toda la aplicación web incluyendo las operaciones directas a la base de datos.
Tipo	Primario.

Fuente: Autor.

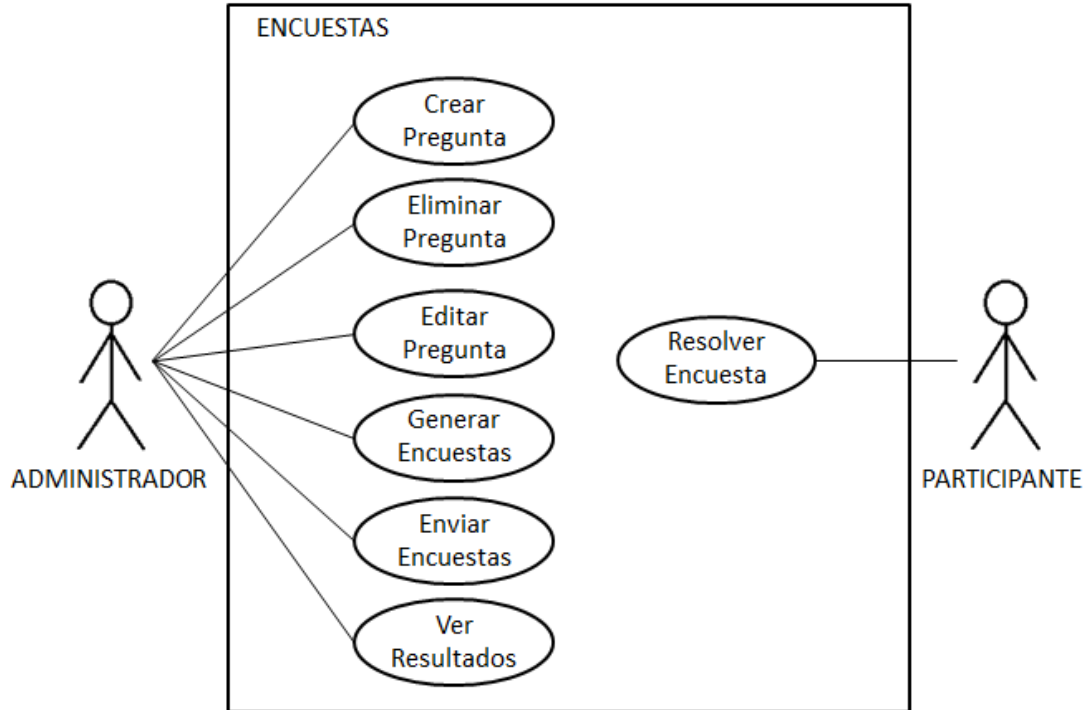
Tabla 17. Descripción de Actores - Participante

Actor	Participante
Caso de Uso	Resolver encuesta.
Descripción	Este usuario se encarga únicamente de resolver la encuesta que le ha sido enviada al correo electrónico.
Tipo	Secundario

Fuente: Autor.

5.15 DIAGRAMA DE CASOS DE USO

Figura 5. Diagrama de Casos de Uso



Fuente: Autor

5.16 DESCRIPCIÓN DE DIAGRAMAS DE CASOS DE USO

Tabla 18. Descripción de Casos de Uso – Crear Pregunta

Título del Caso de Uso	Crear Pregunta	ID del Caso de Uso	CU 001
Descripción General del Caso de Uso o Requisito			
Es iniciado por el administrador de la aplicación al hacer clic en la opción de “Crear Pregunta”			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Iniciar sesión correctamente en la aplicación.			
Flujo normal de Eventos			
	Administrador	Sistema	
1.	Hace clic en la opción ‘Crear Pregunta’.	Muestra en pantalla una ventana con un formulario que contiene los datos necesarios de la nueva pregunta.	
1.1	Hace clic en la opción “Guardar” del formulario.	Muestra todas las preguntas existentes incluyendo la pregunta que acaba de ser creada.	
Condiciones de Salida			
La pregunta queda ubicada en el orden correspondiente según las características dadas.			
Caminos de Excepción/Flujos alternativos			
Ninguno.			

Fuente: Autor.

Tabla 19. Descripción de Casos de Uso – Eliminar Pregunta

Título del Caso de Uso	Eliminar Pregunta	ID del Caso de Uso	CU 002
Descripción General del Caso de Uso o Requisito			
Este caso de uso inicia cuando el administrador decide eliminar una pregunta y elige la opción de “Eliminar” que contiene cada pregunta.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Iniciar sesión correctamente en la aplicación.			
Flujo normal de Eventos			

	Administrador	Sistema
1.	Hace clic en la opción “Eliminar” que contiene la pregunta.	Elimina la pregunta de la base de datos y actualiza la lista de preguntas de la aplicación.
Condiciones de Salida		
Ninguna.		
Caminos de Excepción/Flujos alternativos		
Ninguno.		

Fuente: Autor.

Tabla 20. Descripción de Casos de Uso – Editar Pregunta

Título del Caso de Uso	Editar Pregunta	ID del Caso de Uso	CU 003
Descripción General del Caso de Uso o Requisito			
Este caso de uso se inicia cuando el administrador decide cambiar las características de una pregunta.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Iniciar sesión correctamente en la aplicación.			
Flujo normal de Eventos			
	Administrador	Sistema	
1.	Hacer clic en la opción “Editar” que contiene la pregunta a modificar.	Muestra un formulario editable que contiene los datos actuales de la pregunta seleccionada.	
1.1	Modifica el campo (o campos) que desea cambiar en la pregunta y da clic en “Actualizar”.	Muestra la lista de todas las preguntas existentes en la base de datos incluyendo la pregunta editada.	
Condiciones de Salida			
Ninguna			
Caminos de Excepción/Flujos alternativos			
Ninguno.			

Fuente: Autor.

Tabla 21. Descripción de Casos de Uso – Generar Encuestas

Título del Caso de Uso	Generar Encuestas	ID del Caso de Uso	CU 004
Descripción General del Caso de Uso o Requisito			

Este caso de uso inicia cuando el administrador ha rectificado que todas las preguntas están en orden y decide generar las encuestas.		
Entidades/Actores Involucrados		
Administrador		
Condiciones de Entrada		
Iniciar sesión correctamente en la aplicación. La colección de personas y de preguntas no debe ser vacía.		
Flujo normal de Eventos		
	Administrador	Sistema
1.	Selecciona la opción “Generar Encuestas” en la aplicación.	Relaciona las preguntas a las personas de la base de datos y crea para cada persona una encuesta.
Condiciones de Salida		
Ninguna.		
Caminos de Excepción/Flujos alternativos		
Ninguno.		

Fuente: Autor.

Tabla 22. Descripción de Casos de Uso – Enviar Encuestas

Título del Caso de Uso	Enviar Encuestas	ID del Caso de Uso	CU 005
Descripción General del Caso de Uso o Requisito			
Este caso de uso se inicia cuando el administrador decide enviar las encuestas para que los participantes del proceso de autoevaluación las resuelvan.			
Entidades/Actores Involucrados			
Administrador			
Condiciones de Entrada			
Iniciar sesión correctamente en la aplicación. Las encuestas deben estar generadas. Tener una cuenta de correo electrónico, desde esta cuenta serán enviadas al correo electrónico de los participantes.			
Flujo normal de Eventos			
	Administrador	Sistema	
1.	Hace clic en la opción “Enviar Encuestas” de la aplicación.	Muestra un formulario con los datos necesarios para el envío de las encuestas.	
1.1	Completa el formulario y da clic en “Enviar”	Envía una dirección URL al correo electrónico de cada participante con la dirección de su respectiva encuesta.	

Condiciones de Salida
El atributo “Enviada” de cada encuesta toma el valor de “Si”, esperando a ser respondida.
Caminos de Excepción/Flujos alternativos
Ninguno.

Fuente: Autor.

Tabla 23. Descripción de Casos de Uso – Resolver Encuestas

Título del Caso de Uso	Resolver Encuestas	ID del Caso de Uso	CU 006
Descripción General del Caso de Uso o Requisito			
Cada participante recibe en su correo electrónico una dirección URL con la ruta de su respectiva encuesta para que sea resuelta.			
Entidades/Actores Involucrados			
Participante			
Condiciones de Entrada			
Acceder al correo electrónico y revisar el mensaje que contiene la ruta de la encuesta.			
Flujo normal de Eventos			
	Participante	Sistema	
1.	Hace clic en la URL de la encuesta.	Muestra una ventana que contiene la encuesta del propietario del correo electrónico.	
1.1	Resuelve todas las preguntas de la encuesta.	Almacena las preguntas que son resueltas por el participante en la base de datos.	
Condiciones de Salida			
La encuesta queda resuelta			
Caminos de Excepción/Flujos alternativos			
Ninguno.			

Fuente: Autor.

Tabla 24. Descripción de Casos de Uso – Ver Resultados

Título del Caso de Uso	Ver Resultados	ID del Caso de Uso	CU 007
Descripción General del Caso de Uso o Requisito			
Este caso de uso es iniciado cuando el administrador de la aplicación debe visualizar los datos recolectados por la aplicación para su respectivo análisis.			
Entidades/Actores Involucrados			

Administrador		
Condiciones de Entrada		
Al menos una pregunta de cualquier encuesta debe ser resuelta.		
Flujo normal de Eventos		
	Administrador	Sistema
1.	Hace clic en la opción "Resultados" de la aplicación	Muestra opciones de la clasificación de los resultados (Todos, Egresado, Estudiante y Profesor).
1.1	Hace clic en el tipo de resultados que quiere ver.	Muestra los resultados recogidos de las encuestas y almacenados en la base de datos.
Condiciones de Salida		
Ninguna		
Caminos de Excepción/Flujos alternativos		
Ninguno.		

Fuente: Autor.

6. DISEÑO DE LA APLICACIÓN Y ESTRUCTURA DE DATOS

Terminada la fase de análisis de requisitos se procede a visualizar y definir todo tipo de técnicas de desarrollo de productos software que van a ser implementadas, por ejemplo: patrones de diseño, la estructura de datos, las vistas de la aplicación, entre otros.

6.1 DISEÑO DE LA BASE DE DATOS

Debido a que el desarrollo de esta aplicación web que genera encuestas hace parte de la segunda fase de implementación del Sistema de Información para la Autoevaluación de programas de posgrado, se adopta el diseño de la base de datos creado en la primera fase y las herramientas de implementación de bases de datos utilizadas.

En esta primera fase se toma la decisión de utilizar una base de datos no relacional (No SQL) y hacer uso de la tecnología MongoDB debido a una cantidad considerable de beneficios que otorga al sistema en comparación con las bases de datos relacionales. Algunos de estos beneficios son:

- Las bases de datos no relacionales son dinámicas, esto permite que, en un instante dado, la base de datos pueda hacer cambios en la estructura de sus documentos y estos cambios será reflejados instantáneamente sin problemas de compatibilidad.
- Mejoran el rendimiento en las búsquedas de gran cantidad de datos, al guardar los datos en documentos evita que se deban ejercer operaciones “JOIN” (operación de las bases de datos relacionales) en las consultas.
- No necesitan de servidores costosos para operar de forma óptima.

- Facilidad para mejorar el rendimiento en caso de que sea necesario, solo basta con agregar más recursos en la plataforma Hardware cuando se necesite.

La decisión del uso bases de datos no relacionales y el diseño de la base de datos fueron acciones conjuntas donde participaron estudiantes de pregrado, posgrado y la directora del proyecto en general. Como resultado de este proceso de diseño, la base de datos contiene colecciones como: PERSONAS, GRUPOS_INVESTIGACION, LINEAS_INVESTIGACION, PROPIEDAD_INTELECTUAL, entre otras. Para la aplicación web que genera encuestas se hace énfasis en la colección PERSONAS.

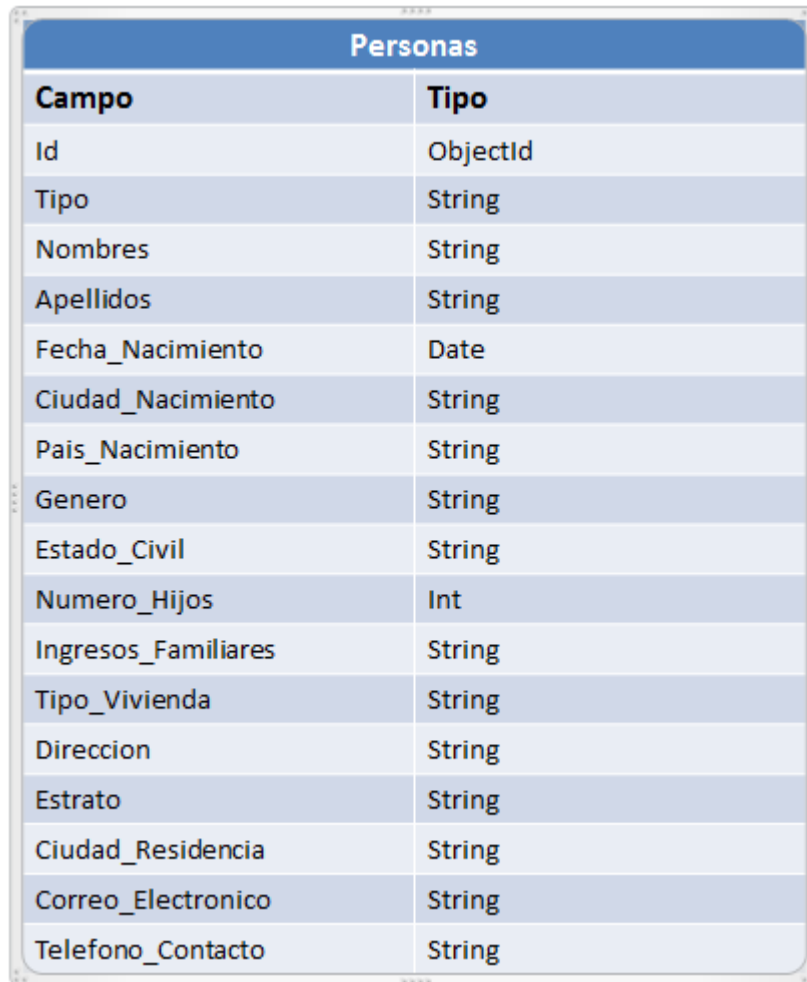
6.2 ESQUEMA DE LA BASE DE DATOS

Las colecciones presentadas a continuación son aquellas que son esenciales para el funcionamiento de la aplicación web que genera encuestas, las demás colecciones de la base de datos son utilizadas para otras operaciones del sistema de Autoevaluación y por este motivo no serán presentadas en este documento¹¹.

En las siguientes figuras se presentan las colecciones de la base de datos de la aplicación de la siguiente forma: El nombre de la colección en la parte superior de la figura, en la columna izquierda los campos de la colección y en la parte derecha de la figura se presenta el tipo que debe tener la variable contenida en ese campo.

¹¹ La base de datos completa del Sistema de Autoevaluación puede ser consultada en el proyecto “COMPONENTE WEB PARA EL REGISTRO Y ANÁLISIS DE LA INFORMACIÓN PARA LA AUTOEVALUACIÓN DE PROGRAMAS DE POSGRADO EN UNIVERSIDADES PÚBLICAS”

Figura 6. Esquema de la base de datos – Colección Personas



Personas	
Campo	Tipo
Id	ObjectId
Tipo	String
Nombres	String
Apellidos	String
Fecha_Nacimiento	Date
Ciudad_Nacimiento	String
Pais_Nacimiento	String
Genero	String
Estado_Civil	String
Numero_Hijos	Int
Ingresos_Familiares	String
Tipo_Vivienda	String
Direccion	String
Estrato	String
Ciudad_Residencia	String
Correo_Electronico	String
Telefono_Contacto	String

Fuente: Autor.

Figura 7. Esquema de la base de datos – Colección Elemento



Elemento	
Campo	Tipo
Id	ObjectId
Enunciado	String
Posicion	Int
RelacionadoCon	HashMap< String, String >
Factor	Int
Caracteristica	Int
Indicador	Int
OpcionesHorizontales	ArrayList< String >
OpcionesVerticales	ArrayList< String >
TipoElemento	String
MultipleRespuesta	Boolean
Obligatorio	Boolean

Fuente: Autor.

Figura 8. Esquema de la base de datos – Colección Encuesta



Encuesta	
Campo	Tipo
Id	ObjectId
IdEncuestado	Personas
Preguntas	List< Elemento >
Enviada	Boolean

Fuente: Autor.

Figura 9. Esquema de la base de datos – Colección Respuesta



Respuesta	
Campo	Tipo
Id	ObjectId
Encuestado	Personas
Enunciado	String
Respuestas	ArrayList<String>
RespuestasMatriz	HashMap<String, String>
Factor	Int
Caracteristica	Int
Indicador	Int

Fuente: Autor.

Las colecciones presentadas en las figuras permiten cumplir las funciones requeridas por la aplicación web de la siguiente forma:

- **Personas:** Contiene los documentos (BSON) de todas las personas que hacen parte del proceso de Autoevaluación. Los atributos de estas personas son usados por la aplicación para generar y enviar las encuestas.
- **Elemento:** Contiene las preguntas creadas por el administrador(a) de la aplicación. La aplicación usa el campo “relacionadoCon” de los documentos de esta colección para asociarlos con los atributos de las personas.
- **Encuesta:** Contiene las encuestas generadas por la aplicación. Cada conjunto de elementos o preguntas asociados a una persona son guardados en documentos que se almacenan en esta colección.
- **Respuesta:** Contiene las respuestas dadas por los participantes del proceso a las encuestas. Las respuestas son guardadas de forma individual a medida que se resuelve cada pregunta, asociando cada respuesta a su respectivo participante para facilitar la clasificación de los resultados al momento de su análisis.

6.3 DISEÑO DE LAS ENTIDADES

El diseño de cada entidad debe coincidir con el diseño de la colección correspondiente de la base de datos. Al implementar cada entidad en una clase JAVA se deben tomar los mismos campos definidos en el diseño de la colección correspondiente y además, se deben asignar las anotaciones correspondientes para que el Framework Morphia pueda leer correctamente cada entidad. Una anotación obligatoria que se usa al comienzo de cada clase JAVA es @Entity, la cual le indica a Morphia que esa clase debe ser leída como una entidad y que las instancias de esa entidad deben ser mapeadas de JAVA a MongoDB o viceversa.

En la siguiente imagen se puede observar el segmento de código de la clase “encuesta” donde se implementa la anotación @Entity.

Figura 10. Segmento de la entidad Encuesta

```
@Entity ( value = "encuesta" )
public class Encuesta implements Serializable {

    @Id
    private ObjectId id;

    @Reference
    private Personas idEncuestado;

    @Embedded
    private List< Elemento > preguntas;

    private boolean enviada;
```

Fuente: Autor

Después que Morphia mapea las entidades y sus instancias, los documentos son almacenados en la base de datos MongoDB de la siguiente forma:

Figura 11. Documento de una encuesta almacenada en MongoDB.

```
{
  "_id" : ObjectId("54b5d7cddc6f45cd25b5f4a3"),
  "className" : "filoen.autoevaluacion.model.Encuesta",
  "idEncuestado" : {
    "$ref" : "personas",
    "$id" : ObjectId("549ec587fcfa26b99fb2fa15")
  },
  "preguntas" : [
    {
      "_id" : ObjectId("54b5d727dc6f45cd25b5f4a1"),
      "enunciado" : "¿Cree que los propósitos del programa se están cumpliendo?",
      "posicion" : 1,
      "relacionadoCon" : {
        "Todos" : "Si"
      },
      "factor" : 1,
      "caracteristica" : 1,
      "indicador" : 5,
      "opcionesVerticales" : [
        "Si ",
        "No"
      ],
      "tipoElemento" : "multiple",
      "multipleRespuesta" : false,
      "obligatorio" : true
    },
    {
      "_id" : ObjectId("54b5d799dc6f45cd25b5f4a2"),
      "enunciado" : "Sugerencias para el programa académico",
      "posicion" : 2,
      "relacionadoCon" : {
        "Todos" : "Si"
      },
      "factor" : 11,
      "caracteristica" : 4,
      "indicador" : 1,
      "obligatorio" : false
    }
  ],
  "enviada" : false
}
```

Fuente: Autor.

6.4 ORGANIZACIÓN DE LOS PAQUETES

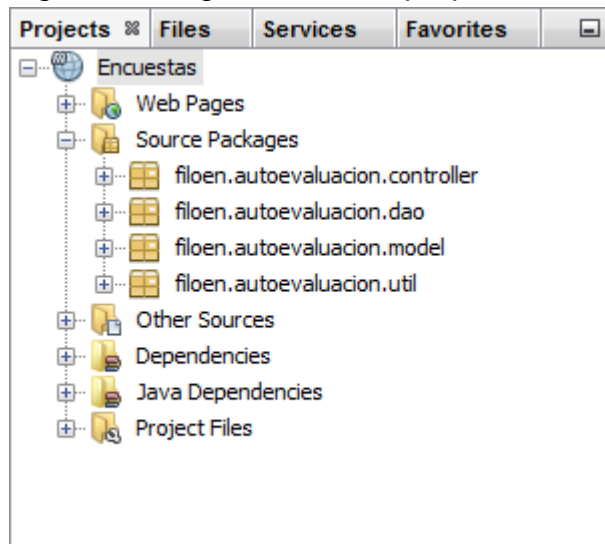
Para cumplir con las técnicas de implementación usadas en la fase uno del proyecto y lograr una integración exitosa de la aplicación con el proyecto general

de Autoevaluación, los paquetes JAVA quedan precedidos con la ubicación *filoen.autoevaluacion* y organizados de la siguiente forma:

- Paquete *filoen.autoevaluacion.controller*: Contiene las clases JAVA encargadas de las operaciones *backend* de la aplicación.
- Paquete *filoen.autoevaluacion.dao*: Contiene los EJB's encargados de las operaciones con la base de datos (guardar, consultar, borrar y actualizar).
- Paquete *filoen.autoevaluacion.modelo*: Contiene las entidades implementadas del modelo de datos.
- Paquete *filoen.autoevaluacion.util*: Contiene métodos que son utilizados con frecuencia en diferentes partes de la aplicación.

La siguiente imagen presenta la organización de los paquetes de la aplicación en el entorno de desarrollo:

Figura 12. Organización de paquetes JAVA en NetBeans



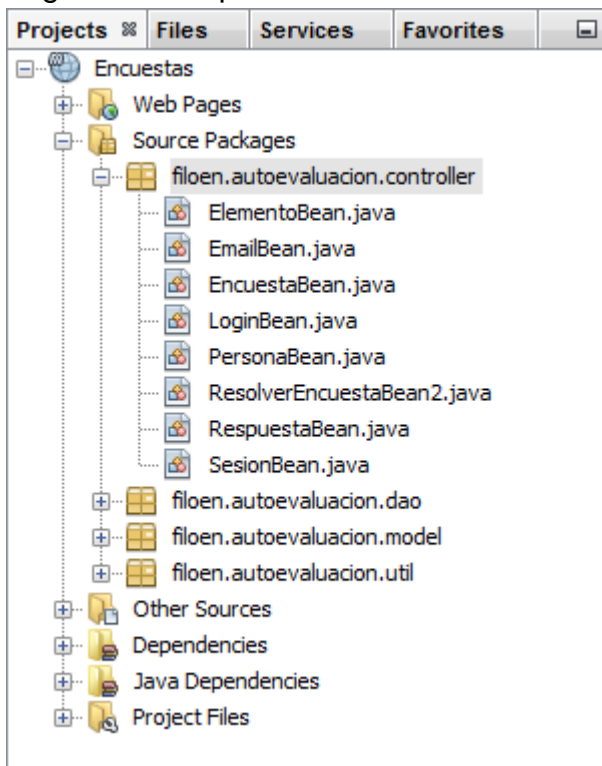
Fuente: Autor

6.5 CLASES DE LÓGICA DE NEGOCIO

Todas las funciones que ofrece la aplicación al administrador y a los participantes están implementadas en las clases JAVA contenidas en el paquete *filoen.autoevaluacion.controller*. Algunas de las funciones que contienen las clases de este paquete son:

- Navegación entre las vistas.
- Validación de datos del usuario y de la aplicación.
- Control de acceso a la aplicación.
- Manejar los eventos iniciados por los componentes.

Figura 13. Paquete *filoen.autoevaluacion.controller*



Fuente: Autor

7. DESARROLLO

Para el desarrollo de la aplicación se ha establecido usar patrones de diseño con una organización adecuada que permita un uso óptimo de los recursos por parte de la aplicación y que a su vez sea compatible con las técnicas usadas en el proyecto general. El uso de estas técnicas de programación facilitará a futuro el mantenimiento de la aplicación.

7.1 MODELO VISTA CONTROLADOR (MVC)

Este patrón de arquitectura de software separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario con el fin de definir y especificar los componentes que serán responsables de un conjunto específico de funciones. Debido a esta característica del patrón MVC de separar los componentes, se decide implementarlo en la aplicación con el fin de manejar una organización y estandarización en la implementación de la misma. Al separar cada uno de los componentes definidos por este patrón de diseño, ofrece la ventaja de localizar conceptualmente y con facilidad el componente que requiera mantenimiento en un momento dado. Para la aplicación web que genera encuestas se crearon los tres componentes (Modelo, Vista y Controlador) definidos por el patrón de diseño.

7.2 DESCRIPCIÓN DE COMPONENTES

7.2.1 Modelo. En este componente se encuentra la representación de la información de la aplicación web, por esta razón gestiona todo tipo de acceso a la información. Es el núcleo de la funcionalidad de la aplicación, de ella depende el comportamiento de las instancias de las clases java creadas para cumplir con las funciones definidas en los requisitos. Constantemente envía información a las

“Vistas” cuando estas así lo requieren. Los controladores envían peticiones de acceso o manipulación de la información al Modelo cada vez que lo necesitan.

7.2.2 Vista. Es la presentación del modelo. Hace uso de las interfaces gráficas para presentar el modelo en un formato adecuado que permite que el usuario interactúe con la aplicación. Las vistas pueden ser notificadas cuando se realizan cambios en el Modelo.

7.2.3 Controlador. Es el componente encargado de manejar todas las acciones del usuario de la aplicación. Cuando se hacen solicitudes de información a la aplicación realiza peticiones al Modelo. Todo controlador está asociado a una “Vista” y se encarga de responder a los eventos activados por los usuarios, se podría decir que el controlador es el intermediario entre la vista y el modelo.

7.3 USO DE PATRONES DE DISEÑO: SINGLETON

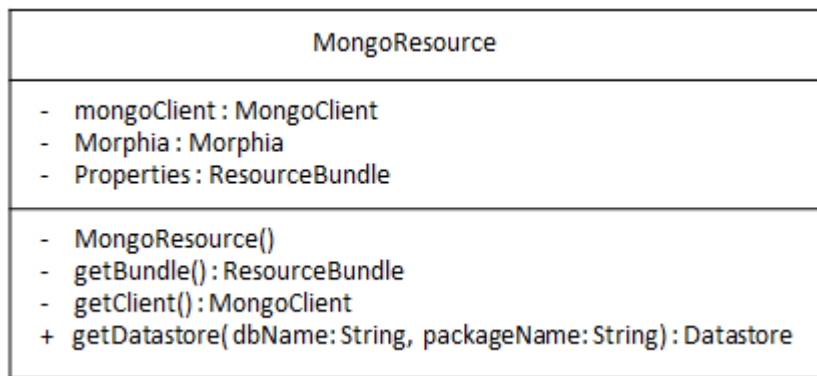
Este patrón de diseño está diseñado para restringir la creación de instancias de una clase, garantizando que una clase solo pueda crear una única instancia y proporcionando un punto de acceso global a ella.

La implementación de este patrón consiste en crear un método que se encarga de instanciar un objeto solo si no existe alguno, además, el constructor de la clase que implementa este patrón de diseño debe definir su método constructor con un alcance privado para evitar que se puedan hacer instancias de la clase desde cualquier parte de la aplicación.

Este patrón de diseño fue implementado en la aplicación web de forma restringiera las conexiones a la base de datos de la aplicación. Con este patrón de diseño implementado, la aplicación provee a los EJB's un solo objeto con los datos de conexión a la base de datos.

En la siguiente figura se presenta el diagrama de la clase *MongoResource*, la cual implementa el patrón de diseño Singleton. En la figura se puede observar que el constructor de la clase es de alcance privado y la clase tiene un método que se encarga de dar acceso global a la única instancia de la clase que provee de los datos de conexión.

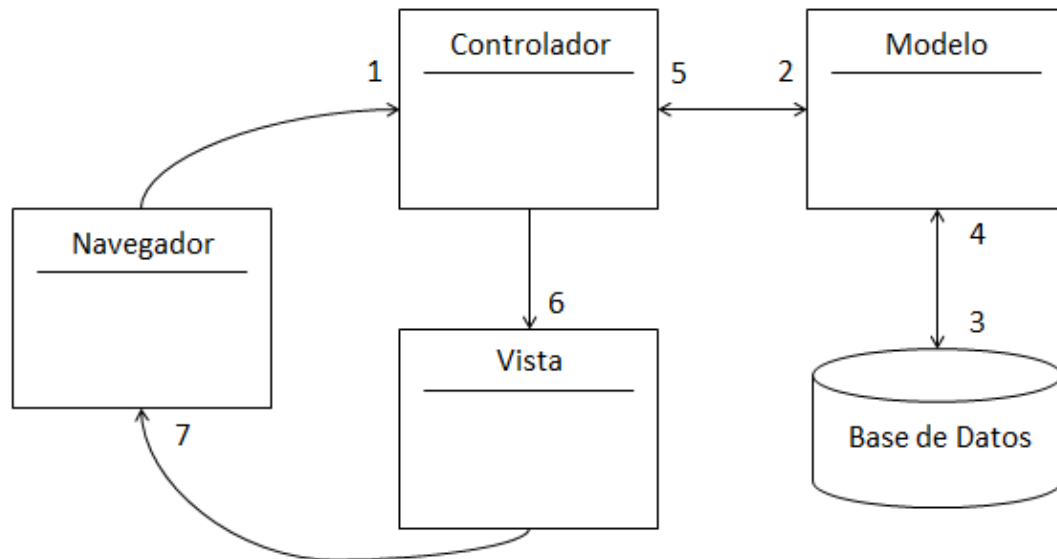
Figura 14. Diagrama de la clase *MongoResource* con Singleton



Fuente: Autor

7.4 INTERACCIÓN DE LOS COMPONENTES

Figura 15. Interacción de los Componentes



Fuente: Autor

1. El controlador recibe una petición por parte del navegador web.
2. El controlador procesa la petición, para esta función puede necesitar los datos de la base de datos y hace referencia al modelo.
3. El modelo accede a la base de datos para realizar la acción correspondiente (insertar, eliminar, actualizar o consultar).
4. El modelo recibe los datos o termina su operación en la base de datos.
5. El modelo confirma al controlador el resultado de su acción sobre la base de datos.
6. Según la confirmación del modelo, el controlador selecciona la vista que se debe mostrar al cliente junto con los elementos asociados.
7. El navegador recibe la vista seleccionada.

8. PRUEBAS

Las pruebas unitarias (llamadas también pruebas modulares) son usadas para verificar el correcto funcionamiento y estructura de cada módulo de la aplicación después de que ha sido codificado. Las pruebas unitarias realizadas a los módulos de la aplicación se hicieron probando la mayoría de los métodos de cada módulo.

Para el desarrollo de las pruebas se utilizó la base de datos de la aplicación y algunos valores de referencia.

Para el módulo *Preguntas* se definieron las siguientes pruebas:

Tabla 25. Pruebas de funcionalidad – Módulo Preguntas

IDENTIFICADO R	TÍTULO	DESCRIPCIÓN	VALORES DE ENTRADA	RESULTADOS ESPERADOS
Prueba1	Creación de una nueva pregunta.	Se realiza la creación de una pregunta nueva con los valores correspondientes.	Enunciado: ¿Cree que los propósitos del programa se cumplen? Opciones: Si, No Tipo: Múltiple Relación: Todos – Si	void
Prueba2	Modificación de una pregunta existente.	Se modifican los valores de los campos de una pregunta.	Prueba1	void

Prueba3	Consulta de todas las preguntas.	Se listan todas las preguntas existentes de la base de datos.	Preguntas almacenadas en la base de datos.	ArrayList<Pregunta>
Prueba4	Eliminación de una pregunta existente.	Se elimina una pregunta específica de la base de datos.	Prueba1	void

Fuente: Autor

Para el módulo *Encuestas* se definieron las siguientes pruebas:

Tabla 26. Pruebas de funcionalidad – Módulo Encuestas

IDENTIFICADOR	TÍTULO	DESCRIPCIÓN	VALORES DE ENTRADA	RESULTADOS ESPERADOS
Prueba5	Generación de encuestas.	Se da la orden a la aplicación de generar las encuestas.	Colección de preguntas y de personas de la base de datos.	ArrayList<Encuesta>
Prueba6	Envío de las encuestas.	Las encuestas generadas se crean con el atributo "Enviada" con valor "No".	Colección de Encuestas y lista de correos electrónicos de los participantes.	void

Fuente: Autor

Para el módulo *Resultados* se definieron las siguientes pruebas:

Tabla 27. Pruebas de funcionalidad – Módulo Resultados

IDENTIFICADOR	TÍTULO	DESCRIPCIÓN	VALORES DE ENTRADA	RESULTADOS ESPERADOS
Prueba7	Ver resultados de todos los participantes.	A la colección de respuestas no se le aplica ningún filtro.	Colección de Respuestas y Tipo: Todos.	ArrayList<Respuesta>
Prueba8	Ver resultados de estudiantes.	A la colección de respuestas se le aplica el filtro por estudiantes.	Colección de Respuestas y Tipo: Estudiante.	ArrayList<Respuesta>

Fuente: Autor

Después de realizar las pruebas se obtuvo los siguientes resultados:

Tabla 28. Pruebas de funcionalidad – Resultados finales

IDENTIFICADOR	TÍTULO	RESULTADOS
Prueba1	Creación de una nueva pregunta.	Exitosa.
Prueba2	Modificación de una pregunta existente.	Exitosa.
Prueba3	Consulta de todas las preguntas.	Exitosa.
Prueba4	Eliminación de una pregunta existente.	Exitosa.
Prueba5	Generación de encuestas.	Exitosa.

Prueba6	Envío de las encuestas.	Exitosa.
Prueba7	Ver resultados de todos los participantes.	Exitosa.
Prueba8	Ver resultados de estudiantes.	Exitosa.

Fuente: Autor.

Las pruebas de usabilidad diseñadas para la aplicación web se hicieron por medio de una entrevista con la persona que será encargada del uso de la aplicación con el rol de Administrador. En la siguiente tabla se presentan los ítems evaluados en este proceso por parte de la persona encuestada, la calificación de cada ítem se da por la asignación de un valor numérico de 1 a 5, donde 1 es el valor más bajo y 5 es el valor más alto.

Tabla 29. Prueba de usabilidad.

PRUEBA DE USABILIDAD					
ÍTEM	CALIFICACIÓN				
	1	2	3	4	5
1. Facilidad de uso de la aplicación.					X
2. Utilidad de la aplicación.					X
3. Confiabilidad de los datos recogidos.					X
4. Agilidad que proporciona al proceso de autoevaluación.					X

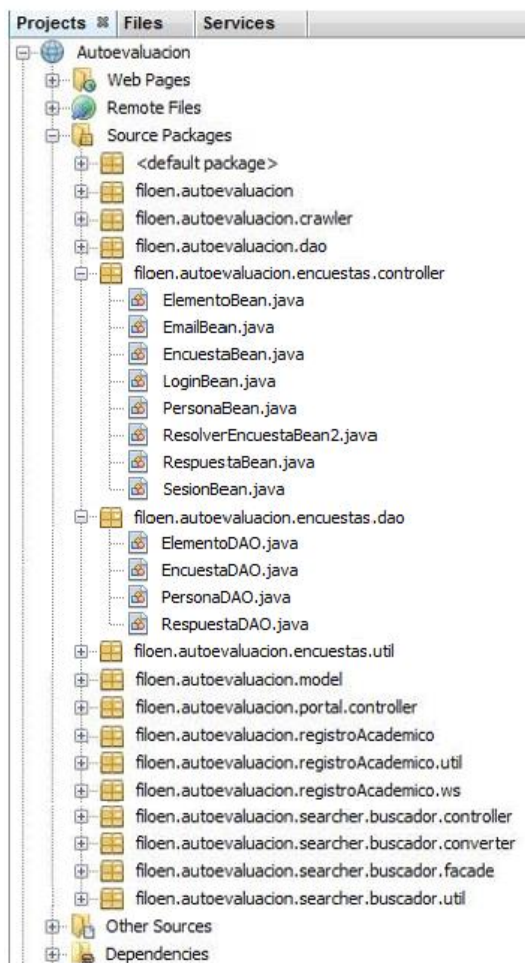
Fuente: Autor.

9. INTEGRACIÓN DE LA APLICACIÓN

Debido a que la aplicación web presentada en este proyecto implementó las mismas técnicas de programación y adoptó la misma estructura y organización definida en el proyecto general, la integración de la aplicación con el proyecto general fue exitosa y funciona correctamente.

En la siguiente figura se presentan los paquetes y las clases correspondientes a la aplicación web que genera encuestas junto con los paquetes y las clases del proyecto general en completo orden.

Figura 16. Integración de la aplicación al proyecto Autoevaluación



Fuente: Autor.

10. CONCLUSIONES

La automatización del proceso de creación de las encuestas por medio de la aplicación web disminuyó en gran cantidad el tiempo invertido por el administrador para esta labor, además, el estudio del proceso actual de generación de encuestas permitió observar que se estaban realizando preguntas que recolectaban información redundante debido a que dicha información ya estaba almacenada en la base de datos, ahora la aplicación web se encarga de este proceso y elimina este tipo de preguntas.

El uso de Frameworks y patrones de diseño disminuyeron en gran medida el tiempo de implementación de la aplicación, también, proporcionaron una organización estandarizada en el proyecto general y por lo tanto, deberían seguirse implementando en proyectos posteriores que estén contenidos en el proyecto de Autoevaluación.

La implementación y el uso de la aplicación web cambian en gran medida la forma en que se realizan las encuestas actualmente en este proceso, generando un nuevo modelo de encuestas gestionado por el administrador que facilita y soporta notoriamente el proceso de Autoevaluación.

El funcionamiento de la aplicación se verificó correctamente por medio de una serie de pruebas, estas pruebas se realizaron con datos obtenidos de procesos anteriores y que ahora se encuentran almacenados para procesos posteriores. Esta verificación generó la satisfacción con la aplicación de la persona encargada del proceso de gestión de encuestas de Autoevaluación.

11. SUGERENCIAS Y RECOMENDACIONES

La información almacenada en la base de datos correspondiente a las respuestas dadas por los participantes del proceso debe ser tratada con suma responsabilidad, debido a que la información almacenada contiene los datos del participante y la respuesta que ha dado a cada pregunta. Se debe garantizar que la información sea utilizada para su propósito principal y no para fines personales.

Se recomienda la integración de la base de datos del proyecto de Autoevaluación con la base de datos del departamento de admisiones de la Universidad Industrial de Santander. Unificando y compartiendo la información almacenada en dichas bases de datos se evita el uso de información duplicada, la cual puede generar problemas lógicos en el proceso si la información cambia en alguna de las dos bases de datos.

BIBLIOGRAFÍA

ARIAS SÁNCHEZ, Jonathan y DUITAMA GARCIA, Oscar David. COMPONENTE WEB PARA EL REGISTRO Y ANÁLISIS DE LA INFORMACIÓN PARA LA AUTOEVALUACIÓN DE PROGRAMAS DE POSGRADO EN UNIVERSIDADES PÚBLICAS. Trabajo de grado Ingeniero de Sistemas. Bucaramanga: Universidad Industrial de Santander. Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas, 2014.

ÇIVICI Ç. PRIMEFACES USER GUIDE 5.1. First Edition. [en línea]. http://www.primefaces.org/docs/guide/primefaces_user_guide_5_1.pdf [citado el 15 de enero de 2015].

COLOMBIA. CONSEJO NACIONAL DE ACREDITACIÓN. Acreditación de programas de posgrado. [en línea]. <http://www.cna.gov.co/1741/article-186363.html> [citado el 15 de enero de 2015]

COLOMBIA. CONSEJO NACIONAL DE ACREDITACIÓN. Sistema Nacional de Acreditación en Colombia. [en línea]. <http://www.cna.gov.co/1741/article-186365.html> [citado el 18 de enero de 2015]

COLOMBIA. CONSEJO NACIONAL DE ACREDITACIÓN. Consejo Nacional de Acreditación. [en línea]. <http://www.cna.gov.co/1741/article-186382.html> [citado el 18 de enero de 2015]

COLOMBIA. CONSEJO NACIONAL DE ACREDITACIÓN. Lineamientos para la Acreditación de Alta Calidad de Programas de Maestría y Doctorado. [en línea]. http://cms-static.colombiaaprende.edu.co/cache/binaries/articles-186363_lineam_MyD.pdf?binary_rand=7259 [citado el 20 de enero de 2015]

COLOMBIA. MINISTERIO DE EDUCACION NACIONAL. Consejo Nacional de Educación -CESU. [en línea]. <http://www.mineduacion.gov.co/1621/article-196487.html> [citado el 22 de enero de 2015]

GOSLING J., JOY B., STEELE G., BRACHA G. The JAVA™ Language Specification. Third Edition. [en línea]. <http://docs.oracle.com/javase/specs/jls/se5.0/jls3.pdf> [citado el 20 de enero de 2015]

MONGODB. Data Modeling Introduction. [en línea]. <http://docs.mongodb.org/manual/core/data-modeling-introduction/> [citado el 22 de enero de 2015]

MONGODB. Introduction to MongoDB. [en línea]. <http://www.mongodb.org/about/introduction/> [citado el 25 de enero de 2015]

ORACLE. JavaServer Faces Technology – Documentation. [en línea]. <http://www.oracle.com/technetwork/java/javaee/documentation/index-137726.html> [citado el 25 de enero 2015]

ORACLE. Enterprise JavaBeans Technology. [en línea]. <http://www.oracle.com/technetwork/java/javaee/ejb/index.html> [citado el 25 de enero de 2015]

WIESEL J. MongoDB desde cero: Modelado de Datos. [en línea]. <http://codehero.co/mongodb-desde-cero-modelado-de-datos/> [citado el 25 de enero de 2015]

ANEXOS

Anexo A: Manual de usuario de la aplicación

[MANUAL DE USUARIO ENCUESTAS AUTOEVALUACIÓN v1.0]

David Jesús Vásquez Rios

Grupo Filosofía y Enseñanza de la Filosofía

2015

CONTENIDO

1. Inicio de Sesión	79
2. Menú general	80
3. Módulo: Preguntas.....	80
3.1 Crear Preguntas	81
3.1.1 Pregunta abierta	82
3.1.2 Pregunta de opción múltiple con única respuesta	83
3.1.3 Pregunta de opción múltiple con múltiple respuesta.....	84
3.1.4 Pregunta de matriz de respuestas	85
3.2 Visualizar preguntas.....	86
3.3 Editar preguntas.....	86
3.4 Eliminar preguntas	87
4. Módulo: Participantes.....	88
4.1 Agregar participantes.....	88
4.2 Visualizar participantes	89
4.3 Editar participantes.....	90
4.4 Eliminar participantes	91
4.5 Buscar participantes.....	91
5. Módulo: Encuestas.....	92
5.1 Generar encuestas	92
5.2 Visualizar encuestas	93
5.3 Generar encuesta específica	93
5.4 Visualizar encuestas	94
5.5 Eliminar encuestas	95
5.6 Enviar encuestas	96
5.7 Resolver encuestas	97
6. Módulo: Resultados	99
6.1 Ver resultados	99
7. Cerrar sesión	100

Manual de Usuario

Encuestas Autoevaluación

Aplicación web que facilita el proceso de autoevaluación de programas de posgrado por medio de encuestas. Por medio de esta aplicación se realizan los procesos de creación, generación y envío de encuestas, además de la recolección de las respuestas dadas. La siguiente figura presenta la interfaz inicial de la aplicación.

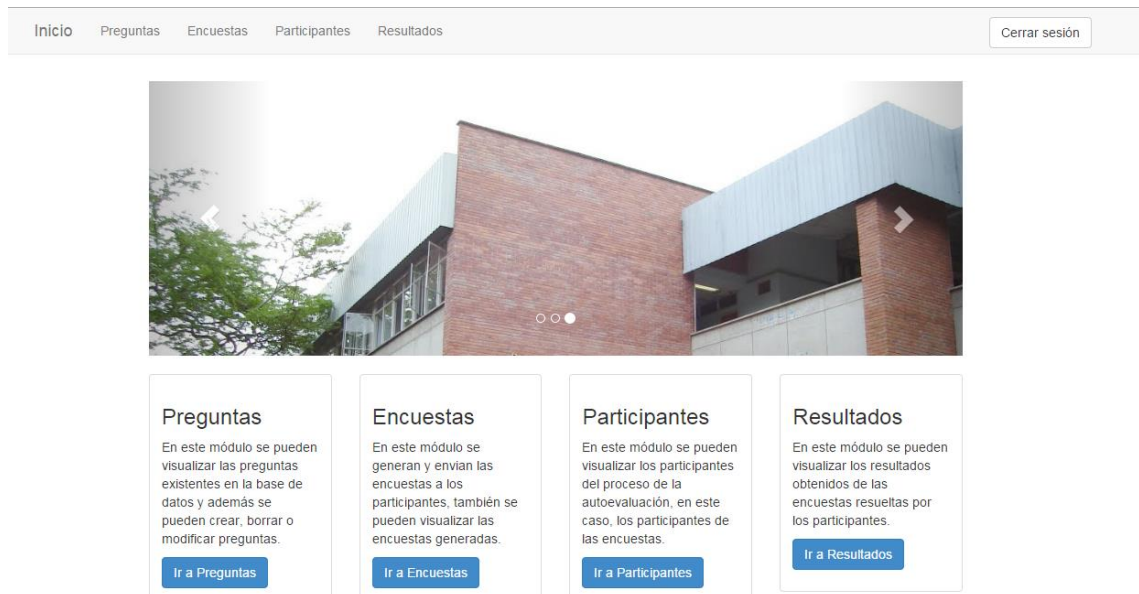
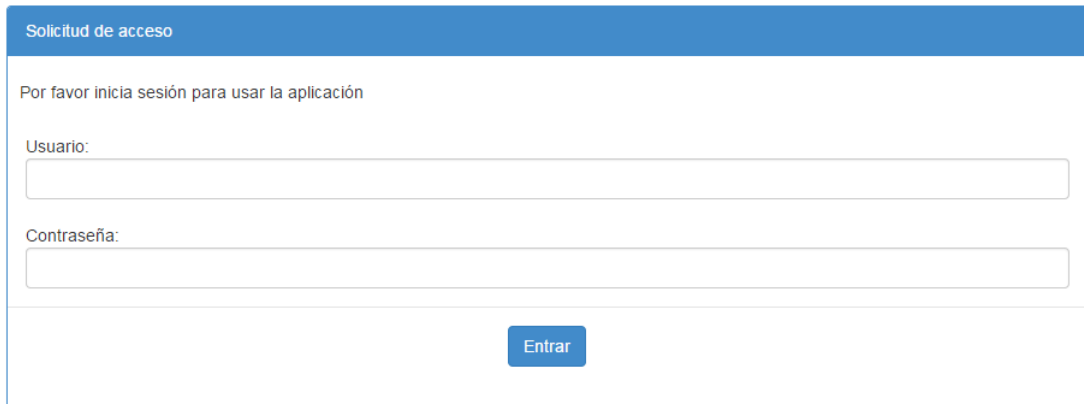


Figura 1. Interfaz inicial de la aplicación

1. Inicio de Sesión



Solicitud de acceso

Por favor inicia sesión para usar la aplicación

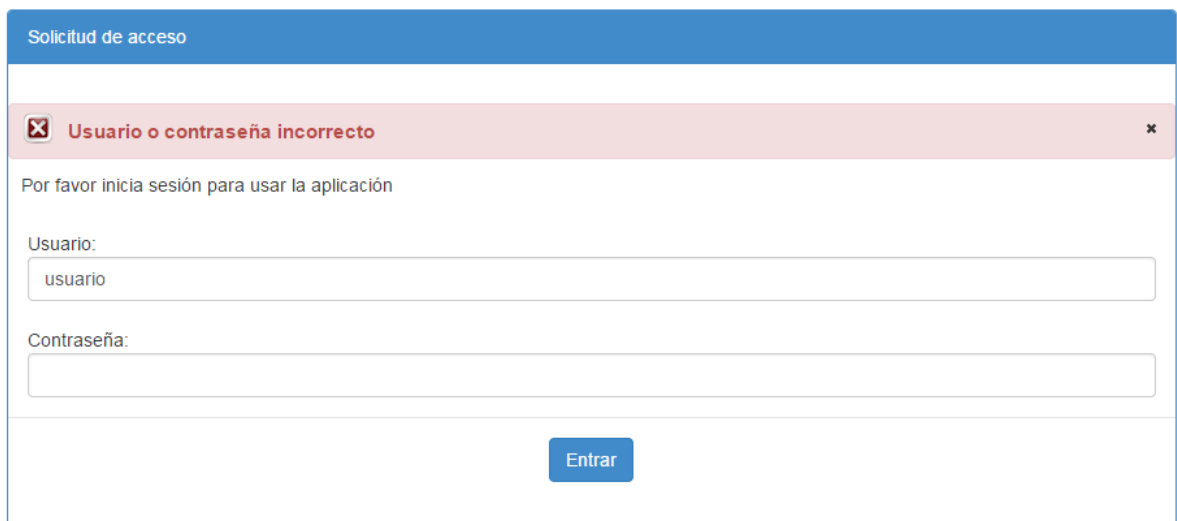
Usuario:

Contraseña:

Figura 2. Formulario de inicio de sesión

En los campos de texto es necesario digitar el respectivo usuario y contraseña correcta del administrador para iniciar sesión en la aplicación.

Después de ingresar los datos y hacer clic en el botón “Entrar” la aplicación se encarga de validar los datos y comprobar que el usuario existe. Si los datos ingresados son erróneos la aplicación muestra una alerta del error.



Solicitud de acceso

✖ Usuario o contraseña incorrecto ✖

Por favor inicia sesión para usar la aplicación

Usuario:

Contraseña:

Figura 3. Error en el inicio de sesión.

Si el problema persiste aun revisando detalladamente los datos de inicio de sesión, el administrador debe contactarse con el desarrollador del sistema.

Si los datos ingresados son correctos, la aplicación le permite al administrador acceder a todas sus funciones.

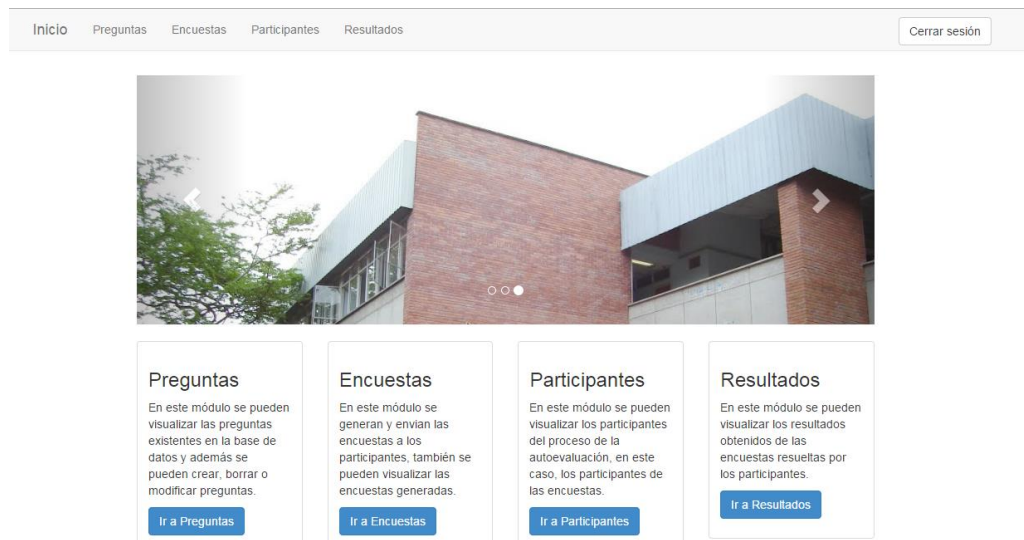


Figura 4. Acceso al sistema exitoso

Al acceder a la aplicación, se muestra el menú principal para administrar la aplicación y una descripción breve de cada módulo.

2. Menú general

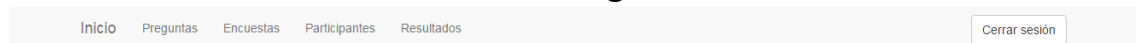


Figura 5. Menú general de la aplicación

Cuando el usuario ingresa a la aplicación se presenta un menú en la parte superior de la pantalla que contiene enlaces a los diferentes módulos de la aplicación. El usuario puede hacer clic en cualquiera de estos enlaces para acceder al módulo esperado.

3. Módulo: Preguntas

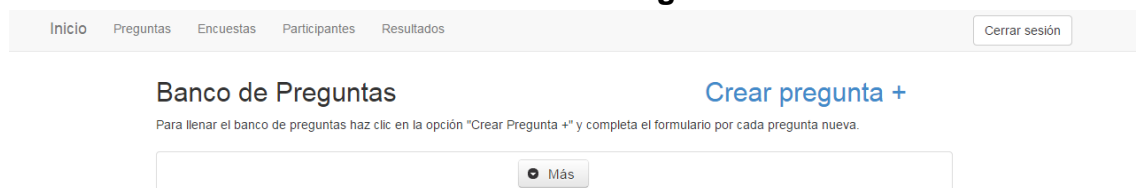


Figura 6. Módulo Preguntas con el banco de preguntas vacío

El módulo Preguntas contiene las funciones necesarias para realizar una completa gestión de las preguntas que van a contener las encuestas. Inicialmente el banco de preguntas se encuentra vacío y se deben insertar las preguntas una por una. Las preguntas quedarán almacenadas en la base de datos, por lo que no será necesario volverlas a insertar.

3.1 Crear Preguntas

El botón "Crear pregunta +" está escrito en un color azul claro y tiene un tamaño de fuente grande y legible.

Figura 7. Opción de crear nueva pregunta

Para crear una nueva pregunta el usuario de la aplicación debe hacer clic en la opción “Crear Pregunta +” en la parte superior derecha de la interfaz del módulo de Preguntas. Después de que el usuario hace clic en la opción, aparecerá un formulario con los datos necesarios de la pregunta. Es importante que en el formulario de cada pregunta se especifique correctamente su relación (característica de cada pregunta que vincula a un participante con las preguntas que debe resolver).

El usuario puede crear cualquiera de los siguientes tipos de pregunta:

3.1.1 Pregunta abierta

Participantes Resultados

Crea una nueva pregunta

• Elige el tipo de la pregunta

Pregunta abierta Opción múltiple Múltiple respuesta Matriz de respuestas

• Ingresa el enunciado de la nueva pregunta

Sugerencias al programa académico

• Obligatoria

• Relacionada con:

Todos Si

Seleccione Seleccione uno

Seleccione Seleccione uno

• Factor: 10 • Característica: 0 • Indicador: 0

Cerrar Guardar

Figura 8. Formulario de pregunta abierta

Las preguntas abiertas son las preguntas que no presentan ningún tipo de opciones de respuesta, la persona encargada de resolver la encuesta puede llenar el campo de la pregunta con el texto que desee.

3.1.2 Pregunta de opción múltiple con única respuesta

The screenshot shows a web form titled "Crea una nueva pregunta" (Create a new question). At the top, there are tabs for "Participantes" and "Resultados". The form is divided into several sections:

- Elige el tipo de la pregunta** (Choose the type of question): Four radio buttons are present: "Pregunta abierta" (unselected), "Opción múltiple" (selected), "Múltiple respuesta" (unselected), and "Matriz de respuestas" (unselected). Below each radio button is a small icon representing the question type.
- Ingresa el enunciado de la nueva pregunta** (Enter the statement of the new question): A text input field containing the question: "¿Considera que los propósitos del programa se están cumpliendo?"
- Opciones de las filas** (Options of the rows): A vertical list of input fields for answer options. The first two are "Si" and "No". There are seven empty input fields below them.
- Obligatoria** (Mandatory): A checkbox that is checked.
- Relacionada con:** (Related to): A section with two columns of dropdown menus. The first column has three dropdowns, all with "Tipo" selected. The second column has three dropdowns, with "Estudiante", "Profesor", and "Seleccione uno" selected.
- Factor:** (Factor): An input field with the value "1".
- Característica:** (Characteristic): An input field with the value "1".
- Indicador:** (Indicator): An input field with the value "1".

At the bottom right, there are two buttons: "Cerrar" (Close) and "Guardar" (Save).

Figura 9. Formulario de pregunta de opción múltiple con única respuesta

Las preguntas de opción múltiple con única respuesta presentan una lista de opciones de respuesta, la persona encargada de resolver la encuesta elige solo una de esas opciones como respuesta a la pregunta.

3.1.3 Pregunta de opción múltiple con múltiple respuesta

The screenshot shows a web form titled "Crea una nueva pregunta" with a close button (X) in the top right corner. The form is divided into several sections:

- Elige el tipo de la pregunta:** Four radio buttons are present: "Pregunta abierta", "Opción múltiple", "Múltiple respuesta" (which is selected), and "Matriz de respuestas". Below each radio button is a small icon representing the question type.
- Ingresa el enunciado de la nueva pregunta:** A text input field containing the text "Indique la(s) razón(es) por la(s) cual(es) escogió el Programa".
- Opciones de las filas:** A vertical list of seven input fields. The first four contain the text "Prestigio", "Calidad", "Costos", and "Proyección laboral". The remaining three are empty.
- Obligatoria:** A checkbox that is checked.
- Relacionada con:** Three dropdown menus. The first two are labeled "Tipo" and both have "Estudiante" selected. The third is labeled "Seleccione" and has "Egresado" selected.
- Factor:** An input field containing the number "1".
- Característica:** An input field containing the number "2".
- Indicador:** An input field containing the number "1".

At the bottom right of the form, there are two buttons: "Cerrar" and "Guardar".

Figura 10. Formulario de pregunta de opción múltiple con múltiple respuesta

Las preguntas de opción múltiple con múltiple respuesta también presentan una lista de opciones de respuesta, pero en este tipo de preguntas la persona encargada de resolverla puede seleccionar más de una opción como su respuesta.

3.1.4 Pregunta de matriz de respuestas

The screenshot shows a web interface for creating a new question. At the top, there are tabs for 'Participantes' and 'Resultados'. The main heading is 'Crea una nueva pregunta'. Below this, there are radio buttons to select the question type: 'Pregunta abierta', 'Opción múltiple', 'Múltiple respuesta', and 'Matriz de respuestas' (which is selected). There are four small icons representing different question types. The next step is to 'Ingresa el enunciado de la nueva pregunta', with the example text 'Califique los siguientes aspectos de los profesores del programa'. Below this, there are two columns of options: 'Opciones de las filas' and 'Opciones de las columnas'. The row options include 'Planeación de clases', 'Planeación de tutorías', 'Uso de recursos didácticos', 'Claridad en explicación', and 'Estímulo a la participación'. The column options include 'Deficiente', 'Mala', 'Regular', 'Buena', and 'Excelente'. There are also checkboxes for 'Obligatoria' (checked) and 'Relacionada con:' with dropdown menus for 'Tipo', 'Seleccione', and 'Estudiante'. At the bottom, there are input fields for 'Factor' (1), 'Característica' (3), and 'Indicador' (2). Finally, there are 'Cerrar' and 'Guardar' buttons.

Figura 11. Formulario de pregunta de matriz de respuestas

Las preguntas de tipo Matriz de respuestas presentan opciones horizontales y verticales, la persona encargada de resolverlas debe seleccionar una opción de cada fila según el valor que se presenta en cada columna.

3.2 Visualizar preguntas

A medida que las preguntas creadas, son almacenadas en la base de datos e inmediatamente se pueden visualizar en la vista del banco de datos.

Banco de Preguntas

[Crear pregunta +](#)

Para llenar el banco de preguntas haz clic en la opción "Crear Pregunta +" y completa el formulario por cada pregunta nueva.

1. ¿Considera que los propósitos del programa se están cumpliendo?

Si

No

Pregunta obligatoria: Si
Relacionada con: Tipo > Profesor; Tipo > Estudiante;
Factor:1 Característica:1 Indicador:1

2. Indique la(s) razón(es) por la(s) cual(es) escogió el Programa

Prestigio

Calidad

Costos

Proyección laboral

Pregunta obligatoria: Si
Relacionada con: Tipo > Estudiante; Tipo > Egresado;
Factor:1 Característica:2 Indicador:1

Figura 12. Banco de preguntas

3.3 Editar preguntas

Cuando el usuario hace clic en el botón "Editar" de alguna de las preguntas que existen en el bando de preguntas, aparecerá un formulario con los datos de la pregunta que se desea modificar y el respectivo botón "Actualizar" para culminar la acción. En la siguiente figura se presenta el formulario para modificar una pregunta.

Editar pregunta

- Tipo de la pregunta
 Pregunta abierta Opción múltiple Múltiple respuesta Matriz de respuestas
- Enunciado
- Opciones de las filas
- Obligatoria
- Relacionada con:

Tipo	Profesor
Tipo	Estudiante
- Factor: • Característica: • Indicador:

Figura 13. Formulario para modificar una pregunta

3.4 Eliminar preguntas

Eliminar

Figura 14. Botón de eliminar pregunta

Para eliminar una pregunta el usuario debe ubicar la pregunta que ya no necesita y hace clic en el respectivo botón de eliminar, la pregunta ya no aparecerá en el banco de preguntas.

4. Módulo: Participantes

The screenshot shows the 'Participantes' module interface. At the top, there is a navigation bar with links for 'Inicio', 'Preguntas', 'Encuestas', 'Participantes', and 'Resultados', along with a 'Cerrar sesión' button. Below the navigation bar, the title 'Participantes' is displayed, followed by a subtitle: 'Las siguientes personas se encuentran actualmente ejerciendo las funciones del programa en proceso de acreditación.' A search bar is present with the text 'Buscar en todos los campos:' and a placeholder 'Palabra clave'. Below the search bar, there is a table with columns for 'Nombres', 'Tipo', 'Correo', 'Editar', and 'Eliminar'. The 'Nombres' column contains an empty input field, and the 'Tipo' column has a dropdown menu set to 'Todos'. The 'Correo' column contains an empty input field. Below the table, a message states 'No hay personas con esa palabra'. At the bottom of the table, there are pagination controls showing '10' items per page. A blue button labeled 'Agregar nuevo participante +' is located below the table.

Figura 15. Módulo Participantes sin datos de participantes

El módulo de Participantes permite realizar diferentes funciones que faciliten al usuario la gestión de los datos de las personas involucradas en el proceso de autoevaluación. Inicialmente la lista de participantes se encuentra vacía, el usuario de la aplicación deberá insertar los datos de cada persona (proceso que solo realizará una sola vez) para llenar la lista de los participantes.

4.1 Agregar participantes

A close-up of the blue button labeled 'Agregar nuevo participante +'.

Figura 16. Botón para agregar un nuevo participante

Para agregar un nuevo participante, el usuario debe hacer clic en el botón "Agregar nuevo participante +". Después de hacer clic en el botón, la aplicación mostrará un formulario donde se deben ingresar los datos del participante.

Nota: Los datos ingresados en este módulo son datos básicos, los datos completos de cada participante se deben ingresar o editar desde el sistema de información general de Autoevaluación.

Participantes Resultados

Agrega un nuevo participante

- Nombres del participante:
- Tipo del participante:
- Estado civil del participante:
- Correo del participante:

Figura 17. Formulario para crear un nuevo participante

4.2 Visualizar participantes

A medida que los datos de los participantes son ingresados y almacenados en la base de datos, en éste módulo (Participantes) se pueden observar los participantes en una lista con algunos de sus datos.

Inicio Preguntas Encuestas Participantes Resultados Cerrar sesión

Participantes

Las siguientes personas se encuentran actualmente ejerciendo las funciones del programa en proceso de acreditación.

Buscar en todos los campos: <input type="text" value="Palabra clave"/>				
Nombres <input type="text"/>	Tipo <input type="text" value="Todos"/>	Correo <input type="text"/>	Editar	Eliminar
Juan Sebastian	Estudiante	juansebastian@gmail.com	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Nicolás	Egresado	nicolas@gmail.com	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
David Jesús	Estudiante	davids2316@gmail.com	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
Edwin Eduardo	Egresado	edwineduardo@hotmail.com	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

Figura 18. Lista de los participantes

4.3 Editar participantes

Si el administrador necesita modificar los datos de un participante, debe hacer clic en el botón “Editar” correspondiente al participante deseado. La aplicación mostrará un formulario con los datos del participante que pueden ser modificados.

Inicio Preguntas Encuestas Participantes Resultados Cerrar sesión

Editar participante

Nombres del participante:

Tipo del participante:

Estado civil del participante:

Correo del participante:

Figura 19. Formulario de editar participante

4.4 Eliminar participantes

Cuando el administrador decida eliminar un participante del proceso, debe hacer clic en el botón “Eliminar” que pertenece al participante deseado de la lista de participantes.

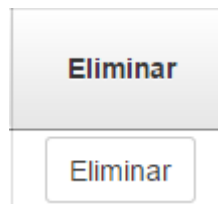
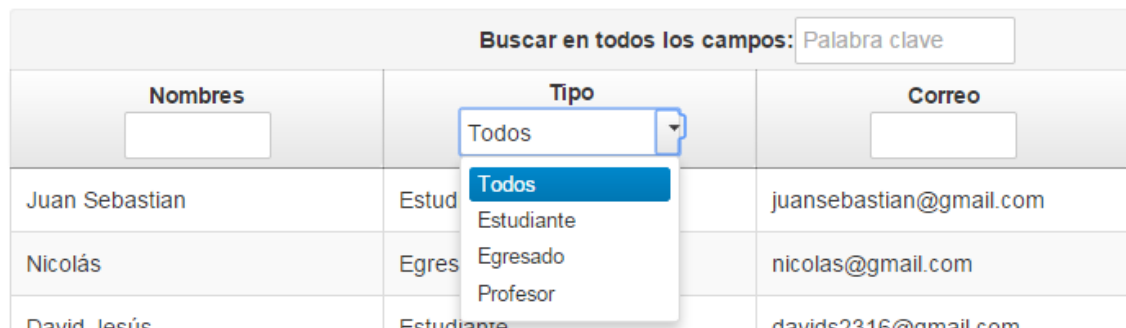


Figura 20. Botón para eliminar un participante

El participante es eliminado de la base de datos y por lo tanto ya no aparecerá en la lista de participantes.

4.5 Buscar participantes

La tabla del módulo “Participantes” no solo permite visualizar los participantes, también permite realizar búsquedas con los datos de los participantes, las búsquedas se pueden realizar usando los diferentes métodos de búsqueda en los diferentes campos.

The image shows a search interface for a table of participants. At the top, there is a search bar with the text 'Buscar en todos los campos:' followed by a text input field containing 'Palabra clave'. Below this is a table with three columns: 'Nombres', 'Tipo', and 'Correo'. The 'Tipo' column has a dropdown menu open, showing options: 'Todos', 'Estudiante', 'Egresado', and 'Profesor'. The table contains three rows of data: Juan Sebastian (Estudiante, juansebastian@gmail.com), Nicolás (Egresado, nicolas@gmail.com), and David Jesús (Estudiante, david2316@gmail.com).

Buscar en todos los campos: Palabra clave		
Nombres	Tipo	Correo
Juan Sebastian	Estud	juansebastian@gmail.com
Nicolás	Egres	nicolas@gmail.com
David Jesús	Estudiante	davide2316@gmail.com

Figura 21. Campos para la búsqueda de participantes

5. Módulo: Encuestas

The screenshot shows the 'Administrar encuestas' (Manage surveys) interface. At the top, there is a navigation bar with links for 'Inicio', 'Preguntas', 'Encuestas', 'Participantes', and 'Resultados', along with a 'Cerrar sesión' (Logout) button. Below the navigation bar, the title 'Administrar encuestas' is displayed. There are two buttons: 'Generar encuestas' (Generate surveys) and 'Generar encuesta específica' (Generate specific survey). A search bar labeled 'Buscar en todos los campos:' contains the text 'Palabra clave'. Below the search bar is a table with the following structure:

Nombre del encuestado	Tipo del encuestado	Encuesta	Enviada	Eliminar encuesta
	Todos			

Below the table, it says 'No existen encuestas' (No surveys exist). At the bottom of the table area, there are pagination controls showing '10' items per page. A blue button labeled 'Enviar encuestas' (Send surveys) is located below the table.

Figura 22. Módulo de encuestas sin encuestas creadas

El módulo Encuestas se encarga de ofrecer las funciones necesarias para la gestión de encuestas. Una vez que la lista de los participantes esté completa y el banco de preguntas este completamente verificado se procede a generar las encuestas.

5.1 Generar encuestas

Para generar las encuestas es necesario que los participantes y las preguntas sean previamente verificados, luego el administrador debe hacer clic en el botón "Generar encuestas" que aparece en la esquina superior izquierda del módulo Encuestas.

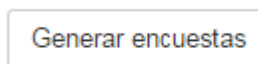


Figura 23. Botón para generar las encuestas

Una vez que el administrador hace clic en el botón, la aplicación solicita una confirmación debido a que es una acción que sobrescribe datos de forma masiva.

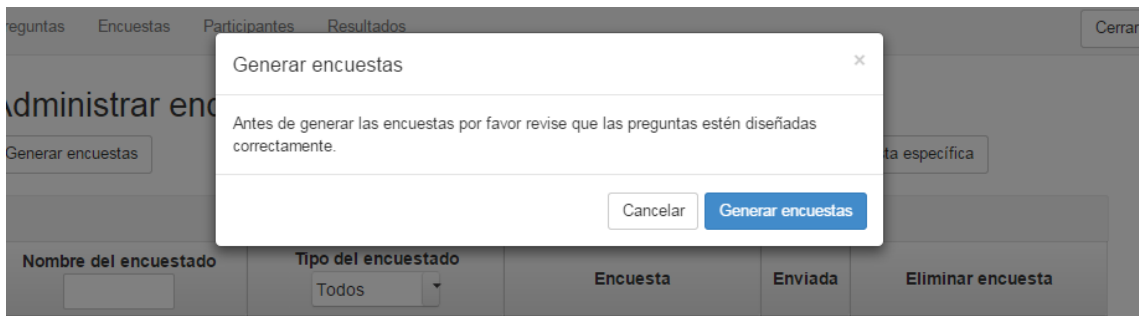


Figura 24. Ventana de confirmación para generar las encuestas

5.2 Visualizar encuestas

Después de generadas las encuestas, el módulo Encuestas contiene una tabla que se encarga de mostrar las encuestas actuales en la base de datos.

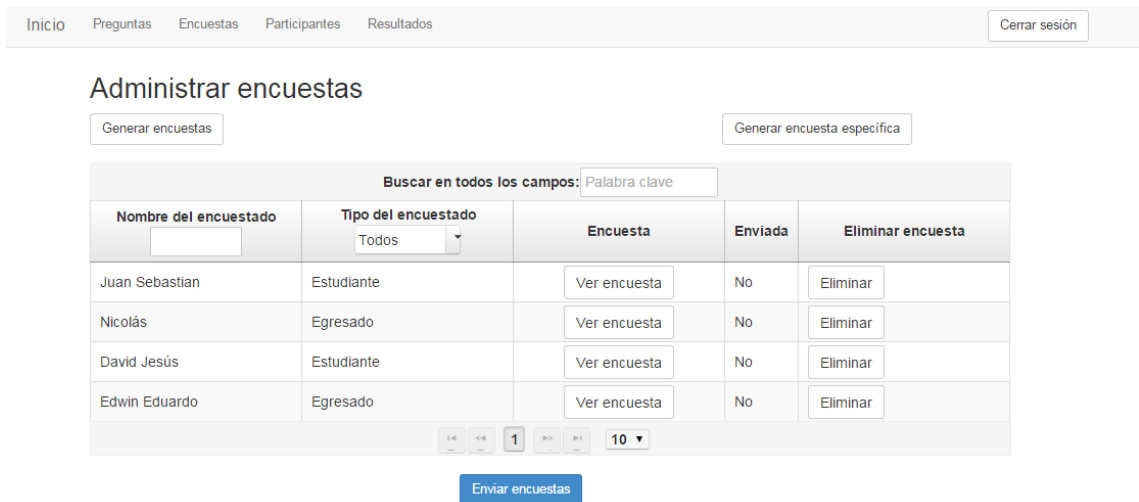


Figura 25. Tabla que muestra las encuestas existentes

5.3 Generar encuesta específica

Si después de realizar el proceso de generar las encuestas se necesita agregar un nuevo participante, el administrador puede hacer clic en el botón “Generar encuesta específica” que se encuentra en la esquina superior derecha del módulo y generar la encuesta solo para el nuevo participante.

Generar encuesta específica

Figura 26. Botón para generar una encuesta específica

Después la aplicación solicitará, por medio de una tabla, el participante para el cual se desea generar la encuesta.

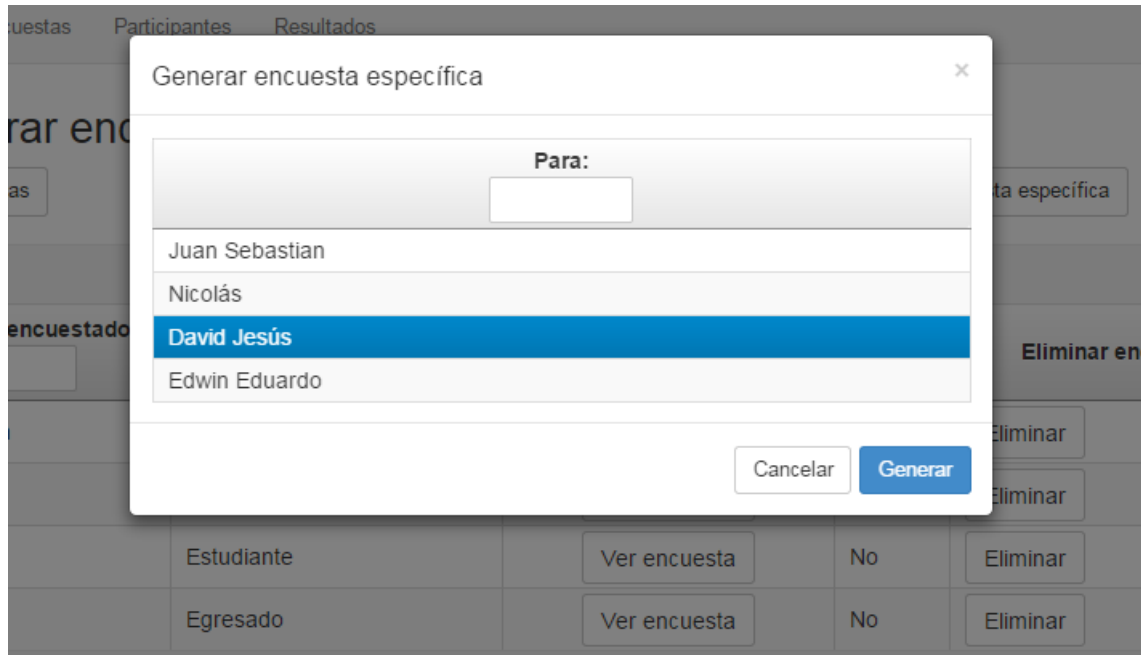


Figura 27. Tabla para generar encuesta específica a un participante

5.4 Visualizar encuestas

Si el administrador desea realizar verificaciones particulares en las encuestas generadas, puede visualizar las encuestas haciendo clic en el botón “Ver encuesta” que contiene cada encuesta en la tabla de Encuestas.

David Jesús	Estudiante	Ver encuesta
-------------	------------	--------------

Figura 28. Encuesta con su respectivo botón “Ver encuesta”

Al hacer clic en el botón se podrán visualizar las preguntas que contiene esa encuesta. Las encuestas se visualizan solo con fines de supervisión, la aplicación en este punto no ha enviado las encuestas y por lo tanto la revisión detallada o visualización de las encuestas son solo acciones opcionales de control.

Inicio Preguntas Encuestas Participantes Resultados Cerrar sesión

Encuesta para David Jesús

1. ¿Considera que los propósitos del programa se están cumpliendo?

Si

No

Pregunta obligatoria: Si

2. Indique la(s) razón(es) por la(s) cual(es) escogió el Programa

Prestigio

Calidad

Costos

Proyección laboral

Pregunta obligatoria: Si

3. Califique los siguientes aspectos de los profesores del programa

Planeación de clases Deficiente Mala Regular Buena Excelente

Planeación de tutorías Deficiente Mala Regular Buena Excelente

Volver a encuestas

Figura 29. Visualización de las preguntas de una encuesta

5.5 Eliminar encuestas

Cada encuesta contiene un botón que le permite al administrador eliminarla si es necesario.

David Jesús	Estudiante	Ver encuesta	No	Eliminar
-------------	------------	---------------------------	----	-----------------------

Figura 30. Encuesta con su respectivo botón "Eliminar"

Después de hacer clic en el botón, la encuesta es eliminada de la base de datos y por lo tanto no se visualizará en la tabla de encuestas y tampoco podrá ser enviada al respectivo participante.

5.6 Enviar encuestas

Después de generar las encuestas, la aplicación las almacena en la base de datos con los datos de cada participante (especialmente su email). Cuando el usuario decide enviar las encuestas, debe hacer clic en el botón “Enviar encuestas” que se encuentra en la parte inferior del módulo de encuestas.

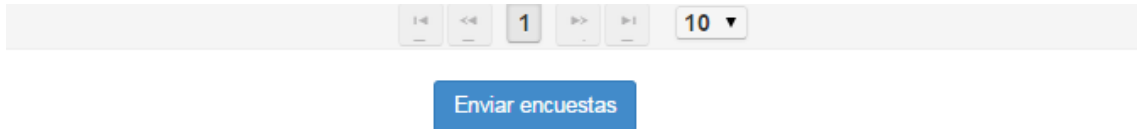


Figura 31. Botón para enviar las encuestas

Después de hacer clic en el botón, la aplicación le solicitará al administrador los datos de envío de encuestas por correo electrónico. El formulario se puede observar en la siguiente figura.

A screenshot of a web application showing a modal dialog box titled 'Enviar encuestas'. The dialog has a close button (X) in the top right corner. It contains the following fields:

- 'Email emisor': A text input field containing 'davids2316@gmail.com'.
- 'Resumen': A section header.
- 'Asunto del mensaje': A text input field containing 'Encuestas Autoevaluación'.
- 'Número de destinatarios': A text input field containing '4'.
- 'Mensaje': A text area containing the text 'Por favor llena la siguiente encuesta' followed by the URL 'http://filosofiyensenanza.uis.edu.co:8080/Encuestas/resolverEncuesta.xhtml?'.

At the bottom right of the dialog are two buttons: 'Cerrar' and 'Enviar'.

Figura 32. Formulario de datos para el envío de encuestas

Una vez enviadas las encuestas por correo, la tabla que muestra las encuestas cambia la columna “Enviada”, la cambia del valor “No” al valor “Si”, lo que indica que el enlace de la encuesta fue enviado exitosamente al email del participante.

Buscar en todos los campos: <input type="text" value="Palabra clave"/>				
Nombre del encuestado	Tipo del encuestado	Encuesta	Enviada	Eliminar encuesta
<input type="text" value=""/>	Todos			
Juan Sebastian	Estudiante	<input type="button" value="Ver encuesta"/>	Si	<input type="button" value="Eliminar"/>
Nicolás	Egresado	<input type="button" value="Ver encuesta"/>	Si	<input type="button" value="Eliminar"/>
David Jesús	Estudiante	<input type="button" value="Ver encuesta"/>	Si	<input type="button" value="Eliminar"/>
Edwin Eduardo	Egresado	<input type="button" value="Ver encuesta"/>	Si	<input type="button" value="Eliminar"/>

Figura 33. Encuestas después de enviadas

Al email de cada participante se envía un correo electrónico con el enlace de la encuesta que debe responder, la siguiente figura muestra un ejemplo de un email.

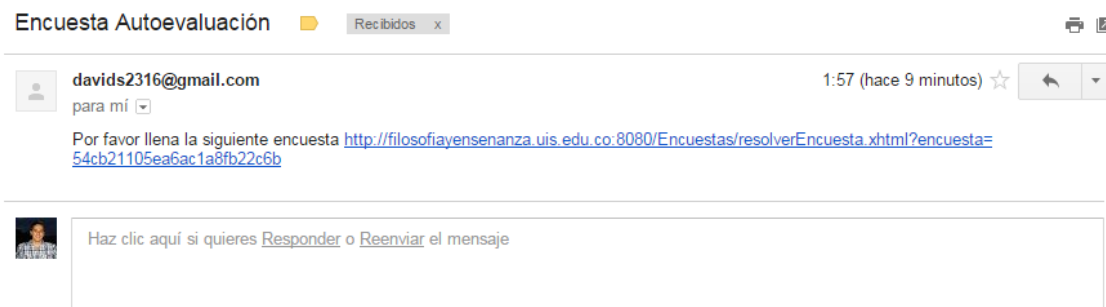


Figura 34. Ejemplo de email recibido por los participantes

5.7 Resolver encuestas

Esta acción pertenece a los participantes. Los participantes deben hacer clic en el enlace que se envía a sus correos, este enlaces los dirige a la aplicación de encuestas y muestra la encuesta que le corresponde al participante.

6. Módulo: Resultados

En el módulo de resultados el administrador podrá visualizar los datos recolectados para su respectivo análisis.

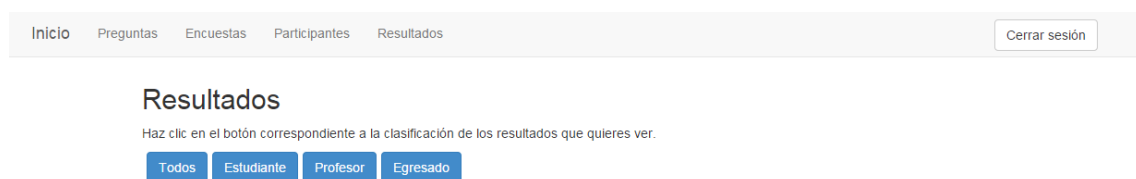


Figura 37. Módulo de resultados

6.1 Ver resultados

El módulo Resultados le permite al administrador visualizar los datos recolectados según la clasificación que desee. La clasificación se basa en el tipo de las personas que participan en el proceso de autoevaluación. El administrador debe hacer clic en el botón respectivo de los resultados que desea ver.



Figura 38. Botones de clasificación de resultados

Al hacer clic en alguno de los botones, la aplicación muestra los datos recolectados y almacenados en la base de datos que cumplan ese criterio de clasificación.

Resultados

Haz clic en el botón correspondiente a la clasificación de los resultados que quieres ver.



Respuestas de Estudiante

¿Considera que los propósitos del programa se están cumpliendo?

Opción	Veces	Porcentaje
Si	1	100 %

Factor 1

Característica 1

Indicador 1

Indique la(s) razón(es) por la(s) cual(es) escogió el Programa

Opción	Veces	Porcentaje
Calidad	1	100 %
Proyección laboral	1	100 %
Prestigio	1	100 %
Costos	1	100 %

Factor 1

Característica 2

Indicador 1

Figura 39. Resultados obtenidos de los estudiantes

Los datos son recolectados a medida que las encuestas son resueltas, para mayor confiabilidad de la información se sugiere analizar los datos una vez que todos los participantes completen sus encuestas.

7. Cerrar sesión

Para cerrar sesión en la aplicación, el administrador debe hacer clic en el botón “Cerrar sesión” que se encuentra la derecha del menú principal de la aplicación.

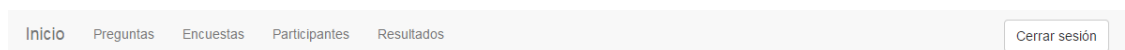


Figura 40. Botón de cerrar sesión en el menú principal

Anexo B: Diagrama de clases UML del modelo de datos

