

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingeniero e Ingeniera de Sistemas**

**TEMA:
DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE
REACTIVOS EN LA CARRERA DE INGENIERÍA DE SISTEMAS DE LA
UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO.**

**AUTORES:
GIOVANNY MAURICIO YANCHA PINDUISACA
GABRIELA YAJAIRA TIPÁN TAPIA**

**TUTOR:
FRANKLIN EDMUNDO HURTADO LARREA**

Quito, febrero del 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros Giovanni Mauricio Yancha Pinduisaca con documento de identificación N° 1722762661, y Gabriela Yajaira Tipán Tapia con documento de identificación N° 1723210470, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE REACTIVOS EN LA CARRERA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO”, mismo que ha sido desarrollado para optar por el título de: Ingeniero e Ingeniera de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

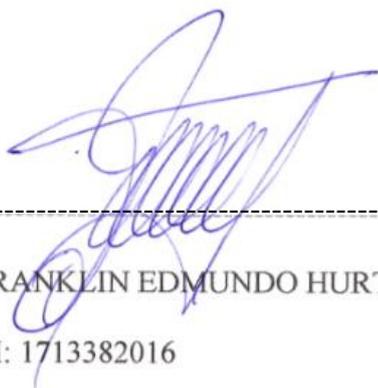

Giovanni Mauricio
Yancha Pinduisaca
CI.1722762661


Gabriela Yajaira
Tipán Tapia
CI.1723210470

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, con el tema: DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE REACTIVOS EN LA CARRERA DE INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO, realizado por Giovanni Mauricio Yancha Pinduisaca y Gabriela Yajaira Tipán Tapia, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero del 2017



FRANKLIN EDMUNDO HURTADO LARREA

CI: 1713382016

Dedicatoria

Dedico este trabajo a Dios por haberme dado las fuerzas, la inteligencia y la sabiduría que necesitaba para culminar el proyecto; porque me enseñó a luchar por mis sueños y objetivos y porque siempre ha sido mi soporte en cada situación de la vida.

Dedico este trabajo a mi esposo y a mi hijo por haberme sostenido con su amor en los momentos difíciles y por haber sacrificado nuestro tiempo juntos para que yo pudiera culminar este objetivo, porque sólo con su presencia me fortalecieron y me motivaron a cumplir este proyecto; los amo con todo mi corazón.

Dedico este trabajo también a mis padres quienes me han afirmado en cada paso que he dado durante toda mi vida, porque sin ellos esto no habría sido posible, porque me han acompañado, me han apoyado y me han impulsado en cada etapa para que yo logre cumplir mis sueños.

Gabriela Yajaira Tipán Tapia

El presente trabajo de titulación primeramente me gustaría agradecerle a Dios y la virgencita de Baños por bendecirme siempre, posterior se lo dedico de manera muy especial a mi padre, que es el principal ejemplo de tenacidad ante la vida, demostrando que los retos te permiten obtener experiencia y forman parte en la vida, su ejemplo me alienta a ser tanto como el, a mi madre que con el cariño inmenso me escuchó y apoyó en las etapas duras, inculcándome la sencillez y humildad que es la mejor presentación de una persona y mi hermana una de las mujeres que más quiero y admiro que siempre está ahí con sus grandes cualidades, su apoyo fundamental fue parte de este trabajo.

A mi familia en general los cuales están pendientes con su apoyo incondicional para motivarme y darme sus consejos que me permitieron llegar a culminar esta meta.

Giovanny Mauricio Yancha Pinduisaca

Agradecimiento

Agradecemos profundamente a nuestro tutor de proyecto de titulación el Ingeniero Franklin Hurtado por habernos guiado con sabiduría y paciencia y por habernos transmitido su conocimiento de una manera desprendida para poder culminar con nuestro trabajo.

Gabriela Yajaira Tipán Tapia
Giovanny Mauricio Yanca Pinduisaca

ÍNDICE

INTRODUCCIÓN	1
Justificación.....	2
Objetivo.....	2
Objetivo General.	2
Objetivos Específicos.....	3
Marco metodológico.	3
Capítulo 1	7
1.1 Alcance.....	7
Elaboración de reactivos.	7
Revisión de reactivos.	7
Uso de reactivos para diferentes tipos de evaluaciones.	7
Generación de reportes.....	7
1.2 Necesidades.....	8
1.3 Proceso actual de manejo de reactivos.....	8
1.4 Requerimientos	10
1.4.1 Visión del software.	10
1.4.2 Especificación de requerimientos.	12
Capítulo 2.....	17
Marco teórico	17
2.1. Proceso de software.....	17
2.2. Calidad de software.....	18
2.3 Aplicaciones web	19
2.4 Herramientas de programación	20
2.4.1 Java Server Faces.	20
2.4.2 Primefaces.....	21
2.4.3 Hibernate.....	21
2.4.4 JBoss.	21
2.4.5 Javascript.....	22
2.5 Metodologías de desarrollo de software	22
2.5.1 Scrum.	23
2.6 Teoría de reactivos	25
2.7 BPMN	27
Capítulo 3.....	28
Diseño del aplicativo.....	28
3.1 Casos de uso.....	28
3.2 Diagrama de la base de datos	35
3.3 Diagrama de clases.....	38
3.4 Diseño de interfaces	40
Capítulo 4.....	42
Construcción y pruebas	42
4.1 Estándares	42
4.2 Modificaciones.....	43
4.3 Diagrama del proceso automatizado	44
4.4 Arquitectura del sistema.....	48
4.5 Modelado de interfaces	51

4.6 Código relevante	55
4.6.1 Métodos de acceso al sistema.	55
4.6.2 Métodos para los reactivos.....	56
4.6.3 Métodos para almacenar las respuestas.....	57
4.6.4 Método para añadir imágenes.	58
4.6.5 Métodos para generar examen.	59
4.6.6 Método para generar el examen en un archivo pdf.	60
4.6.7 Método para la comparación de la similitud de los reactivos.	61
4.7 Diccionario de datos.....	63
4.8 Pruebas	66
4.8.1 Pruebas unitarias.	67
4.8.2 Pruebas funcionales.....	69
4.8.3 Pruebas de integración.	74
4.8.4 Pruebas de rendimiento.....	75
CONCLUSIONES	81
RECOMENDACIONES	83
GLOSARIO DE TÉRMINOS	84
LISTA DE REFERENCIAS	85

ÍNDICE DE TABLAS

Tabla 1. MR001 Ingresar al sistema	12
Tabla 2. MR002 Permitir la parametrización de reactivos	12
Tabla 3. MR003 Permitir la parametrización de docentes y materias	13
Tabla 4. MR004 Elaborar reactivos	13
Tabla 5. MR005 Revisar reactivos	14
Tabla 6. MR006 Generación de evaluaciones.....	14
Tabla 7. MR007 Generación de exámenes complexivos o de fin de carrera.....	15
Tabla 8. MR008 Rendir evaluaciones o exámenes complexivos o de fin de carrera.	15
Tabla 9. MR009 Generar reportes.....	15
Tabla 10. Caso de uso 1: Inicio de sesión	29
Tabla 11. Caso de uso 2: Parametrización de reactivos	30
Tabla 12. Caso de uso 3: Administración de docentes por materia	30
Tabla 13. Caso de uso 4: Elaboración de reactivos.....	31
Tabla 14. Caso de uso 5: Revisión de reactivos.....	32
Tabla 15. Caso de uso 6: Generación de evaluaciones	32
Tabla 16. Caso de uso 7: Generación de exámenes complexivos o de fin de carrera	33
Tabla 17. Caso de uso 8: Rendir evaluación	33
Tabla 18. Caso de uso 9: Generación de reportes	34
Tabla 19. REACTIVOS PREGUNTA APROBACIÓN	63
Tabla 20. REACTIVOS OPCIONES RESPUESTAS APROBADAS	65
Tabla 21. Plantilla de pruebas	67
Tabla 22. Prueba 1.....	67
Tabla 23. Prueba 2.....	68
Tabla 24. Prueba 3.....	68
Tabla 25. Prueba 4.....	69
Tabla 26. Prueba 5.....	70
Tabla 27. Prueba 6.....	70
Tabla 28. Prueba 7.....	71
Tabla 29. Prueba 8.....	72
Tabla 30. Prueba 9.....	72

ÍNDICE DE FIGURAS

Figura 1. Product Backlog del proyecto.....	4
Figura 2. Sprint 1.....	5
Figura 3. Sprint 2.....	5
Figura 4. Procesos manuales.....	9
Figura 5. Diagrama general de casos de uso.....	28
Figura 6. Diagrama de la base de datos.....	37
Figura 7. Diagrama de clases.....	39
Figura 8. Diseño de la pantalla de parametrización de reactivos.....	40
Figura 9. Diseño de la pantalla de ingreso de usuarios.....	40
Figura 10. Diseño de la pantalla de actualización de usuarios.....	41
Figura 11. Diseño de la pantalla de elaboración de reactivos.....	41
Figura 12. Proceso automatizado del módulo de Administración de la aplicación ...	45
Figura 13. Proceso automatizado roles docente, comisión y estudiante.....	47
Figura 14. Diseño de la arquitectura general de la aplicación.....	48
Figura 15. Capa vista dentro del proyecto.....	49
Figura 16. Capa modelo dentro del paquete en el proyecto.....	50
Figura 17. Clase modelo de mapeo entre la aplicación y la base de datos.....	50
Figura 18. Capa control dentro del paquete en el proyecto.....	51
Figura 19. Ingreso y actualización de la parametrización de reactivos.....	52
Figura 20. Parametrización ingreso y actualización de docentes.....	52
Figura 21. Parametrización ingreso y actualización de materias.....	53
Figura 22. Elaboración y actualización de los reactivos.....	53
Figura 23. Pantalla de ingreso del reactivo.....	54
Figura 24. Pantalla que muestra el ingreso de respuestas por reactivo.....	54
Figura 25. Método de la clase usuario sistema.....	55
Figura 26. Método que realiza la encriptación de la contraseña.....	55
Figura 27. Método que parametriza las pantallas dependiendo del rol de usuario....	55
Figura 28. Permite manejar los controladores en varias clases de la mismas.....	56
Figura 29. Método que obtiene la fecha actual del sistema.....	56
Figura 30. Método que ingresa el reactivo escrito desde la interfaz.....	57
Figura 31. Método general y recuperación del código del reactivo ingresado.....	57
Figura 32. Método que ingresa las respuestas por cada reactivo.....	58
Figura 33. Método que permite añadir imágenes como anexos.....	58
Figura 34. Librería ITEXT que genera los exámenes en PDF.....	59
Figura 35. Genera el número de reactivos para el examen automático.....	59
Figura 36. Librería ITEXT que genera los exámenes en PDF.....	60
Figura 37. Parte del método de añadir reactivos automáticos de acuerdo al porcentaje de sus ejes.....	60
Figura 38. Instancia del método que codifica el contenido HTML para añadir a PDF.....	61
Figura 39. Parte del método que añade las respuestas de los reactivos en el documento PDF.....	61
Figura 40. Algoritmo que realiza la comparación entre dos cadenas de caracteres...	62
Figura 41. Pruebas realizadas del algoritmo para la comparación de cadenas.....	63
Figura 42. Instancia del método que permite comparar el reactivo ingresado.....	63

Figura 44. Se realizó una prueba de rendimiento con 1400 usuario en 10 segundos 76
Figura 45. Se realizó una prueba de rendimiento con 100 usuario en 10 segundos .. 77
Figura 46. Se realizó una prueba de rendimiento con 100 usuario en 10 segundos .. 77
Figura 47. Se realizó una prueba de rendimiento con 100 docentes en 10 segundos 78
Figura 48. Se realizó una prueba de rendimiento con 100 docentes en 10 segundos 79
Figura 49. Se realizó una prueba de rendimiento con 1000 usuarios en 10 segundos80
Figura 50. Se realizó una prueba de rendimiento con 1000 usuarios en 10 segundos80

Resumen

En el presente proyecto se desarrolló una aplicación web de gestión de reactivos para la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Campus Sur, con el objetivo de facilitar al docente el manejo de cada uno de los reactivos creados para sus materias y a su vez, tener un mejor control de los reactivos enviados a revisión para su futuro uso en exámenes complexivos o de fin de carrera.

En el capítulo I se presenta el tema del proyecto, los antecedentes, la justificación, el alcance, objetivos y marco metodológico.

En el capítulo II se presenta el marco teórico en el cual se indica el ciclo de vida que se debe seguir para conseguir un producto de software, su calidad, aplicaciones web, las herramientas de programación utilizadas, la metodología Scrum y teoría de reactivos.

En el capítulo III se presenta el análisis y el diseño de la aplicación web, procesos manuales y automáticos, especificación de requerimientos, casos de uso, modelado de la base de datos y diseño de interfaces de la aplicación desarrollada.

En el capítulo 4 se presenta la construcción del software con los procesos automatizados al culminar el desarrollo, los métodos de programación más importantes utilizados, la arquitectura utilizada y el diccionario de los datos utilizados; adicional se presentan diferentes pruebas que se realizaron durante y después del desarrollo de la aplicación.

Finalmente se indican las conclusiones y recomendaciones obtenidas al culminar el desarrollo del proyecto de gestión de reactivos.

Abstract

In the present project of a web application of question management was developed for the Systems Engineering career at the Salesian Polytechnic University Campus Sur, with the objective of facilitating to the teacher the management of each of the questions created for their subjects and in turn, have a better control of the questions submitted for review for future use in complex or end of career evaluations.

Chapter I presents the project theme, background, justification, scope, objectives and methodological framework.

Chapter II presents the theoretic framework that indicates the life cycle that must be followed to get a software product, its quality, web applications, programming tools used, Scrum methodology and questions theory.

Chapter III presents the analysis and design of the web application, manual and automatic processes, specification of requirements, use cases, database modeling and interface design of the developed application.

Chapter 4 presents the construction of software with automated processes at the end of development, the most important programming methods used, the architecture used and the dictionary of the data used; Additional tests are presented that were made during and after the development of the application.

Finally, the conclusions and recommendations obtained at the end of the development of the reagent management project are indicated.

INTRODUCCIÓN

Hoy en día, debido a los crecientes avances tecnológicos y a las necesidades actuales de las instituciones educativas a nivel mundial, se han logrado automatizar un sin número de procesos que realizan tanto los docentes como los estudiantes, con el objetivo de conservar recursos como costos y tiempo.

El suministrar a las instituciones educativas de recursos tecnológicos como apoyo para sus labores diarias tiene como objetivo mejorar los procesos de enseñanza y aprendizaje proveyendo de esta manera, una educación de alta calidad.

El uso de aplicaciones web dentro de las instituciones educativas ha permitido optimizar las tareas que tienen los docentes facilitando su gestión al contar con datos centralizados y, a su vez, agilizando las tareas asignadas a los estudiantes; de ahí la importancia del uso de la tecnología como apoyo en cada campo incluido el educativo.

Por lo tanto, el objetivo del proyecto de titulación de gestión de reactivos es lograr una automatización de los procesos actuales y una centralización de la información con el fin de facilitar las tareas de docentes y estudiantes concentrando todos los datos en la aplicación web.

Antecedentes

Cada semestre los docentes de la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana deben formular un cierto número de reactivos y enviarlos mediante correo electrónico a una comisión para su revisión, el resultado de esta revisión puede demorar un tiempo considerable debido a que los integrantes de la comisión deben revisar el correo electrónico y enviar la respuesta al respectivo docente sin contar con un plazo determinado de contestación, por lo cual se generan

tiempos muertos; debido al desarrollo del proceso actual, se dificulta tener un rápido procesamiento en lo que respecta a la revisión de reactivos, lo cual eventualmente puede provocar un escenario de desorganización ya que se pueden presentar demoras entre la elaboración y revisión de los reactivos y también ocasionar una pérdida de información debido a la gran cantidad de la misma y a la falta de herramientas para su gestión; por lo tanto, este control manual muestra una dificultad en la comunicación entre los docentes y la comisión para realizar sus funciones de una manera eficaz, eficiente, actualizada y ordenada.

Justificación

Las aplicaciones web brindan amplios servicios de resguardo de datos y son accesibles desde cualquier lugar; al contar con este tipo de tecnología se ha propuesto desarrollar una aplicación web que gestione la creación y uso de reactivos con lo cual se pretende conseguir:

Contar con un flujo de procesos secuenciales y automatizados que eviten la pérdida de información

Disminuir el tiempo de respuesta entre la elaboración de reactivos y su revisión

Uso efectivo de los reactivos para los diferentes eventos de evaluación

Contar con inventarios de reactivos vigentes y dados de baja

Objetivo

Objetivo General.

Desarrollar una aplicación web para la gestión de reactivos en la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana sede Quito.

Objetivos Específicos.

Diseñar la aplicación web de gestión de reactivos de modo que se consideren todos los requerimientos establecidos para su adecuado funcionamiento.

Implementar la aplicación utilizando las herramientas de software libre Java y PostgreSQL.

Verificar que los reactivos ingresados por los docentes no se repitan con los elaborados anteriormente y que reposan en el sistema.

Implementar reportes de reactivos vigentes y dados de baja, de elaboración de reactivos por parte de los docentes y del número total de reactivos por su estado actual.

Marco metodológico.

Para la elaboración del proyecto de gestión de reactivos se utilizó la metodología ágil de desarrollo de software Scrum; dicha metodología fue escogida por las siguientes razones:

El equipo de trabajo fue de dos personas

La posibilidad de cambios de forma y fondo fueron considerados como altamente probables

La exigencia del cliente para obtener un producto tecnológico de alta calidad en un tiempo corto

Scrum cuenta con tres tipos de participantes, que en el contexto del presente proyecto se lo ha manejado de la siguiente manera:

Dueño del producto, rol asignado al docente encargado de la denominada comisión, quien coordina la gestión de reactivos.

Experto Scrum, responsabilidad asignada al docente tutor del proyecto, quien veló y asesoró a los miembros del equipo, para que el proyecto se desarrolle guiándose en la metodología Scrum.

Equipo de trabajo conformado por dos desarrolladores.

Usuarios finales, los cuales son los beneficiarios de la aplicación web desarrollada y que aportaron con su opinión y conocimiento desde el inicio del desarrollo; dichos usuarios son la comisión y la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana Campus Sur.

Se realizó un Product Backlog con el dueño del producto para definir las tareas que se debían realizar, a continuación, se indica una parte del mismo:

Product Backlog

PRODUCT BACKLOG			
Descripción	Sprint asignado	Fecha real	Observaciones
Ingreso al sistema con cuatro tipos de roles: Administrador, docente, comisión de reactivos, estudiante	Sprint 1	15/2/2016	Se inició el desarrollo del producto tecnológico sin la definición de roles para el ingreso al sistema, por lo cual se tuvo que realizar la modificación ya que se estaba manejando el ingreso de forma global
Docente pueda elaborar reactivos para usarlos en evaluaciones de sus diferentes materias	Sprint 1	15/2/2016	
Docente pueda elaborar reactivos para enviarlos a revisión y estos puedan ser usados para exámenes de fin de carrera o complejivos	Sprint 1	15/2/2016	
Administrador de la aplicación pueda realizar la parametrización de docentes	Sprint 2	27/4/2016	Se redefinió el módulo de administración ya que al inicio se manejaba únicamente la parametrización de reactivos y surgió la necesidad de realizar la parametrización de docentes, materias y estudiantes; para lo cual el tutor de la tesis nos brindó su ayuda entregándonos los distributivos con la información de docentes y estudiantes y la información de las materias se las obtuvo de la página web de la UPS.
Administrador de la aplicación pueda realizar la parametrización de materias	Sprint 2	27/4/2016	
Administrador de la aplicación pueda realizar la parametrización de estudiantes	Sprint 2	27/4/2016	
Administrador de la aplicación pueda realizar la parametrización de reactivos	Sprint 2	27/4/2016	

Figura 1. Product Backlog del proyecto
Elaborado por: Gabriela Tipán y Giovanni Yancha

De acuerdo a la metodología Scrum se elaboró también el Sprint Backlog en el cual se han definido las actividades o tareas a realizarse en cada sprint o iteración, en la siguiente figura se detalla una parte del mismo.

SPRINT 1	
Requerimiento	Tarea del Sprint
Ingreso al sistema con cuatro tipos de roles: Administrador, docente, comisión de reactivos, estudiante	Diagrama de base de datos Diagrama de clases Crear interfaz de inicio de sesión Definir roles de administrador, docente, comisión de reactivos, estudiante
Docente pueda elaborar reactivos para usarlos en evaluaciones de sus diferentes materias	Elaboración módulo de elaboración de reactivos
Docente pueda elaborar reactivos para enviarlos a revisión y estos puedan ser usados para exámenes de fin de carrera o complexivos	Elaboración módulo de envío de reactivos a revisión

Figura 2. Sprint 1

Elaborado por: Gabriela Tipán y Giovanni YanCHA

SPRINT 2	
Requerimiento	Tarea del Sprint
Administrador de la aplicación pueda realizar la parametrización de docentes	Definir la forma de obtener la información de docentes Crear interfaz de administración de docentes Definir la forma de cargar la información Definir el diseño del esquema de la base de datos para docentes
Administrador de la aplicación pueda realizar la parametrización de materias	Definir la forma de obtener la información de materias Crear interfaz de administración de materias Definir la forma de cargar la información Definir el diseño del esquema de la base de datos para materias y sus categorías
Administrador de la aplicación pueda realizar la parametrización de estudiantes	Definir la forma de obtener la información de estudiantes Crear interfaz de administración de estudiantes Definir la forma de cargar la información Definir el diseño del esquema de la base de datos para estudiantes
Administrador de la aplicación pueda realizar la parametrización de reactivos	Definir los parámetros generales para el ingreso de reactivos Crear interfaz de administración de parametrización de reactivos

Figura 3. Sprint 2

Elaborado por: Gabriela Tipán y Giovanni YanCHA

El Product Backlog cuenta con 18 requerimientos que se han cumplido durante el desarrollo del proyecto, para lo cual se han definido 5 sprints, cada uno con un detalle de las tareas a realizarse en cada iteración. Para esta planificación se mantuvieron reuniones con el dueño del producto y se documentaron tres entrevistas:

1. En la primera entrevista del desarrollo de la aplicación con el responsable de la Comisión de reactivos, a la cual en adelante se la denominará la comisión, se presentó un prototipo de diseño con las pantallas de ingreso, revisión y parametrización. Posterior a esto se procedió con la construcción de la aplicación y a agendar una nueva revisión de las pantallas definitivas en la web.
2. En la segunda entrevista se presentó la elaboración de reactivos y sus respectivas opciones: en dicha entrevista se acordó que primero se ingrese el reactivo en general y posteriormente se ingrese una respuesta a la vez ya que era la mejor forma dinámica de ingresar de 4 a 7 opciones de respuestas de acuerdo a la parametrización de reactivos realizada en la aplicación web; se levantó la observación con el cambio de que se deba manejar un número fijo de respuestas para cada reactivo y poder dar mayor facilidad al subir las respuestas al sistema.
3. En la tercera entrevista se verificaron los cambios para su validación y revisión, se presentó el módulo de revisión de reactivos, se aprobó y se solicitó que la revisión y envío de reactivos sean notificados por mail a los miembros de la comisión o docentes, dependiendo del caso.

Capítulo 1

1.1 Alcance

El proyecto se ha planteado con la finalidad de desarrollar una aplicación web, para gestionar los procesos de elaboración de reactivos, su revisión y uso para sus diferentes eventos de evaluación. La solución planteada cuenta con los siguientes módulos:

Administración.

Permite controlar el acceso y uso de la aplicación web, estableciendo perfiles de acuerdo al cargo del usuario que ingresa al sistema dependiendo de si es docente, integrante de la comisión o estudiante.

Elaboración de reactivos.

Permite la elaboración de reactivos por parte de los docentes para que puedan ser utilizados en sus materias o puedan ser enviados a la comisión para su revisión.

Revisión de reactivos.

Permite la revisión de los reactivos elaborados anteriormente por los docentes, de esta manera se cuenta con una base de datos que almacena reactivos para los diferentes eventos de evaluaciones.

Uso de reactivos para diferentes tipos de evaluaciones.

Permite la generación de exámenes en base al banco de reactivos vigentes para su uso en evaluaciones semestrales por parte de los docentes y exámenes complexivos por parte de la comisión.

Generación de reportes.

Permite la generación de reportes del proceso de cumplimiento de la carga de los reactivos por parte de los docentes, la generación de un inventario de reactivos vigentes y dados de baja y un reporte del número total de reactivos por su estado actual.

1.2 Necesidades

La carrera, a través de su consejo, identificó la necesidad de contar con una aplicación web que facilite la elaboración de reactivos para que los docentes puedan crearlos y luego usarlos en sus pruebas, y así ser utilizados en los diferentes exámenes que la carrera de Ingeniería de Sistemas ejecuta.

El responsable de la comisión indicó que era necesario que la aplicación web desarrollada contara con un módulo que permitiera una revisión de reactivos eficiente y eficaz de los reactivos que cada docente envía y a su vez mantener un control sobre el estado de los mismos cuando han sido enviados a revisión para su utilización en diferentes exámenes de la carrera.

1.3 Proceso actual de manejo de reactivos

Una vez realizadas las especificaciones de las necesidades de la carrera de Ingeniería de Sistemas y de la comisión, se detallan los procesos manuales que se llevaban a cabo antes del desarrollo de la aplicación web como se indica en la figura No. 4.

En dichos procesos las tareas, tanto de los docentes, como de la comisión y estudiantes se realizaba de manera manual; es decir, el docente elaboraba reactivos en hojas de papel y los enviaba a la comisión para que aprueben o nieguen los mismos, caso contrario los utilizaba para generar evaluaciones de sus propias materias. La comisión aprobaba los reactivos recibidos o a su vez los rechazaba y se devolvían al docente. Las hojas de papel eran entregadas a los estudiantes quienes rendían las diferentes evaluaciones de esta manera.

Procesos manuales

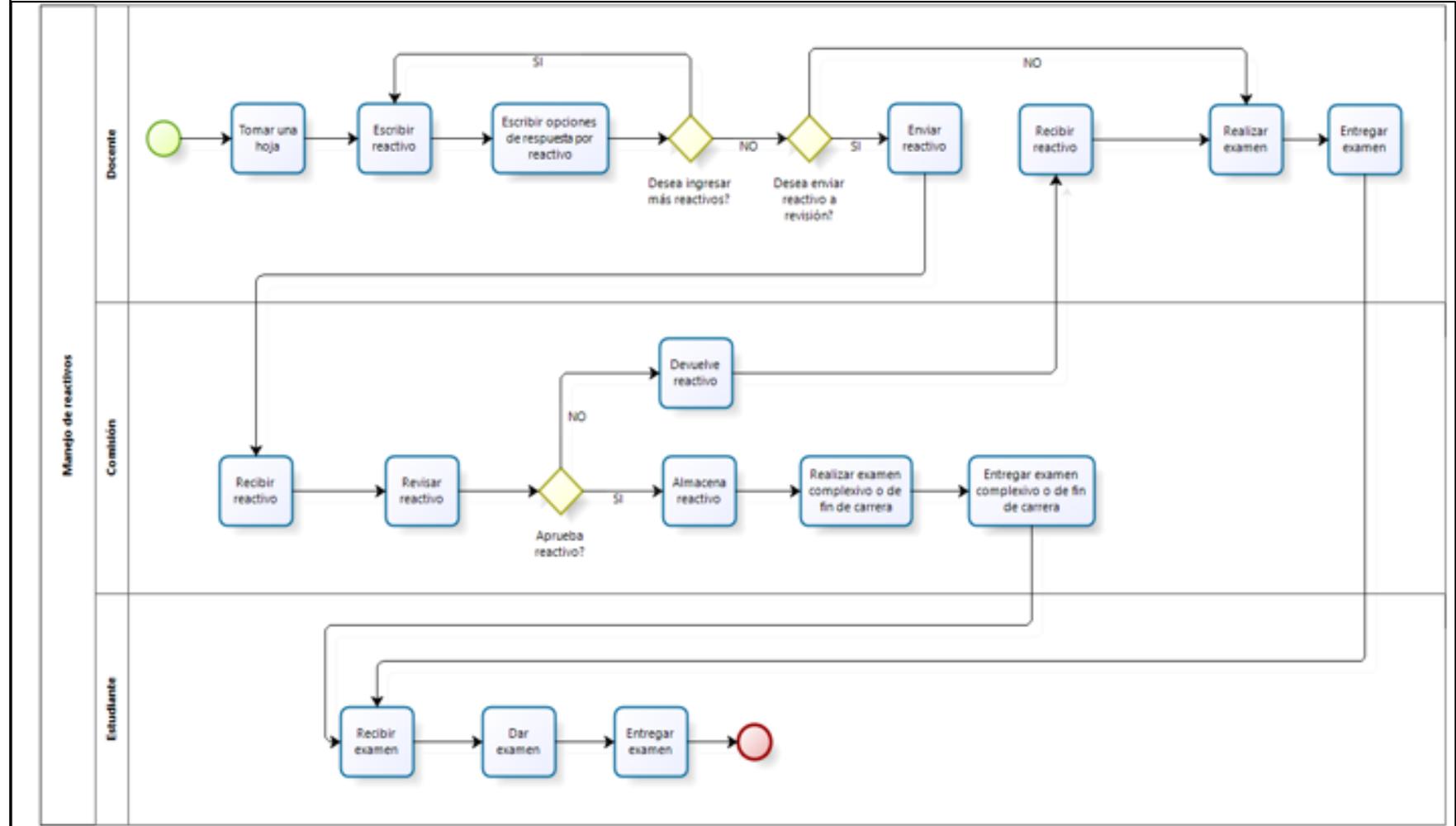


Figura 4. Procesos manuales

Elaborado por: Gabriela Tipán y Giovanni Yancha

1.4 Requerimientos

1.4.1 Visión del software.

De acuerdo a la identificación de los intereses del Responsable de la comisión se definieron los siguientes requerimientos:

La necesidad de realizar el módulo de elaboración de reactivos con el fin de facilitar a los docentes de la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana la elaboración de reactivos de una manera efectiva para su utilización y la entrega de los mismos para su uso por parte de la comisión.

La creación del módulo de Administración de Reactivos en el cual los integrantes de la comisión realizan la revisión de los mismos para su utilización en la creación de diferentes tipos de exámenes complexivos o de fin de carrera; el módulo de administración de reactivos creado brinda a los usuarios integrantes de la comisión una interfaz que permite analizar los reactivos que los docentes envían a revisión y de acuerdo al análisis realizado la aplicación web permite que la comisión apruebe o niegue los mismos; cabe indicar que la comisión puede realizar las siguientes acciones con respecto a un reactivo:

Aprobar un reactivo para su utilización

Rechazar un reactivo

Modificar un reactivo y aprobarlo para su utilización

Negar un reactivo y enviarlo al docente para su corrección

El siguiente módulo es la utilización de los reactivos creados que tiene dos tipos de utilidades:

Los docentes realizan la parametrización de evaluaciones que desean tomar a sus estudiantes, la interfaz permite generar dicha evaluación con las características definidas por el docente en la parametrización como número de preguntas a

generarse, descripción, período actual, fecha de inicio y fin para rendir la evaluación, tiempo de duración y calificación.

Los integrantes de la comisión utilizan los mismos para la creación de exámenes complexivos o de fin de carrera, para que posteriormente los estudiantes rindan dichos exámenes. Se encuentran definidos dos tipos de exámenes, manuales y automáticos. Los exámenes manuales se generan seleccionando los reactivos de acuerdo a las materias previamente escogidas, mientras que los exámenes automáticos se generan con reactivos escogidos aleatoriamente y que no hayan sido utilizados.

Una vez creada la evaluación o exámenes complexivos o de fin de carrera por parte del docente o por parte de la comisión se indicará a los estudiantes para que ingresen a la aplicación web y rindan el examen.

Con el fin de facilitar la correcta utilización de la aplicación web se definió crear un perfil de administrador para que se centralice la administración de usuarios, materias y reactivos durante cada semestre de la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana campos Sur.

Se definieron cinco secciones para las funciones del administrador de la aplicación web:

1. Administración Docente: Permite añadir o editar a los docentes y activar o desactivar a los mismos para su acceso al sistema.
2. Administración de Materias: Permite el ingreso del nivel, materia, el docente asignado a dicha materia, categoría a la que pertenece la materia, carrera a la que pertenece la materia.
3. Parametrización general de reactivos: Permite a la comisión parametrizar las características generales que tendrán los reactivos como período, descripción del

período, bimestre, porcentaje de similitud entre los reactivos creados, número de respuestas por reactivo, y los campos opcionales como objetivo o logro de aprendizaje, unidad o tema, grado de dificultad del reactivo y argumento.

4. Administración de alumnos: Permite añadir o editar a los alumnos y activar o desactivar a los mismos para su acceso al sistema.

5. Generación de reportes: Permite generar reportes de reactivos vigentes y dados de baja, de elaboración de reactivos por parte de los docentes y del número total de reactivos por su estado actual.

1.4.2 Especificación de requerimientos.

A continuación, se describen los requerimientos funcionales de la aplicación web para la elaboración, revisión y administración de reactivos utilizando un esquema de tablas en las cuales se ingresan los siguientes puntos:

Número de requerimiento: Se ingresa la letra MR seguida de tres números que ascenderán secuencialmente.

Nombre de requerimiento: La funcionalidad que tendrá el requerimiento

Descripción: Se describe brevemente la funcionalidad que tendrá el requerimiento

Tabla 1.
MR001 Ingresar al sistema

Número	MR001
Nombre	Ingresar al sistema
Descripción	En la opción de administración de la aplicación web se debe definir el acceso a docentes que pueden elaborar y/o revisar reactivos y estudiantes que pueden rendir evaluaciones.

Nota: Esta tabla contiene especificación de requerimientos para el ingreso al sistema

Tabla 2.
MR002 Permitir la parametrización de reactivos

Número	MR002
---------------	-------

Nombre	Permitir la parametrización de reactivos
Descripción	La aplicación web debe permitir que se realice la parametrización bimestral del período del semestre, el porcentaje de similitud de los reactivos ingresados y el número de respuestas que tendrá cada reactivo.

Nota: Esta tabla contiene especificación de requerimientos para la parametrización de reactivos

Tabla 3.

MR003 Permitir la parametrización de docentes y materias

Número	MR003
Nombre	Permitir la parametrización de docentes y materias
Descripción	La aplicación web debe permitir al administrador asignar un docente a cada materia de la carrera. Existen dos formas de asignar docentes a las materias: Manual: El usuario escoge al docente que asignará a una materia. Automático: Se realiza la carga de un archivo que contiene la relación de materias con docentes.

Nota: Esta tabla contiene especificación de requerimientos de parametrización de docentes y materias

Tabla 4.

MR004 Elaborar reactivos

Número	MR004
Nombre	Elaborar reactivos
Descripción	La aplicación web debe permitir al docente realizar la elaboración de reactivos ingresando para cada uno las siguientes características definidas por la comisión: Opcionales: ✓ Ingresar unidad o tema del reactivo que se está elaborando

	<ul style="list-style-type: none"> ✓ Ingresar el argumento para la creación del reactivo ✓ Ingresar la dificultad del reactivo creado ✓ Ingresar el objetivo o logro de aprendizaje del reactivo que se está creando <p>Obligatorias:</p> <ul style="list-style-type: none"> ✓ Ingresar reactivo ✓ Ingresar bibliografía ✓ Añadir de ser necesario una imagen como guía para el reactivo que se está elaborando ✓ Guardar el reactivo creado ✓ Añadir respuestas para el reactivo creado ✓ Ingresar el texto o justificación de cada respuesta agregada ✓ Seleccionar la respuesta correcta ✓ Guardar respuestas ✓ Salir
--	--

Nota: Esta tabla contiene especificación de requerimientos para elaborar reactivos

Tabla 5.
MR005 Revisar reactivos

Número	MR005
Nombre	Revisar reactivos
Descripción	La comisión debe validar cada reactivo enviado por los docentes para aprobarlos, modificarlos o rechazarlos.

Nota: Esta tabla contiene especificación de requerimientos para la revisión de reactivos

Tabla 6.
MR006 Generación de evaluaciones

Número	MR006
---------------	-------

Nombre	Generar evaluaciones
Descripción	Los docentes generan reactivos que serán utilizados para las evaluaciones en sus materias.

Nota: Esta tabla contiene especificación de requerimientos para generación de evaluaciones

Tabla 7.

MR007 Generación de exámenes complexivos o de fin de carrera

Número	MR007
Nombre	Generar exámenes complexivos o de fin de carrera
Descripción	<p>La comisión genera dos tipos de exámenes:</p> <p>Examen automático: La comisión escoge los ejes, materias, número de reactivos y porcentaje de reactivos para cada eje con lo cual se genera de manera aleatoria el examen.</p> <p>Examen manual: La comisión escoge las materias, número de reactivos y reactivos con lo cual se genera el examen.</p>

Nota: Esta tabla contiene especificación de requerimientos para la generación de exámenes

Tabla 8.

MR008 Rendir evaluaciones o exámenes complexivos o de fin de carrera

Número	MR008
Nombre	Rendir examen
Descripción	Los estudiantes ingresan a la aplicación para rendir las evaluaciones creadas por los docentes o los exámenes creados por la comisión.

Nota: Esta tabla contiene especificación de requerimientos para rendir evaluaciones o exámenes

Tabla 9.

MR009 Generar reportes

Número	MR009
Nombre	Generar reportes o inventarios
Descripción	El administrador de la aplicación genera reportes de reactivos

	vigentes y dados de baja, de elaboración de reactivos por parte de los docentes y del número total de reactivos por su estado actual.
--	---

Nota: Esta tabla contiene especificación de requerimientos para generar reportes o inventarios

Capítulo 2

Marco teórico

Es de gran importancia entender las herramientas que han sido utilizadas para el desarrollo del proyecto, con el fin de dar al lector una visión amplia y clara de la aplicación desarrollada; para lo cual se presentan conceptos que permitirán entender el ciclo de vida del proyecto desde el análisis hasta su construcción e implementación.

2.1. Proceso de software

La aplicación de la ingeniería de software da como resultado un producto de software que comprende el software desarrollado con la documentación registrada de dicho desarrollo.

Un modelo de proceso es utilizado como guía para la creación de un producto de software; cabe recalcar que el modelo de proceso puede cambiar de acuerdo a cada empresa. (Mendez, 2009)

Para lograr el producto de software, se debe seguir una serie o conjunto de pasos o procedimientos llamados procesos de software, con los cuales se obtendrá un producto de alta calidad. Un proceso de desarrollo de software es el flujo o ciclo de vida del mismo, abarcando las siguientes etapas:

Análisis: Es definir los requerimientos del programa que se va a desarrollar, esta etapa es muy importante ya que permite entender claramente las necesidades del negocio y da un enfoque claro de las características que debe tener el producto que se elabore. En esta etapa se debe identificar el problema para proponer las soluciones.

Diseño: Se plasman las características ya identificadas del producto que se está desarrollando, determina el funcionamiento de la aplicación. En esta etapa se diseñan bosquejos de lo que será el producto desarrollado.

Construcción: Es trasladar a codificación las características ya establecidas, tomando como base los diseños creados en la etapa anterior; para lo cual se debe definir un lenguaje de programación y un gestor de base de datos y demás herramientas necesarias.

“En esta fase se desarrollará la parte final de la aplicación, haciendo uso de templates para el correcto desarrollo de las interfaces. Luego de esto se deberá presentar al cliente

el producto final, de modo que pueda dar nuevas sugerencias del software y adaptarlas

posteriormente.” (Tixe & Vivero, 2012)

Pruebas: Esta etapa permite descubrir errores que se hayan cometido en alguna otra etapa de elaboración del producto de software. Las pruebas se deben ejecutar mientras se está desarrollando el producto con el objetivo de realizar correcciones a tiempo ya que si no se hace podría ocasionar pérdidas en tiempo, costos, calidad, entre otros.

Liberación o estabilización: Es la instalación del software desarrollado en las máquinas de los usuarios.

Mantenimiento: Llamado también como etapa de post producción en la cual se corrigen problemas identificados una vez que se implementa el producto de software.

2.2. Calidad de software

“La creciente preocupación por la calidad en la industria del software tiene como objetivo principal el desarrollo sistemático de productos y servicios de mejor calidad y el cumplimiento de las necesidades y expectativas de los clientes” (Echeverry, Cabrera, & Ayala, 2008)

La ingeniería de software abarca métodos, procesos y procedimientos para la construcción y mantenimiento de sistemas de software. Comprende todos los aspectos para la creación de software desde la obtención de requerimientos hasta su mantenimiento, una vez puesto en un ambiente de producción; con lo cual se consigue obtener un software de alta calidad con características como usabilidad ya que debe ser un software que facilite su uso al usuario, por lo cual debe contar con una interfaz apropiada y además con la documentación que guíe al usuario y mantenimiento porque debe ser un software que se pueda modificar de acuerdo a las necesidades del cliente.

Se puede evidenciar el aumento en la exigencia de calidad que los clientes de productos de software están solicitando actualmente; para cumplir con las expectativas de los clientes las empresas desarrolladoras de software deben reafirmar sus bases desde el inicio de un proyecto y mantenerlas mientras dure su desarrollo. (Echeverry, Cabrera, & Ayala, 2008)

2.3 Aplicaciones web

Una aplicación web es un producto de software que contiene herramientas que los usuarios pueden utilizar mediante un navegador a través de un servidor web. Contiene elementos que permiten una comunicación activa entre los usuarios y la información, esto permite que el usuario acceda a los datos de modo interactivo.

Una de las principales características de las aplicaciones web es la portabilidad ya que pueden ser ejecutadas desde muchas plataformas debido a su gran compatibilidad.

La aplicación web emplea una arquitectura cliente servidor, esta comunicación es posible mediante el protocolo Hypertext Transfer Protocol (HTTP), este método permite el intercambio de información entre el cliente y el servidor lo cual permite

que muchos clientes se encuentren conectados al servidor utilizando las funciones de la aplicación web ya que trabajan de manera paralela en la misma.

La Web 2.0 se refiere a la interacción entre los usuarios con diferentes páginas que se encuentran en la web; al contrario, la web 1.0 que contaba únicamente con páginas web que mostraban información estática sin posibilidad que los usuarios aporten con sus comentarios o conocimientos ya que eran producidos y modificados por una sola persona.

La Web 2.0 comprende aplicaciones y diferentes páginas en la red que brindan servicios interactivos que permiten al usuario tener el control de su propia información; ha sido un progreso que ha ido desde aplicaciones tradicionales a aplicaciones enfocadas en el usuario final.

La Web 2.0 permite que los usuarios puedan acceder de una manera fácil y gratuita, facilita el uso de diferentes páginas con la posibilidad de que cada usuario tenga participación con el contenido y comentarios del mismo. (Palchevich, 2008)

2.4 Herramientas de programación

Para el desarrollo de la aplicación web se utilizaron lenguajes y herramientas de programación descritos a continuación.

2.4.1 Java Server Faces.

Java Server Faces (JSF) es un framework que permite crear aplicaciones basadas en un modelo vista controlador, permite la creación de interfaces de usuario a través de vistas que reúnen componentes gráficos. (Dept. Ciencia de la Computación e IA, 2014)

Java Server Faces permite simplificar la integración en desarrollo de interfaces del cliente basadas en la web.

JSF envía muchas peticiones al servidor, para lo cual para optimizar incorpora etiquetas AJAX, lo que permite actualizar sus componentes selectivamente otorgando dinamismo a las páginas.

2.4.2 Primefaces.

Es una librería de componentes para JSF para facilitar la creación de aplicaciones web amigables con el usuario final, siendo una librería de código abierto (OPEN SOURCE) para JSF posee una gran cantidad de componente que permite un manejo dinámico en las páginas web.

El manejo de componentes Primefaces dentro de la web es muy ligero, lo cual es una ventaja y permite manejar un buen rendimiento de la aplicación.

2.4.3 Hibernate.

Es una herramienta que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación para el mapeo relacional de objetos y una solución de administración de persistencia o capa de persistencia para la plataforma Java.

Hibernate maneja las consultas de la información mediante su propio lenguaje HQL(Hibernate Query Language) potente siendo orientado a objetos lo cual comprende herencia, polimorfismo y asociación. (González, 2003)

Persistencia es una tarea que permite serializar objetos java estructurados en forma de árbol a una base de datos de forma tabular y viceversa, surge de la necesidad de mapear la columnas y registros de la base de datos para poder optimizar en velocidad y eficiencia.

2.4.4 JBoss.

Es un servidor de aplicaciones web utilizado en Java, es decir, es la plataforma para el desarrollo e implementación de software. Permite que la aplicación esté disponible

y brinda servicios que permiten su mantenimiento y el acceso a la información sin licencia; es decir, tiene una licencia de código abierto(GPL/LGPL), orientado a arquitectura de servicios, permite el manejo de servicios de persistencia de objeto/relaciones y consulta para java. (Nieto, 2011)

2.4.5 Javascript.

Es un lenguaje de programación orientado a objetos que permite al desarrollador presentar una aplicación web dinámica, donde los cambios o modificaciones son actualizadas inmediatamente. Una ventaja de JavaScript es que no es necesario compilar el código desarrollado para que la aplicación web funcione, sino que se ejecuta directamente desde un navegador. (Méndez & Vizúete, 2010)

2.5 Metodologías de desarrollo de software

Existen varios marcos de trabajo que pueden ser utilizados para el desarrollo de software que contiene herramientas, procesos y procedimientos para brindar un soporte y una guía durante el desarrollo de un producto tecnológico. Una metodología de desarrollo de software permite administrar y desarrollar un proyecto de manera clara y ordenada, organizando el trabajo por etapas y definiendo acciones en cada una de ellas y funciones para cada integrante. Para que una metodología sea ágil debe permitir realizar entregas que serán incrementales y que se adapten a modificaciones solicitadas por el cliente. (Letelier, s.f.)

Cabe recalcar que no todas las metodologías son aplicables a todos los proyectos, por lo cual se debe tener claro cuál de ellas es la que mejor se adapta en cada caso. Existe una variedad de metodologías para desarrollar software entre las cuales se encuentran RUP, XP, Crystal Methodologies y SCRUM.

2.5.1 Scrum.

“Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.” (Schwaber & Sutherland, 2013)

Scrum es una metodología ágil de desarrollo de software con la cual se tiene un conjunto de pasos para obtener un producto tecnológico de alta calidad de forma iterativa e incremental; Scrum permite trabajar tanto en el desarrollo como en el mantenimiento de software. Define un marco para la administración de proyectos, es utilizada cuando los requerimientos pueden ir siendo modificados mientras se desarrolla el software.

Scrum utiliza iteraciones llamadas sprints en las cuales se desarrolla un entregable al cliente, cada sprint tiene una duración de hasta 4 semanas en las cuales se desarrolla el ejecutable a ser entregado y se realizan reuniones diarias entre los desarrolladores para evaluar las entregas realizadas y los puntos faltantes.

En el desarrollo de la aplicación web de gestión de reactivos se utilizó la metodología Scrum ya que era necesario la interacción constante entre los desarrolladores con el cliente y las reuniones diarias entre los desarrolladores con el objetivo de realizar entregables de alta calidad en cada sprint.

Scrum tiene las siguientes características:

Flexibilidad a cambios que se puedan dar al presentar los entregables al cliente.

En cada entregable el cliente puede utilizar los módulos desarrollados de manera completa ya que las entregas son completamente funcionales, lo cual implica que se obtendrá un producto de alta calidad.

La comunicación diaria entre el equipo de trabajo permite obtener mayor productividad en los mismos; dentro de la comunicación entre el equipo de trabajo se deben tomar en cuenta los siguientes puntos:

Se debe realizar una reunión de planificación de cada sprint

Debe realizarse siempre una reunión diaria de aproximadamente 15 minutos

Debe hacerse una revisión del sprint realizado en conjunto con el cliente y los invitados que el dueño del producto defina

Debe realizarse una retrospectiva del sprint con el fin de evaluar el desempeño del equipo de trabajo

El equipo de trabajo es multifunción y está conformado por dos desarrolladores que a la vez se dividen en dueño del producto y Scrum Master. (Schwaber & Sutherland, 2013)

El dueño del producto gestiona el Product Backlog o lista del producto la cual contiene una lista de las tareas que debe ejecutar cada miembro del equipo.

El Scrum Master se encarga de controlar que el equipo de desarrollo adopte como guía a la metodología Scrum verificando que se cumplan los pasos y tareas que la misma contiene.

Product Backlog es un detalle de los requerimientos que se tienen para todo el desarrollo del proyecto. (Gallego, s.f.)

El Sprint Backlog es una lista que contiene todas las tareas pendientes de cada Sprint.

2.6 Teoría de reactivos

Para el correcto desarrollo de una aplicación web enfocada en gestionar reactivos, es necesario contar con información acerca de su elaboración; existen diversas recomendaciones que se utilizan para la generación de reactivos que incluyen buenas prácticas para evitar redundancia de información. La gestión de reactivos supone conocer métodos y técnicas utilizados con el objetivo de elaborar reactivos de una manera adecuada. (Velde, 2014)

Es importante mantener una buena redacción ya que el reactivo creado debe ser claro para que el estudiante entienda lo que se le quiere preguntar y a su vez, pueda escoger la respuesta correcta. (Valdivia, s.f)

El objetivo principal de un reactivo es medir el conocimiento que un estudiante tiene sobre cierto tema, por lo cual se deben omitir palabras que puedan causar confusión y verificar que el nivel de elaboración del reactivo vaya acorde al conocimiento del estudiante. (Aparicio, Zurutuza, & Iparraguirre, s.f)

Un reactivo es una pregunta que se debe contestar solicitando que se ingrese una respuesta correcta. Para el desarrollo de la aplicación web de gestión de reactivos se utilizó el formato de reactivos de opción múltiple, es decir, el docente elabora el reactivo y proporciona múltiples opciones para que el estudiante seleccione la respuesta correcta.

Existen diferentes tipos de reactivos que se suelen utilizar en las evaluaciones, entre los cuales se encuentran:

Reactivos de opción múltiple, los mismos que son utilizados por la Universidad Politécnica Salesiana y consiste en escoger la respuesta correcta de entre varias opciones.

Reactivos de verdadero – falso, comprende en indicar si una afirmación es correcta o incorrecta.

Reactivos de ordenamiento, consiste en ordenar de inicio a fin un grupo de respuestas para darle sentido a una oración.

Reactivos de apareamiento, comprende en unir dos respuestas para darle sentido a una oración.

Reactivos de llenado de espacios en blanco, comprende en llenar los espacios en blanco que se encuentran dentro de una oración.

Reactivos de completamiento de texto, consiste en escribir la respuesta de una pregunta definida. (Valdivia, s.f)

Uno de los requerimientos del cliente partió de la necesidad de que los reactivos elaborados por los docentes no se repitan; para lo cual se utilizó un método de comparación semántica que analiza la similitud entre los reactivos que se están elaborando con los reactivos elaborados anteriormente; si se detecta que los reactivos son similares el sistema emite un mensaje indicando que se superó el porcentaje de similitud de reactivos.

Es recomendable pilotear las evaluaciones realizadas, teniendo en cuenta varios aspectos en la elaboración de reactivos:

Grado de dificultad de cada reactivo

Tiempo de duración de cada reactivo de acuerdo al grado de dificultad de los mismos.

Número de reactivos que debe tener una evaluación

Evaluaciones con instrucciones claras para los estudiantes

Si bien es cierto, es imposible adelantarse a todas las situaciones que se puedan presentar, el pilotear las evaluaciones permite ir mejorando cada vez más la calidad de pruebas realizadas.

2.7 BPMN

BPMN es una notación gráfica que permite visualizar el flujo de los procesos de una empresa y su implementación, lo cual facilita a los involucrados entender los pasos de cada proceso. (Bizagi, 2014)

Modelar utilizando BPMN sirve como herramienta para realizar optimización de procesos de una empresa ya que se tiene un enfoque visual claro de la secuencia de las actividades de la organización y la información involucrada en cada paso. (White & Miers, 2009)

BPMN otorga una amplia y detallada explicación gráfica de cada proceso y subprocesos de la empresa, indicando un inicio y un fin en los mismos, con lo cual se puede realizar un buen análisis y diseño de software.

Capítulo 3

Diseño del aplicativo

3.1 Casos de uso

A continuación, se indica el diagrama de casos de uso en la figura No. 5, que muestra los diferentes escenarios en los cuales interactúan los actores intervinientes en la aplicación web desarrollada.

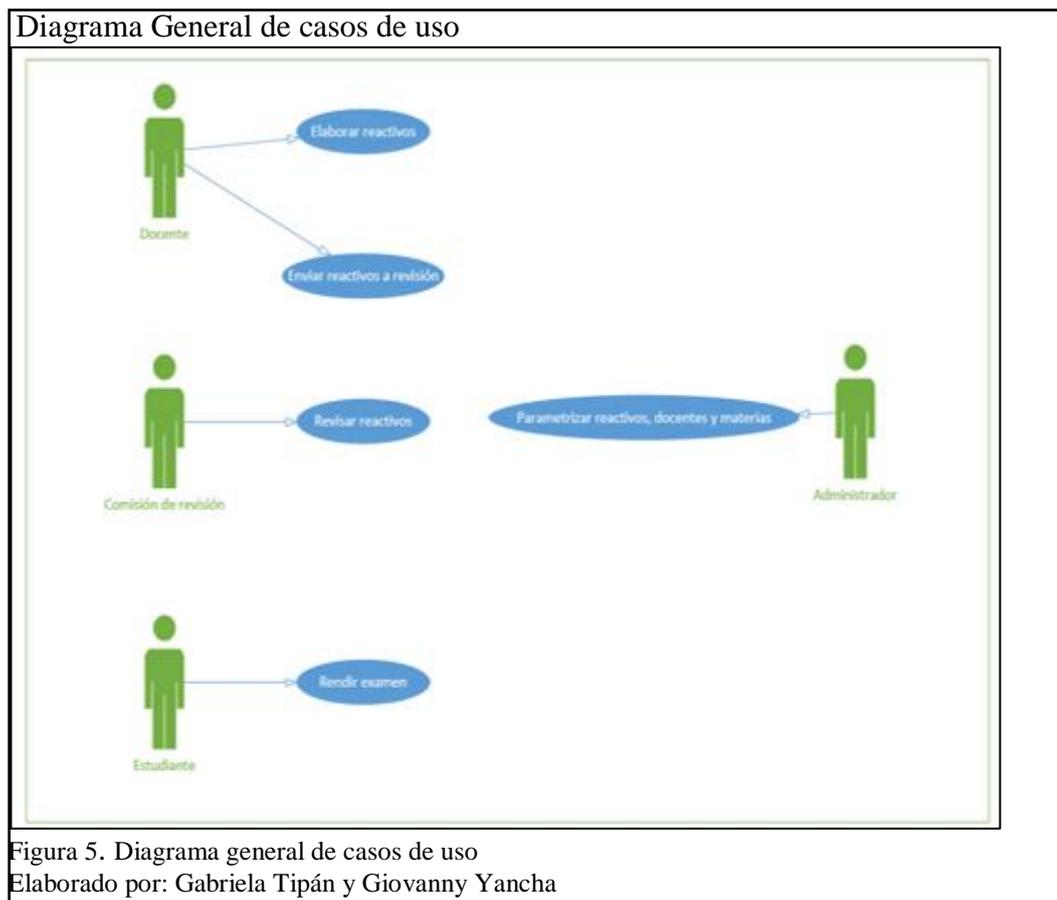


Figura 5. Diagrama general de casos de uso
Elaborado por: Gabriela Tipán y Giovanni Yancha

Se han establecido los siguientes casos de uso de la aplicación web que contienen la secuencia de pasos de la misma con sus respectivos actores.

Administrador de la aplicación: Es la persona que administra la aplicación web con el objetivo de permitir el acceso a personas autorizadas y realizar la parametrización de periodos, docentes, materias y exámenes.

Docente: Es la persona que elabora los exámenes para sus materias y envía reactivos que serán utilizados para diferentes tipos de exámenes.

Comisión de reactivos: Es un grupo seleccionado de docentes que realizan la revisión de los reactivos enviados por los docentes para aprobar, modificar o rechazar los mismos.

Estudiante: Es la persona que ingresa a la aplicación para rendir una evaluación de cierta materia o un examen complejo o de fin de carrera.

Tabla 10.

Caso de uso 1: Inicio de sesión

Caso de uso	Inicio de sesión
Descripción	Se define el acceso que los usuarios tienen al sistema, incluyendo modificaciones como edición, activación o desactivación de usuarios.
Actores	Administrador de la aplicación Usuarios con diferentes roles o permisos
Flujo normal	Agregar a los usuarios que acceden al sistema: <ol style="list-style-type: none"> 1. Ingresar al sistema 2. Ingresar nuevo usuario 3. Ingresar cédula 4. Ingresar nombres 5. Ingresar apellidos 6. Ingresar correo electrónico Modificar usuarios: <ol style="list-style-type: none"> 7. Editar usuario 8. Activar usuario 9. Desactivar usuario
Flujo alternativo	1A. El sistema indica usuario incorrecto

Nota: Caso de uso de inicio de sesión

Tabla 11.
Caso de uso 2: Parametrización de reactivos

Caso de uso	Parametrización de reactivos
Descripción	Parametrizar las características que tienen los reactivos
Actores	Administrador de la aplicación
Flujo normal	<ol style="list-style-type: none"> 1. Ingresar al sistema 2. Ingresar el código del período 3. Ingresar el período 4. Ingresar el bimestre 5. Ingresar el porcentaje de similitud 6. Ingresar el número de respuestas por reactivo 8. Guardar los cambios
Flujo alternativo	1A. El sistema indica opción ingresada incorrecta

Nota: Caso de uso de parametrización de reactivos

Tabla 12.
Caso de uso 3: Administración de docentes por materia

Caso de uso	Administración de docentes por materia
Descripción	Asignar docentes a las diferentes materias de la carrera
Actores	Administrador de la aplicación
Flujo normal	<p>Asignación manual:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema 2. Escoger materia 3. Escoger docente a asignar a la materia 4. Guardar cambios <p>Asignación automática:</p> <ol style="list-style-type: none"> 1. Ingresar al sistema 2. Cargar archivo de relación de materias con docentes

	3. Guardar cambios
Flujo alternativo	1A. El sistema indica usuario inválido

Nota: Caso de uso de administración de docentes por materia

Tabla 13.

Caso de uso 4: Elaboración de reactivos

Caso de uso	Elaboración de reactivos
Descripción	Elaborar reactivos que se utilizarán para que los estudiantes rindan los diferentes tipos de evaluaciones o exámenes complexivos o de fin de carrera.
Actores	Docentes
Flujo normal	<p>1.Ingresar al sistema</p> <p>Opcionales:</p> <p>2.Ingresar unidad o tema del reactivo que se está elaborando</p> <p>3.Ingresar el argumento para la creación del reactivo</p> <p>4.Ingresar la dificultad del reactivo creado</p> <p>5.Ingresar el objetivo o logro de aprendizaje del reactivo que se está creando.</p> <p>Obligatorias:</p> <p>6.Ingresar reactivo</p> <p>7.Ingresar bibliografía</p> <p>8.Añadir de ser necesario una imagen como guía para el reactivo que se está elaborando</p> <p>9.Guardar el reactivo creado</p> <p>10.Añadir respuestas para el reactivo creado</p> <p>11.Ingresar el texto o justificación de cada respuesta agregada</p> <p>12.Seleccionar la respuesta correcta</p>

	13. Guardar cambios
Flujo alternativo	1A. El sistema indica usuario inválido 8A. Desplegará un mensaje indicando los formatos válidos para añadir imagen 12A. El sistema indica que se debe marcar la respuesta correcta

Nota: Caso de uso de elaboración de reactivos

Tabla 14.

Caso de uso 5: Revisión de reactivos

Caso de uso	Revisión de reactivos
Descripción	La comisión debe revisar los reactivos enviados por los docentes para aprobarlos, modificarlos o rechazarlos.
Actores	Comisión de reactivos
Flujo normal	1. Ingresar al sistema 2. Revisar reactivo 3. Se acepta, modifica, devuelve para corrección o rechaza reactivo 4. Guardar cambios
Flujo alternativo	1A. El sistema indica usuario incorrecto

Nota: Caso de uso de revisión de reactivos

Tabla 15.

Caso de uso 6: Generación de evaluaciones

Caso de uso	Generación de evaluaciones
Descripción	Los docentes generan reactivos que serán utilizados para las evaluaciones en sus materias.
Actores	Docentes
Flujo normal	1. Ingresar al sistema

	2. Elaborar reactivo 3. Escoger los reactivos para la generación de la evaluación 4. Guardar cambios
Flujo alternativo	1A. El sistema indica usuario incorrecto

Nota: Caso de uso de generación de evaluaciones

Tabla 16.

Caso de uso 7: Generación de exámenes complexivos o de fin de carrera

Caso de uso	Generación de exámenes complexivos o de fin de carrera
Descripción	La comisión genera exámenes complexivos o de fin de carrera
Actores	Comisión de reactivos
Flujo normal	1. Ingresar al sistema 2. Escoger materias 3. Escoger número de reactivos Examen automático: 4. Generar examen Examen manual: 4. Escoger los reactivos que serán tomados en cuenta para el examen 5. Generar examen
Flujo alternativo	1A. El sistema indica usuario incorrecto

Nota: Caso de uso de generación de exámenes

Tabla 17.

Caso de uso 8: Rendir evaluación

Caso de uso	Rendir evaluación
Descripción	Ingresar a la aplicación para rendir diferentes tipos de exámenes
Actores	Estudiantes

Flujo normal	<ol style="list-style-type: none"> 1. Ingresar al sistema 2. Rendir examen 3. Guardar cambios
Flujo alternativo	1A. El sistema indica usuario incorrecto

Nota: Caso de uso para rendir evaluación

Tabla 18.

Caso de uso 9: Generación de reportes

Caso de uso	Generar reportes
Descripción	Genera reportes de reactivos vigentes y dados de baja, de elaboración de reactivos por parte de los docentes y del número total de reactivos por su estado actual.
Actores	Administrador de la aplicación
Flujo normal	<ol style="list-style-type: none"> 1. Ingresar al sistema <p>Reporte de reactivos vigentes y dados de baja:</p> <ol style="list-style-type: none"> 2. Escoger opción de inventario de reactivos 3. Escoger período 4. Escoger bimestre 5. Escoger nivel 6. Escoger materias 7. Escoger tipo (utilizado, no utilizado, todos) 8. Escoger generar reporte <p>Reporte de elaboración de reactivos por parte de los docentes:</p> <ol style="list-style-type: none"> 2. Escoger opción de Reactivos por Docente 3. Escoger período 4. Escoger bimestre 5. Escoger docente

	6. Escoger Generar reporte Reporte del número total de reactivos por su estado actual: 2. Escoger opción Número de Reactivos por Estado 3. Escoger período 4. Escoger bimestre 5. Escoger docente 6. Escoger estado (Elaborado, pendiente, aprobado, rechazado, corrección por parte de la comisión, corrección por parte del Docente) 7. Escoger Generar reporte 8. Guardar cambios
Flujo alternativo	1A. El sistema indica usuario incorrecto

Nota: Caso de uso para generación de reportes

3.2 Diagrama de la base de datos

Para el diseño del esquema se basó en el modelo de entidad-relación publicado por Peter Chen en 1976 como la mejor práctica, ya que en base a los elementos del modelo como entidades, atributos y relaciones permiten representar totalmente la información de la base de datos. (Kroenke, 2003)

En el diseño de esquema se presentaron los desafíos de las relaciones que debe tener obligatoriamente un reactivo como el docente que lo crea, materia o subcategoría, entre otros; esto conllevó a que se deba estructurar un esquema de relaciones que manejen docentes con sus respectivas materias o subcategorías, grupos y niveles que maneja regularmente la universidad para adaptarse a la estructura de esta.

El diseño del esquema está orientado a considerar que no solo debe soportar los datos de una carrera sino para otras, en base a la teoría de reactivos se consideró que debe

soportar más de un tipo de reactivos que de ser necesario en otras posteriores versiones considerar tales reactivos.

Una buena práctica es manejar nomenclaturas con el objetivo de poder entender y manejar más rápido y de manera efectiva los tipos de elementos que se maneja en la base de datos.

Nomenclatura de base de datos:

1. Las tablas deben nombrarse usando la siguiente nomenclatura. -

<VERBO/ADJETIVO>_<DESCRIPCION>

Ejemplo:

REACTIVOS_PREGUNTA

2. Los campos de una tabla corresponden a los atributos de una entidad, describen propiedades de la misma.

Las columnas deben ser nombradas según los lineamientos a continuación:

<iniciales del nombre de la tabla>_<tipo>_<descripción>.

Ejemplo:

RP_A1_REACTIVO

RP_DA_FECHA_CREACION

3. El campo clave de una tabla de nombrarse con las siguientes consideraciones.

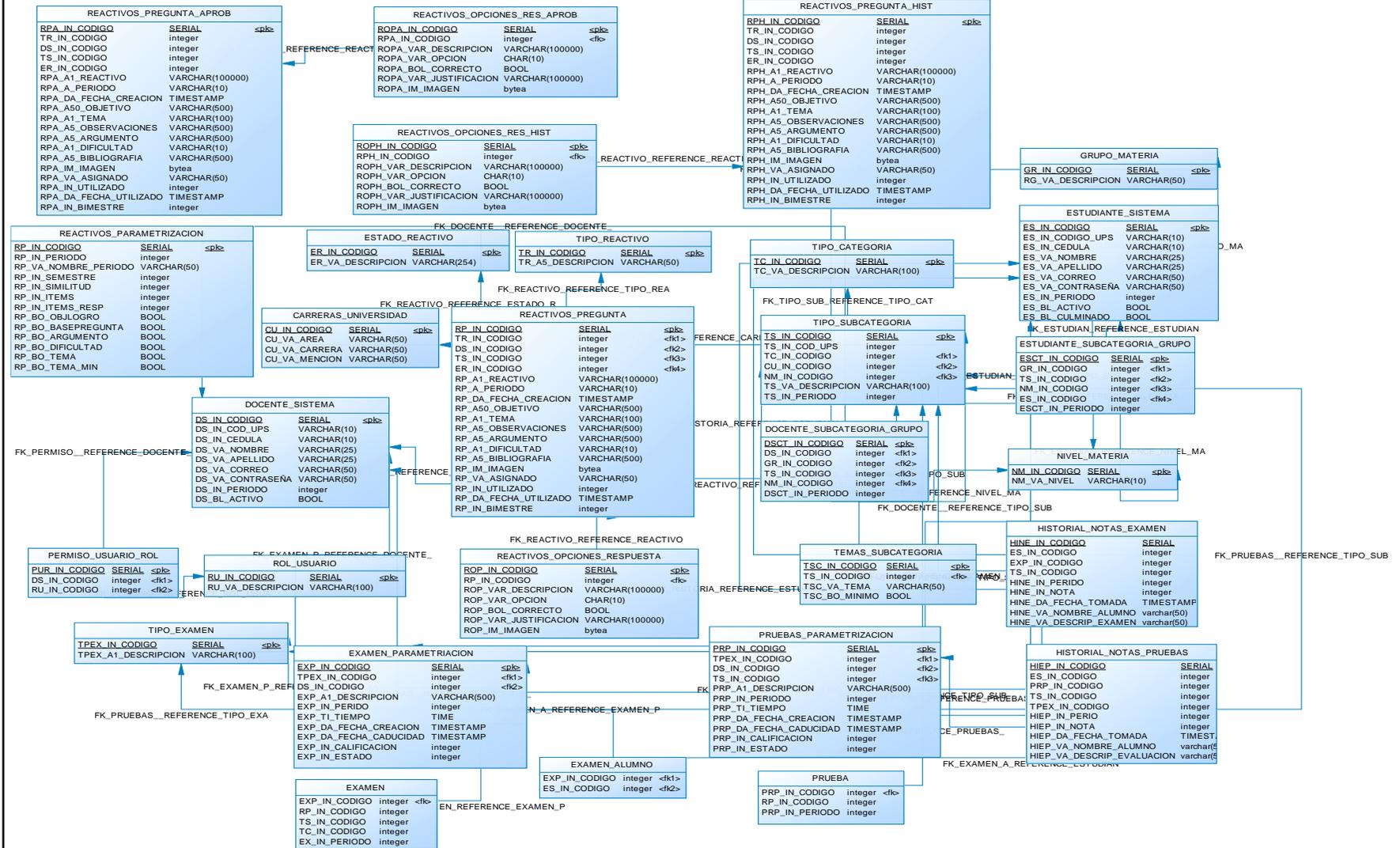
<iniciales del nombre de la tabla>_<IN>_<descripción>

Ejemplo: Para una tabla de REACTIVOS_PREGUNTA se definirían las claves:

RP_IN_CODIGO

A continuación, en la figura No. 6 se muestra el esquema de la base de datos relacional de la aplicación web para la administración de reactivos, con sus respectivas tablas, campos y relaciones.

Diagrama de base de datos



3.3 Diagrama de clases

Un diagrama de clases permite presentar la estructura de la aplicación, en el presente proyecto se estructuró el diagrama en base al modelo MVC que utiliza el enfoque orientado a objetos, por lo cual se tienen tres tipos de clases más relevantes para el sistema.

Interface Denominada como vista o view, es la que permite la interacción con el usuario y la aplicación.

Control. Maneja la interacción entre las bases de datos, aplicación y vista, se ejecutan los métodos con la lógica del sistema.

Model. Clases del mapeo de las tablas correspondiente a la base de datos, estas manejan las persistencias automáticamente.

El diagrama de clases presentado es el general con las clases definidas posterior a su fase de desarrollo, ya que mediante esta etapa surgieron cambios necesarios.

Como se indica en la figura No. 7 se presenta el diagrama de clases que contiene las clases y las relaciones entre las mismas.

3.4 Diseño de interfaces

En esta sección se indica el diseño de las pantallas de la aplicación web con la funcionalidad que tendrá cada una:

1. En la figura No. 8 se muestra el diseño de la parametrización de reactivos en la cual el administrador del sistema puede parametrizar los mismos ingresando el número de respuestas de los exámenes, período, semestre, porcentaje de similitud.

Diseño parametrización de reactivos

PARAMETRIZACIÓN DE REACTIVOS

Período:

Semestre:

Porcentaje de Similitud:

Número de reactivos:

Número de respuestas:

Figura 8. Diseño de la pantalla de parametrización de reactivos
Elaborado por: Gabriela Tipán y Giovanni Yancha

2. El administrador de la aplicación parametriza la información personal de los docentes que acceden al sistema. En la figura No. 9 se muestra cómo se agrega la información de docentes y en la figura No. 10 se modifica la misma.

Diseño ingreso usuario

ADMINISTRACIÓN DE USUARIOS

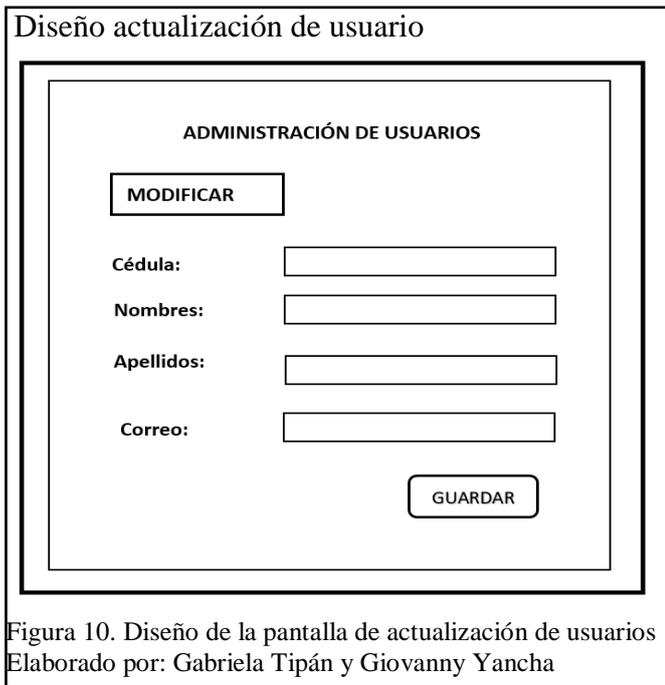
Cédula:

Nombres:

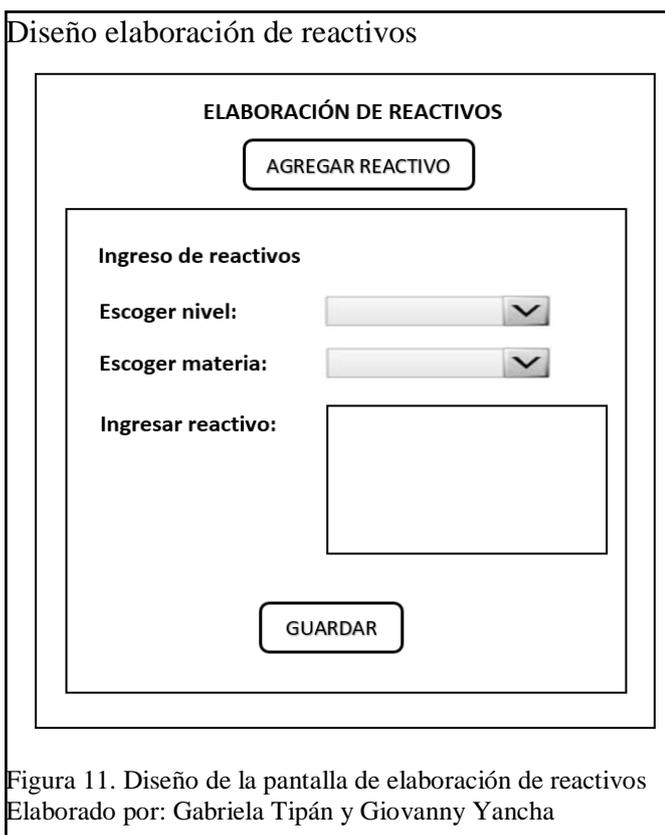
Apellidos:

Correo:

Figura 9. Diseño de la pantalla de ingreso de usuarios
Elaborado por: Gabriela Tipán y Giovanni Yancha



3. En la figura No. 11 se muestra la elaboración de reactivos por parte de los docentes. Despliega la pantalla en la cual se ingresan uno a uno los reactivos elaborados por el docente.



Capítulo 4

Construcción y pruebas

4.1 Estándares

Existen diversos estándares o principios que ayudan a definir las características de estructura del código y tablas de base de datos que se están creando; son reglas de buenas prácticas emitidas como recomendaciones que suelen ser utilizadas a diario en las labores profesionales.

Para el desarrollo de la aplicación web de gestión de reactivos se utilizaron dos estilos: Pascal y Camel para definir el manejo de mayúsculas y minúsculas.

A continuación se detallan ejemplos de clases, variables, métodos, entre otros; con los cuales se aplicaron los estilos antes mencionados.

1. Notación Pascal para nombrar las clases

```
public class ReactivosPregunta { }
```

2. Notación Pascal para nombrar los métodos

```
void GuardarPregunta(string strNombre) { }
```

3. Notación de Camell nombrar variables y parámetros de los métodos

```
int intCodRespImage= 0;

void ConsultarPregunta(int intCodigoReactivo) {

    string _strMensajeReactivo = "Pregunta N° " + intCodigoReactivo; }
```

4. Utiliza el prefijo “_”, para variables locales.

```
string _strDescripcion;

int _intCodigoReactivo;
```

5. Nombres de Clases o Métodos y funciones deben seguir el siguiente estándar:

```
<Acción/Verbo><descripción>

ConsultarReactivo();
```

```
ActualizarReactivo();
```

6. Los comentarios deben estar al mismo nivel que el código:

```
// Instancia la clase del modelo usuario
```

```
private UsuarioSistema usrsistem = new UsuarioSistema();
```

7. No se deben usar guiones bajos (_) para nombres de variables globales.

4.2 Modificaciones

Una vez definidos los estándares se continúa con el proceso de construcción y pruebas en el cual se mantuvieron reuniones con el responsable de la comisión identificando en las mismas ciertos cambios que debían realizarse en cuanto a estructura y funcionalidad de los módulos de la aplicación web.

Se presentó el avance del ingreso de reactivos y se levantaron los siguientes requerimientos:

Inicialmente se contaba con dos pantallas, una para el ingreso de reactivos, y la otra pantalla para el ingreso de ítems por reactivos. El cambio sugerido fue que se realice el ingreso de los reactivos y sus respuestas en la misma pantalla con la posibilidad de agregar un anexo. Se realizó la modificación en la cual en la misma pantalla se dividen dos tabs, uno para el ingreso del reactivo y otro para el ingreso de las respuestas.

Se solicitó que cada campo tenga su validación de campos vacíos.

Surgió la necesidad de que el reactivo se maneje por estados, de manera que se asignaron los siguientes estados:

Elaborado

Pendiente de revisión

Aprobado

Rechazado

Corrección por parte de la comisión

Corrección por parte del docente

Se modificó el esquema inicial de la base de datos

4.3 Diagrama del proceso automatizado

A continuación, se describe el diagrama del proceso automatizado en la aplicación web para el manejo de reactivos, en el cual se presenta el funcionamiento de los procesos de elaboración y revisión de reactivos y el uso de los mismos para diferentes tipos de evaluaciones y exámenes.

Como se muestra en la figura No. 12, el módulo de Administración de la aplicación permite realizar la parametrización de reactivos, materias y docentes y modificación de usuarios que tienen acceso a la aplicación web.

Proceso automatizado módulo de Administración

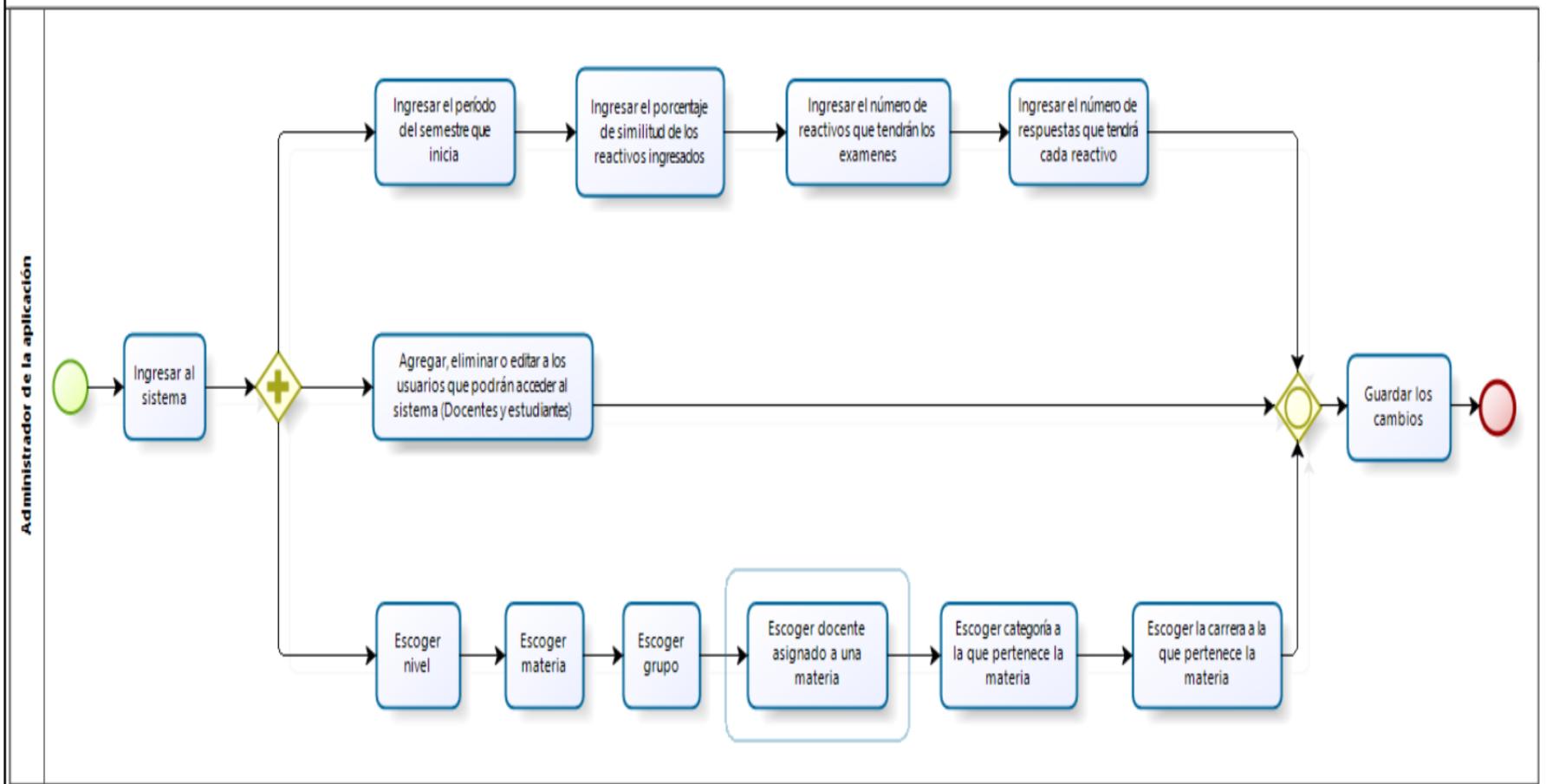


Figura 12. Proceso automatizado del módulo de Administración de la aplicación
Elaborado por: Gabriela Tipán y Giovanny Yancha

En la figura No. 13 se indica como el proceso continúa con la elaboración de reactivos por parte de los docentes, en el mismo se decide si el reactivo es enviado a revisión o si será utilizado por el docente para la elaboración de pruebas y/o exámenes propios de su materia.

En caso de que la decisión del docente sea enviar el reactivo a revisión, dicho reactivo es transferido al módulo de revisión para que la comisión apruebe, modifique o rechace reactivos.

Si el reactivo es aprobado o modificado por la comisión, el mismo podrá ser utilizado en diferentes tipos de exámenes; caso contrario, si el reactivo es negado sin opción de corrección, podrá ser utilizado por el docente en evaluaciones de sus materias.

Proceso automatizado módulo de manejo de reactivos

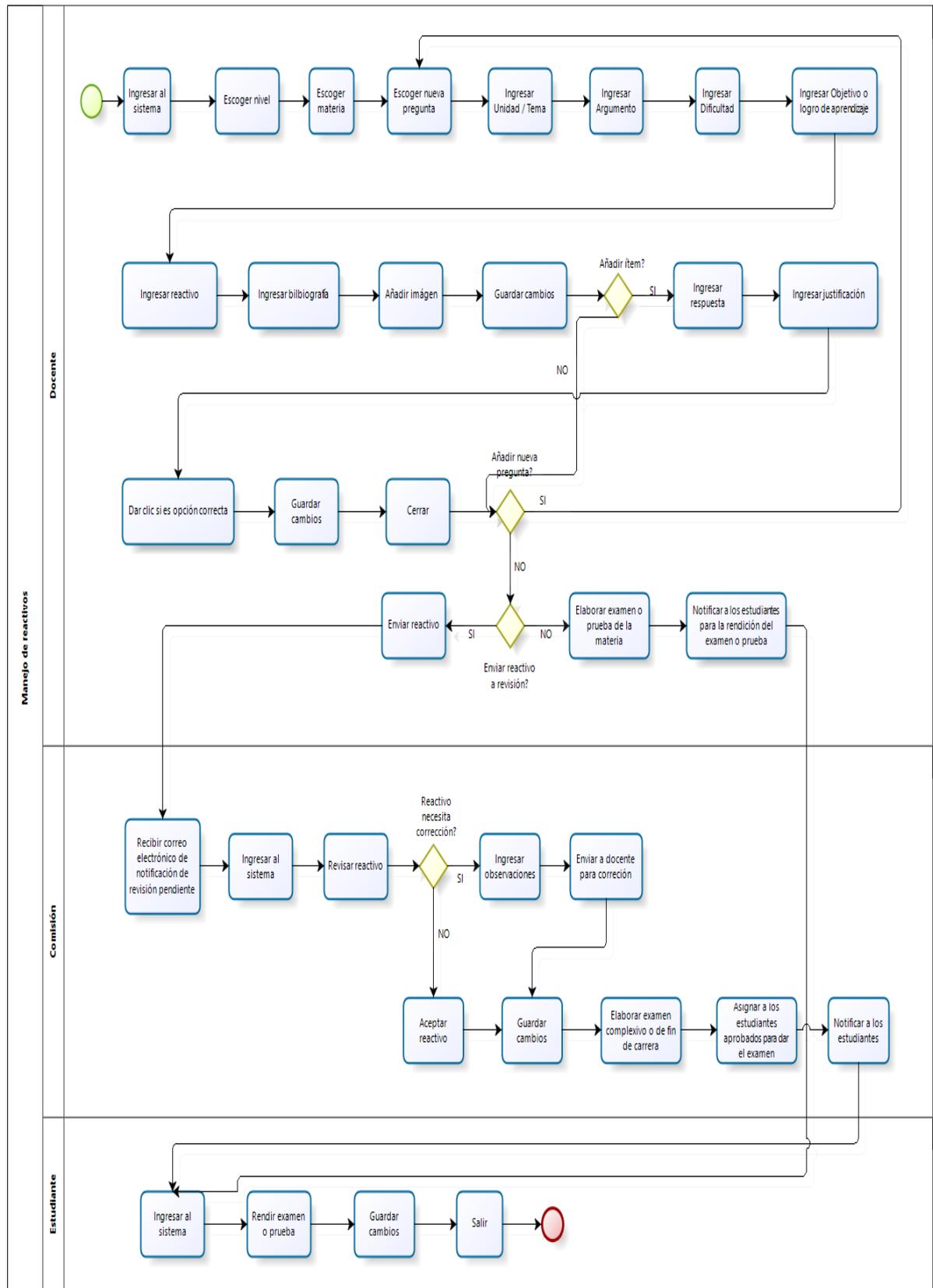
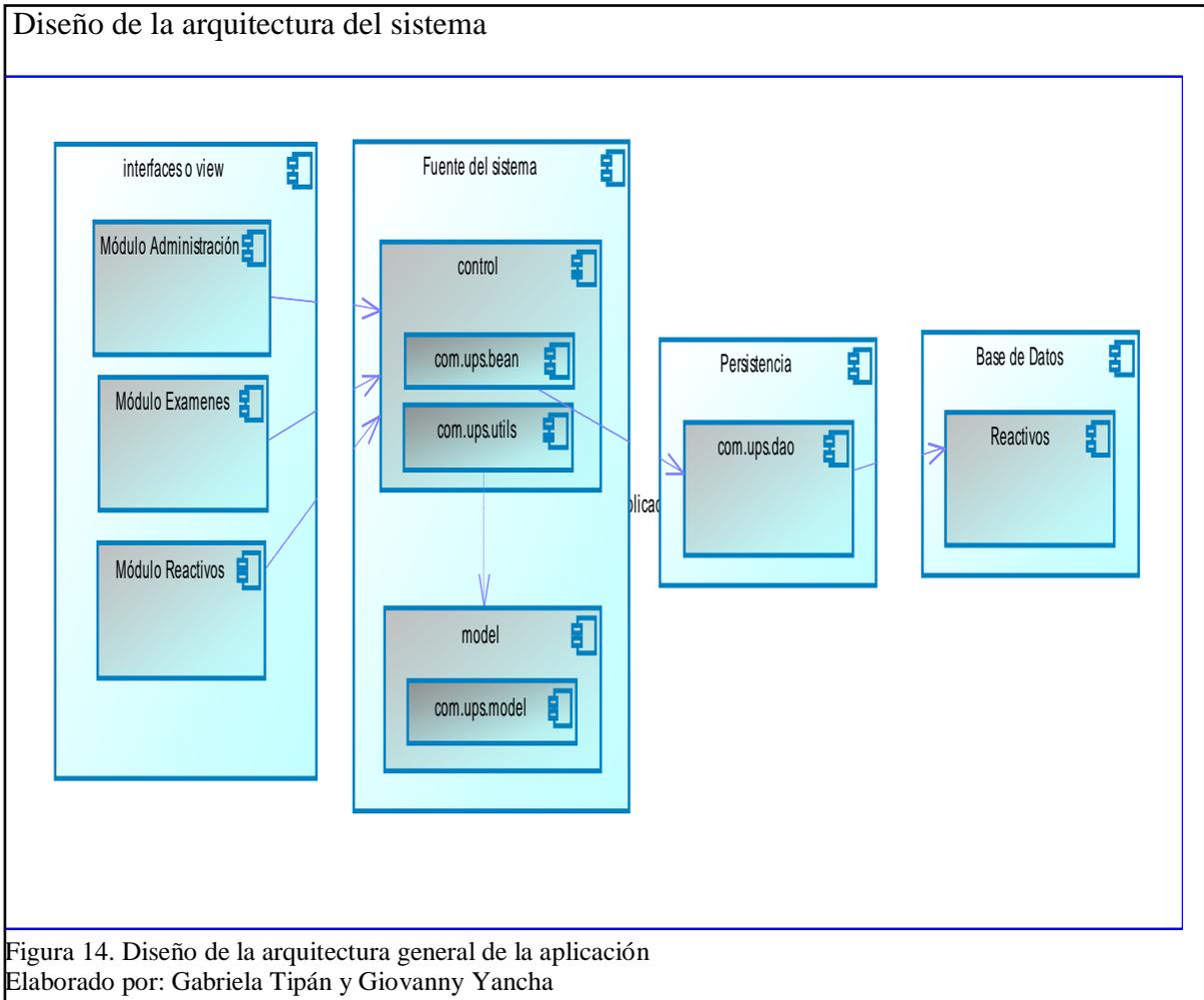


Figura 13. Proceso automatizado roles docente, comisión y estudiante
 Elaborado por: Gabriela Tipán y Giovanni Yancha

4.4 Arquitectura del sistema

Para la arquitectura del sistema se empleó el (MVC) modelo vista controlador lo cual permitió mantener de una forma estructurada la construcción y manejo de la aplicación.



En la figura No. 14 se definió la arquitectura general de la aplicación en base al MVC, para una mejor forma de manejo de cada uno de los elementos del proyecto se manejó un estándar para los paquetes que contienen las clases, pero también define donde las clases se localizan en la estructura jerárquica del directorio.

Paquetes:

Por defecto todos los paquetes se han escrito en minúsculas y sin utilizar caracteres especiales. El paquete base fue definido como `com.ups`, en este paquete no se ha definido ninguna clase.

Se tiene, así mismo, otro nivel extra dentro del paquete definido con el nombre del proyecto o del módulo

- A. com.ups.model (model)
- B. com.ups.bean (control)
- C. com.ups.dao (persistencia)
- D. com.ups.utils (utils)

En la capa Modelo se manejó el modelado de las tablas de base de datos.

En la vista se han utilizado las interfaces o las páginas de la aplicación en este caso las páginas XHTML para JSF.

En los controladores se manejaron los managed Bean y utils que son los que permiten la interacción entre la vista con los modelos y la lógica del negocio.

En la capa Vista, tal como se muestra en la figura No. 15, se han utilizado los ficheros XHTML con etiquetas especiales que definen componentes JSF, dichos componentes se convierten al final en código que se pasa al navegador para que lo muestre al usuario.

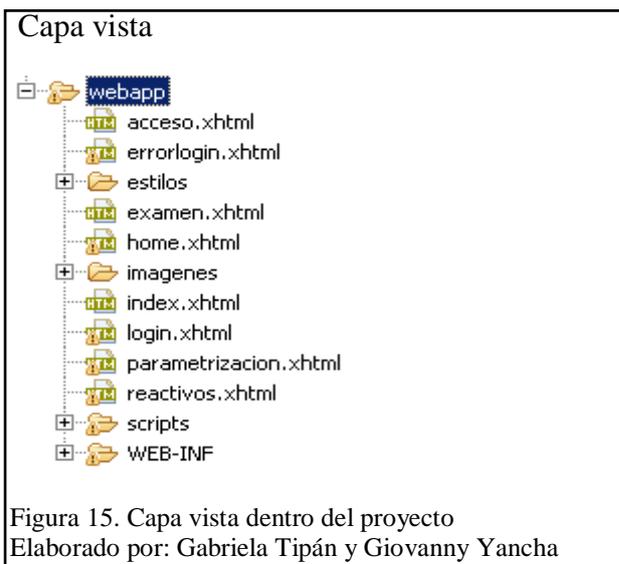


Figura 15. Capa vista dentro del proyecto
Elaborado por: Gabriela Tipán y Giovanni Yancha

Modelo: Permite el mapeo de las tablas de base de datos con la aplicación mediante Hibernate, estos se ubican en el paquete de model, tal como se muestra en la figura No.

16

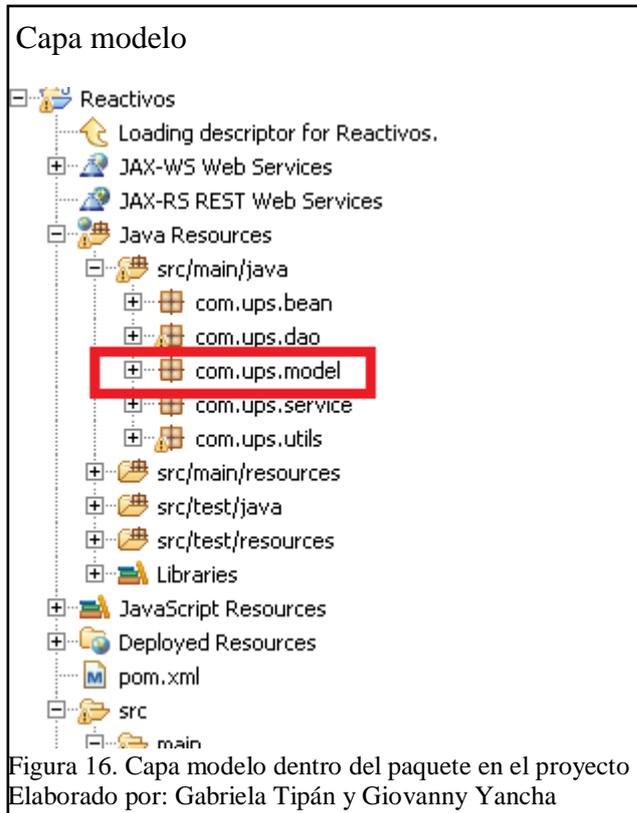


Figura 16. Capa modelo dentro del paquete en el proyecto
Elaborado por: Gabriela Tipán y Giovanni Yancha

Contiene todos los modelos que hacen referencia a todas las tablas que se tiene en BDD, estos están ubicados en el paquete de model como por ejemplo el de docente, como se indica en la figura No. 17

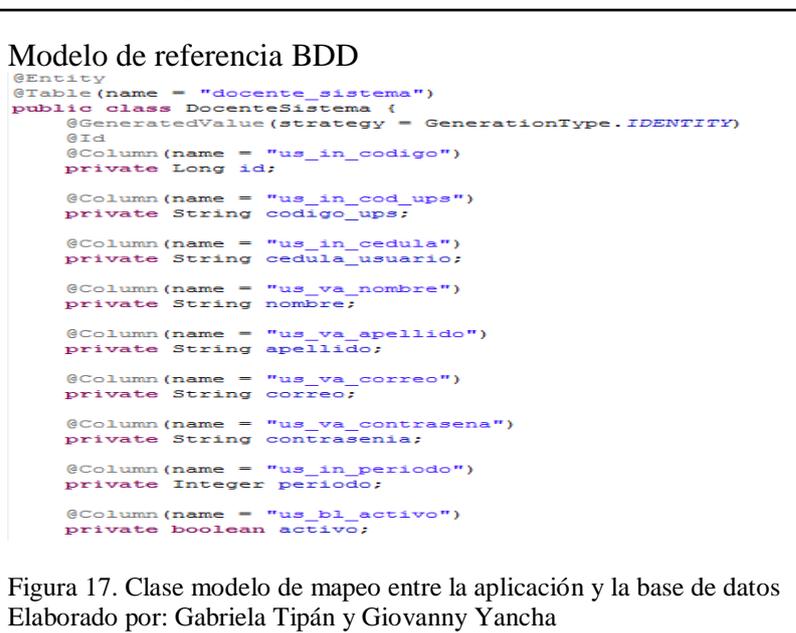


Figura 17. Clase modelo de mapeo entre la aplicación y la base de datos
Elaborado por: Gabriela Tipán y Giovanni Yancha

En el controlador están los beans y utils que gestionan la lógica de la aplicación de los reactivos y se conectan con la capa vista para su manejo, también manejan los métodos que fueron necesarios para el proyecto como por ejemplo el de acceder a la aplicación, método para verificar datos, creación de reportes, comparación de la similitud de reactivos; estos se ubican en el paquete bean como se indica en la figura No. 18

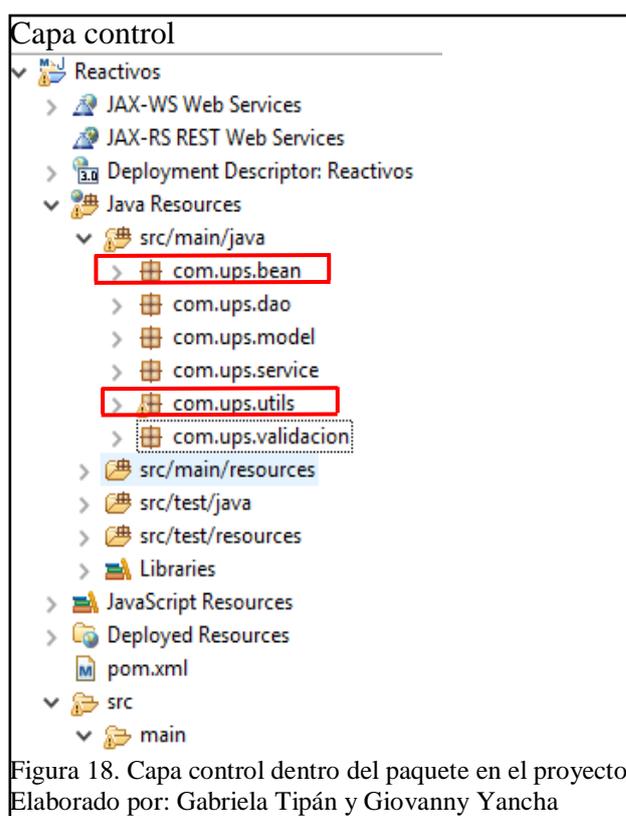
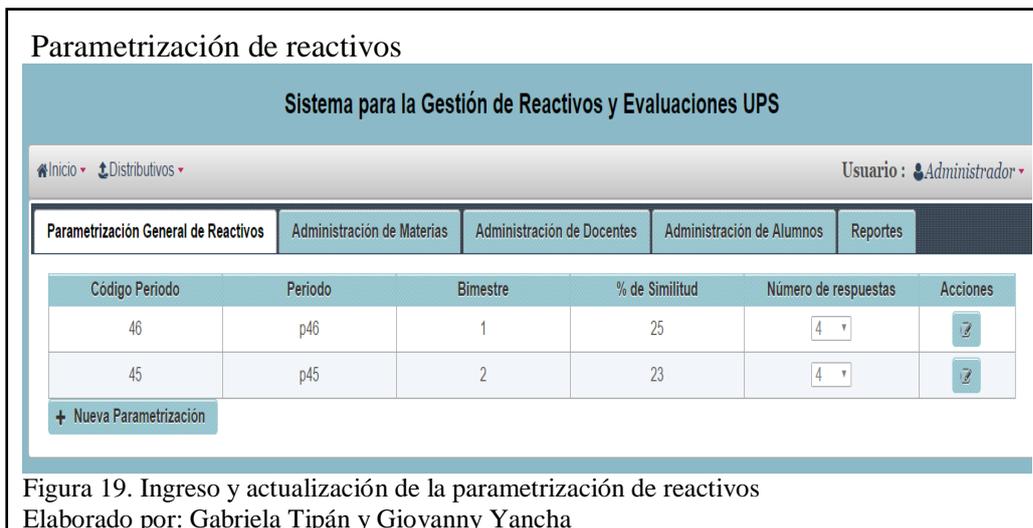


Figura 18. Capa control dentro del paquete en el proyecto
Elaborado por: Gabriela Tipán y Giovanni Yancha

4.5 Modelado de interfaces

Finalizada la construcción de la aplicación web para el manejo de reactivos, se presentan las pantallas más relevantes con su funcionalidad.

1. En la figura No. 19 se muestra que el administrador del sistema puede parametrizar los reactivos ingresando el número de respuestas por reactivo, período, bimestre, porcentaje de similitud y campos opcionales.



2. El administrador de la aplicación parametriza la información de los docentes que acceden al sistema como se muestra en la figura No. 20



3. El administrador de la aplicación parametriza las materias existentes en cada semestre, como se puede visualizar en la figura No. 21

Administración de materias

Sistema para la Gestión de Reactivos y Evaluaciones UPS

Inicio ▾ Distributivos ▾ Usuario : **Administrador** ▾

Parametrización General de Reactivos | **Administración de Materias** | Administración de Docentes | Administración de Alumnos | Reportes

SubCategorías o Materias

Nivel: Nivel 1 ▾

Sub Categoría	Categoría	Carrera	Acciones
ALGEBRA LINEAL	MATEMÁTICAS PARA INGENIERIA ▾	Gestión ▾	
ANTROPOLOGIA CRISTIANA	SOCIAL ▾	Gestión ▾	
CALCULO DIFERENCIAL	MATEMÁTICAS PARA INGENIERIA ▾	Gestión ▾	
INTRODUCCION A LA INFORMATICA	PROGRAMACION ▾	Gestión ▾	
PROGRAMACION I	PROGRAMACION ▾	Gestión ▾	
TECNICAS DE INVESTIGACION	SOCIAL ▾	Gestión ▾	

+ Nueva Sub Categoría

Figura 21. Parametrización ingreso y actualización de materias

Elaborado por: Gabriela Tipán y Giovanni Yancha

4. En la figura No. 22 se muestra la elaboración de reactivos por parte de los docentes y se puede visualizar de acuerdo al nivel y materia.

Sistema para la Gestión de Reactivos y Evaluaciones UPS

Inicio ▾ Usuario : **WALTER FERNANDO** ▾

Elaboración de Reactivos | Revisión de Reactivos | Elaboración de Exámenes Complejos | Lista de Exámenes Complejos | Elaboración de Evaluaciones

Listado de Evaluaciones

Nivel: Nivel 2 ▾ Materias: ESTRUCTURA DE DATOS ▾

(1 of 1) 5 ▾

Pregunta	Bibliografía	Dificultad	Estado	Acciones
que es estructura de datos periodo 46	teste	Alta	Elaborado ▾	

(1 of 1) 5 ▾

+ Nuevo Reactivo Envío Revisión

Figura 22 Elaboración y actualización de los reactivos

Elaborado por: Gabriela Tipán y Giovanni Yancha

5. A continuación, en la figura No. 23 se observa que despliega la pantalla en la cual se ingresan uno a uno los reactivos elaborados por el docente.

Ingreso de Reactivos

Ingreso de Reactivos

Unidad/Tema: * Tema: Temas: Mínimos:

Argumento: * Argumento: Dificultad: *

Objetivo o logro de Aprendizaje: *

Reactivo: *

Bibliografía: * [Insertar Imagen en el Texto](#)

Añadir Imagen como Anexo:

Figura 23. Pantalla de ingreso del reactivo
Elaborado por: Gabriela Tipán y Giovanni Yancha

6. En la figura No. 24 se visualiza el ingreso de las respuestas para cada reactivo creado

Ingreso de Reactivos

Ingreso de Items

Respuesta	Justificación	Opción	Correcto	Acciones
No se encontrarán respuestas				

Descripción Respuestas

Opción: Correcto:

Respuesta: *

[Insertar Imagen en el Texto](#)

javascript:void('Subindice')

Figura 24. Pantalla que muestra el ingreso de respuestas por reactivo
Elaborado por: Gabriela Tipán y Giovanni Yancha

4.6 Código relevante

A continuación, se agrega el código fuente más importante utilizado para el desarrollo de la aplicación web.

4.6.1 Métodos de acceso al sistema.

En la figura No. 25 se muestra el método que se utiliza para la verificación y validación de los usuarios que acceden a la aplicación, este código fuente es muy importante ya que, al tratarse de una página de gestión de reactivos deben utilizarse métodos de seguridad para el control de ingreso.

Métodos de acceso al sistema

```
// Metodos de la clase Usuario Sistema|
public String VerificarDatos() throws Exception {
```

Figura 25. Método de la clase usuario sistema
Elaborado por: Gabriela Tipán y Giovanni Yancha

Utiliza la librería proporcionada por DigestUtils, para realizar la encriptación devuelve el valor como una cadena hexadecimal de 32 caracteres. El método recibe la contraseña la encripta y la envía para validarla o compararla contra la información almacenada en la base de datos, una parte del método se muestra en la figura No. 26

Método de encriptación

```
String strContraseñaEncriptada = DigestUtils.md5Hex(this.usrsistem.getContrasenia());
this.usrsistem.setContrasenia(strContraseñaEncriptada);
lstusr = usrsitDAO.VerificarDatos(this.usrsistem);
```

Figura 26. Método que realiza la encriptación de la contraseña
Elaborado por: Gabriela Tipán y Giovanni Yancha

Si la validación es afirmativa recupera los datos del usuario y valida que tenga asignado un rol para el sistema el cual permite ver u ocultar las opciones a las cuales tiene acceso el usuario, como se muestra en la figura No. 27

Método para la comprobación del rol

```
if(lstrolusuario.size() != 0) {
    // metodo para comprobar rol de usuario
    Permisos(Integer.parseInt(""+lstrolusuario.get(0).getId()));
}
```

Figura 27. Método que parametriza las pantallas dependiendo del rol de usuario
Elaborado por: Gabriela Tipán y Giovanni Yancha

4.6.2 Métodos para los reactivos.

La aplicación desarrollada, a través del siguiente método permite almacenar o guardar un reactivo.

Utiliza las instancias de la clase controladores o Bean de usuario del sistema, parametrización y tipo de reactivos de los cuales recupera la información necesaria para guardar y parametrizar los mismos, como se visualiza en la figura No. 28

```
Instancia de los Bean

// Intancias de las clases beans
@ManagedProperty("#{tipoReactivoBean}")
private TipoReactivoBean tipoReactivoBean;
@ManagedProperty("#{parametrizacionReactivosBean}")
private ParametrizacionReactivosBean parametrizacionReactivosBean;
@ManagedProperty("#{usuarioSistemaBean}")
private UsuarioSistemaBean usuarioSistemaBean;
```

Figura 28. Permite manejas los controladores en varias clases de la mismas
Elaborado por: Gabriela Tipán y Giovanni Yancha

Se visualiza en la figura No. 29 la utilización del método sql.date para obtener la fecha actual del servidor

```
| Método SQL date
// obtiene la fecha actual de la aplicación
java.sql.Date dtfechaActual = new Date(Calendar.getInstance()
    .getTimeInMillis());
```

Figura 29. Método que obtiene la fecha actual del sistema
Elaborado por: Gabriela Tipán y Giovanni Yancha

El método GuardarPregunta toma los datos ingresados y también los campos que el usuario no ve y que son necesarios para la parametrización del reactivo ingresado como se puede apreciar en la figura No. 30

Método de guardar reactivo

```
@SuppressWarnings("static-access")
public void GuardarPregunta() {
    java.sql.Timestamp _sqlDate;
    reapre.setIntCodUsuarioSistema(Integer.parseInt(usuarioSistemaBean
        .getLstusr().get(0).getId().toString()));
    _sqlDate = new java.sql.Timestamp(dtfechaActual.getTime());
    reapre.setDatFechaCreacion(_sqlDate);
    reapre.setPeriodo(Integer.parseInt(parametrizacionReactivosBean
        .getLstparametrizacion().get(0).getPeriodo().toString()));
    reapre.setIntCodTipoReactivo(Integer.parseInt(tipoReactivoBean
        .getLsttiporeactivos().get(0).getId().toString()));
    reapre.setIntCodEstadoreactivo(1);
    reapre.setStrUsuarioCreo(usuarioSistemaBean.getLstusr().get(0)
        .getNombre().toString()
        + " "
        + usuarioSistemaBean.getLstusr().get(0).getApellido()
        .toString());
    reapre.setIntBimestre(parametrizacionReactivosBean
        .getLstparametrizacion().get(0).getBimestres());
    System.out.println("codigo materia: "
        + reapre.getIntCodTipoSubCategoria());
    try {
        reapredao.GuardarPregunta(reapre);
        ListarPreguntasUpdate(reapre.getIntCodTipoSubCategoria());
        FacesContext.getCurrentInstance().addMessage(
            null,
            new FacesMessage(FacesMessage.SEVERITY_INFO,
                "Reactivo Guardado", "Ingreso Items"));
    } catch (Exception e) {
        // TODO: handle exception
    }
}
```

Figura 30. Método que ingresa el reactivo escrito desde la interfaz
Elaborado por: Gabriela Tipán y Giovanni Yancha

Posterior al ingreso se puede consultar el reactivo ingresado para visualizarlo en pantalla y obtener el código único del mismo, lo cual permite asociar las respuestas u opciones a un reactivo.

4.6.3 Métodos para almacenar las respuestas.

El método visualizado en la figura No. 31 permite almacenar las opciones de respuesta que se agregan a un reactivo. Recupera el código único del reactivo que fue creado para asociarle a las opciones que se le va a asignar al reactivo.

Método para almacenar respuestas

```
public void GuardarRespuesta() throws IOException {
    FacesContext fc = FacesContext.getCurrentInstance();
    Map<String, String> params = fc.getExternalContext()
        .getRequestParameterMap();
    // obtiene el codigo del Reactivo Ingresado
    int intCodReactivo = Integer.parseInt(params.get("id1").toString());
```

Figura 31. Método general y recuperación del código del reactivo ingresado
Elaborado por: Gabriela Tipán y Giovanni Yancha

Posteriormente el método llama a una sentencia for a través de la cual recorre el número de opciones del reactivo de acuerdo al número parametrizado inicialmente cuyo valor es entre 4-7 y lo almacena uno a uno.

A continuación, en la figura No. 32 se muestra una parte del ingreso, el método que almacena la información en la base de datos es: `rearesdao.GuardarRespuesta(reares)`

```
Método del almacenamiento en BDD

try {
    if(!validarSelecionRespuesta(parametrizacionReactivosBean
        .getLstparametrizacion().get(0).getRespuestas())){
        for (int i = 0; i < parametrizacionReactivosBean
            .getLstparametrizacion().get(0).getRespuestas(); i++) {
            if (i == 0) {
                reares.setStropcion(stropA);
                reares.setStrRespuesta(strdesA);
                reares.setStrJustificacion(strjusa);
                reares.setImagen(imageDatA);
                reares.setBlcorrecto(blA);
                rearesdao.GuardarRespuesta(reares);
            } else if (i == 1) {
                reares.setStropcion(stropB);
                reares.setStrRespuesta(strdesB);
                reares.setStrJustificacion(strjusB);
                reares.setImagen(imageDatB);
                reares.setBlcorrecto(blB);
                rearesdao.GuardarRespuesta(reares);
            }
        }
    }
}
```

Figura 32. Método que ingresa las respuestas por cada reactivo
Elaborado por: Gabriela Tipán y Giovanni Yancha

4.6.4 Método para añadir imágenes.

Se creó un método que permite almacenar temporalmente un anexo antes de ser insertado definitivamente a la base de datos.

El método utiliza un evento Ajax el cual obtiene de manera automática el contenido del archivo para almacenarlo en una variable tipo bytes e insertarlo definitivamente en la base de datos, como se muestra en la figura No. 33

```
Método para añadir imágenes

if (event.getFile() != null) {
    if (event.getComponent().getId().equals("imagenA")) {
        File files = new File(event.getFile().getFileName());
        // crea el byte para almacenar en BDD
        imageDatA = new byte[(int) event.getFile().getSize()];
        // Asigna el contenido de la imagen a Byte
        imageDatA = event.getFile().getContents();
        strnombreimagenA = event.getFile().getFileName();
    } else if (event.getComponent().getId().equals("imagenB")) {
```

Figura 33. Método que permite añadir imágenes como anexos
Elaborado por: Gabriela Tipán y Giovanni Yancha

Primero se obtiene la longitud del archivo para transformar el mismo a bytes.

El objetivo de que el anexo agregado a un reactivo se almacene temporalmente antes de ser almacenado definitivamente se debe a que, para almacenar el reactivo como nuevo registro deben llenarse todos los campos y si se inserta la imagen a la base de datos sin antes validar los otros campos, este campo se ingresará vacío (null).

4.6.5 Métodos para generar examen.

En esta clase se definen los métodos necesarios tanto para el ingreso, parametrización y generación de cualquier tipo de examen.

El método utiliza el componente ITEXT para java que permite generar documentos PDF y pre visualizar el mismo en pantalla antes de guardar como se puede visualizar en la figura No. 34

Librería para Documentos

```
import com.itextpdf.text
```

Figura 34. Librería ITEXT que genera los exámenes en PDF
Elaborado por: Gabriela Tipán y Giovanni Yancha

En la figura No. 35 se muestra el método para añadir el número de preguntas que tendrá un examen

Método añadir número de reactivos

```
// Añade el numero de preguntas por materias despues de calcular el  
// porcentaje  
public void AñadirNumeroPreguntas(int intcodigo) {
```

Figura 35. Genera el número de reactivos para el examen automático
Elaborado por: Gabriela Tipán y Giovanni Yancha

Es importante saber que cada materia está asignada a un eje, cada eje tiene un porcentaje establecido, por lo cual se creó un método para que, al generar un examen automático, se calcule el número de preguntas por materias o Sub Categorías asignadas al eje y dependiendo del porcentaje que corresponde.

Partiendo del número total de preguntas por examen y después de establecer un porcentaje a cada eje, se realiza un cálculo en base al número de materias que tiene dicho eje y se le calcula un número de preguntas.

Estos métodos permitieron que el docente no añada los reactivos de forma manual al elegir generar el examen automático.

4.6.6 Método para generar el examen en un archivo pdf.

El método visualizado en la figura No. 36 utiliza el componente itext para java que permite generar documentos pdf y pre visualizar el mismo en pantalla antes de guardar.

Método para generar el examen PDF

```
// Genera el examen en PDF  
public void GenerarExamenPDF () {
```

Figura 36. Librería ITEXT que genera los exámenes en PDF
Elaborado por: Gabriela Tipán y Giovanni Yancha

Después de haber parametrizado los ejes, las materias y las preguntas que se desea para el examen, el método genera el examen en pdf. El método primero recorre los ejes y los añade al documento. Recupera el código de materia y consulta todos los reactivos de la misma como se muestra en la figura No. 37

Parte del método de añadir reactivos automáticos

```
for (int i = 0; i < numerodeejes; i++)  
{  
    if (matrizejes[i][0] == intcodigo)  
    {  
        intnumeroPreguntas = (matrizejes[i][2] * intNumeroIngresoReactivos) / 100;  
        intnumeroPreguntasMateria = (int) intnumeroPreguntas  
            / matrizejes[i][1];  
        intnumeroPreguntasMateriadecimal = (double) intnumeroPreguntas  
            / matrizejes[i][1];  
        intnumerodecimal = intnumeroPreguntasMateriadecimal  
            - intnumeroPreguntasMateria;  
        intnumeroPreguntasRestantes = (int) (intnumerodecimal * matrizejes[i][1]);  
        // parte donde asigna laspreguntas cuando es exacto  
        if (intnumeroPreguntas % matrizejes[i][1] == 0)  
        {  
            for (int j1 = 0; j1 < matrizejesmaterias.length; j1++)  
            {  
                if (matrizejesmaterias[j1][0] == intcodigo)  
                {  
                    matrizejesmaterias[j1][2] = intnumeroPreguntasMateria;  
                    System.out.println("Eje: "  
                        + matrizejesmaterias[j1][0] + " Materia "  
                        + matrizejesmaterias[j1][1]  
                        + " Num Preguntas "  
                        + matrizejesmaterias[j1][2]);  
                }  
            }  
        }  
    }  
}
```

Figura 37. Parte del método de añadir reactivos automáticos de acuerdo al porcentaje de sus ejes
Elaborado por: Gabriela Tipán y Giovanni Yancha

Como se puede apreciar en la figura No. 38 se utiliza un método que permite codificar el contenido HTML de los reactivos que se generara en el pdf ya que al almacenar en base de datos se guarda como contenido HTML.

Método para codificar contenido HTML

```
// Método donde elimina el contenido html para añadir al documento PDF
obj.ContenidoHTML2(pdf, lstpreguntaApr.get(k).getStrPreguntaReactivo(), writer);
```

Figura 38. Instancia del método que codifica el contenido HTML para añadir a PDF
Elaborado por: Gabriela Tipán y Giovanni Yancha

Se añaden las respuestas por reactivo y los anexos en caso de que el usuario haya seleccionado anteriormente la opción de ver anexos como se puede apreciar en la figura No. 39

Método de para añadir respuestas

```
// For donde añade las respuestas por pregunta
for ( int t = 0; t < lstrespreguntaApr.size(); t++ ) {
    // Método donde elimina el contenido html para añadir al documento PDF

    pdf.add(new Paragraph(lstrespreguntaApr
        .get(t).getStropcion().trim()
        + "-"));
    obj.ContenidoHTML2(pdf, lstrespreguntaApr.get(t).getStrRespuesta(), writer);
    // Valida si tiene anexo o desea ver con archivos adjuntos
    if ( lstrespreguntaApr.get(t).getImagen() != null && anexosver == true )
    {
        Image imagen1 = Image
            .getInstance(lstrespreguntaApr
                .get(t).getImagen());
        imagen1.setAlignment(Element.ALIGN_CENTER);
        imagen1.scalePercent(10f);
        pdf.add(imagen1);
    }
}
h++;
```

Figura 39. Parte del método que añade las respuestas de los reactivos en el documento PDF
Elaborado por: Gabriela Tipán y Giovanni Yancha

4.6.7 Método para la comparación de la similitud de los reactivos.

Basado en el análisis léxico que en lingüística es el porcentaje de similitud entre conjunto de cadenas, caracteres o palabras; como parte del requerimiento en la

validación del ingreso de reactivos en el cual la aplicación permita una comparación mediante la parametrización de un límite de porcentaje de similitud entre el reactivo que se ingresa con los que reposan en el repositorio de reactivos.

Se utilizó el algoritmo de LetterPairSimilarity en comparación con otro propuesto, se realizaron pruebas de comparación, siendo este el más óptimo ya que refleja valores más reales.

El algoritmo fue impulsado por los siguientes requisitos:

“Un verdadero reflejo de la similitud léxica - las cadenas con pequeñas diferencias deben ser reconocidas como similares. En particular, una superposición significativa de subcadenas debería indicar un alto nivel de similitud entre las cadenas.

Una robustez a los cambios de orden de las palabras - dos cadenas que contienen las mismas palabras, pero en un orden diferente, debe ser reconocido como similar. Por otro lado, si una cadena es sólo un anagrama aleatorio de los caracteres contenidos en el otro, entonces debería (usualmente) ser reconocido como diferente.

Independencia del lenguaje - el algoritmo debería trabajar no sólo en inglés, sino en muchos idiomas diferentes” (Arnott & White, s.f.)

El algoritmo trabaja con la siguiente fórmula, esta usa pares de caracteres en las dos cadenas de caracteres como se visualiza en la figura No. 40

Algoritmo de similitud

$$\text{similarity}(s1, s2) = \frac{2 \times |\text{pairs}(s1) \cap \text{pairs}(s2)|}{|\text{pairs}(s1)| + |\text{pairs}(s2)|}$$

Figura 40. Algoritmo que realiza la comparación entre dos cadenas de caracteres
Elaborado por: Gabriela Tipán y Giovanni Yancha

En la figura No. 41 se puede apreciar el ejemplo realizado en la consola de java del algoritmo utilizado para la comparación entre las cadenas de caracteres:

```

Pruebas del algoritmo de comparación
Algoritmo de LetterPairSimilarity ....
Pruebas de similitud :
ingrese el texto 1:
prueba de similitud
ingrese el texto 2:
prueba de similitud caracteres
Valor devuelto: 76.0
Valor devuelto: 75.67567567567568

```

Figura 41. Pruebas realizadas del algoritmo para la comparación de cadenas
Elaborado por: Gabriela Tipán y Giovanni Yancha

Dentro del proyecto se utiliza el método de validación de reactivos, el cual compara la similitud del reactivo ingresado, como se indica en la figura No. 42

```

Método de comparación
public Boolean ComparacionReactivo(String strDescripcionIngreso, int _intTipoGuardar, long intCodigoReactivo, int intCodSubCategoria) {

```

Figura 42. Instancia del método que permite compara el reactivo ingresado
Elaborado por: Gabriela Tipán y Giovanni Yancha

4.7 Diccionario de datos

Contiene un detalle de todos los datos que intervienen en el sistema.

Tabla 19.

REACTIVOS PREGUNTA APROBACIÓN

Nombre del campo	Tipo	Longitud	Descripción	Clave
RPA_IN_CODIGO	SERIAL	-	Código único del reactivo o pregunta del sistema	Clave primaria
TR_IN_CODIGO	INTEGER	-	Código único del tipo de reactivo	-
DS_IN_CODIGO	INTEGER	-	Código único del docente que crea el reactivo	-

TS_IN_CODIGO	INTEGER	-	Código único del tipo de sub categoría o materia asignado	-
ER_IN_CODIGO	INTEGER	-	Código único del estado del reactivo asignado	-
RPA_A1_REACTIVO	VARCHAR	100000	Código único de la opción respuesta del reactivo	-
RPA_A_PERIODO	VARCHAR	10	Descripción del reactivo o pregunta	-
RPA_DA_FECHA_CREACION	TIMESTAMP	-	Fecha de creación del reactivo	-
RPA_A50_OBJETIVO	VARCHAR	500	Objetivo del reactivo	-
RPA_A1_TEMA	VARCHAR	100	Tema de reactivo	-
RPA_A5_OBSERVACIONES	VARCHAR	500	Observaciones del reactivo	-
RPA_A5_ARGUMENTO	VARCHAR	500	Argumento del reactivo	-
RPA_A1_DIFICULTAD	VARCHAR	10	Dificultad del reactivo	-

D	R			
RPA_A5_BIBLIOGRA FIA	VARCHA R	500	Bibliografía del reactivo	-
RPA_IM_IMAGEN	BYTEA	-	Anexo del reactivo tipo imagen	-
RPA_VA_ASIGNADO	VARCHA R	50	Descripción del usuario que creo el reactivo	-
RPA_IN_UTILIZADO	INTEGER	-	Si reactivo está vigente o dado de baja	-
RPA_DA_FECHA_UT ILIZADO	TIMESTA MP	-	Fecha en la cual un reactivo es utilizado	-
RPA_IN_BIMESTRE	INTEGER	-	Bimestre en el cual el reactivo es utilizado	-

Nota: Contiene los datos de la tabla REACTIVOS PREGUNTA APROBACIÓN

Tabla 20.

REACTIVOS OPCIONES RESPUESTAS APROBADAS

Nombre del campo	Tipo	Longitud	Descripción	Clave
ROPA_IN_CODIGO	SERIAL	-	Código único de la opción respuesta del reactivo	Clave primaria
RPA_IN CODIGO	INTEGER	-	Código único del reactivo o	Clave secundaria

			pregunta	
ROPA_VAR_DESCRIPCION	VARCHAR	100000	Descripción de la opción respuesta	-
ROPA_VAR_OPCION	CHAR	10	Descripción de la opción	-
ROPA_BOL_CORRECTO	BOOL	-	True o false en el caso de ser correcta o incorrecta la opción	-
ROPA_VAR_JUSTIFICACION	VARCHAR	100000	Descripción de la justificación para la respuesta	-
ROPA_IM_IMAGEN	BYTEA	-	Anexo de la opción tipo imagen	-

Nota: Contiene los datos de la tabla REACTIVOS OPCIONES RESPUESTAS APROBADAS

4.8 Pruebas

Existen diferentes tipos de pruebas para evaluar la calidad de la aplicación web desarrollada.

Para registrar los tipos de pruebas con los resultados obtenidos se creó la siguiente plantilla:

Tabla 211.
Plantilla de pruebas

ID	Código de la prueba realizada. Se establece el ID de la prueba con las siguientes iniciales: PR (Prueba) seguida de un número secuencial que aumentará desde el número 1
Nombre	Nombre de la prueba realizada
Descripción	Descripción de la prueba realizada
Pasos	Pasos para la ejecución de la prueba
Resultado	Resultado de la prueba realizada
Errores	Errores presentados
Responsable	Responsable de ejecutar la prueba

Nota: Plantilla para ingresar los datos de las pruebas realizadas

A continuación, se detallan los tipos de pruebas a realizarse.

4.8.1 Pruebas unitarias.

Las pruebas unitarias permitieron evaluar el desarrollo del producto tecnológico por módulos separados.

Tabla 22.
Prueba 1

ID	PR1
Nombre	Módulo de administración
Descripción	Se realiza el ingreso al módulo de administración de

	la aplicación web para evaluar las opciones que se tienen en el mismo.
Pasos	Ingresar al sistema como administrador Identificar las opciones que se habilitan al ingresar Escoger cada una de las opciones y probarlas
Resultado	Resultado de la prueba realizada
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Prueba en módulo de administración

Tabla 23.
Prueba 2

ID	PR2
Nombre	Módulo de elaboración de reactivos
Descripción	Se realiza el ingreso al módulo de elaboración de reactivos de la aplicación web para evaluar las opciones que se tienen en el mismo.
Pasos	Ingresar al sistema como docente Identificar las opciones que se habilitan al ingresar Escoger cada una de las opciones y probarlas
Resultado	Resultado de la prueba realizada
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Prueba en módulo de elaboración de reactivos

Tabla 24.
Prueba 3

ID	PR3
Nombre	Módulo de revisión de reactivos
Descripción	Se realiza el ingreso al módulo de revisión de reactivos de la aplicación web para evaluar las opciones que se tienen en el mismo.
Pasos	Ingresar al sistema como usuario de la comisión Identificar las opciones que se habilitan al ingresar Escoger cada una de las opciones y probarlas
Resultado	Resultado de la prueba realizada
Errores	No se presentaron errores
Responsable ejecución	Giovanny Yancha

Nota: Prueba en módulo de revisión de reactivos

4.8.2 Pruebas funcionales.

Las pruebas funcionales realizadas en el software desarrollado ayudaron a verificar su correcta funcionalidad.

Tabla 25.
Prueba 4

ID	PR6
Nombre	Ingreso al sistema con usuarios autorizados
Descripción	Intentar ingresar al sistema con usuarios autorizados
Pasos	Intentar loguearse con usuarios autorizados que tienen acceso a la aplicación
Resultado	Se verificó que el sistema permite que los usuarios autorizados pueden acceder.

Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas para ingresar al sistema con usuarios autorizados

Tabla 26.
Prueba 5

ID	PR7
Nombre	Roles establecidos para usuario docente
Descripción	Verificar que los usuarios puedan acceder a la aplicación web y se desplieguen únicamente las opciones de acuerdo a su rol
Pasos	-Ingresar al sistema como docente -Verificar las opciones que el usuario puede visualizar
Resultado	Se verificó que el sistema permite a los docentes visualizar las opciones únicamente de acuerdo a su rol.
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas con rol de docente

Tabla 27.
Prueba 6

ID	PR8
Nombre	Roles establecidos para usuario administrador
Descripción	Verificar que el usuario administrador pueda acceder a la aplicación web y se desplieguen únicamente las

	opciones de acuerdo a su rol
Pasos	-Ingresar al sistema como administrador -Verificar las opciones que el usuario puede visualizar
Resultado	Se verificó que el sistema permite al usuario administrador visualizar las opciones de acuerdo a su rol.
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas con rol de administrador

Tabla 28.
Prueba 7

ID	PR9
Nombre	Roles establecidos para usuario de la comisión
Descripción	Verificar que cada usuario pueda acceder a la aplicación web y se desplieguen únicamente las opciones de acuerdo a su rol
Pasos	-Ingresar al sistema como comisión -Verificar las opciones que el usuario puede visualizar
Resultado	Se verificó que el sistema permite a los usuarios que son parte de la comisión visualizar las opciones de acuerdo a su rol.
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas con rol de usuario de la comisión

Tabla 29.
Prueba 8

ID	PR10
Nombre	Roles establecidos para usuario estudiante
Descripción	Verificar que cada usuario pueda acceder a la aplicación web y se desplieguen únicamente las opciones de acuerdo a su rol
Pasos	-Ingresar al sistema como estudiante -Verificar las opciones que el usuario puede visualizar
Resultado	Se verificó que el sistema permite a los estudiantes visualizar únicamente las opciones de acuerdo a su rol.
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas con rol de estudiante

Tabla 30.
Prueba 9

ID	PR12
Nombre	Revisión y aprobación de reactivos
Descripción	Revisión de reactivos y aprobación para su uso en diferentes exámenes
Pasos	-Se realiza la revisión de reactivos enviados por los docentes

	-Se aprueban reactivos para su uso en elaboración de exámenes.
Resultado	-La aplicación web permitió visualizar todos los reactivos que los docentes enviaron para revisión. -La aplicación web permitió aprobar reactivos para su posterior uso en la elaboración de exámenes.
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas de revisión y aprobación de reactivos

Tabla 35.
Prueba 10

ID	PR15
Nombre	Elaboración de exámenes
Descripción	Elaboración de exámenes por parte de la comisión
Pasos	Se realiza la elaboración de exámenes definiendo el número de preguntas y escogiendo usuarios o niveles que accederán a los mismos.
Resultado	La aplicación web permitió realizar la elaboración de diferentes tipos de exámenes para que los estudiantes rindan los mismos.
Errores	No se presentaron errores
Responsable	Giovanny Yancha

Nota: Pruebas de elaboración de exámenes

4.8.3 Pruebas de integración.

Tabla 36.
Prueba 11

ID	PR16
Nombre	Elaboración de reactivos para exámenes
Descripción	Ingreso al sistema para crear reactivos utilizados en exámenes complexivos
Pasos	Se ingresa al sistema con rol de docente, se elaboran reactivos y se envían a revisión. Se ingresa al sistema con rol de comisión, se aprueban reactivos y se genera examen complexivo.
Resultado	La aplicación web permitió realizar la elaboración de reactivos para enviarlos a revisión. Los reactivos se aprobaron y se generó de manera exitosa el examen complexivo.
Errores	No se presentaron errores
Responsable ejecución	Giovanny Yanca

Nota: Pruebas de elaboración de reactivos para exámenes

Tabla 37.
Prueba 12

ID	PR17
Nombre	Elaboración de reactivos para evaluaciones de la

	materia
Descripción	Ingreso al sistema para crear reactivos utilizados en pruebas de la materia
Pasos	Se ingresa al sistema con rol de docente, se elaboran reactivos y utilizan para generar evaluaciones de las propias materias
Resultado	La aplicación web permitió realizar la elaboración de reactivos para usarlos en evaluaciones de las materias de los docentes
Errores	No se presentaron errores
Responsable ejecución	Giovanny Yancha

Nota: Pruebas de elaboración de reactivos para evaluaciones de las materias del docente

4.8.4 Pruebas de rendimiento.

Para poder realizar las pruebas de rendimiento se utilizó la herramienta JMeter, la cual está desarrollada en java y permite medir el comportamiento de servidores. Dicha aplicación permite establecer un número específico de usuarios para simular en la prueba el período de tiempo y el número de veces que se ejecutará la misma.

4.8.4.1 Prueba 1.

La primera prueba se estableció con 1400 usuarios en un periodo de 10 segundos, cada hilo se inició 0,007 segundos después de que el hilo anterior fue comenzado y se repitió sólo una vez

Prueba 1 de rendimiento con 1400 usuarios tabla de resumen

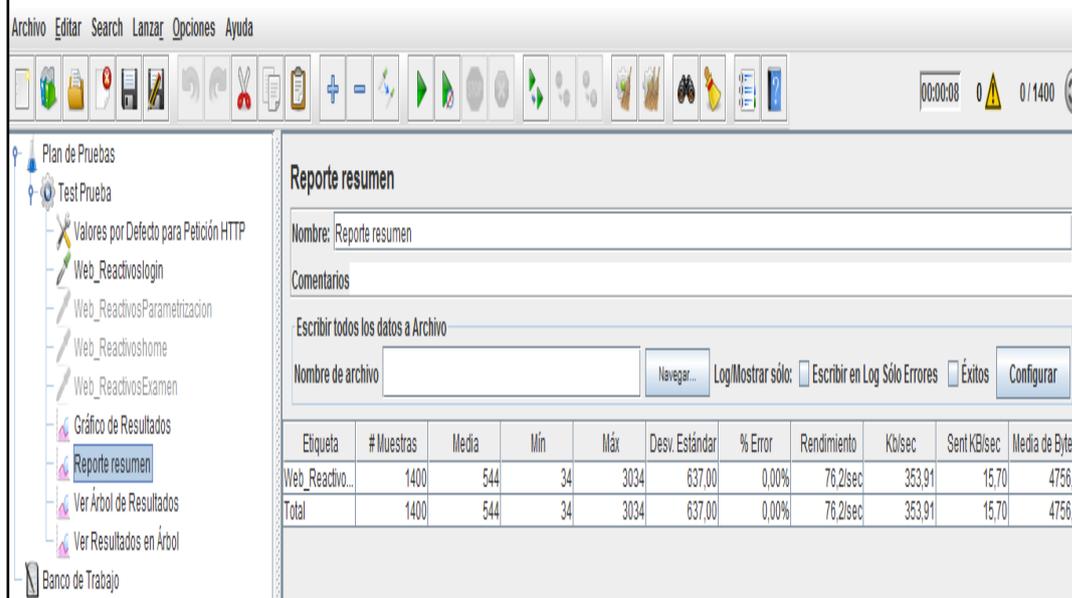


Figura 43. Se realizó una prueba de rendimiento con 1400 usuario en 10 segundos

Elaborado por: Gabriela Tipán y Giovanni Yancha

Prueba 1 de rendimiento con 1400 usuarios gráfico

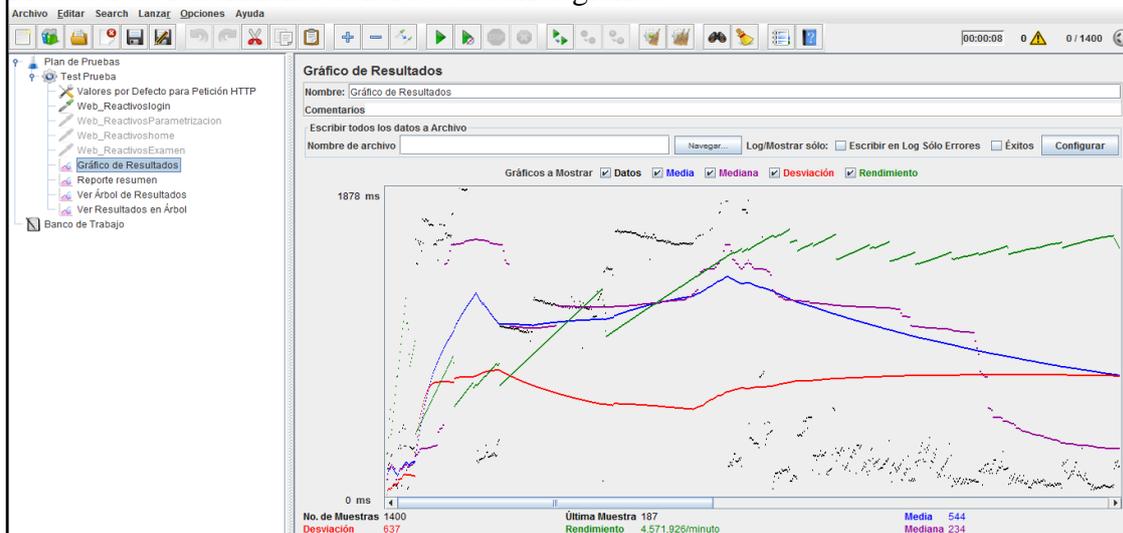


Figura 43. Se realizó una prueba de rendimiento con 1400 usuario en 10 segundos

Elaborado por: Gabriela Tipán y Giovanni Yancha

En la figura No. 43 se muestra un resumen que permite ver que el rendimiento es óptimo ya que en resumen muestra que la media de respuesta fue de 544 milisegundos; y en la figura No. 44 el rendimiento estuvo debajo de 1078 milisegundos.

4.8.4.2 Prueba 2.

Se realizó la segunda prueba de la página de parametrización estableciendo los siguientes parámetros:

Se establecieron 100 usuarios en un periodo de 10 segundos y se repitió una sola vez para comprobar la respuesta de la página evaluada; para la prueba se tomó en cuenta que la parametrización es visible únicamente para el rol del administrador por lo cual se establecieron 100 usuarios como un máximo, aunque con el rol de administrador el acceso a la página sería máximo de dos usuarios.

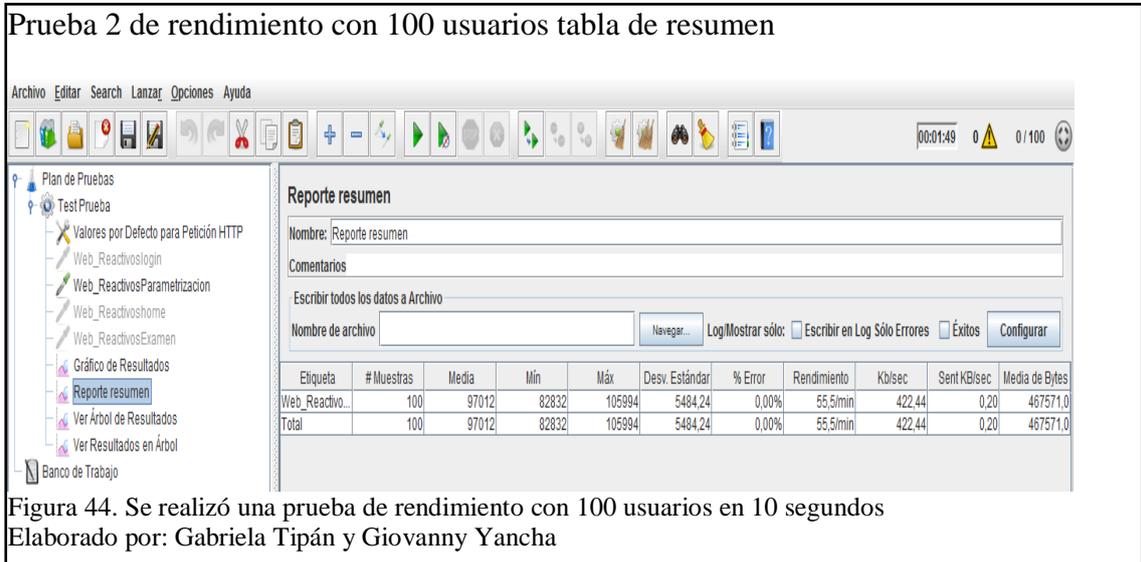


Figura 44. Se realizó una prueba de rendimiento con 100 usuarios en 10 segundos
Elaborado por: Gabriela Tipán y Giovanni Yancha



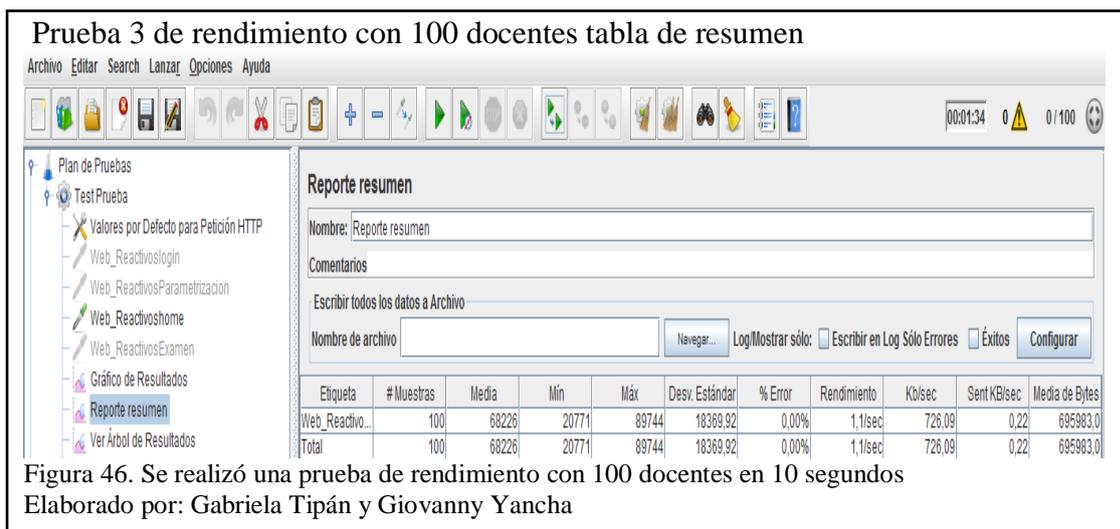
Figura 45. Se realizó una prueba de rendimiento con 100 usuario en 10 segundos
Elaborado por: Gabriela Tipán y Giovanni Yancha

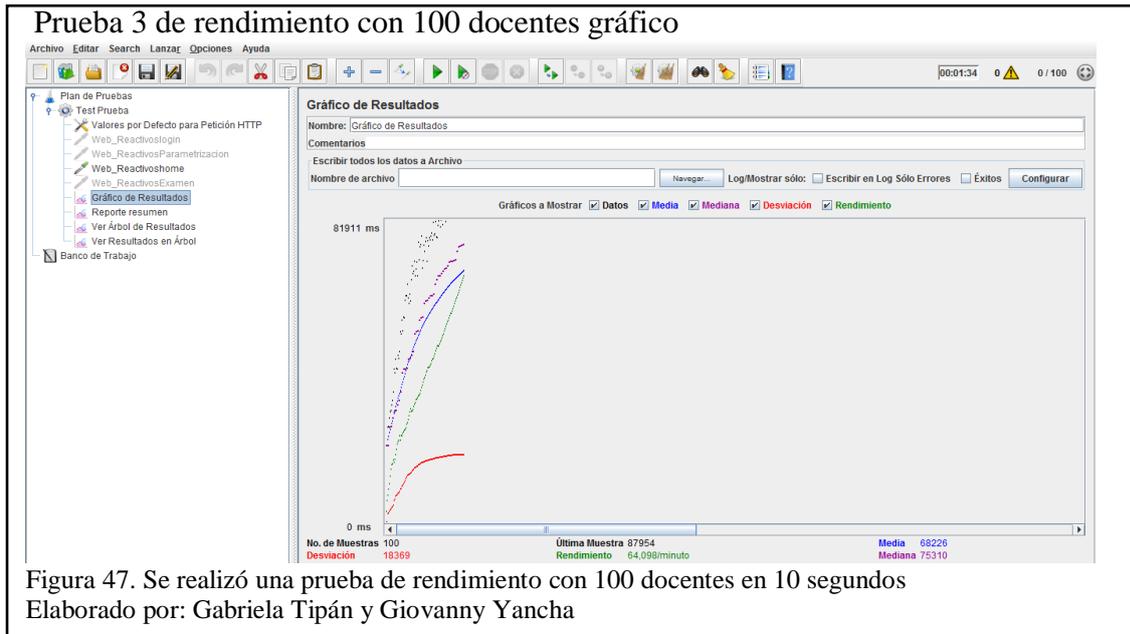
En la figura No. 45 se muestra un resumen que permite ver que el rendimiento fue óptimo al realizar la prueba con 100 usuarios teniendo en cuenta que a la página analizada por default accederán máximo 2 usuarios; la media de respuesta fue de 82832 milisegundos equivalentes a 0,82 segundos de respuesta por usuario y en la figura No. 46 el rendimiento fue por debajo de 99428 milisegundos.

4.8.4.3 Prueba 3.

Para esta tercera prueba de la página de los roles de docentes y comisión se establecieron los siguientes parámetros:

Se establecieron 100 usuarios (docentes) en un periodo de 10 segundos repitiéndose una sola vez para comprobar la respuesta de la aplicación; teniendo en cuenta que la página es visible únicamente para el rol de comisión y docentes, razón por la cual se establecieron 100 usuarios como máximo aunque realmente el número de docentes que accederán a la página es de aproximadamente 40.





En la figura No. 47 se muestra un resumen que permite ver que el rendimiento fue óptimo, la media de respuesta fue de 68226 milisegundos equivalentes a 0,68 segundos de respuesta por usuario y en la figura No. 48 el rendimiento fue por debajo de 81911 milisegundos.

4.8.4.4 Prueba 4.

Para la cuarta prueba de la página de exámenes se establecieron los siguientes parámetros:

Se establecieron 1000 usuarios en un periodo de 10 segundos repitiéndose una sola vez para comprobar la respuesta de la aplicación, a esta página se simuló el acceso de los estudiantes a rendir las diferentes evaluaciones, actualmente se tienen 625 estudiantes en la carrera de Ingeniería de Sistemas.

Prueba 4 de rendimiento con 1000 usuarios tabla de resumen

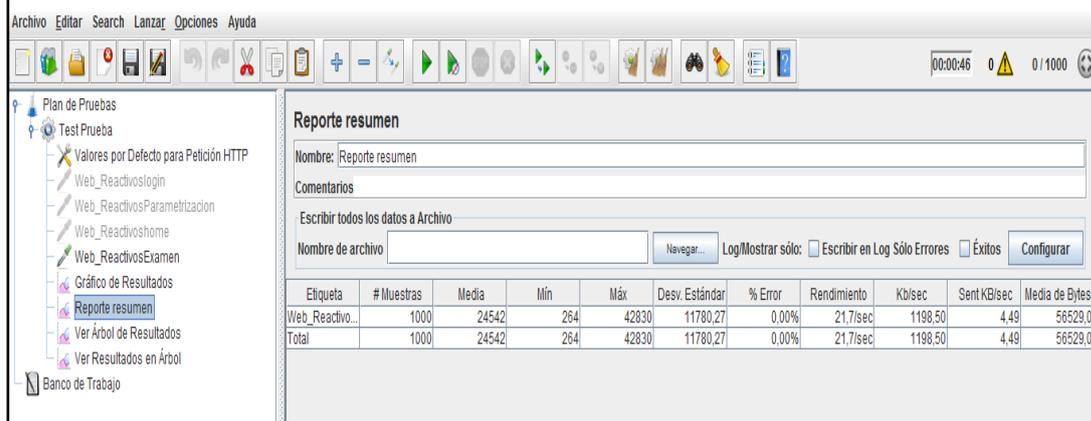


Figura 48. Se realizó una prueba de rendimiento con 1000 usuarios en 10 segundos
Elaborado por: Gabriela Tipán y Giovanni Yancha

Prueba 4 de rendimiento con 1000 usuarios gráfico

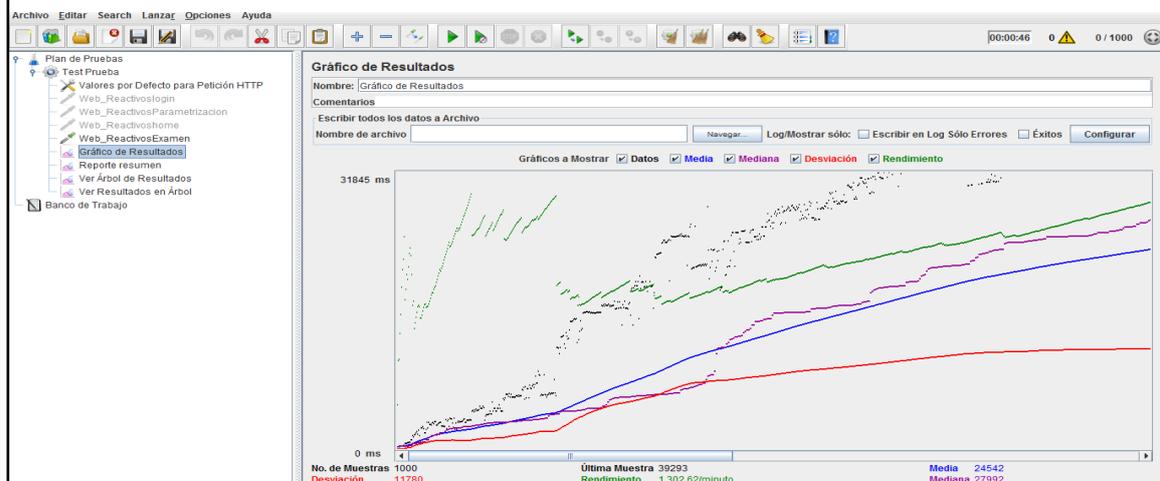


Figura 49. Se realizó una prueba de rendimiento con 1000 usuarios en 10 segundos
Elaborado por: Gabriela Tipán y Giovanni Yancha

En la figura No. 49 se muestra un resumen que permite ver que el rendimiento fue óptimo ya que en resumen muestra que la media de respuesta fue de 24542 milisegundos equivalentes a 0,24 segundos de respuesta por usuario y en la figura No. 50 el rendimiento estuvo por debajo de 31845 milisegundos.

CONCLUSIONES

- El uso de la metodología SCRUM permitió la constante comunicación con el cliente, lo cual ayudó a obtener una clara especificación de requerimientos para el desarrollo del proyecto obteniendo así una gran adaptación y flexibilidad;
- permitió una mejor mitigación de riesgos al tener dividido al proyecto en iteraciones o sprints; los riesgos fueron los cambios de especificaciones que pudieron haber ocasionado pérdidas de recursos como costos y tiempo.
- El uso de la herramienta Hibernate en el desarrollo de la aplicación informática, permitió lograr gran facilidad en el manejo de la base de datos con los objetos, además, permitió manejar de forma automatizada y transparente la persistencia dependiendo de la forma con la que se realicen las consultas, lo cual significó un aporte significativo para la gestión de la arquitectura MVC.
- El uso de Primefaces como framework para JSF en el diseño de la páginas ofreció componentes de calidad, con lo cual se consiguió que el manejo de la parametrización, ingreso de reactivos, visualización de las evaluaciones sean amigables con el usuario.
- Realizar pruebas de rendimiento con la herramienta Jmeter permitió validar errores al someter la aplicación a una carga más amplia de usuarios de forma concurrente, posteriormente, una vez vistos los resultados, se corrigieron estas anomalías y se validó el funcionamiento de la aplicación con la cantidad de usuarios necesarios, logrando con esto, un desempeño óptimo.

- El manejo de Hibernate es muy práctico en la comunicación con la base de datos, sin embargo es recomendable para el manejo de ésta, tener conocimientos básicos sobre su uso en su propio lenguaje HQL ya que esta sintaxis no es igual a la que normalmente se ejecuta en los motores de bases de datos.

RECOMENDACIONES

- Para hacer uso de la metodología ágil SCRUM para desarrollo de software es indispensable que el equipo de trabajo se encuentre comprometido con el proyecto ya que el logro de los objetivos en cada iteración dependen únicamente de cada uno de los miembros.
- Como manejo de buenas prácticas en el desarrollo de proyectos tecnológicos es recomendable manejar estándares para todos los componentes del proyecto, base de datos, modelos de proyecto y código fuente, lo cual permita una fácil comprensión.
- En base al desarrollo del proyecto, específicamente sobre el requerimiento de la comparación de reactivos, el algoritmo compara las cadenas en base al lenguaje semántico; es recomendable para esta comparación utilizar un algoritmo que permita implementar inteligencia artificial para obtener así un resultado más real.
- El uso de Hibernate ofrece la implementación del CRUD, que son operaciones o métodos rápidos que permiten crear la sentencia HQL con el objetivo de optimizar las consultas con persistencia, por lo que es recomendable el uso en aplicaciones orientadas a la web con al menos cientos de peticiones cuyos milisegundos y latencia son primordiales.

GLOSARIO DE TÉRMINOS

HTTP. - Hypertext Transfer Protocol o en español protocolo de transferencia de hipertexto

MVC. - Modelo-Vista-Controlador

XHTML.- Extensible HyperText Markup Language, es básicamente HTML expresado como XML válido.

JSF. - Java Server Faces

AJAX. - Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML)

HQL. - Hibernate Query Language

JEE. - Java Plataform, Enterprise Edition

CRUD .- Create, Read, Update y Delete o Insertar, leer, actualizar y eliminar

GPL. - Licencia Pública General

LGPL. - Licencia Pública General Reducida de GNU

HTML. - Siglas en inglés de HyperText Markup lenguaje de marcas de hipertexto

BDD .- Base de datos

LISTA DE REFERENCIAS

- A. K., & W. S. (s.f.). *How to Strike a Match*. Obtenido de Catalysoft:
<http://www.catalysoft.com/articles/StrikeAMatch.html>
- Aparicio, E. A., Zurutuza, M. J., & Iparraguirre, C. L. (s.f.). *Manual para la elaboración de reactivos*. Obtenido de Manual para la elaboración de reactivos:
<http://www.setse.org.mx/ReformaEducativa/Rumbo%20a%20la%20Primera%20Evaluaci%C3%B3n/Factores%20de%20Evaluaci%C3%B3n/Did%C3%A1cticos/Manual%20para%20la%20elaboraci%C3%B3n%20de%20reactivosDGB.pdf>
- Bizagi. (2014). *BPMN 2.0*. Obtenido de BPMN 2.0:
<http://resources.bizagi.com/docs/BPMNbyExampleSPA.pdf>
- Dept. Ciencia de la Computación e IA. (26 de Junio de 2014). *Introducción a JavaServer Faces*. Obtenido de Introducción a JavaServer Faces:
<http://www.jtech.ua.es/j2ee/publico/jsf-2012-13/sesion01-apuntes.html#Caracter%C3%ADsticas+de+JSF>
- Echeverry, A. L., C. C., & Ayala, L. V. (Septiembre de 2008). *INTRODUCCIÓN A LA CALIDAD DE SOFTWARE*. Obtenido de INTRODUCCIÓN A LA CALIDAD DE SOFTWARE:
https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&sqi=2&ved=0ahUKEwjy8le51uDQAhUMfiYKHAEbAfAQFggdMAE&url=https%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Farticulo%2F4745899.pdf&usq=AFQjCNHatauvQ3_fcspZq0K7L2ag8ISjvQ&bvm=bv.140496471,d.eWE&c
- Gallego, M. T. (s.f.). *Metodología Scrum*. Obtenido de GESTIÓN DE PROYECTOS INFORMÁTICOS:
<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- González, H. S. (21 de Marzo de 2003). *Manual Hibernate*. Obtenido de Manual Hibernate:
<http://static1.1.sqspcdn.com/static/f/923743/14427535/1317484934257/ManualHibernate.pdf?token=Ofaih5C6RC19fhfTKNUJbXH8A%3D>
- Kroenke, D. M. (2003). *Procesamiento de bases de datos : fundamentos, diseño e implementación*. México, D.F.: Pearson Educación, 2003.
- Letelier, P. (s.f.). *IntroduccionProcesoSW*. Obtenido de IntroduccionProcesoSW:
https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjSwqGQkuDQAhUI5iYKHRUvBpoQFggkMAI&url=http%3A%2F%2Fwww.dsic.upv.es%2Fasignaturas%2Ffacultad%2Flsi%2Fdoc%2FIntroduccionProcesoSW.doc&usq=AFQjCNGAyOzvXjChM6izslal_eLoQhtk2Q&bvm=bv
- Méndez, F. C., & Vizueté, A. G. (2010). *Tesis "DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN PORTAL WEB PARA EL CONTROL Y SEGUIMIENTO ACADÉMICO DE LOS ESTUDIANTES DE SYSTEM CENTROS DE FORMACIÓN"*. Obtenido de Tesis "DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN PORTAL WEB PARA EL CONTROL Y SEGUIMIENTO ACADÉMICO DE LOS ESTUDIANTES DE SYSTEM CENTROS DE

- FORMACIÓN": <http://dspace.ups.edu.ec/bitstream/123456789/292/13/UPS-CT001848.pdf>
- Mendez, G. (2009). *Proceso Software y Ciclo de Vida*. Obtenido de Proceso Software y Ciclo de Vida: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/02-ProcesoCicloDeVida.pdf>
- Nieto, F. C. (Septiembre de 2011). *Tesis "IMPLEMENTACIÓN DE UN CLUSTER A NIVEL DE APLICACIÓN EN LA EMPRESA CASA MOELLER MARTÍNEZ C.A. UTILIZANDO UNA PLATAFORMA Y HERRAMIENTAS OPEN SOURCE"*. Obtenido de Tesis "IMPLEMENTACIÓN DE UN CLUSTER A NIVEL DE APLICACIÓN EN LA EMPRESA CASA MOELLER MARTÍNEZ C.A. UTILIZANDO UNA PLATAFORMA Y HERRAMIENTAS OPEN SOURCE": <http://bibdigital.epn.edu.ec/bitstream/15000/4086/1/CD-3854.pdf>
- Palchevich, D. R. (Junio de 2008). *Nuevas tecnologías Web 2.0: Hacia una real democratización de la información y el conocimiento*. Obtenido de Nuevas tecnologías Web 2.0: Hacia una real democratización de la información y el conocimiento: <http://eprints.rclis.org/11814/1/Rodriguez-DianaTRABAJOelis.pdf>
- RUIZ, F. R. (s.f.). *WEB 2.0. UN NUEVO ENTORNO DE APRENDIZAJE EN LA RED*. Obtenido de WEB 2.0. UN NUEVO ENTORNO DE APRENDIZAJE EN LA RED: <http://dim.pangea.org/revistaDIM13/Articulos/pacoruz.pdf>
- S. K., & S. J. (Julio de 2013). *La Guía de Scrum*. Obtenido de La Guía Definitiva de Scrum: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Tixe, V. C., & Vivero, A. E. (2012). *Tesis "DESARROLLO DE UNA METODOLOGÍA PARA LA CONSTRUCCIÓN DE UNA APLICACIÓN WEB MEDIANTE LA INTEGRACIÓN DE JSF Y EJB. CASO PRÁCTICO: HOSPITAL PROVINCIAL GENERAL DOCENTE RIOBAMBA"*. Obtenido de Tesis "DESARROLLO DE UNA METODOLOGÍA PARA LA CONSTRUCCIÓN DE UNA APLICACIÓN WEB MEDIANTE LA INTEGRACIÓN DE JSF Y EJB. CASO PRÁCTICO: HOSPITAL PROVINCIAL GENERAL DOCENTE RIOBAMBA": <http://dspace.esPOCH.edu.ec/handle/123456789/2538>
- Valdivia, O. A. (s.f). *Instrumentos de Evaluación*. Obtenido de Instrumentos de Evaluación: http://depa.fquim.unam.mx/dsa/PAIDOS/AC05-Elaboracion_de_Reactivos.pdf
- Velde, H. V. (2014). *APRENDER A PREGUNTAR, PREGUNTAR PARA APRENDER*. Obtenido de APRENDER A PREGUNTAR, PREGUNTAR PARA APRENDER: https://www.upf.edu/cquid/_pdf/saber_preguntar_vandvelde.pdf
- White, S., & Miers, D. (2009). *Guía de Referencia y Modelado BPMN*. Obtenido de Guía de Referencia y Modelado BPMN: http://www.futstrat.com/books/book_images/Guia%20de%20Referencia%20y%20Modelado%20BPMN%20chap1-2.pdf