

**DESARROLLO DE LA NUEVA VERSIÓN DEL SISTEMA DE INFORMACIÓN DE
PREGUNTAS, QUEJAS, RECLAMOS Y SUGERENCIAS PARA LA
UNIVERSIDAD INDUSTRIAL DE SANTANDER**

CHRISTIAN FERNANDO PACHECO JAIMES

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FÍSICO - MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

2010

**DESARROLLO DE LA NUEVA VERSIÓN DEL SISTEMA DE INFORMACIÓN DE
PREGUNTAS, QUEJAS, RECLAMOS Y SUGERENCIAS PARA LA
UNIVERSIDAD INDUSTRIAL DE SANTANDER**

CHRISTIAN FERNANDO PACHECO JAIMES

**Proyecto de Grado presentado como requisito para optar al título de
Ingeniero de Sistemas**

DIRECTOR

Ing. ENRIQUE TORRES LÓPEZ.

CODIRECTOR

Ing. JORGE ELIÉCER VIDAL RODRIGUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FÍSICO - MECÁNICAS

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

2010

DEDICATORIA

A Dios que me ha dado la vida,

*A mis padres que dieron lo mejor de sí mismos para hacerme un ser
humano de bien,*

A mis hermanas que me han acompañado y apoyado en todo momento,

A todos aquellos que han pasado por mi vida y dejado una huella.

Christían Fernando Pacheco Jaimes.

AGRADECIMIENTOS

Un agradecimiento especial a Dios, Padre todo poderoso, que me dio la vida y cada cosa que he podido lograr a lo largo de mi existencia.

A mis padres que han sido el motor fundamental durante este proceso formativo, que se culmina con grandes éxitos para mi vida. Ellos han entregado cada minuto de su vida para que no me falte nada y les agradezco que me hayan dado la oportunidad de ser un profesional.

A mis hermanas que durante toda mi vida han estado al lado mío, y han sido esas compañeras inseparables en mi caminar. Me han guiado, formado y hasta aguantado en muchas etapas de mi vida.

A todos y cada uno de mis amigos que con su presencia iluminan mi camino y lo llenan de alegría y felicidad.

A la División de Servicios de Información de la Universidad Industrial de Santander por la oportunidad de terminar mi formación profesional con esta Entidad.

Un agradecimiento muy especial al Ingeniero Enrique Torres López, por su entrega inmejorable en el acompañamiento de todo éste proceso, por su paciencia y sus consejos que serán de gran importancia en mi desarrollo como profesional.

A la Dirección de Control Interno y Evaluación de Gestión de la Universidad Industrial de Santander, por la confianza depositada en mí para la realización de éste proyecto.

Y en general a la comunidad académica de la Universidad Industrial de Santander por haber hecho de mí un profesional.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	20
PARTE I. FUNDAMENTOS.....	22
1. PRESENTACION DEL PROYECTO.....	22
1.1 DESCRIPCION DEL PROYECTO.	22
1.1.1 OBJETIVO GENERAL.....	22
1.1.2 OBJETIVOS ESPECÍFICOS.....	22
1.2 JUSTIFICACION.....	25
1.2.1 ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA.....	25
1.2.2 IMPACTO.....	26
1.2.3 VIABILIDAD.....	26
2. MARCO TEÓRICO.....	27
2.1 DIRECCION DE CONTROL INTERNO Y EVALUACION DE GESTION.....	28
2.1.1 MISIÓN.....	28
2.1.2 ANTECEDENTES.....	28
2.2 GENERALIDADES SISTEMA DE GESTION DE CALIDAD.	30
2.2.1 POLÍTICA DE CALIDAD DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.....	30
2.2.2 OBJETIVOS DE LA CALIDAD.....	30
2.2.3 PROCESOS DE EVALUACIÓN.....	31
2.2.4 PROCEDIMIENTO DE QUEJAS Y RECLAMOS.....	32
2.3 GENERALIDADES DEL ENTORNO DE DESARROLLO	38
2.3.1 APLICACIONES ORIENTADAS A LA WEB.....	38
2.3.1.1 LA WEB COMO SISTEMA DE INFORMACIÓN.....	38
2.3.1.2. TECNOLOGÍAS WEB.....	40
2.3.1.3 PLATAFORMAS DE DESARROLLO WEB.....	42

2.3.1.4 WEB 2.0.	43
2.3.2 DISEÑOS CON EL ESTÁNDAR UML.	44
2.3.2.1 INTRODUCCIÓN A UML	44
2.3.2.2 ELEMENTOS COMUNES A TODOS LOS DIAGRAMAS.	45
2.3.2.3 DIAGRAMAS DE CASOS DE USO.	47
2.3.2.4 DIAGRAMAS DE SECUENCIA.	49
2.3.2.5 DIAGRAMA DE ACTIVIDADES.....	50
2.3.2.6 DIAGRAMA DE CLASES.	51
2.3.2.7 DIAGRAMA ENTIDAD – RELACIÓN	53
2.3.3 JAVA ENTERPRISE EDITION 5.0 (JEE 5).....	54
2.3.4 ENTORNO DE DESARROLLO.....	55
2.3.4.1 JAVA SERVER FACES (JSF).....	55
2.3.4.2 OBJECT RELATIONAL MAPPING / JPA	56
2.3.4.3 EJB 3.0	56
2.3.5 SERVIDOR DE APLICACIONES – JBOSS.....	57
2.3.6 ENTERPRISE ARCHITECT	57
2.3.7 CONSTRUCCIÓN DE PROTOTIPOS DE SOFTWARE.	58
<u>3. METODOLOGÍA DE DESARROLLO.....</u>	<u>60</u>
3.1 CICLO DE VIDA DEL PROYECTO.....	60
3.1.1 ANÁLISIS DE REQUERIMIENTOS:	60
3.1.2 DISEÑO.....	61
3.1.3 IMPLEMENTACIÓN DE LA APLICACIÓN.	61
3.1.4 PRUEBAS DEL SOFTWARE.	62
3.1.5 AJUSTES.	62
3.2 METODOLOGIA DE DESARROLLO DEL PROYECTO.....	63
3.2.1 MODELO DE CONSTRUCCIÓN DE PROTOTIPOS.....	63
3.2.2 ESTRUCTURA DEL MODELO DE CONSTRUCCIÓN DE PROTOTIPOS.	64
3.2.3 PROCEDIMIENTO PARA LA METODOLOGÍA PLANTEADA.....	65

<u>PARTE II. DESARROLLO DEL SISTEMA.....</u>	<u>67</u>
<u>4. APLICACIÓN DE LA METODOLOGÍA.....</u>	<u>67</u>
4.1 LEVANTAMIENTO DE REQUERIMIENTOS	67
4.1.1 DESCRIPCIÓN GENERAL.....	67
4.1.2 DEFINICIONES Y SIGLAS.	68
4.1.3 FUNCIONES QUE DEBE CUMPLIR EL SISTEMA.	69
4.1.3.1 DEL REGISTRO DE LA SOLICITUD	69
4.1.3.2 ADMINISTRACIÓN	72
4.1.3.3 PERSONAL UAA.	75
4.2 ESTÁNDARES DE LA DIVISIÓN DE SERVICIOS DE INFORMACIÓN.....	76
4.2.1 ASPECTOS GENERALES.....	76
4.2.2 DOCUMENTACIÓN DE LOS DIAGRAMAS DE DISEÑO	80
4.2.3 SINTAXIS DE NOMBRES EN JAVA.....	81
4.2.4 DOCUMENTACIÓN	85
4.2.5 CAPA DE PRESENTACIÓN.....	85
4.3 DIAGRAMAS UML	91
4.3.1 DIAGRAMA DE CASOS DE USO.....	91
4.3.1.1 IDENTIFICACIÓN DE LOS ACTORES	91
4.3.1.2 CASOS DE USO POR ACTOR	92
4.3.2 DIAGRAMAS DE SECUENCIAS	106
4.3.3 DIAGRAMA DE CLASES	110
4.3.4 DIAGRAMA DE DATOS.....	111
4.3.5 DIAGRAMA DE ACTIVIDADES.....	113
4.4 PROTOTIPO INICIAL.....	115
4.4.1 GENERALIDADES DEL PROTOTIPO INICIAL	115
4.4.2 INTERFAZ RESULTADO DEL PROTOTIPO INICIAL.....	116
4.5 PROTOTIPO FINAL.....	120
4.5.1 GENERALIDADES DEL PROTOTIPO FINAL	120

4.5.3 PROYECTO ENMARCADO EN EL ESQUEMA DE SEGURIDAD DE LA UIS.....	131
4.5.3.1 ESTRUCTURA DE LA BASE DE DATOS SOPORTE	131
4.5.3.2 ENTORNO DE NAVEGACIÓN.....	134
4.5.3.3 ENTORNO DE CONTROL DE DATOS.....	134
4.5.3.4 AUDITORÍA	135
4.5.4 ORGANIZACIÓN DE DIRECTORIOS.	135
4.5.5 DOCUMENTACIÓN DE PROGRAMAS FUENTE.....	139
<u>5. CONCLUSIONES.....</u>	<u>150</u>
<u>6. RECOMENDACIONES</u>	<u>152</u>
<u>7. BIBLIOGRAFÍA.....</u>	<u>153</u>

LISTADO DE FIGURAS

Figura 1. Mapa de Procesos UIS.....	32
Figura 2. Diagrama de Flujo Procedimiento de Quejas y Reclamos en la Universidad Industrial de Santander.....	35
Figura 3. Elementos de un Sistema de Información.....	39
Figura 4. Historia de UML.....	45
Figura 5. Ejemplo de una Nota UML.....	46
Figura 6. Ejemplo de una Dependencia UML.....	46
Figura 7. Ejemplo de Diagrama de Caso de Uso.....	48
Figura 8. Ejemplo de Diagrama de Secuencia.....	49
Figura 9. Ejemplo Diagrama de Actividades.....	50
Figura 10. Ejemplo de Diagrama de Clases.....	52
Figura 11. Ejemplo Diagrama Entidad-Relación.....	53
Figura 12. Modelo de Construcción de Prototipos.....	64
Figura 13. Estructura del Modelo de Construcción de Prototipos.....	64
Figura 14. Plantilla Principal División de Servicios de Información.....	85
Figura 15. Plantilla de Contenido División de Servicios de Información.....	86
Figura 16. Plantilla de Contenido con Formulario. División de Servicios de Información.....	87
Figura 17. Casos de Uso Ciudadano.....	92
Figura 18. Caso de Uso: Registrar Solicitud.....	93
Figura 19. Casos de Uso asociados con Personal Control Interno Permisos totales.....	95
Figura 20. Caso de Uso. Gestionar tabla Soporte.....	96
Figura 21. Caso de Uso Leer Solicitud.....	98
Figura 22. Caso de Uso. Evaluar Respuesta UAA.....	100
Figura 23. Caso de Uso: Ver Informes Estadísticos. Personal CI-UIS AR.....	102

Figura 24. Casos de Uso Personal UAA.....	104
Figura 25. Diagrama de Secuencia para el caso de uso. Registrar Solicitud.	107
Figura 26. Diagrama de Secuencia. Caso de uso Leer Solicitud CI.	108
Figura 27. Diagrama de Secuencia para el caso de Uso. Leer Solicitud UAA.	109
Figura 28. Diagrama de Clases sistema PQRS.....	110
Figura 29. Diagrama de Datos del Sistema de PQRS.	111
Figura 30. Diagrama de Actividades Registro de la Solicitud.	113
Figura 31. Diagrama Actividades Ver Solicitud.....	114
Figura 32. Vista de Registro de Solicitud. Prototipo Inicial.....	116
Figura 33. Vista entrada control Interno. Prototipo Inicial	117
Figura 34. Vista de la Solicitud por parte de Control Interno. Prototipo Inicial.	118
Figura 35. Vista Respuesta enviada por UAA. Prototipo Inicial.	119
Figura 36. Vista Registro Solicitud. Prototipo Final.....	121
Figura 37. Vista de la confirmación del recibo de la Solicitud. Prototipo Final.	122
Figura 38. Vista Formulario Rastreo Solicitud. Prototipo Final.....	123
Figura 39. Vista Rastreo de la Solicitud. Prototipo Final	123
Figura 40. Vista Solicitudes Nuevas Control Interno. Prototipo Final.....	124
Figura 41. Vista Detalle de la Solicitud. Prototipo Final.	125
Figura 42. Vista Lista Solicitudes con Respuesta de UAA. Prototipo Final.....	126
Figura 43. Vista Respuesta enviada por la UAA. Prototipo Final.....	127
Figura 44. Vista consulta de solicitudes. Prototipo Final.....	128
Figura 45. Vista Lista de Solicitudes UAA. Prototipo Final.....	129
Figura 46. Vista Respuesta de la Solicitud por la UAA. Prototipo Final.	130

LISTADO DE TABLAS

Tabla 1. Diagrama Flujo Procedimiento Quejas y Reclamos UIS	37
Tabla 2. Descripción Caso de uso: Registrar Solicitud	94
Tabla 3. Descripción de Caso de Uso. Gestionar Tablas Soporte	97
Tabla 4. Descripción Caso de Uso: Leer Solicitud	99
Tabla 5. Descripción Caso de Uso: Evaluar Respuesta UAA.	101
Tabla 6. Descripción Caso de Uso. Ver Informes Estadísticos.....	103
Tabla 7. Caso de Uso. Leer Solicitud UAA.	105

LISTADO DE ANEXOS

Anexo A. Formato de recepcion de quejas y reclamos.....	154
Anexo B. Formato de recepcion de Sugerencias.....	157
Anexo C. CD Anexo al Documento.	

RESUMEN

1. **TITULO:** DESARROLLO DE LA NUEVA VERSION DEL SISTEMA DE INFORMACIÓN DE PREGUNTAS, QUEJAS, RECLAMOS Y SUGERENCIAS PARA LA UNIVERSIDAD INDUSTRIAL DE SANTANDER. *

2. **AUTOR:** PACHECO JAIMES CHRISTIAN FERNANDO**

3. **PALABRAS CLAVE:** Sistema de Información Web, Preguntas, Quejas, Reclamos, Sugerencias, Veeduría Ciudadana, Aplicaciones Cliente Servidor, Java, Modelo de Construcción de Prototipos.

4. DESCRIPCION:

Ante las exigencias del sistema de gestión de calidad de la Universidad Industrial de Santander y de las normas ISO 9001:2008 y NTCGP1000:2009 en las cuales se encuentra certificada la Universidad, se hace necesario la implementación del sistema de veeduría ciudadana, el cual ha funcionado hasta la actualidad, pero este sistema ha sufrido diversos cambios a medida que ha pasado el tiempo y presenta problemas de eficiencia, facilidad de uso y poco adaptable a los cambios.

Es por esto que se hace necesario la realización de un proyecto cuyo objetivo general sea: "Realizar el análisis de requerimientos, diseño, implementación, puesta en marcha y documentación del nuevo sistema de Información para la recepción y trámite de las quejas y reclamos, preguntas y sugerencias que facilite las labores de control interno y la evaluación de gestión de la Universidad Industrial de Santander."

Los requerimientos fueron definidos por el cliente, que en éste caso es la Dirección de Control Interno y Evaluación de Gestión de la Universidad Industrial de Santander, con quienes se formuló el diseño y la posterior evaluación del producto final entregado, desarrollado el modelo de construcción de prototipos como metodología.

Este nuevo producto se realizo con los estándares de calidad y eficiencia de la División de Servicios de Información de la Universidad Industrial de Santander en los desarrollos de los nuevos sistemas de información.

* Trabajo de Investigación

** Facultad de Ingenierías Físico – Mecánicas, Escuela de Ingeniería de Sistemas e Informática. Director: Ingeniero Enrique Torres López. Codirector: Ingeniero Jorge E. Vidal Rodríguez.

SUMMARY

1. **TITLE:** DEVELOPMENT OF THE NEW VERSION OF THE QUESTION, COMPLAINT, AND SUGGESTION INFORMATION SYSTEM AT THE INDUSTRIAL UNIVERSITY OF SANTANDER. *

2. **AUTHOR:** PACHECO JAIMES CHRISTIAN FERNANDO**

3. **KEYWORDS:** Web Information System, Questions, Complaints, Suggestions, Citizen Review Board, Client Serving Applications, Java, Prototype Constructions Model.

4. DESCRIPTION:

Before the demands of the quality management system of the Industrial University of Santander and the norms in which the university finds itself certified, it becomes necessary to implement the quality management system, which has worked until now, but has also suffered various problems over time, and presents problems of efficiency, facility of use, and adaptability to changes.

Therefore, it has become necessary to bring to life a project with the general objective of realizing the analysis of requirements, design, implementation, start up, and documentation of the new information system for the reception and processing of complaints, questions, and suggestions, that facilitates the labor of internal control and the evaluation of management at the Industrial University of Santander.

The requirements were defined by the customer, which in this case is the Internal Control Management and Evaluation of Management, Industrial University of Santander, with whom he made the design and subsequent evaluation of the final product delivered, developed the building model prototyping as a methodology

This new product came about with the standards of quality and efficiency of the Division of Information Services of the Industrial University of Santander during the ongoing development of the new information systems.

* Project

** Physique-Mechanics Sciences Department, Computer Science Engineering, Engineer Enrique Torres Lopez. Engineer Jorge E. Vidal Rodríguez.

TERMINOS Y DEFINICIONES

DCIEG : Dirección de Control Interno y Evaluación de Gestión.

Ciudadano : Persona de la comunidad que no es funcionario ni proveedor ni estudiante de la universidad.

Control Interno: Personal de la Dirección de Control Interno y Evaluación de Gestión.

Estudiante UIS : Persona matriculada en algún programa académico de la UIS.

Funcionario UIS : Personal vinculado a la Universidad mediante contrato de trabajo.

Proveedor UIS : Persona vinculada a la Universidad mediante un contrato de suministro de productos o servicios de manera indirecta.

PQRS : Pregunta, Queja, Reclamo o Sugerencia.

Solicitante : Ciudadano, Funcionario UIS, Estudiante UIS o Proveedor UIS que formula una PQRS.

UAA : Unidad Académica y Administrativa de la Universidad Industrial de Santander.

Queja : Es el medio por el cual el beneficiario pone de manifiesto su inconformidad con la actuación de determinado funcionario o con la forma y condiciones en que se preste o no un servicio.

Reclamo : Es una solicitud del beneficiario con el objeto de que se revise una actuación administrativa y/o académica, motivo de su inconformidad, y se tome una decisión final justa y equitativa

Sugerencia : Es una insinuación o formulación de ideas tendientes al mejoramiento de un servicio o de la misma Institución.

Pregunta : Enunciado interrogativo que se emite con intención de obtener algún tipo de información

Interfaz : La idea fundamental en el concepto de interfaz, es el de mediación. La interfaz es lo que “media”, lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador. Esto implica, además, que se trata de un sistema de traducción, ya que los dos se comunican con lenguajes diferentes: verbo-icónico en el caso del hombre y binario en el caso de la máquina.

Java : Lenguaje de Programación que se caracteriza por tener una arquitectura que permite que el código escrito funcione en multitud de sistemas operativos sin ser modificado.

Método : Es una operación que define como se comporta un objeto.

Módulo : Se utiliza como sinónimo de Subsistema.

Sistema de Información: Aplicación comercial para el computador. Está constituida por la base de datos, los programas de aplicación, los procedimientos manuales y automatizados, e incluye los sistemas computacionales que realizan procesamiento.

Clase : Una clase de objetos describe un grupo de objetos con propiedades similares, con relaciones comunes entre otros y con una semántica común.

INTRODUCCIÓN

En la actualidad, el mundo moderno y digitalizado, obliga a las organizaciones, ya sean de carácter público o privado a ofrecer facilidades de comunicación entre las entidades y los diferentes usuarios que interactúan con ellas.

Tomando en cuenta esto y que la constitución y la ley colombiana exigen que sus entidades cuenten con un mecanismo de comunicación, donde los usuarios puedan expresar sus preguntas, quejas, reclamos y sugerencias siendo éstas atendidas de manera oportuna¹, la Universidad Industrial de Santander a través de la Dirección de Control Interno y Evaluación de Gestión ha implementado y certificado a la universidad en las normas del sistema de gestión de calidad ² que expresan de manera clara la necesidad de establecer un mecanismo de comunicación con los usuarios y atender sus solicitudes.

Por todo lo anterior, la universidad dando cumplimiento a estas directrices normativas y tecnológicas, implantó el sistema de veeduría ciudadana, el cual se encuentra funcionando actualmente.

Sin embargo, el sistema actual no cumple con los requisitos necesarios de efectividad y facilidad que le debe ofrecer al usuario final, no es lo suficientemente flexible, y se ha ido adaptando en varias ocasiones a los nuevos requerimientos que se han detectado, o bien para corregir fallas o para contemplar nuevos

¹ Constitución Política de Colombia de 1991, Art. 23, 74 y 209
Código Contencioso Administrativo, Derecho de Petición, Art. 4 – 42.
Ley 30 de 1992, Régimen de Educación Superior.
Ley 87 de 1993, Normas de Control Interno.
Ley 190 de 1995, Estatuto Anticorrupción, Art. 49 – 52 al 55.
Ley 850 de 2003, Reglamenta Veedurías Ciudadanas.
Ley 962 de 2005, Anti trámites.

² ISO 9001:2008 , NTCGP1000:2009

escenarios, lo que ha ocasionado que no sean viables más modificaciones, haciendo necesario un proceso completo de reingeniería, que involucrara el escenario actual, tiene en cuenta escenarios futuros, detectando los posibles cambios y la flexibilidad ante ellos.

Este proyecto pretende diseñar, implementar e implantar una nueva versión del sistema de veeduría ciudadana, cambiando el esquema hacia las diferentes solicitudes que puede hacer un usuario: Preguntas, quejas, reclamos y sugerencias, y darle trámite de manera efectiva teniendo como premisa que cada solicitud es importante y debe resolverse de manera efectiva y en el menor tiempo posible.

Para la realización de éste proyecto se contó con el apoyo de la División de Servicios de Información (DSI) de la Universidad Industrial de Santander, entidad encargada de la administración y desarrollo de la tecnología de información, recibiendo todo el soporte y asesoría necesarios para ello, utilizando los estándares definidos por la DSI para el desarrollo de los nuevos sistemas de información y herramientas tales como Informix, Java EE 5, J-boss seam y diferentes tipos de software ya licenciados por la UIS.

En el transcurso del desarrollo del proyecto se realizaron diversas reuniones con la unidad directamente beneficiada, Dirección de Control Interno y Evaluación de Gestión para refinar prototipos y producto final, proceso que se describirá en el presente documento.

PARTE I. FUNDAMENTOS

1. PRESENTACION DEL PROYECTO

1.1 DESCRIPCION DEL PROYECTO.

1.1.1 Objetivo General. Realizar el análisis de requerimientos, diseño, implementación, puesta en marcha y documentación del nuevo sistema de Información para la recepción y trámite de las quejas y reclamos, preguntas y sugerencias que facilite las labores de control interno y la evaluación de gestión de la Universidad Industrial de Santander.

1.1.2 Objetivos Específicos. El sistema de información de preguntas, quejas, reclamos y sugerencias tiene como objetivos específicos:

- Efectuar el proceso de recolección y análisis de requerimientos propios del sistema, mediante las siguientes alternativas:
 - Realizando reuniones periódicas con los funcionarios de la Dirección de Control Interno y Evaluación de Gestión.
 - Elaboración de documentos escritos en prosa, y gráficos que permitan establecer con claridad las necesidades propias del sistema.
 - Verificando los documentos donde se plantea la necesidad y la estructura del proceso de recepción de preguntas, quejas, reclamos y sugerencias bajo el esquema de gestión de calidad en el cual ha sido certificada la universidad.

- Llevar a cabo el diseño que permita definir plenamente el sistema bajo el estándar del lenguaje de modelado unificado(UML), a través de:
 - El diseño del comportamiento y relaciones entre los casos de uso, mediante el diagrama de casos de uso.
 - El diseño de los flujos de trabajo paso a paso y de operación, mediante el diagrama de Actividades.
 - El diseño de la estructura del sistema, especificando las clases, los atributos y las relaciones entre ellos, mediante el diagrama de clases.
 - El diseño de la base de datos, mediante el diagrama Entidad-Relación.
 - El diseño de la interacción entre los objetos del sistema, mediante el diagrama de Secuencia.

- Desarrollar e implementar el sistema de información de preguntas, quejas, reclamos y sugerencias teniendo en cuenta las siguientes características generales:
 - Será desarrollado en entorno web, para facilitar su interacción con la comunidad universitaria y con la ciudadanía en general.
 - Se desarrollará en lenguaje de programación Java 5 y manejador de base de datos Informix.
 - El sistema recibirá las solicitudes de los usuarios, clasificándolas en: Pregunta, Queja, Reclamo o Sugerencia y las enviará a Dirección de Control Interno y Evaluación de Gestión.
 - La Dirección de Control Interno y Evaluación de Gestión, por medio del sistema de información, hará el traslado de la solicitud a la unidad académico-administrativa que le corresponda dar respuesta.
 - La respuesta a la solicitud por parte de la unidad académico administrativa será enviada nuevamente a la Dirección de Control Interno y Evaluación de Gestión, unidad que decidirá si es pertinente, si lo es se publicará dando respuesta al usuario solicitante, si no lo

- es, se devolverá con las recomendaciones necesarias para que la unidad académico administrativa responsable revise la respuesta y realice los ajustes necesarios.
- Todo el trámite se debe hacer por medio del sistema, facilitando la comunicación entre las dependencias de la universidad.
 - Se guardará un registro de todas las solicitudes hechas y las correspondientes respuestas emitidas por las dependencias, para que puedan ser consultadas por los funcionarios de la Dirección de Control Interno y Evaluación de Gestión, y si es pertinente por toda la comunidad en general.
 - Se elaborarán informes estadísticos de las solicitudes realizadas, por rango de fechas, tipo de solicitud, unidades académico administrativas responsables, tiempo de respuesta, entre otros.
- Efectuar las pruebas necesarias, para garantizar el correcto funcionamiento del sistema, teniendo en cuenta que:
 - Las pruebas deben hacerse permanentemente durante el desarrollo de la aplicación.
 - Dentro de los posibles errores están: Lógica en la Programación, malinterpretación de los requerimientos, diseños defectuosos; lo que implica un proceso de refinamiento.
 - Realizar los procesos de refinamiento y puesta en marcha, de tal manera que:
 - El usuario final quede conforme con el resultado obtenido, después de un proceso iterativo de análisis, revisión y ajuste por parte de él mismo.
 - Las necesidades del usuario se mantengan dentro del alcance y definiciones establecidas al comienzo del proyecto.
 - Documentar y capacitar al usuario final y el personal técnico encargado del mantenimiento del sistema de tal manera que :

- El funcionamiento del sistema quede claro para el usuario final, y no genere inconvenientes su uso por parte del personal de la Dirección de Control Interno y Evaluación de Gestión, ni para la Ciudadanía en General.
- Las especificaciones técnicas queden claras, para facilitar las labores de mantenimiento cuando éstas sean necesarias en el sistema.
- Quede registro de todo lo efectuado durante el desarrollo del sistema de información, para que futuros procesos de reingeniería puedan ser efectuados sin mayores traumatismos.

1.2 JUSTIFICACION.

1.2.1 Antecedentes y descripción del problema.

Debido a las exigencias de la certificación del Sistema de Gestión de Calidad para la Universidad Industrial de Santander relacionadas con las actividades de control interno y de evaluación de la gestión, se hizo necesario la creación independiente al módulo de veeduría ciudadana, del módulo de quejas y reclamos, preguntas y sugerencias, con un enfoque de mayor cobertura, con mejores facilidades de seguimiento, con un esquema de alta seguridad, un módulo de auditoría que permita el traceo de las actividades realizadas y consultas que permitan el suministro de información a los estamentos interesados en conocerla.

Esta solución reemplazará al actual sistema, el cual, según la Dirección de Control Interno y Evaluación de Gestión, no fue concebida para satisfacer los requerimientos exigidos por las normas de calidad bajo las cuales se encuentra certificada la Universidad Industrial de Santander (ISO 9001:2008 Y NTCGP 1000:2009), puesto que su desarrollo fue anterior a éstos procesos.

Lo anterior explica la razón por la cual se hace necesaria la implementación de un sistema moderno que cumpla con los estándares del sistema de gestión de calidad siendo éste más sencillo y amigable para el usuario, tanto de Dirección de Control Interno y Evaluación de Gestión como de la ciudadanía en general.

La idea general de éste proyecto es hacer un sistema a través de un proceso de reingeniería con el cual se genere un producto que elimine las limitaciones actuales y multiplique el rendimiento efectivo del sistema.

1.2.2 Impacto.

Este sistema es un medio que le permite a la comunidad universitaria y a la ciudadanía en general, expresar su opinión acerca del cumplimiento de los objetivos institucionales en virtud de los principios de eficiencia, transparencia y de los derechos de participación e información.

Este sistema será usado por todo tipo de usuarios, tanto ciudadanos externos a la universidad, como personal perteneciente a la comunidad universitaria. Su promedio actual de uso oscila alrededor de 300 solicitudes trimestrales, y éstas aumentan en épocas especiales de la vida universitaria como: matrículas, grados, horarios, y eventos no programados como paros; las cuales deben ser procesadas y administradas por la Dirección de Control interno y Evaluación de Gestión y las diferentes unidades académico-administrativas de la universidad.

1.2.3 Viabilidad.

Para la realización de éste proyecto se cuenta con herramientas de libre distribución como el Jboss Developer Studio 3.0 G.A, el servidor de aplicaciones Jboss-seam productos patrocinados por la compañía Red Hat Software Inc, la cual tiene un contrato de soporte con la Universidad Industrial de Santander. Esto conlleva a muchas facilidades durante su desarrollo, éstas herramientas evolucionan en términos de calidad y seguridad, puesto que hay miles de personas en todo el mundo trabajando en ellas a través de internet, con la formación de grupos de discusión que brindan información y capacitación a los desarrolladores a través del intercambio de conocimiento acerca de los distintos productos,

Aquellas herramientas que no son de libre distribución ya han sido adquiridos y licenciados por la Universidad Industrial de Santander.

En cuanto a hardware se cuenta con instalaciones adecuadas, con los equipos requeridos y el soporte tecnológico necesario para el desarrollo del mismo, además se tiene la supervisión por parte del director del proyecto y la colaboración del equipo de trabajo de la División de Servicios de información, organismo encargado de la generación y el uso de soluciones informáticas de la más alta calidad técnica que facilitan el proceso de modernización institucional.

2. MARCO TEÓRICO

2.1 DIRECCION DE CONTROL INTERNO Y EVALUACION DE GESTION.

2.1.1 Misión.

La Dirección de Control Interno y Evaluación de Gestión tiene como propósito la evaluación, seguimiento y monitoreo permanente al grado de avance y desarrollo del Sistema de Control Interno en cada uno de sus componentes y del Sistema de Gestión de la Calidad, implementados en la Institución, conforme a los planes de acción que se establezcan para ello, en búsqueda del autocontrol, la autorregulación, y la autogestión de cada una de las Unidades que conforman la universidad, todo esto liderado por un equipo de auditores internos competentes y calificados para realizar el acompañamiento requerido.

2.1.2 Antecedentes.

2.1.2.1 Control Interno y Fiscal.

Con la nueva Constitución Política de 1991 se puso de manifiesto la importancia del control como mecanismo que permite evaluar la administración pública de todas las entidades e individuos involucrados con el manejo del dinero y bienes del estado.

Se señalan en ella los principios que son la igualdad, eficacia, economía y publicidad y valoración de los costos ambientales utilizando la descentralización, delegación y desconcentración de las funciones, ordenando además que la administración pública, en todos sus deberes tendrá un control interno que se ejercerá en los términos que señale la ley.

Ya con la ley marco del sistema de control interno (ley 87/93) se determinan las normas para ejercer el control interno en las entidades y organismos estatales, dando mayor soporte jurídico al establecimiento de un adecuado sistema de control interno que garantice la gestión de la administración y de todos los individuos involucrados para lograr el manejo eficiente, eficaz y económico de los recursos.

2.1.2.2 Estructura Organizativa.

Mediante Acuerdo del Consejo Superior No.070 de Noviembre 28 de 2005, fue creada la Oficina de Control Interno y Evaluación de Gestión, haciendo énfasis en los criterios de descentralización, participación ciudadana y autonomía teniendo como preferencia la nueva Constitución Nacional, las leyes y decretos reglamentarios del servicio público de la educación superior; la ley de ciencia y tecnología; el programa nacional para la modernización del estado; y el estatuto general de la Universidad.

La nueva estructura refleja una Oficina de Evaluación y Control de Gestión, que cumple las funciones de control interno, ubicada en el más alto nivel jerárquico de la institución.

2.1.2.3 Sistema de Control Interno.

El Sistema de Control Interno, es el sistema integrado por el esquema de organización y el conjunto de planes, métodos, principios, normas, procedimientos

y mecanismos de verificación y evaluación adoptados por la institución, con el fin de procurar que todas las actividades, operaciones y actuaciones, así como la administración de la información y los recursos se realicen de acuerdo con las normas constitucionales y legales dentro de las políticas trazadas por la dirección y en la atención a las metas u objetivos fijados por la Alta Dirección de la UIS.

2.2 GENERALIDADES SISTEMA DE GESTION DE CALIDAD.

2.2.1 Política de calidad de la Universidad Industrial de Santander.³

La Universidad Industrial de Santander, como ente público y autónomo, está comprometida con la eficacia, la eficiencia, la efectividad y el mejoramiento continuo de sus procesos, en consonancia con el Proyecto Institucional, para lograr la satisfacción de las necesidades de sus beneficiarios y fortalecer el desarrollo de la Educación Superior. Promueve para ello una cultura de calidad basada en el autocontrol, la oportunidad en el servicio y la asertividad en la comunicación de todas las acciones universitarias.

2.2.2 Objetivos de la Calidad.⁴

1. Garantizar, en forma adecuada y pertinente, el apoyo al desarrollo de los procesos misionales de la UIS.
2. Gestionar y administrar la capacidad disponible de la Universidad para maximizar los resultados de los procesos misionales.
3. Consolidar una cultura de resolución de problemas presentes y potenciales en los procesos de la Universidad.

³ Manual de Calidad, Universidad Industrial de Santander. Código MDI.01 Versión 6, página 11

⁴ Manual de Calidad, Universidad Industrial de Santander. Código MDI.01 Versión 6, página 11

4. Lograr en los beneficiarios una percepción altamente favorable de los productos y servicios ofrecidos por la Universidad.
5. Consolidar en los servidores de la Universidad la apropiación y empoderamiento de sus actividades para alcanzar los objetivos misionales.
6. Dar a los beneficiarios productos y servicios en los tiempos requeridos, acorde a sus necesidades.
7. Proporcionar a los beneficiarios información veraz, clara y respetuosa, que facilite su interacción con la Institución.

2.2.3 Procesos de Evaluación.

En el marco de procesos que hacen parte del sistema de gestión de la calidad de la Universidad Industrial de Santander (Ver Figura. Mapa de Procesos UIS) se encuentran los procesos de evaluación, que son el conjunto de procesos relacionados directamente con las actividades de control y evaluación del desempeño del Sistema de Gestión de la Calidad, así como con las actividades destinadas a evaluar la Calidad Académica de la Institución. Dentro de estos procesos se encuentran:

- CA: Gestión de la Calidad Académica
- SE: Seguimiento Institucional

El segundo, SE: Seguimiento Institucional, involucra dentro de sus procedimientos todos aquellos que tienen que ver con la recepción y trámite de las quejas, reclamos y sugerencias, generando el procedimiento formal establecido institucionalmente para darles trámite, procedimiento que mencionará en el apartado siguiente.



Figura 1. Mapa de Procesos UIS.⁵

2.2.4 Procedimiento de Quejas y Reclamos⁶

Enmarcado dentro de los procesos de seguimiento institucional, el procedimiento de quejas y reclamos tiene como objetivo:

Gestionar las quejas y reclamos de manera oportuna, confidencial y objetiva, con el fin de garantizar la mejora en las actividades desarrolladas de acuerdo con la normatividad vigente.

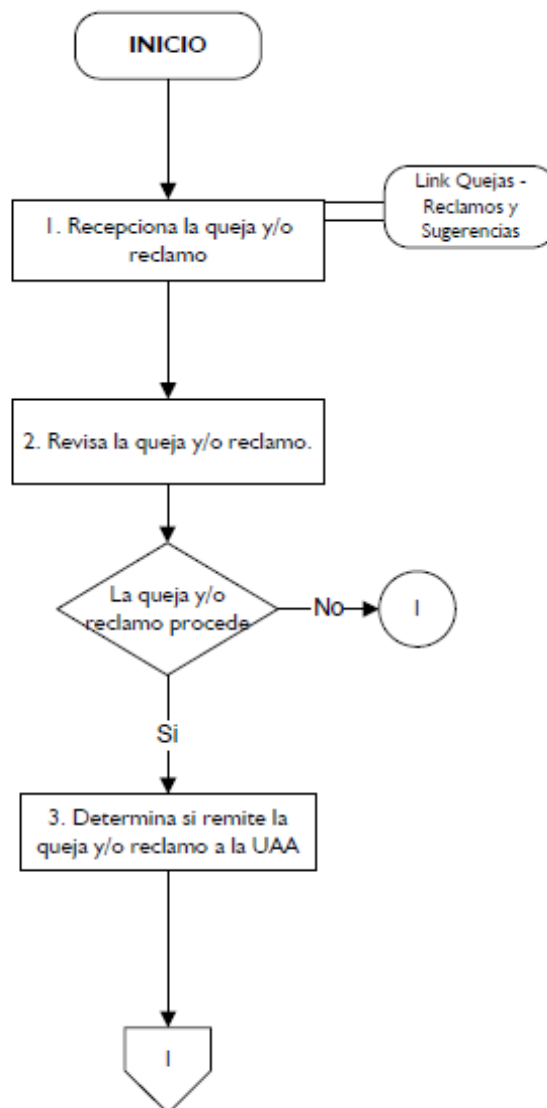
Este procedimiento que inicia desde la recepción de la queja y/o reclamo hasta el cierre de la misma, aplica a toda la comunidad en general.

⁵ Manual de Calidad, Universidad Industrial de Santander. Código MDI.01 Versión 6, página 16

⁶ Proceso de Seguimiento Institucional, Universidad Industrial de Santander. Código PSE.04 Versión 3.

A continuación se describirá mediante diagrama de flujo el proceso vigente aprobado mediante resolución de rectoría número 1859 del 4 de diciembre de 2007, siendo éste actualizado en su tercera versión el 15 de abril de 2009.

Diagrama de Flujo procedimiento de Quejas y Reclamos en la Universidad Industrial de Santander



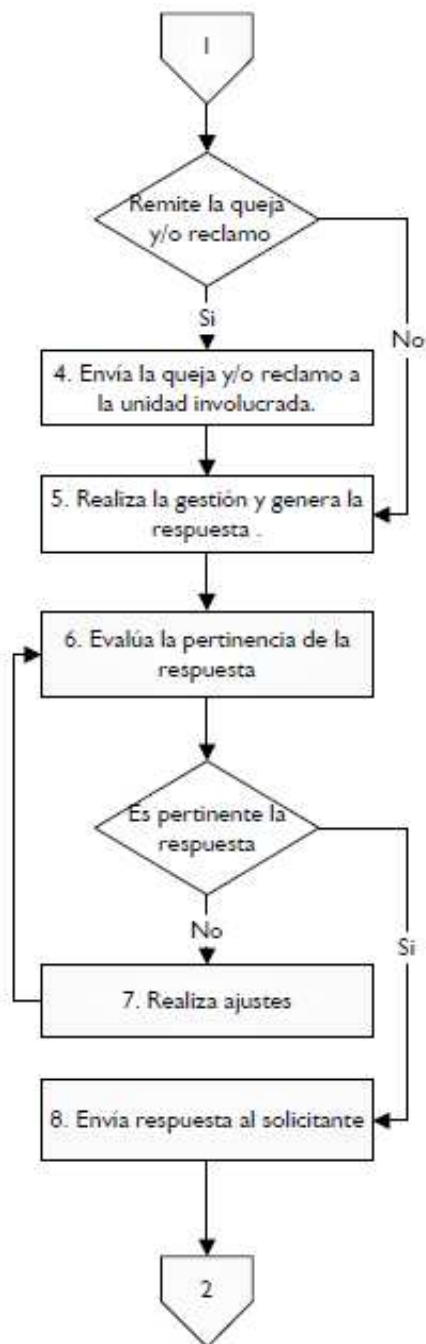




Figura 2. Diagrama de Flujo Procedimiento de Quejas y Reclamos en la Universidad Industrial de Santander

Descripción del Diagrama de Flujo.

Definiciones y Abreviaturas.

Queja y/o Reclamo: Expresión de la insatisfacción hecha a la organización, relativa a sus productos y/o servicios.

U.A.A: Unidad Académico Administrativa

Sugerencia: Es una propuesta que se presenta para incidir o mejorar un proceso cuyo objeto está relacionado con la prestación de un servicio o el cumplimiento de una función pública.

DESCRIPCION DIAGRAMA DE FLUJO		
PASO	DESCRIPCION	RESPONSABLES
1	Recepción de la queja y/o reclamo. Para interponer la queja el solicitante ingresa a la Página Web Institucional, (Link Quejas - Reclamos y Sugerencias y el Sistema le confirmará el recibido de la queja y/o reclamo e informa el número consecutivo asignado. En caso que la queja sea recibida telefónica o personalmente, debe diligenciar el formato FSE.06 (Anexo A y B) y se ingresa al Sistema	Director Control Interno y Evaluación de Gestión
2	Revisa la queja y/o reclamo y determina si esta procede. En caso en que la queja y/o reclamo no proceda, informa al solicitante y termina el proceso.	Director Control Interno y Evaluación de Gestión
	Determina si remite la queja y/o reclamo a la U.A.A involucrada.	Director Control Interno y Evaluación de Gestión
4	Envía la queja y/o reclamo a la U.A.A involucrada	Director Control Interno y Evaluación de Gestión
5	Realiza la gestión y genera la respuesta a la queja y/o reclamo (Plazo máx. 5 días hábiles). Si la situación presentada lo requiere, se debe realizar un análisis de las causas que generaron la queja y/o reclamo y establecer un Plan de Acción para evitar que vuelva a ocurrir.	Director U.A.A.
6	Evalúa la pertinencia de la respuesta y realiza las observaciones correspondientes.	Director Control Interno y Evaluación de Gestión

DESCRIPCION DIAGRAMA DE FLUJO (continuación)		
PASO	DESCRIPCION	RESPONSABLES
7	En caso que la respuesta no sea pertinente, realiza los ajustes necesarios y vuelve al numeral 6.	Director U.A.A.
8	Envía respuesta al solicitante	Director Control Interno y Evaluación de Gestión
9	En caso de existir Plan de Acción para dar solución a la queja y/o reclamo, se realiza seguimiento y verificación a las acciones que hayan propuesto las unidades.	Director Control Interno y Evaluación de Gestión
10	Cierra la queja	Director Control Interno y Evaluación de Gestión
11	Presenta el Informe de Estado de las quejas y/o reclamos a la Alta Dirección	Director Control Interno y Evaluación de Gestión

Tabla 1. Diagrama Flujo Procedimiento Quejas y Reclamos UIS

2.3 GENERALIDADES DEL ENTORNO DE DESARROLLO

2.3.1 Aplicaciones orientadas a la Web

En los primeros días de la Web, los sitios Web consistían de páginas estáticas, permitiendo una interacción limitada con el usuario. Al comienzo de los años 90, estas limitaciones fueron superadas cuando los servidores Web fueron reemplazados para permitir comunicaciones a través del desarrollo de fragmentos de código que eran ejecutados del lado del servidor. A partir de entonces las aplicaciones dejaron de ser estáticas y solamente editadas por aquellos “gurús” del HTML y se permitieron a usuarios normales interactuar con las aplicaciones por primera vez.

Este fue un paso fundamental para llegar a la Web que hoy en día conocemos. Sin la interacción no existiría el comercio electrónico, el Web-mail , Internet-banking, blogs, comunidades online.

2.3.1.1 La Web como Sistema de Información.⁷

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo de la Web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet. Los sitios Web, como expresión de sistemas de información, deben poseer los siguientes componentes:

⁷ Rodríguez Perojo Keilyn, Ronda León Rodrigo. El Web como Sistema de Información. http://bvs.sld.cu/revistas/aci/vol14_1_06/aci08106.htm

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.

Se pudiese definir sistema de información como el conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías:

- Personas
- Datos
- Actividades o técnicas de trabajo.
- Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos (incluyendo procesos manuales y automáticos) dando lugar a información más elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización en función de sus objetivos.

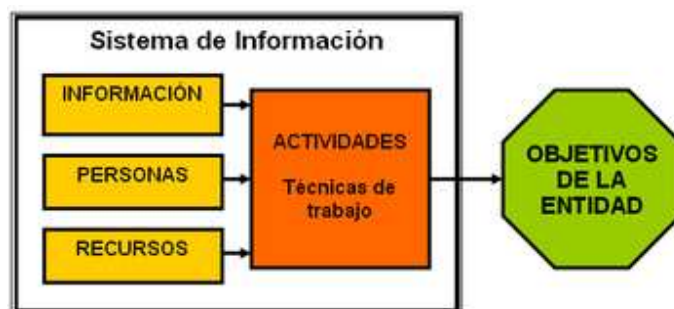


Figura 3. Elementos de un Sistema de Información⁸

⁸ Fuente: http://es.wikipedia.org/wiki/Archivo:Esquema_sistema_de_informacion.png

Normalmente el término es usado de manera errónea como sinónimo de sistema de información informático, en parte porque en la mayoría de los casos los recursos materiales de un sistema de información están constituidos casi en su totalidad por sistemas informáticos, pero siendo estrictos, un sistema de información no tiene por qué disponer de dichos recursos (aunque en la práctica esto no suele ocurrir). Se podría decir entonces que los sistemas de información informáticos son una subclase o un subconjunto de los sistemas de información en general⁹.

Actualmente, los sistemas de información se encuentran al alcance de las grandes masas de usuarios por medio de Internet; así se crean las bases de un nuevo modelo, en el que los usuarios interactúan directamente con los sistemas de información para satisfacer sus necesidades de información.

2.3.1.2. Tecnologías Web.¹⁰

Inicialmente, era difícil la construcción de aplicaciones sofisticadas. La primera generación de aplicaciones Web era primitiva, en general basada en formularios con información y aplicaciones de búsqueda. Incluso estas aplicaciones básicas requerían de un alto seniority para su construcción.

A través del tiempo, el conocimiento necesario para construir aplicaciones ha sido reducido. Hoy en día, es relativamente sencillo construir aplicaciones sofisticadas utilizando las modernas plataformas y lenguajes, como pueden ser PHP, .NET o Java.

⁹ Tomado de: http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n

¹⁰ López Gustavo, Blanco Ignacio. Tesis en Ingeniería Informática.
<http://materias.fi.uba.ar/7500/blanco-tesisingenieriainformatica.pdf>

Primera generación – CGI

Common Gateway Interface (CGI) fue la tecnología reinante desde aproximadamente 1993 hasta fines de los '90 cuando los lenguajes de scripting comenzaron a ganar importancia.

CGI trabaja encapsulando la información provista por el usuario en variables de ambiente. Estas luego son accedidas por scripts o programas desarrollados comúnmente en Perl o C. Estos programas procesan la información provista por los usuarios, y luego envían código HTML con la información procesada a la salida estándar, que a su vez es capturada por el servidor Web y pasada al usuario.

Scripting.

La falta de manejos de sesiones y control de autorización por parte de CGI impidió el desarrollo de aplicaciones Web comerciales con esa tecnología.

Los desarrolladores Web comenzaron entonces a utilizar lenguajes de script, como JavaScript o PHP para resolver esos problemas. Básicamente los lenguajes de script son ejecutados en el servidor Web y como son no compilados son desarrollados e implementados más fácilmente.

Los lenguajes de script tienen algunas desventajas:

- La mayoría de los lenguajes no son tipados y no promueven buenas prácticas de programación.
- Son más lentos en comparación con los lenguajes compilados (a veces hasta 100 veces más lentos).
- Es difícil (no imposible) escribir aplicaciones de múltiples capas porque en general las capas de presentación, aplicación y datos residen en la misma máquina, limitando de esta forma la escalabilidad y seguridad.

- La mayoría no soporta nativamente métodos remotos o llamadas a Web services, lo que hace difícil la comunicación entre servidores de aplicación y con Web services externos.

De cualquier manera a pesar de las desventajas aplicaciones grandes y frecuentemente accedidas han sido desarrolladas utilizando lenguajes de script, como eGroupWare (egroupware.org), que está escrita en PHP. Además muchas aplicaciones de Internet banking han sido desarrolladas en ASP.

Los lenguajes de script incluyen, ASP, Perl, Cold Fusion y PHP. De cualquier manera, muchos de esos podrían ser considerados como lenguajes interpretados híbridos, en particular las últimas versiones de PHP y Cold Fusion.

2.3.1.3 Plataformas de desarrollo web.

Una vez que los lenguajes de script alcanzaron los límites de performance y escalabilidad, los proveedores más grandes evolucionaron hacia la plataforma de Sun J2EE y cuya evolución se encuentra en JEE6 y la de Microsoft .NET.

J2EE

- Utiliza el lenguaje Java para producir aplicaciones Web.
- Permite la creación de grandes aplicaciones distribuidas.
- Provee un buen control de sesión y manejo de autorización.
- Permite la creación de aplicaciones de múltiples capas.

Una de las desventajas de J2EE es que posee una curva de aprendizaje importante, lo que provoca una difícil inserción de diseñadores Web y programadores en sus primeros pasos.

.NET

Microsoft actualizó su tecnología ASP a ASP.NET que imita a J2EE en muchas maneras.

- Simplifica la creación de aplicaciones pequeñas a programadores que se están iniciando y a diseñadores gráficos.
- Permite la creación de grandes aplicaciones distribuidas.
- Provee un buen control de sesión y manejo de autorización.
- Permite a los programadores la utilización de su lenguaje de programación favorito, el que es compilado a código nativo.

La elección entre J2EE y .NET es dependiente de la plataforma. Las aplicaciones J2EE teóricamente pueden ser ejecutadas en la mayoría de las plataformas, desde Linux a AIX, MacOS X o Windows.

2.3.1.4 Web 2.0.

El concepto original de la Web (en este contexto, llamada Web 1.0) eran páginas estáticas

HTML que no eran actualizadas frecuentemente. El éxito de las punto com dependía de webs más dinámicas (a veces llamadas Web 1.5) donde los CMS servían páginas HTML dinámicas creadas al vuelo desde una actualizada base de datos. En ambos sentidos, el conseguir hits (visitas) y la estética visual eran considerados como unos factores muy importantes.

Los propulsores de la aproximación a la Web 2.0 creen que el uso de la Web está orientado a la interacción y redes sociales, que pueden servir contenido que explota los efectos de las redes con o sin crear webs interactivas y visuales. Es

decir, los sitios Web 2.0 actúan más como puntos de encuentro, o webs dependientes de usuarios, que como webs tradicionales.

2.3.2 Diseños con el estándar UML.¹¹

2.3.2.1 Introducción a UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT y OOSE (Object-oriented software engineering). UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

¹¹ Ferré Grau Xavier, Sánchez S. María Isabel. Desarrollo Orientado a Objetos con UML. Facultad de Informática - UPM

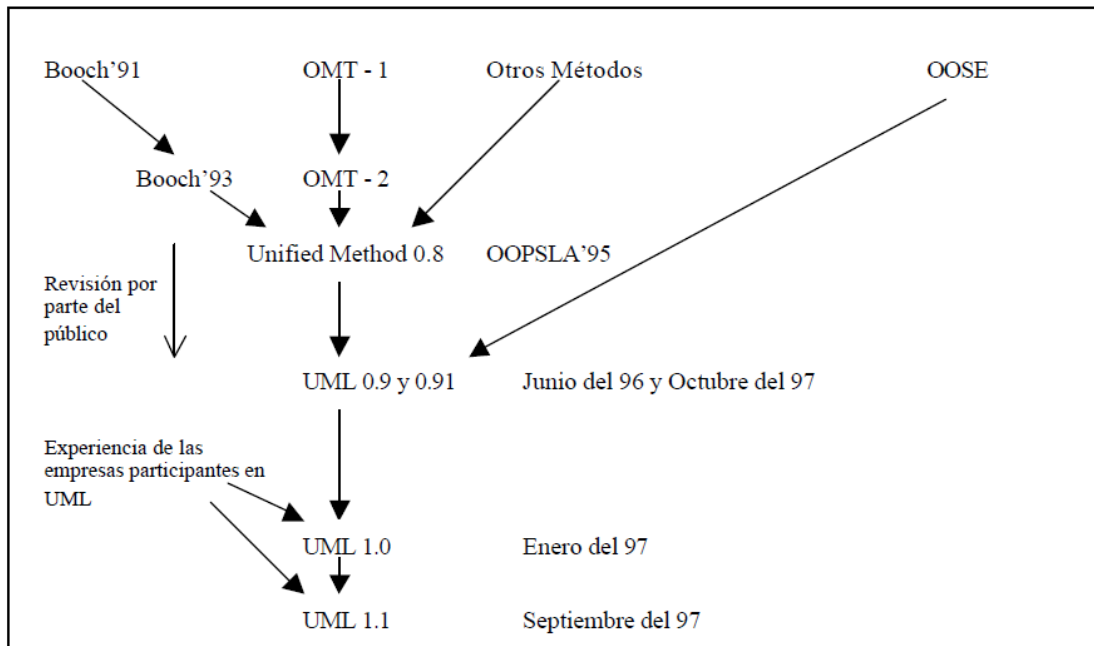


Figura 4. Historia de UML.

2.3.2.2 Elementos comunes a todos los diagramas.

Nota

Una nota sirve para añadir cualquier tipo de comentario a un diagrama o a un elemento de un diagrama. Es un modo de indicar información en un formato libre, cuando la notación del diagrama en cuestión no nos permite expresar dicha información de manera adecuada.

Una nota se representa como un rectángulo con una esquina doblada con texto en su interior.

Puede aparecer en un diagrama unida a un elemento por medio de una línea discontinua. Puede contener restricciones, comentarios, el cuerpo de un procedimiento, etc.

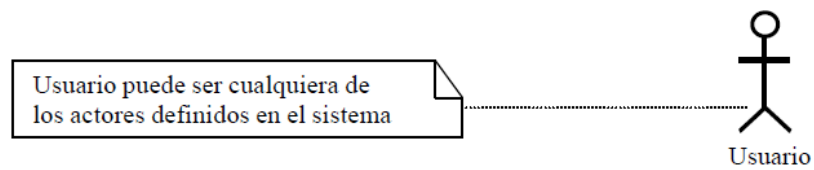


Figura 5. Ejemplo de una Nota UML.

Dependencias

La relación de dependencia entre dos elementos de un diagrama significa que un cambio en el elemento destino puede implicar un cambio en el elemento origen (por tanto, si cambia el elemento destino habría que revisar el elemento origen).

Una dependencia se representa por medio de una línea de trazo discontinuo entre los dos elementos con una flecha en su extremo. El elemento dependiente es el origen de la flecha y el elemento del que depende es el destino (junto a él está la flecha).

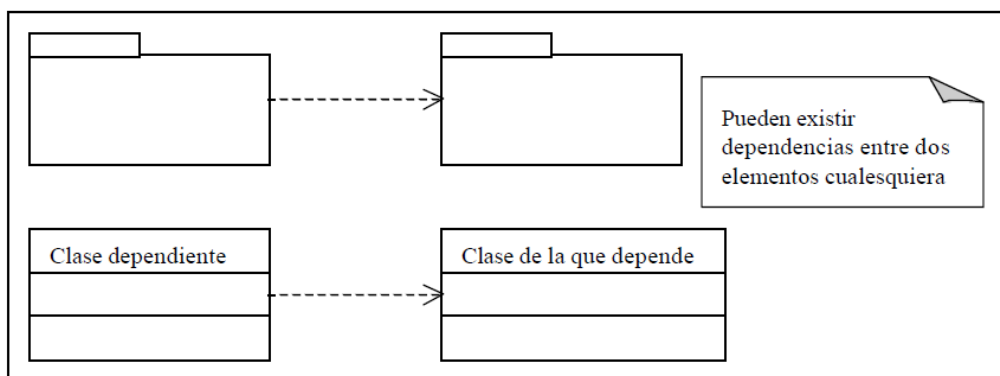


Figura 6. Ejemplo de una Dependencia UML.

2.3.2.3 Diagramas de Casos de Uso.

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Elementos del diagrama de casos de uso.

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

Actores

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo.

Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).

Casos de Uso

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

Relaciones entre Casos de Uso

Entre dos casos de uso puede haber las siguientes relaciones:

- Extiende: Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- Uso: Cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<uso>> según sea el tipo de relación.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

En la Figura se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.

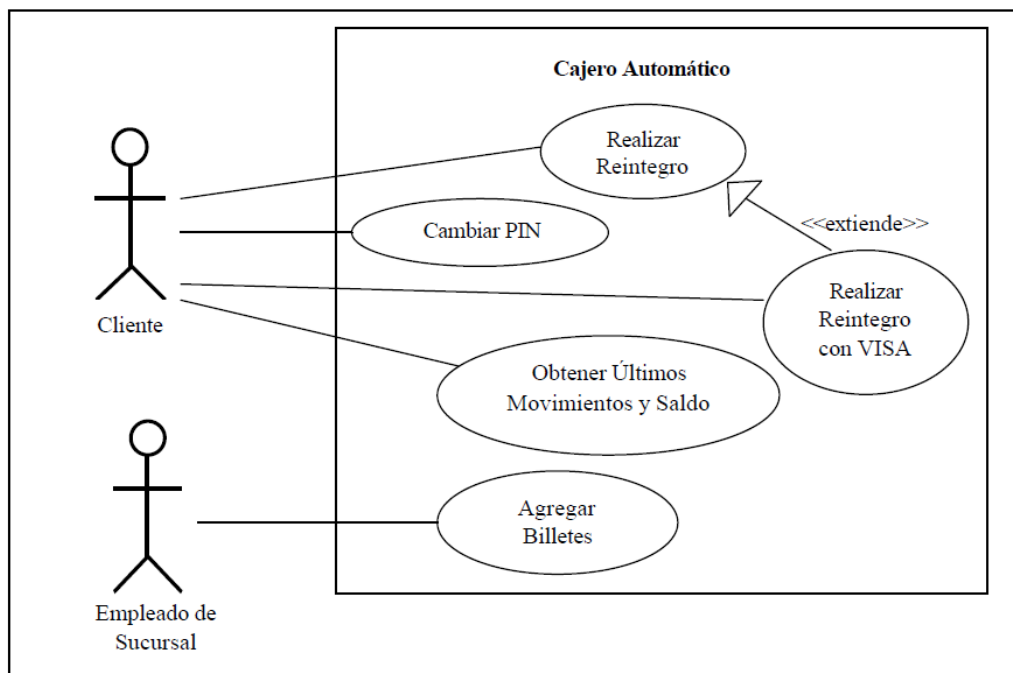


Figura 7. Ejemplo de Diagrama de Caso de Uso

2.3.2.4 Diagramas de Secuencia.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.

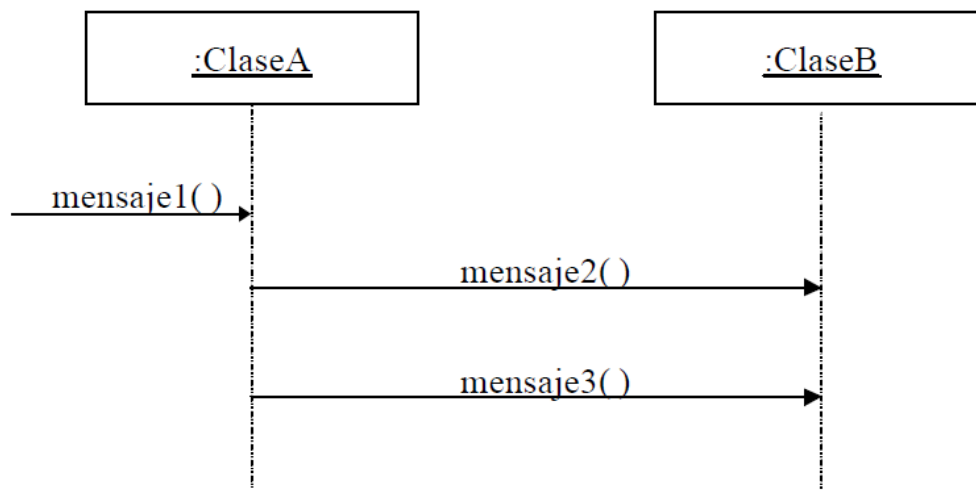


Figura 8. Ejemplo de Diagrama de Secuencia

2.3.2.5 Diagrama de Actividades

La idea es generar una especie de diagrama Pert, en el que se puede ver el flujo de actividades que tienen lugar a lo largo del tiempo, así como las tareas concurrentes que pueden realizarse a la vez. El diagrama de actividades sirve para representar el sistema desde otra perspectiva, y de este modo complementa a los anteriores diagramas vistos.

Gráficamente un diagrama de actividades será un conjunto de arcos y nodos.

Desde un punto de vista conceptual, el diagrama de actividades muestra cómo fluye el control de unas clases a otras con la finalidad de culminar con un flujo de control total que se corresponde con la consecución de un proceso más complejo.

Por este motivo, en un diagrama de actividades aparecerán acciones y actividades correspondientes a distintas clases. Colaborando todas ellas para conseguir un mismo fin.

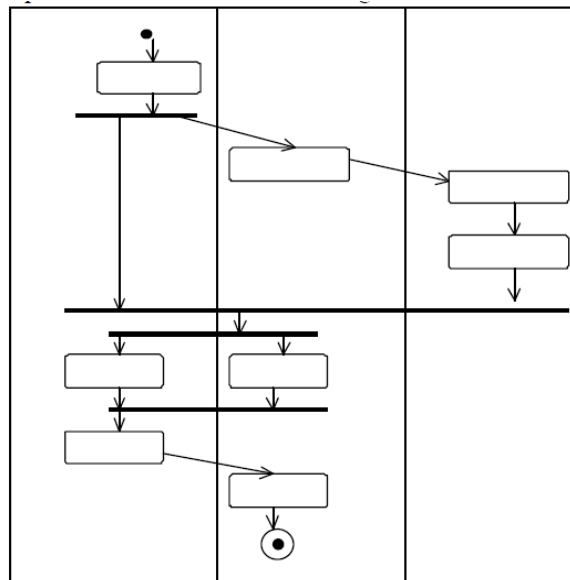


Figura 9. Ejemplo Diagrama de Actividades

2.3.2.6 Diagrama de Clases¹².

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Representación de: - Requerimientos en entidades y actuaciones. - La arquitectura conceptual de un dominio - Soluciones de diseño en una arquitectura - Componentes de software orientados a objetos

- Propiedades también llamados atributos o características, son valores que corresponden a un objeto, como color, material, cantidad, ubicación. Generalmente se conoce como la información detallada del objeto. Suponiendo que el objeto es una puerta, sus propiedades serían: la marca, tamaño, color y peso.
- Operaciones comúnmente llamados métodos, son aquellas actividades o verbos que se pueden realizar con/para este objeto, como por ejemplo abrir, cerrar, buscar, cancelar, acreditar, cargar. De la misma manera que el nombre de un atributo, el nombre de una operación se escribe con minúsculas si consta de una sola palabra. Si el nombre contiene más de una palabra, cada palabra será unida a la anterior y comenzará con una letra mayúscula, a excepción de la primera palabra que comenzará en minúscula. Por ejemplo: abrirPuerta, cerrarPuerta, buscarPuerta, etc.
- Interfaz es un conjunto de operaciones que permiten a un objeto comportarse de cierta manera, por lo que define los requerimientos mínimos del objeto. Hace referencia a polimorfismo.

¹² Fuente: http://es.wikipedia.org/wiki/Diagrama_de_clases

- Herencia se define como la reutilización de un objeto padre ya definido para poder extender la funcionalidad en un objeto hijo. Los objetos hijos heredan todas las operaciones y/o propiedades de un objeto padre. Por ejemplo: Una persona puede especializarse en Proveedores, Acreedores, Clientes, Accionistas, Empleados; todos comparten datos básicos como una persona, pero además cada uno tendrá información adicional que depende del tipo de persona, como saldo del cliente, total de inversión del accionista, salario del empleado, etc.

Diagrama de Clases

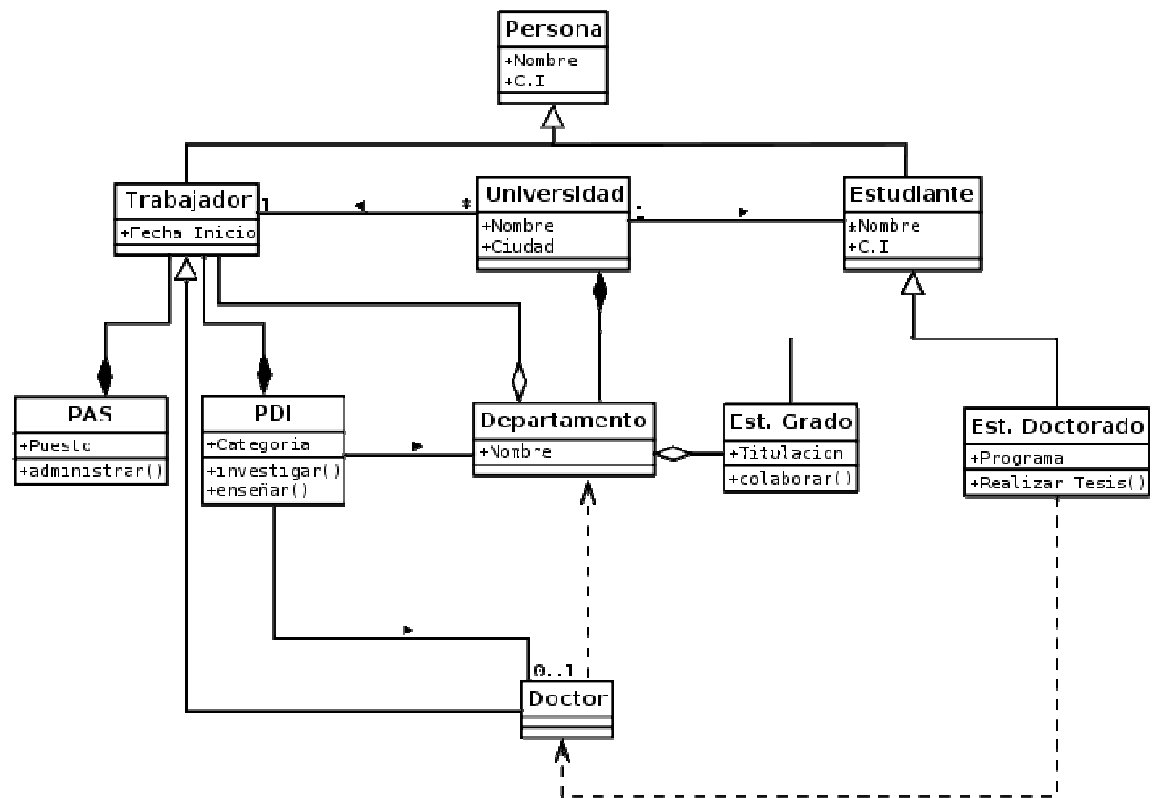


Figura 10. Ejemplo de Diagrama de Clases.

Al diseñar una clase se debe pensar en cómo se puede identificar un objeto real, como una persona, un transporte, un documento o un paquete. Estos ejemplos de clases de objetos reales, es sobre lo que un sistema se diseña. Durante el proceso del diseño de las clases se toman las propiedades que identifican como único al objeto y otras propiedades adicionales como datos que corresponden al objeto.

2.3.2.7 Diagrama Entidad – Relación¹³

Un diagrama o modelo entidad-relación (a veces denominado por su siglas, E-R "Entity relationship", o, "DER" Diagrama de Entidad Relación) es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

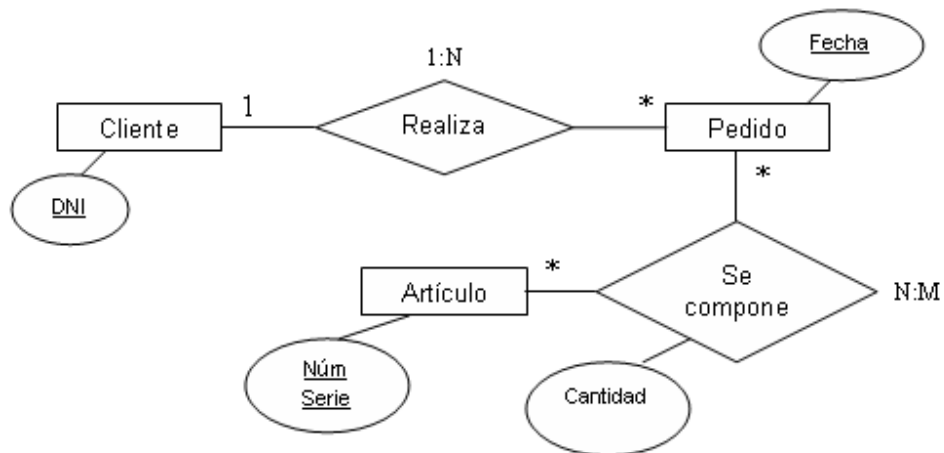


Figura 11. Ejemplo Diagrama Entidad-Relación.

¹³ Fuente: http://es.wikipedia.org/wiki/Diagrama_entidad-relaci%C3%B3n

El Modelo Entidad-Relación.

- Se elabora el diagrama (o diagramas) entidad-relación.
- Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama.

Dado lo rudimentario de esta técnica se necesita cierto entrenamiento y experiencia para lograr buenos modelos de datos.

El modelado de datos no acaba con el uso de esta técnica. Son necesarias otras técnicas para lograr un modelo directamente implementable en una base de datos.

Brevemente:

- Transformación de relaciones múltiples en binarias.
- Normalización de una base de datos de relaciones (algunas relaciones pueden transformarse en atributos y viceversa).
- Conversión en tablas (en caso de utilizar una base de datos relacional).

2.3.3 Java Enterprise Edition 5.0 (JEE 5)

Introducción a Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de

ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

Java EE5

Java Enterprise Edition 5 (Java EE 5) se centra en hacer más fácil el desarrollo, sin embargo, conserva la riqueza de la plataforma J2EE 1.4. Ofrece funciones como Java Server Faces (JSF) y la tecnología de servicios web API, Java EE 5 hace que la codificación sea más simple y directa, pero mantiene el poder que ha establecido a Java EE como la primera plataforma para servicios web y desarrollo de aplicaciones empresariales.

El SDK de Java EE 5 y Java Application Platform SDK prestan apoyo a las especificaciones Java EE 5, y el SDK características adicionales, tales como tiempo de ejecución de Open ESB, Portlet Container, y Sun Java System Access Manager.

2.3.4 Entorno de Desarrollo

2.3.4.1 Java Server Faces (JSF)

Java Server Faces es una solución integral al problema de proveer una experiencia de usuario rica y a la vez sencilla en aplicaciones web. Para los desarrolladores de software, JSF provee una API estandarizada, fácil de usar y orientada a objetos, permitiendo crear interfaces de usuario con componentes reutilizables. Esto también repercute en la consistencia de tales interfaces, brindándole al usuario una experiencia de máxima calidad.

Su principal ventaja consiste en que el desarrollador sólo debe aprender el modelo de interfaces de usuario de JSF una sola vez, lo que le permitirá usar cualquier componente que cumpla los estándares de este modelo, aún si tales componentes provienen de terceros.

2.3.4.2 Object Relational Mapping / JPA

Más conocida por su sigla JPA, es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sí pasaba con EJB2, y permitir usar objetos regulares.

2.3.4.3 EJB 3.0

Enterprise JavaBeans (EJB) es una arquitectura de componentes para la construcción de aplicaciones empresariales ejecutadas en servidores. Tiene por propósito proveer una forma estándar de implementar este tipo de aplicaciones, haciéndose cargo de aspectos comunes y repetitivos como la persistencia, la integridad transaccional y la seguridad, permitiendo que el desarrollador pase a preocuparse exclusivamente por la lógica del negocio en sí.

JBoss AS fue de los primeros servidores de aplicaciones en adoptar las especificaciones de EJB 3.0. Este modelo de EJB simplifica el desarrollo eliminando la necesidad de una interfaz “Home” y descriptores de despliegue,

reemplazándolos por anotaciones. Facilita la implementación de la persistencia por medio del api JPA.

2.3.5 Servidor de Aplicaciones – Jboss

El servidor de aplicaciones JBoss es una herramienta certificada para el desarrollo de aplicaciones empresariales Java. Su madurez y el esfuerzo de muchos desarrolladores, e incluso las sugerencias que han realizado muchos usuarios, han permitido que JBoss AS (Application Server) se popularice ampliamente y sea común en los currículos de muchos desarrolladores. Encuestas recientes muestran que es el servidor de aplicaciones más popular actualmente.

Es reconocido por soportar los estándares más recientes. De hecho, es el primer servidor de aplicaciones en alcanzar la certificación J2EE 1.4 cuando salió su versión 4.0 (actualmente va en la versión 5, liberada a finales de 2008). Pero JBoss no sólo marca la pauta en la adopción de estándares con su servidor de aplicaciones, sino en la imposición de los mismos. Recientemente se le eligió para hacer parte del Java Community Process (JCP). Además en los últimos años ha estado a la cabeza del desarrollo de Java Enterprise llegando a establecerse en todas las especificaciones de requerimientos de Java (Java Specification Requests, JSRs).

2.3.6 Enterprise Architect

Enterprise Architect es una herramienta desarrollada por Sparx Systems que ofrece la capacidad de realizar el modelado de un proyecto y apoyar el desarrollo del mismo durante todo su ciclo de vida. Enterprise Architect logra esto usando UML, pero también puede generar código en varios lenguajes de acuerdo a algunos de estos diagramas.

Su valor como herramienta radica en la capacidad de permitir a los ingenieros desarrolladores comunicar sus ideas y su visión sobre los proyectos facilitando la administración y la redistribución de esta información.

2.3.7 Construcción de Prototipos de Software.

Un prototipo es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y, enterarse más acerca del problema y sus posibles soluciones. Un prototipo de software apoya a dos actividades del proceso de ingeniería de requerimientos:

- Obtención de requerimientos: les permite adquirir nuevas ideas para los requerimientos y encontrar áreas fuertes y débiles del software.
- Validación de requerimientos. El prototipo puede revelar errores y omisiones en los requerimientos. La construcción de prototipos puede utilizarse como un análisis de riesgo y una técnica de reducción. Un modelo iterativo del proceso, como el desarrollo incremental, se utiliza junto con un lenguaje diseñado para el desarrollo rápido de aplicaciones. Por lo tanto, las técnicas utilizadas para desarrollar un prototipo para validar los requerimientos también se utilizan para desarrollar el sistema de software mismo.

Ventajas:

- Al demostrar las funciones del sistema se identifican las discrepancias entre los desarrolladores del software y los usuarios.
- Durante el desarrollo del prototipo el personal del desarrollo de software puede darse cuenta de que los requerimientos son inconsistentes y/o están incompletos.

- Se dispone rápidamente de un sistema que funciona y demuestra la factibilidad y usabilidad de la aplicación a administrar.

En sistemas grandes y complejos una forma de resolver la dificultad de evaluación es utilizar un enfoque evolutivo para el desarrollo de sistemas. Esto significa proporcionar al usuario un sistema incompleto y después modificarlo y aumentarlo en el momento en que los requerimientos del usuario sean claros.

El enfoque de construcción de prototipos desechables es para ayudar a refinar y clasificar la especificación del sistema. El prototipo se escribe, evalúa y modifica. La evaluación del prototipo informa del desarrollo de la especificación detallada del sistema que se incluye en el documento de requerimientos de este. Una vez que se ha redactado la especificación, el prototipo ya no es útil y se desecha.

Prototipo No funcional ¹⁴

Es un modelo no funcional a escala configurado para probar ciertos aspectos de diseño. Un ejemplo de este enfoque es un modelo a escala completa de un automóvil que se usa para pruebas en un túnel de viento. El tamaño y forma del automóvil son precisos, pero el automóvil no es funcional. En este caso solo se incluyen las características del automóvil que son fundamentales para la prueba en el túnel de viento.

Un modelo no funcional a escala de un sistema de información podría producirse cuando la codificación requerida por las aplicaciones es demasiado extensa para incluirse en el prototipo, pero se puede conseguir una idea útil del sistema a través de la elaboración de un prototipo de la entrada y la salida. En este caso, el procesamiento, debido al excesivo costo y el tiempo requerido, no podría incluirse en el prototipo. Sin embargo, aún se podrían tomar algunas decisiones sobre la utilidad del sistema con base en la entrada y la salida incluidas en el prototipo.

¹⁴ Fuente: Kendall, Julie E. Analisis y Diseño de sistemas. Capitulo 6. Página 153

3. METODOLOGÍA DE DESARROLLO

3.1 CICLO DE VIDA DEL PROYECTO.

A continuación se hace una descripción de las diferentes actividades que se llevaron a cabo durante el transcurso del proyecto, buscando conceptualizar la metodología que se aplicó en el desarrollo del nuevo sistema de información de preguntas, quejas, reclamos y sugerencias para la Universidad Industrial de Santander.

3.1.1 Análisis de Requerimientos:

El análisis de requerimientos es la tarea que plantea la asignación de software a nivel de sistema y el diseño de programas.

Todo el proceso de análisis de requerimientos se especificó, junto con los funcionarios de la Dirección de Control Interno y Evaluación de Gestión, las funcionalidades y comportamiento que debían tener los programas desarrollados en el transcurso del proyecto, se definió la interfaz y el esquema de navegación entre páginas junto con otros elementos del sistema y se establecieron los estándares de diseño que el sistema debe mantener.

El análisis de requerimientos permite la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño procedimental. Finalmente, la especificación de requerimientos suministra los medios para valorar la calidad de los programas, una vez que se haya construido.

En esta etapa se hicieron reuniones periódicas con los funcionarios de la Dirección de Control Interno y Evaluación de Gestión, para que fuesen ellos los

que definan las características del software que se desea implantar y que se adapte plenamente a las exigencias del modelo de gestión de calidad en la que ha sido certificada la universidad.

3.1.2 Diseño

El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo).

En esta etapa de diseño se hace una traducción de los requisitos a una representación del software donde se pueda evaluar su calidad antes de que comience la codificación.

El diseño se efectuó, mediante modelos UML (Lenguaje de Modelado Unificado) dónde se incluyeron los Diagramas de Clases, de Actividades, de Entidad-Relación y de Secuencia, utilizando la herramienta Enterprise Architect licenciada por la Universidad Industrial de Santander a través de División de Servicios de Información.

3.1.3 Implementación de la Aplicación.

El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código se llevó a cabo en esta tarea.

En esta etapa se procede a generar el software que se ha diseñado. Se realizó teniendo en cuenta los parámetros establecidos por la División de Servicios de Información en cuanto a los estándares técnicos y de calidad que caracterizan las aplicaciones que son generadas para el servicio de la Universidad, teniendo como base el Lenguaje de programación JAVA 5, frameworks como: seam, java server

faces (JSF), Enterprise Java Beans (EJB 3.0) e Informix como motor de base de datos.

3.1.4 Pruebas del software.

Son los procesos que permiten verificar y revelar la calidad de un producto software.

Las pruebas de software se integran dentro de las diferentes fases del ciclo de desarrollo del software establecidas en la ingeniería de software.

Una vez generado el código, comienzan las pruebas, proceso utilizado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Las pruebas se centran en los procesos lógicos internos del software, asegurando que todas las sentencias sean probadas y asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.

Estas pruebas se aplicaron de forma permanente a lo largo del desarrollo del proyecto por parte del equipo de trabajo, y se abrió un espacio dónde los usuarios finales interactuaron con la aplicación con el objetivo que detectaran posibles fallos que hayan sido omitidos en el momento del desarrollo.

3.1.5 Ajustes.

Es el proceso de mejora del sistema tomando como base las sugerencias y observaciones que se plantearon en el periodo de pruebas, aquí también se incluyen las correcciones a los posibles fallos detectados, mejoras en la interfaz y el diseño de la misma.

Este proceso está incluido dentro del proceso iterativo de refinamiento que se le dio al sistema, para adaptarse plenamente a las necesidades del cliente que en el presente caso es la Dirección de control interno y evaluación de gestión.

3.2 METODOLOGIA DE DESARROLLO DEL PROYECTO.

3.2.1 Modelo de Construcción de Prototipos.

Teniendo como base las actividades anteriormente descritas, se dispuso como metodología de desarrollo el MODELO DE CONSTRUCCION DE PROTOTIPOS.

Se eligió esta metodología debido a que es muy frecuente que los clientes/usuarios que están solicitando el sistema, en este caso la Dirección de Control Interno y Evaluación de Gestión, definan un conjunto de objetivos generales para el software, pero no identifican los requisitos detallados de entrada, proceso o salida.

En otros casos, el responsable del desarrollo del software puede no estar seguro de la eficacia de un algoritmo, o no haber comprendido plenamente el requerimiento del usuario.

Para éstas y otras muchas situaciones, un paradigma de construcción de prototipos puede ofrecer el mejor enfoque, ya que la entrega de prototipos, que hacen parte integral del proyecto en su conjunto, permitirán la corrección temprana de errores o la redefinición del sistema en caso de ser necesario, y los prototipos tanto funcionales como no funcionales permitirán la familiarización del usuario con el sistema que se está desarrollando.

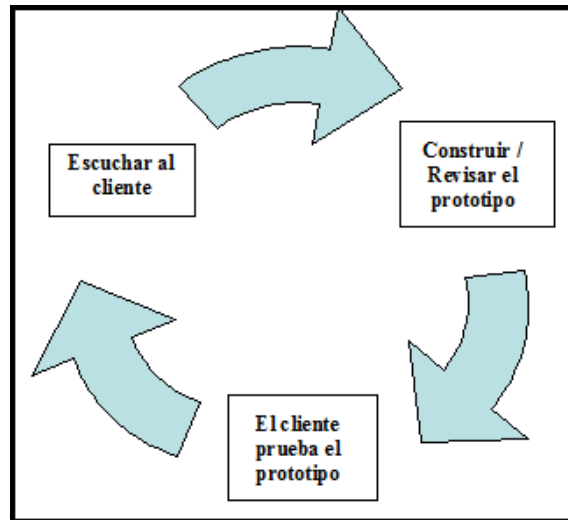


Figura 12. Modelo de Construcción de Prototipos

3.2.2 Estructura del modelo de construcción de prototipos.

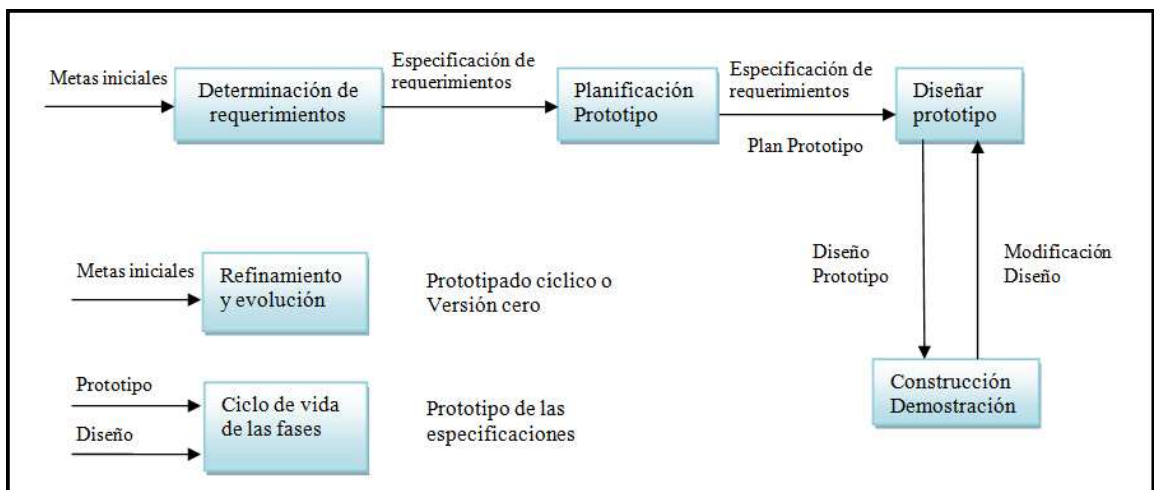


Figura 13. Estructura del Modelo de Construcción de Prototipos

Ésta metodología planteada, permitió que el proyecto se llevara a cabo de manera eficiente, debido a:

- En la creación de los prototipos iniciales se trabajó con unas ideas aproximadas de lo que pretendía alcanzar la Dirección de Control Interno y Evaluación de Gestión, éstas ideas dieron resultado a un producto o prototipo inicial, el cual evolucionó dando como resultado un prototipo más maduro, que cumple con todos los requerimientos del cliente.
- Con el uso del modelo de prototipos se dio la facilidad de mejorar, de manera temprana los prototipos, teniendo en cuenta las sugerencias del usuario solicitante del proyecto, de manera que se pudieran cubrir a cabalidad las necesidades por las cuales se desarrolló el proyecto.
- Es importante conocer de forma temprana si los diferentes prototipos de los servicios, cumplen a satisfacción con los requerimientos expuestos por los funcionarios de la Dirección de Control Interno y Evaluación de Gestión.
- En este sistema de desarrollo no se libera un prototipo sin haber realizado todas las pruebas necesarias del mismo, pero los inconvenientes resultan inevitables y se da la facilidad de detectarlos y corregirlos de manera oportuna.

3.2.3 Procedimiento para la metodología planteada.

- a. La construcción de prototipos comienza con la recolección de los requisitos. La Dirección de Control Interno y Evaluación de Gestión de la Universidad Industrial de Santander (quienes son los usuarios administradores del sistema) cuentan de manera verbal lo que desean alcanzar con la nueva versión y los problemas de la actual versión que inevitablemente deben ser subsanados.

- b. El desarrollador y usuario se reúnen y definen los objetivos globales para el software, identifican todos los requisitos conocidos y perfilan las áreas en donde será necesaria una mayor definición.
- c. Teniendo la abstracción del sistema por parte del desarrollador, se realiza todo el diseño de los objetos mediante los diagramas UML: Casos de uso, de clases, de secuencia, entidad – relación y actividades.
- d. Luego se produce el Diseño del Prototipo que se enfoca sobre la representación de los aspectos del software visibles al usuario (por ejemplo, métodos de entrada y formatos de salida), en ésta etapa se presenta al usuario el diseño tanto de los diagramas UML como el prototipo NO funcional, para que éste lo apruebe y se adecúe plenamente a su solicitud.
- e. Se procede con la construcción del prototipo que se ha propuesto.
- f. El prototipo es evaluado por el usuario y se utiliza para refinar los requisitos de tal forma que el usuario identifique aspectos que deban ser replanteados o mejorados.
- g. Se produce un proceso iterativo en el que el prototipo es “afinado” (Refinamiento del prototipo) para que satisfaga las necesidades del usuario, al mismo tiempo que facilita al que lo desarrolla una mejor comprensión de lo que hay que hacer y poder entregar el producto final requerido.

PARTE II. DESARROLLO DEL SISTEMA.

4. APLICACIÓN DE LA METODOLOGÍA

4.1 LEVANTAMIENTO DE REQUERIMIENTOS

Se presentan los requerimientos, generados después de las primeras reuniones con el cliente, en donde se pudo abstraer la idea principal de lo que se quiere alcanzar durante el desarrollo del proyecto, que sirvió como base para las primeras etapas del análisis y diseño del nuevo sistema de información de preguntas, quejas, reclamos y sugerencias.

4.1.1 Descripción General.

Este sistema debe permitir la formulación de las Quejas, Reclamos, Sugerencias y solicitud de información (Preguntas) por parte de quien la realiza (ciudadano, estudiante, funcionario o proveedor) y permitir la gestión y tratamiento de las PQRS por parte de la Dirección de Control Interno y Evaluación de Gestión o de las unidades Académicas y Administrativas involucradas. Es importante aclarar, que la recepción de las PQRS se centrará en la Dirección de Control Interno y Evaluación de Gestión, quienes se encargarán de gestionarla y dar respuesta o de remitirlas a las unidades académicas y administrativas correspondientes, de acuerdo con su contenido.

Este módulo, que actualmente está integrado con Veeduría Ciudadana, será independiente de éste y tendrá por nombre “Quejas, Reclamos, Sugerencias y Preguntas”, por tanto, debe aparecer en la página Web de la UIS una pestaña con el título Quejas, Reclamos, Sugerencias y Preguntas, en la cual se realiza la gestión de las mismas por parte de todas las partes interesadas.

Los requisitos de este módulo se describen a continuación.

4.1.2 Definiciones y siglas.

DCIEG	:	Dirección de Control Interno y Evaluación de Gestión.
Ciudadano	:	Persona de la comunidad que no es funcionario ni proveedor ni estudiante de la universidad.
Control Interno:		Personal de la Dirección de Control Interno y Evaluación de Gestión.
Estudiante	:	Persona matriculada en algún programa académico de la UIS.
Funcionario	:	Personal vinculado a la Universidad mediante contrato de trabajo.
Proveedor	:	Persona vinculada a la Universidad mediante un contrato de suministro de productos o servicios de manera indirecta.
PQRS	:	Pregunta, Queja, Reclamo o Sugerencia.
QR	:	Queja o Reclamo.
Solicitante	:	Ciudadano, Funcionario UIS, Estudiante UIS o Proveedor UIS que formula una PQRS.
UAA	:	Unidad Académica y Administrativa.

4.1.3 Funciones que debe cumplir el sistema.

El módulo debe brindar soporte a la gestión de la DCIEG, al personal de la UAA y dar respuesta oportuna a los requerimientos de los Solicitantes:

4.1.3.1 Del registro de la solicitud

Diligenciamiento de la solicitud.

1. Cuando la persona ingresa a esta sección debe llenar un formulario que contiene:

- Primer Nombre* : campo para datos alfabéticos, donde se especificará el primer nombre del solicitante.
- Segundo Nombre : campo para datos alfabéticos, donde se especificará el segundo nombre del solicitante.
- Primer Apellido* : campo para datos alfabéticos, donde se especificará el primer apellido del solicitante.
- Segundo Apellido : campo para datos alfabéticos, donde se especificará el segundo apellido del solicitante.
- Identificación* : campo numérico, donde se especificará el número de documento de identidad del solicitante se debe aclarar que este campo es necesario para que pueda realizar seguimiento a la queja.
- Correo electrónico* : campo que solo admitirá formatos de correo electrónico, donde se especificará el correo electrónico del solicitante.

- Dirección* : campo alfanumérico, donde se especificará la dirección de contacto con el solicitante
- Tipo de Solicitante* : menú desplegable con opciones para elegir: funcionario UIS, estudiante UIS, ciudadano o proveedor UIS.
- Adjuntar archivo : campo que permite subir archivos de tamaño especificado por la capacidad disponible para tal servicio y de los formatos especificados, pero que sea adecuado para ofrecer la posibilidad de adjuntar documentos que complementen la queja o reclamo. Los formatos permitidos son pdf, Word , Excel e imágenes.
- Tipo de Solicitud* : campo de selección de opciones a través de un menú desplegable en donde se escoge si la solicitud es una Queja- Reclamo, una Sugerencia o una Pregunta; la opción por defecto no debe permitir enviar el formulario.
- Motivos de la Solicitud*: menú desplegable con opciones para elegir los motivos de la solicitud éstos varían de acuerdo al tipo de solicitud que va a ingresar el usuario, pueden ser agregados o modificados por el Administrador, cuando lo estimen conveniente), algunos de ellos son:
- Percepción del servicio.
 - Infraestructura.
 - Oportunidad en el servicio.
 - Información veraz.
 - Atención al beneficiario.

Disponibilidad de recursos.

Comunicación.

Transparencia en el manejo de recursos.

Asesoría confiable

Otros.

- Descripción* : campo para redacción de la Solicitud, se debe dar la posibilidad de tener a disposición 2000 caracteres.
- Ayuda en línea : El sistema debe permitir la ayuda en línea para explicar al Solicitante el funcionamiento del módulo PQRS. La redacción y contenido está a cargo de DCIEG.

2. Luego de diligenciar el Formulario de la PQRS y registrarla, debe aparecer un certificado al solicitante en el que se muestre el número de radicado de la solicitud (consecutivo), que durante todo el proceso no puede cambiar.

(*) Campos obligatorios

Seguimiento de la PQRS

Permite realizar seguimiento por parte del Solicitante de la QR, Sugerencia o Pregunta con su número de identificación y el número de radicado dado en el momento del registro, que al mismo tiempo debió hacersele llegar al correo electrónico del solicitante; se muestra si son correctos los datos digitados (es decir si coincide con la información registrada en la base de datos), el estado de la PQRS y la respuesta si ésta ya se hubiera dado por los encargados de hacerlo, y todas las PQRS que el solicitante hubiese realizado previamente.

4.1.3.2 Administración

Al módulo de administración tienen acceso funcionarios de la DCIEG y las UAA de la Universidad, con diferentes permisos. Para ingresar al módulo se debe tener un formulario con dos campos:

Login:

Contraseña:

Para las personas autorizadas se encuentran los siguientes perfiles, los cuales tendrán acceso al sistema bajo el nuevo esquema de seguridad que la División de Servicios de Información esta próxima a poner en marcha:

- Control Interno (CI) con permisos totales: con todos los permisos dentro de este módulo tales como: reporte de autorizados de Control Interno y personal U.A.A activos, direccionar PQRS a las diferentes UAA, responder PQRS, evaluar respuesta de las PQRS dada por las UAA., agregar observaciones (comunicación interna entre CI y UAA), si la respuesta es pertinente entonces enviar respuesta al solicitante. Además, puede realizar reportes y ver estadísticas.
- Control Interno (CI) con permisos restringidos: Con autorización sólo para consultar reportes y obtener estadísticas de todas las dependencias (personal de CI).
- UAA: tiene permiso para ver solo las quejas de su dependencia y darle respuesta. Realizar reportes y ver estadísticas.

Control Interno con permisos totales.

Esta sección debe traer las siguientes opciones:

Reporte de autorizados de Control Interno y personal UAA:

A través de una tabla, permite ver todos los autorizados del sistema habilitados para dar trámite a las PQRS. Tiene un buscador: buscar por dependencia, por perfil de autorizado, por nombre.

Gestión de las PQRS.

1. Reporte de PQRS: aparecen en una tabla todas las PQRS que se han realizado en el portal Web, así:

N° consecutivo, Tipo de Solicitud, Motivo de la Solicitud, Solicitante, Estado de la solicitud, fecha y hora de registro.

Los estados de cada solicitud serán definidos así:

- Pendiente: cuando la queja no ha sido revisada por Control Interno.
- Confirmado: cuando Control Interno ha leído la queja.
- En trámite: cuando la solicitud se encuentra en proceso interno para dar respuesta a la misma.
- Enviada a Solicitante: cuando Control Interno envía finalmente la respuesta al correo electrónico suministrado por el Solicitante.

Si está en el estado pendiente: la opción permitida debe ser ver (aparece el contenido de la solicitud)

Si está en el estado confirmado: debe permitir responder al Solicitante, o enviar a UAA, con lugar para colocar información por parte de Control Interno, que será información de carácter interno y no puede ser vista en la respuesta final al

Solicitante, es decir, que tiene dos campos para texto, UAA: en el que se envían observaciones, sugerencias, etc., a la UAA y el campo Solicitante donde se digita la respuesta final al Solicitante.

La opción enviar respuesta a Solicitante debe enviar automáticamente un correo electrónico al Solicitante y el almacenamiento de la respectiva respuesta para su consulta posterior de todos los autorizados para tal acción.

Además, esta sección debe permitir imprimir, guardar y copiar, realizar reportes con los siguientes filtros: número de radicado, tipo de solicitud, motivo de la solicitud, UAA, estado de la queja, por fechas y tipo de Solicitante.

Consultas

Al dar clic sobre la opción VER SOLICITUD permite visualizar el contenido de la PQRS, las observaciones que envió CI a UAA sobre la queja, la respuesta de UAA y toda la información sobre la PQRS en una ventana que fácilmente permita volver al reporte general de las PQRS, de igual forma se debe permitir en esta opción poder filtrar todas las solicitudes, tanto las que están en trámite como las que se encuentran cerradas por haber sido dada respuesta al solicitante.

El filtro debe ser por:

- UAA.
- Rango de fechas
- Tipo de Solicitud
- Motivos de la Solicitud.

Estadísticas

Debe permitir al administrador seleccionar entre los tipos de solicitud, y poder visualizar, entre otros, los siguientes datos estadísticos:

- Porcentaje y totalidad por atributo de las quejas y reclamos.

- Por UAA para PQRS.
- Por meses, trimestres, semestres y años para PQRS.

Mantenimiento al Sistema.

El usuario de Control Interno con permisos totales también tendrá la posibilidad de hacerle mantenimiento a las tablas soporte, en dónde se puede redefinir parámetros básicos para el funcionamiento del sistema, tales como:

- Tipos de Solicitud.
- Tipos de Solicitantes.
- Motivos de la Solicitud y las relaciones de éstos con los tipos de solicitud asociados.

Control Interno con permisos restringidos:

Dos secciones: una para ver reportes y otra para estadísticas de todas las PQRS

- Puede imprimir, guardar y copiar.
- Podrá ver las estadísticas, enunciadas en el apartado anterior para los usuarios con permisos totales.

4.1.3.3 Personal UAA.

Cuando el personal de la UAA ingrese a su módulo puede ver las PQRS que han llegado a su dependencia. Debe tener opción para responder y enviar esta respuesta a Control Interno, en donde pueden hacer observaciones y determinar la pertinencia de la respuesta, si no es adecuada pueden enviarla otra vez a la UAA donde deben volver a responder.

De igual manera puede hacer reportes, reportes filtrados y estadísticas para las PQRS que se encuentran en su dependencia.

4.2 ESTÁNDARES DE LA DIVISIÓN DE SERVICIOS DE INFORMACIÓN.¹⁵

4.2.1 Aspectos Generales

Interfaz de desarrollo

El IDE de desarrollo a utilizar es el JBoss Developer Studio, el cual debe ser instalado en la carpeta por defecto del instalador.

Este y todos los programas necesarios se pueden descargar, por el personal autorizado, del equipo establecido para tal fin.

Servidor de aplicaciones

En cuanto al servidor de aplicaciones de desarrollo se debe utilizar el mismo que se encuentra en los servidores de producción y desarrollo, el cual debe ser instalado en la carpeta C:\jboss-5.0.0.GA.

JAVA

La versión del compilador de JAVA debe ser la 1.5, la cual debe ser instalada en el directorio C:\java1.5.

SEAM

La versión del seam a utilizar es la 2.1.2.GA. (jboss-seam-2.1.2.GA.zip).

Espacio de trabajo

El espacio de trabajo se debe crear en C:\workspace.

¹⁵ Fuente: Estandares de la División de Servicios de Información de la Universidad Industrial de Santander.

El nombre del proyecto para los JPA debe estar conformado de la siguiente manera:

[Sistema]Entidades

Por ejemplo: AcademicoEntidades (La primera letra de cada palabra en mayúscula).

El repositorio para los JPA debe estar conformado de la siguiente manera:

[Sistema]JPA

Por ejemplo: AcademicoJPA (La primera letra de cada palabra en mayúscula).

El Server name en el IDE de desarrollo se debe llamar JBoss 5 y el nombre del JBoss Runtime Environment se debe llamar JBoss 5.

Servidor de versiones

Para su configuración se debe instalar en el JBoss Developer Studio los siguientes paquetes:

- subclipse-site-1.4.7
- ajdt_1.6.1a_for_eclipse_3.4
- org.tmatesoft.svn_1.2.1.eclipse

Una vez instalados se debe solicitar el nombre de usuario y la contraseña al Ingeniero encargado.

Cualquier inquietud acerca de la instalación y configuración del servidor de versiones, dirigirse con el Ingeniero encargado.

También se debe instalar el siguiente programa: TortoiseSVN-1.5.8.15348-win32-svn-1.5.5.msi para conectarse con el servidor de versiones de la documentación.

Plantillas, estilos, imágenes y formateador

Descargar del servidor de versiones de documentación las carpetas Estilos, Plantillas e Imágenes y copiarlas en la careta view de la aplicación.

Patrones a utilizar

Los patrones a utilizar corresponden a cada una de las capas implementadas:

- Capa de presentación: Modelo Vista Controlador, el cual es implementado por Java Server Faces (JSF).
- Capa de lógica de negocio: Session Façade
- Capa de persistencia: Entity Access Object (EAO)

Rich Faces

La versión a utilizar es la incluida en el jboss-seam-2.1.2.

Código HTML

Está prohibido el uso de etiquetas HTML en las páginas.

Anotaciones en la entidad

Las anotaciones en la entidad se deben colocar antes del método get correspondiente y no en la declaración del atributo.

Llaves compuestas

Se debe utilizar la anotación EmbeddedId, la cual obliga a declara un objeto del tipo de la llave dentro del EJB de entidad.

HashCode e Equals en EJB de entidad

Se debe utilizar únicamente los campos que conforman la llave primaria. En el caso de las llaves compuestas, el atributo que hace referencia a ésta.

JPA externos

Para utilizar los JPA externos (academico.jar, recursos-humanos.jar, etc), éstos deben copiarse en la carpeta raíz. En el caso de Windows C:\, para Linux \).

En cada uno de los archivos persistence.xml de su aplicación, se debe utilizar <jar-file>file:/jpa.jar</jar-file>.

Por ejemplo:

```
<jar-file>file:/academico.jar</jar-file>
```

```
<jar-file>file:/recursos-humanos.jar</jar-file>
```

```
<jar-file>file:/general-UIS.jar</jar-file>
```

Servicios

Para crear un servicio se debe tener en cuenta las siguientes reglas:

- La interfaz debe contener la anotación Remote
- La implementación de la interfaz debe contener la anotación RemoteBinding con su respectivo nombre jndi (igual al utilizado por la anotación Name). Por ejemplo:

```
@RemoteBinding(jndiBinding = "ConsultarGenerales")
```

Para utilizar el servicio se debe utilizar la anotación EJB indicando el nombre jndi. De acuerdo al ejemplo anterior sería:

```
@EJB(mappedName = "ConsultarGenerales")
```

4.2.2 Documentación de los Diagramas de Diseño

Casos de Uso.

Para el desarrollo del modelo de casos de uso, se debe realizar un diagrama de casos de usos por módulo del sistema a implementar siguiendo el estándar propuesto por el Lenguaje Unificado de Modelado 2.1 (UML). Se deben tener en cuenta los siguientes los siguientes puntos:

Identificación de Actores

Se identifican con el rol que desempeñan en el sistema.

Diagrama de Casos de Uso

El diagrama de casos de uso se tiene que realizar con la herramienta Enterprise Architect. Cada caso de uso constituye un flujo completo de eventos especificando la interacción que toma lugar entre el actor y el sistema.

Casos de Uso

Se deben identificar con una acción. Los verbos que se pueden utilizar se encuentran al final del documento.

La información mínima requerida por cada caso de uso es la siguiente:

- Descripción completa
- Precondiciones y postcondiciones
- Descripción del escenario básico y alternos
- Diagrama de clases
- Si el caso de uso es complejo se debe incluir:
- Diagrama de secuencia y/o diagrama de actividades

4.2.3 Sintaxis de Nombres en Java

Reglas de sintaxis generales

- En esta sección se especifican las reglas de sintaxis generales para todos los identificadores (nombres de variables, clases, métodos, etc.)
- Todos los nombres de los identificadores deben estar en español.
- Siempre se deben utilizar nombres que sean claros, concretos y libres de ambigüedades. Usando palabras completas evitando acrónimos y abreviaturas.
- Los nombres deben estar definidos sin espacios en blanco, sin guiones (_ , -), ni comillas (" , '), sin operadores (+ , - , / , *), sin tildes, utilizar la n en vez de la ñ y sin caracteres especiales.
- No se debe utilizar la mayúscula para diferenciar entre identificadores distintos. Ejemplo: contador y Contador.
- No se deben diferenciar dos identificadores solo con numerales en cualquier posición. Ejemplo: contador1, contador2, 1contador, 2contador.
- Las siguientes partículas están prohibidas en la declaración de los nombres de identificadores: artículos (el, la, los, unos, unas, un), determinantes demostrativos (este, ese, aquel, aquellos), cardinales (uno, dos, etc.), pronombres de cualquier tipo (yo, tu, el, me, te, se, este, ese, mi, tu, su, etc.).
- Se deben utilizar máximo 5 palabras por nombre, las 3 primeras palabras van completas, a partir de la cuarta palabra se quitan las vocales a la palabra exceptuando la última vocal y la primera si la palabra empieza por vocal. No se utiliza ningún separador entre las palabras, se separa cada palabra utilizando su primera letra en mayúscula. Ejemplo: hacerMantenimientoConsultaUsros, hacerMantenimientoAsignaturasCntxtto.

Clases

- Los nombres de las clases deben iniciar siempre en mayúscula, deben ser simples y descriptivos.
- Los nombres de las clases de Entidad deben ser sustantivos en singular.
- Para las clases de Entidad siempre se debe definir la anotación `@Table` que indica la tabla de la base de datos relacionada para su persistencia. Además se debe utilizar el parámetro `name` de la anotación `@Entity` para identificar el EJB (`@Entity(name = "xxx")`). El nombre de la clase puede ser distinto al nombre de la tabla y debe seguir el estándar de identificadores mencionado en el numeral anterior.
- Los nombres de los EJB utilizados por el patrón `Session Façade` está conformado por un verbo autorizado (Ver anexo de verbos). Además debe incluir al final las letras "EJB". Ejemplo: `RegistrarMatriculaEstudianteEJB`.
- La interfaz asociada al EJB debe llevar el mismo nombre del EJB sin el sufijo "EJB". Ejemplo: `RegistrarMatriculaEstudiante`.
- Para los EJB de entidad, cuando la clase representa una relación entre dos entidades, el nombre se forma uniendo los nombres de las entidades involucradas.
- Los nombres de las clases que representan excepciones terminaran siempre con el sufijo "EXCEPCION".
- El nombre de la clase no contendrá detalles sobre la implementación interna de la misma. Por ejemplo `ArrayEstudiantes` no es nombre válido.

Métodos

- Se deben utilizar los verbos autorizados para su codificación.
- El nombre de los métodos debe iniciar con minúscula la primera palabra, las siguientes palabras inician en mayúscula, sin separadores.

- La primera palabra del nombre de los métodos debe ser un verbo en infinitivo y debe representar una acción o comportamiento de la clase.
- El nombre del método debe describir claramente el comportamiento del mismo.
- En lo posible no se deben usar verbos genéricos aplicables a todo como: procesar, gestionar, manejar. Ejemplo: procesarEstudiante(), gestionarCliente(), en este caso el verbo no aclara el cometido real del método.

Paquetes

- El nombre de los paquetes debe iniciar con minúscula la primera palabra, las siguientes palabras inician en mayúscula, sin separadores.
- La estructura de los paquetes es la siguiente:

co.edu.uis.[sistema].[aplicación].[módulo].[caso de uso]

Ejemplo:

co.edu.uis.financiero.egresos.contratacion.ordenarPrestacionServicio.

- La estructura para el paquete donde van a estar las entidades comunes para todos los sistemas es el siguiente:

co.edu.uis.sistema.entidades

- La estructura para el paquete donde van a estar los servicios comunes para todos los sistemas es el siguiente:

co.edu.uis.sistema.servicios

Variables

- Para el nombre de las variables utilizar palabras completas en singular (máximo tres palabras). El nombre deber ir en plural cuando la variable representa una lista o un conjunto de elementos.
- Si las palabras no son suficientes para la descripción, se debe hacer un comentario, al frente de la variable.
- Las constantes (final) van en mayúscula sostenida separando las palabras por guión de piso (_).
- Para los argumentos, deben iniciar siempre con la letra “a” y posteriormente el nombre según lo establecido anteriormente.
- Para las instancias de las clases, las entidades llevan el mismo nombre de la clase, solo que la palabra inicial va en minúscula. Si se necesita más de una instancia, para diferenciarlas se debe adicionar una palabra que identifique el rol que desempeña. Ejemplo: Estudiante estudiantePregrado, Estudiante estudiantePostgrado.
- La variable del EntityManager se debe llamar em.
- La variable de tipo FacesMessages se debe llamar facesMessages.

Librerías (JAR)

El nombre de las librerías no sigue el mismo estándar de las clases. Éstas se deben escribir en minúscula, separando cada palabra con un guión (-). Ejemplo: recursos-humanos.jar.

Nombres de archivos

Se sigue el mismo estándar descrito en las reglas de sintaxis generales.

4.2.4 Documentación

La documentación relacionada con el diseño del sistema reside en la base de datos de Enterprise Architect.

La documentación del código fuente se debe hacer en cada una de las clases, siguiendo el estándar de JAVADOC y documentando la definición de la clase, descripción de los métodos get y set y descripción de los parámetros de entrada y salida de cada uno de los métodos que componen la clase.

4.2.5 Capa de Presentación

Plantilla principal

La plantilla principal de una página es la siguiente:

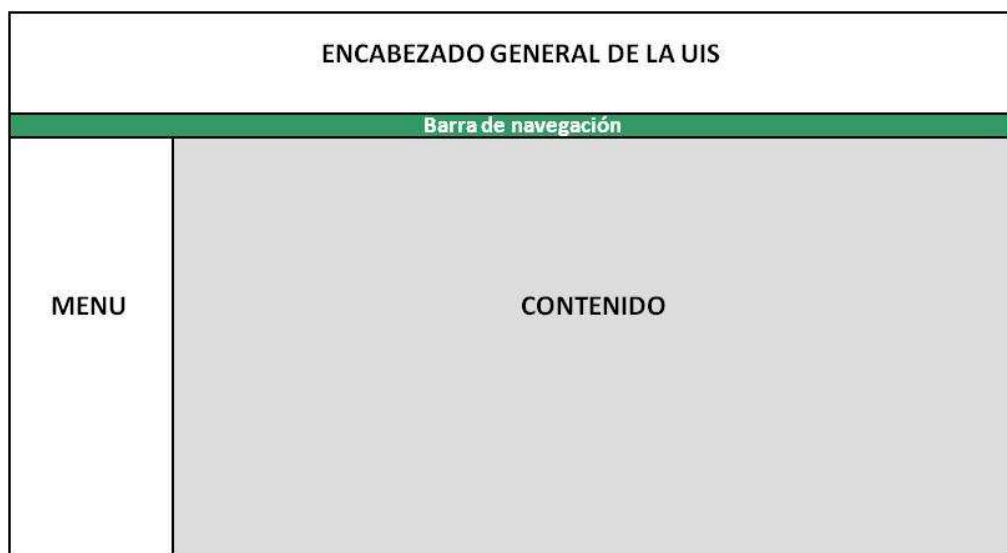


Figura 14. Plantilla Principal División de Servicios de Información

Contenido

Esta sección se refiere al caso de uso implementado. La estructura de esta sección es:

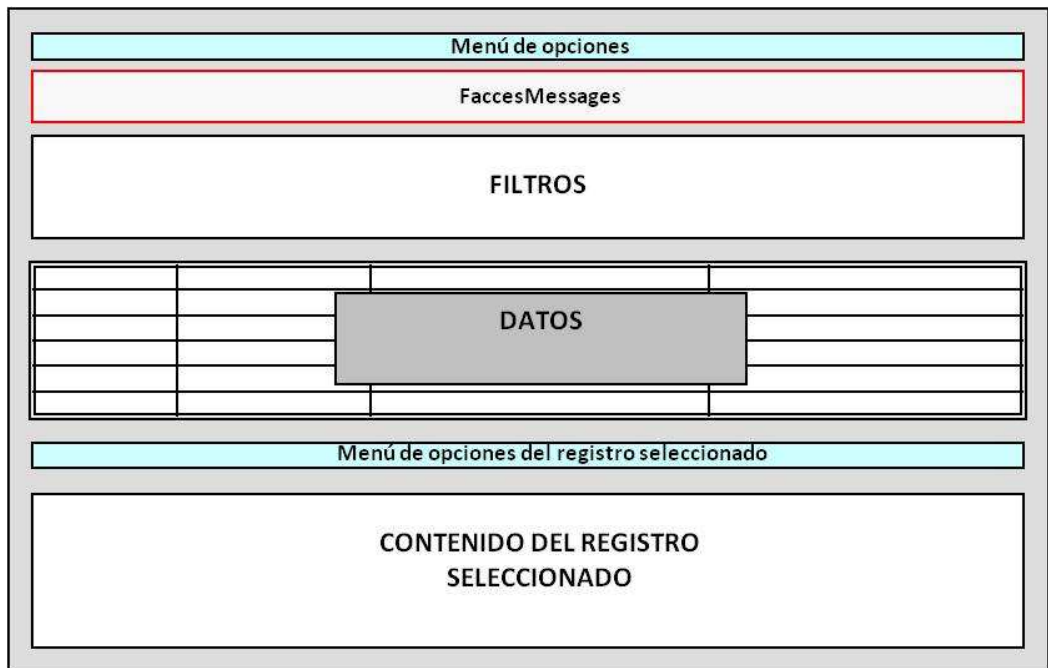


Figura 15. Plantilla de Contenido División de Servicios de Información.

Para las páginas que muestran formularios:

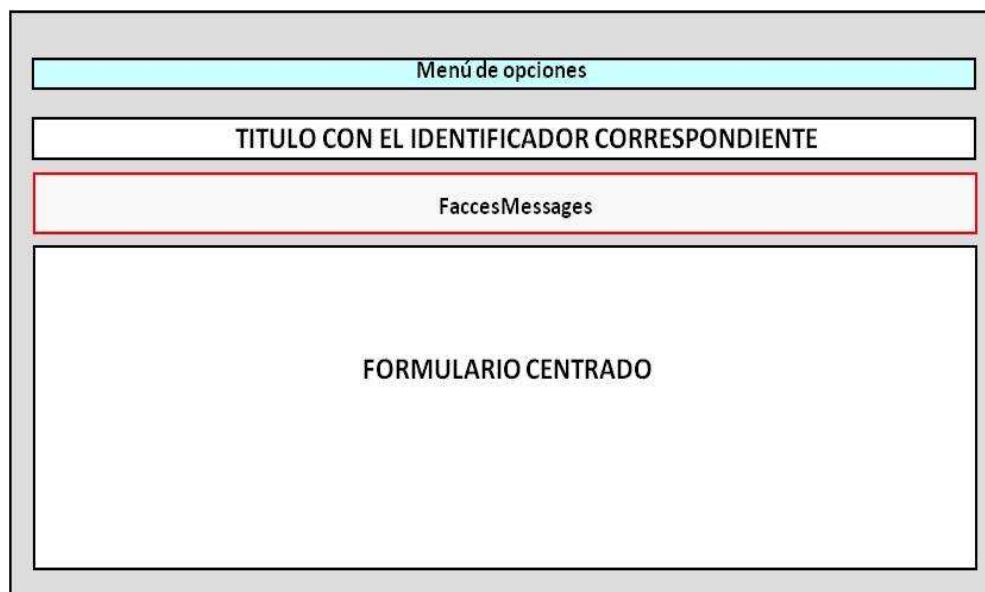


Figura 16. Plantilla de Contenido con Formulario. División de Servicios de Información.

Paginación

Para la paginación se debe tener en cuenta el tipo de consulta que se va a realizar y la memoria consumida por ésta en el servidor de base de datos. De acuerdo a esto la aplicación decide el número de registros máximos que debe retornar la consulta.

Por ejemplo, se quiere consultar los estudiantes de la sede Bucaramanga. De acuerdo al análisis realizado, no hay un consumo alto de memoria, por lo que se decide retornar 100 estudiantes máximos. Esto indica que cuando se implemente el método de consulta en JPQL, el parámetro `setMaxResults` debe tener el valor de 100.

Texto

Existen cinco tipos de plantillas para mostrar texto en la página. Éstas se encuentran dentro de la carpeta Plantillas.

- etiqueta.xhtml: Texto o datos.
- etiquetaColumna.xhtml: El título de la columna en una tabla
- titulo.xhtml: Títulos
- etiquetaNavegacion.xhtml: Utilizado en la barra de navegación
- mostrar.xhtml: Permite visualizar dos etiquetas (etiqueta, dato) al mismo tiempo. Su objetivo es la de visualizar datos como si fuera un formulario.

La sintaxis para utilizar las plantillas son:

```
<s:decorate template=" ../Plantillas/[nombrePlantilla]">
  <ui:define name="label">
    #{FormatoWeb.mostrarMensaje('primerNombre', true)}
  </ui:define>
</s:decorate>
```

Donde nombrePlantilla puede ser etiqueta.xhtml, etiquetaColumna.xhtml, etiquetaTitulo.xhtml o etiquetaNavegacion.xhtml.

Para la plantilla mostrar.xhtml:

```
<s:decorate template=" ../Plantillas/mostrar.xhtml">
  <ui:define name="label">
    #{FormatoWeb.mostrarMensaje('primerNombre', true)}
  </ui:define>
  <h:outputText
    value="#{formatoWeb.usuario.primerNombre}" />
</s:decorate>
```

Edición

Para capturar cualquier tipo de dato en una caja de texto se debe utilizar la plantilla edicion.xhtml de la siguiente manera:

```
<s:decorate id="dcr[identificador]"
    template="/Plantillas/edicion.xhtml">

    <ui:define name="label">
        #{FormatoWeb.mostrarMensaje('primerNombre', true)}
    </ui:define>

    <h:inputText id="txt[nombreCajaDeTexto]" value="[valor]"
        required="true">

        <a:support event="onblur"
            reRender="dcr[identificador]"/>
    </h:inputText>

</s:decorate>
```

dcr[identificador] es el nombre del elemento decorate. Por ejemplo: dcrPrimerNombre.

txt[nombreCajaDeTexto] es el nombre de la caja de texto. Por ejemplo: txtPrimerNombre.

Tablas estáticas

Se debe usar el control panelGrid de Java Server Faces.

Tablas dinámicas

Se debe usar el control dataTable de Rich Faces, agregando las siguientes líneas de programación en su definición:

```
onRowMouseOver="this.style.backgroundColor='#E1E1E1'"
```

```
onRowMouseOut="this.style.backgroundColor='{a4}Skin.tableBackgroundColor}'
```

Las cuales indican el color de la fila cuando el puntero del mouse esta sobre ésta.

Listas desplegadas

Para cualquier tipo de lista se debe utilizar el control de Java Server Faces.

Mensajes del sistema

Los mensajes del sistema son textos enviados por los EJB de sesión, ya sea de confirmación de una acción o errores en el procedimiento. Dichos errores se deben mostrar en la etiqueta FacesMessages la cual debe estar definida al comienzo de la página después del menú si existiera éste. El código es el siguiente:

```
<h:messages globalOnly="true" styleClass="message"/>
```

Para mostrar errores de validación en los formularios, éstos deben aparecer al frente de cada control y no globalmente.

4.3 DIAGRAMAS UML

Dentro del proceso de análisis y diseño del sistema de información se generaron los diagramas UML, que se presentarán a continuación, es de destacar que durante el transcurso del proyecto el diseño se fue adaptando a medida que se fue perfeccionando el prototipo inicial.

Aquí se presentarán los esquemas básicos de cada uno de los diagramas que se generaron, producto del diseño, con el fin de ilustrar el fondo y la forma como fueron producidos.

La documentación completa del diseño, se encuentra en el CD anexo, que hace parte de este informe.

4.3.1 Diagrama de Casos de Uso.

4.3.1.1 Identificación de los Actores

Dentro del sistema de Preguntas, Quejas, Reclamos y Sugerencias se pueden identificar los siguientes actores, cada uno con roles, acciones y permisos distintos:

- **Ciudadano:** Son todas aquellas personas que registran las solicitudes dentro del sistema, las cuales se pueden clasificar e identificar en el sistema como Estudiantes UIS, Funcionarios UIS, Proveedores UIS y ciudadanía en general.
- **Personal de Control interno con permisos totales (Personal CI-UIS AT)**
Son los funcionarios de la Dirección de Control Interno y Evaluación de Gestión de la UIS, que tienen acceso a todas las opciones administrativas del sistema, entre las cuales están: Mantenimiento Tablas soporte, Leer y

darle acción a cada una de las Solicitudes, revisar las respuestas de las Unidades.

- **Personal Control Interno con permisos restringidos (Personal CI-UIS AR):** Son los funcionarios de la Dirección de Control Interno y Evaluación de Gestión de la UIS, que sólo tienen acceso a la consulta e impresión de las estadísticas.
- **Personal Unidad Académico Administrativa UIS (Personal UAA):** Son todos los jefes de cada una de las Unidades Académico Administrativas de la UIS.

4.3.1.2 Casos de uso por Actor

Actor: Ciudadano

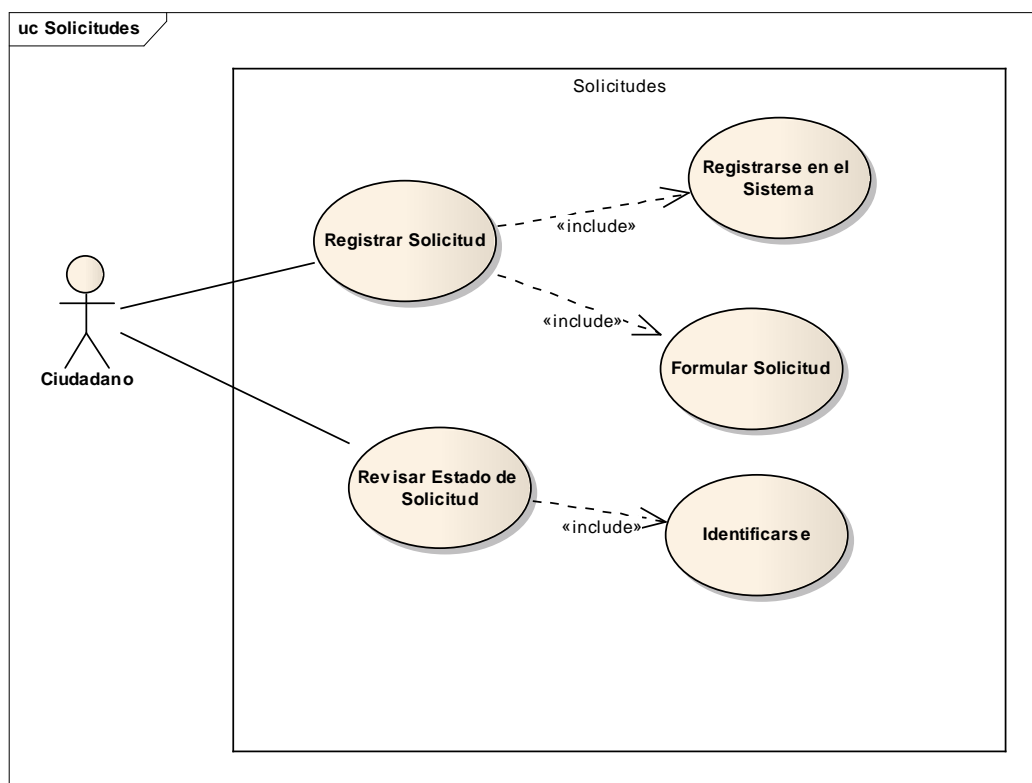


Figura 17. Casos de Uso Ciudadano

Se identifican dos casos de uso asociados a Ciudadano:

- Registrar Solicitud
- Revisar Estado Solicitud

Aquí se documenta la información del caso de uso: Registrar Solicitud, los demás casos de uso se pueden encontrar documentados en el CD que hace parte de éste documento en la carpeta \Sistema PQRS\Documentación UML\Diagramas Casos de Uso\Ciudadano

Caso de uso: Registrar Solicitud

El Ciudadano Registra en el Sistema su solicitud: Queja, Reclamo, pregunta o Sugerencia, describiendo a cabalidad lo que quiere manifestar

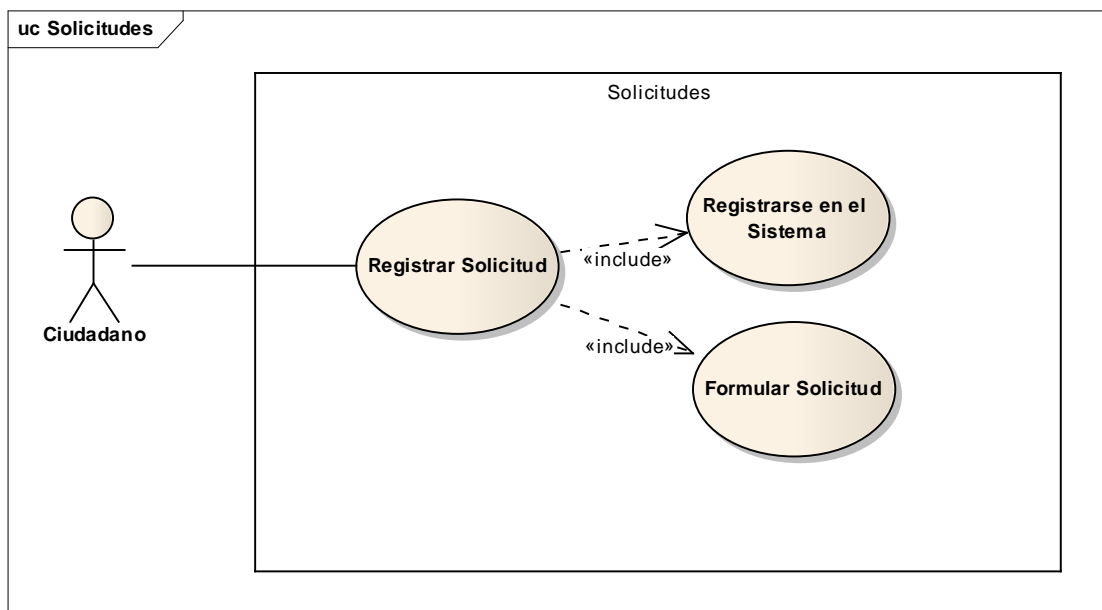


Figura 18. Caso de Uso: Registrar Solicitud

CASO DE USO	REGISTRAR SOLICITUD	
Objetivo	El Ciudadano Registra en el Sistema su Solicitud : Queja , Reclamo, Pregunta o Sugerencia, describiendo a cabalidad lo que quiere manifestar	
Actores	Ciudadano	
Pre condiciones	Haber ingresado sus datos personales, y describir la solicitud	
Post Condiciones	Generar el recibo de confirmación de la solicitud con el correspondiente número de radicado, enviarlos al email	
Pasos	Acción del Actor	Respuesta del Sistema
	Ingresar al Sistema PQRS	
		Presenta el Formulario de Inscripción de la Solicitud
	Digita sus Datos personales, el tipo de Solicitud, el motivo de la solicitud y la descripción de la solicitud, en caso de existir ingresará también Archivos para soportar su Solicitud	
		Presenta la opción de Registrar o Cancelar
	Registra la Solicitud	
		Genera un número de radicado y lo envía como soporte al email suministrado
	Cancelar Solicitud	
	Presenta de nuevo el formulario de Inscripción de la Solicitud	
Variaciones		
Extensiones	El sistema accede a la base de datos, y consulta si el usuario se registro previamente, si es así le trae los datos sin necesidad de volverlos a digitar, si es un Estudiante, valida que lo sea y muestra su información, igual si es un Funcionario de la Universidad	

Tabla 2. Descripción Caso de uso: Registrar Solicitud

Actor: Personal de Control interno con permisos totales (Personal CI-UIS AT)

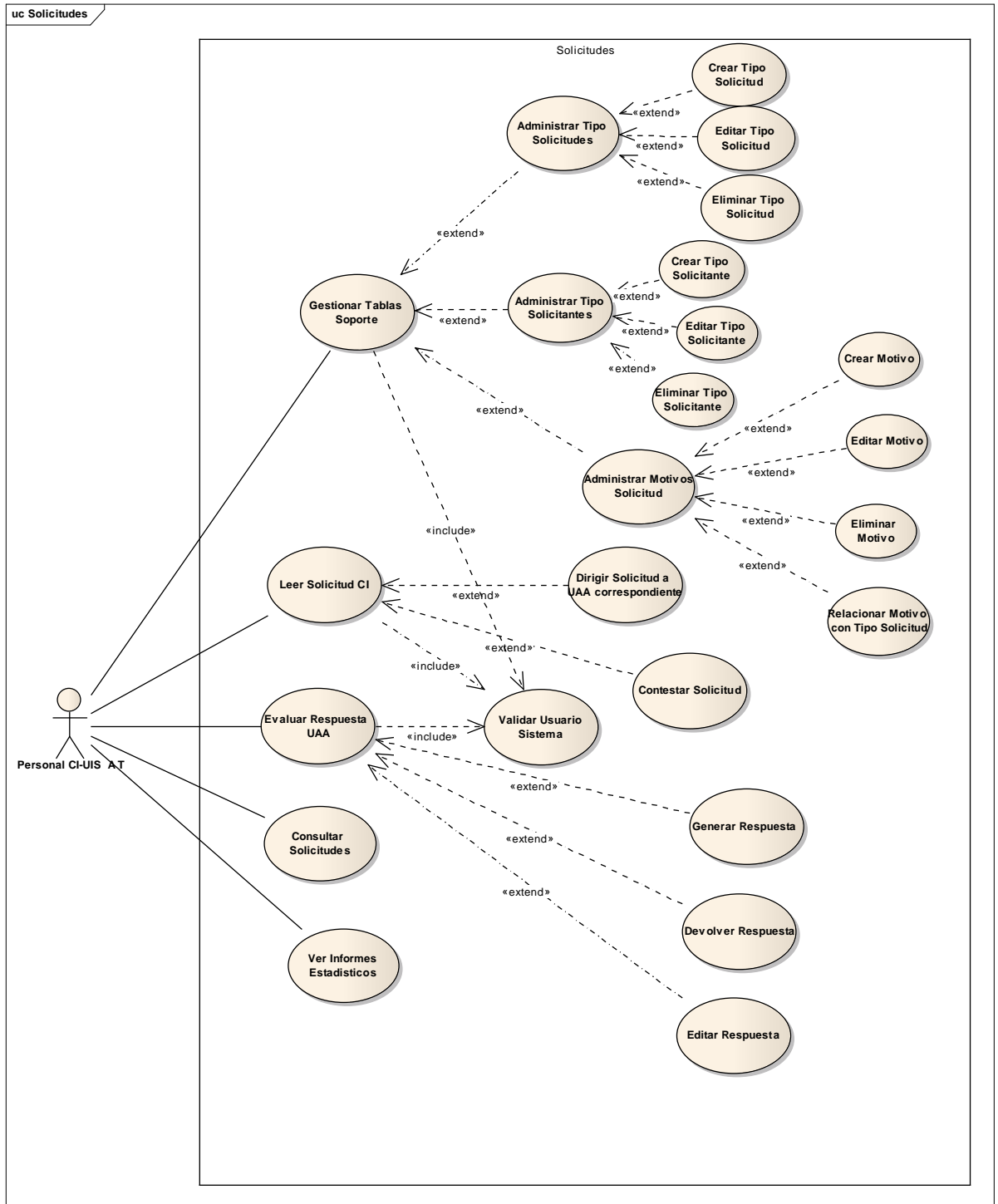


Figura 19. Casos de Uso asociados con Personal Control Interno Permisos totales

Sin duda alguna éste es el actor que mayor interacción tiene con el sistema y el cual define parámetros importantes para el funcionamiento del mismo. Se identifican los siguientes casos de uso para Personal de control Interno con permisos totales:

- Gestionar Tablas Soporte
- Leer Solicitud
- Evaluar Respuesta UAA
- Consultar Solicitudes
- Ver Informes Estadísticos

Caso de Uso: Gestionar Tablas Soporte

Mediante este caso de uso el usuario tendrá la opción de mantener actualizado el sistema, a través de las tablas que lo soportan, y de ésta manera darle dinamismo al sistema, disminuyendo la dependencia del usuario con el desarrollador.

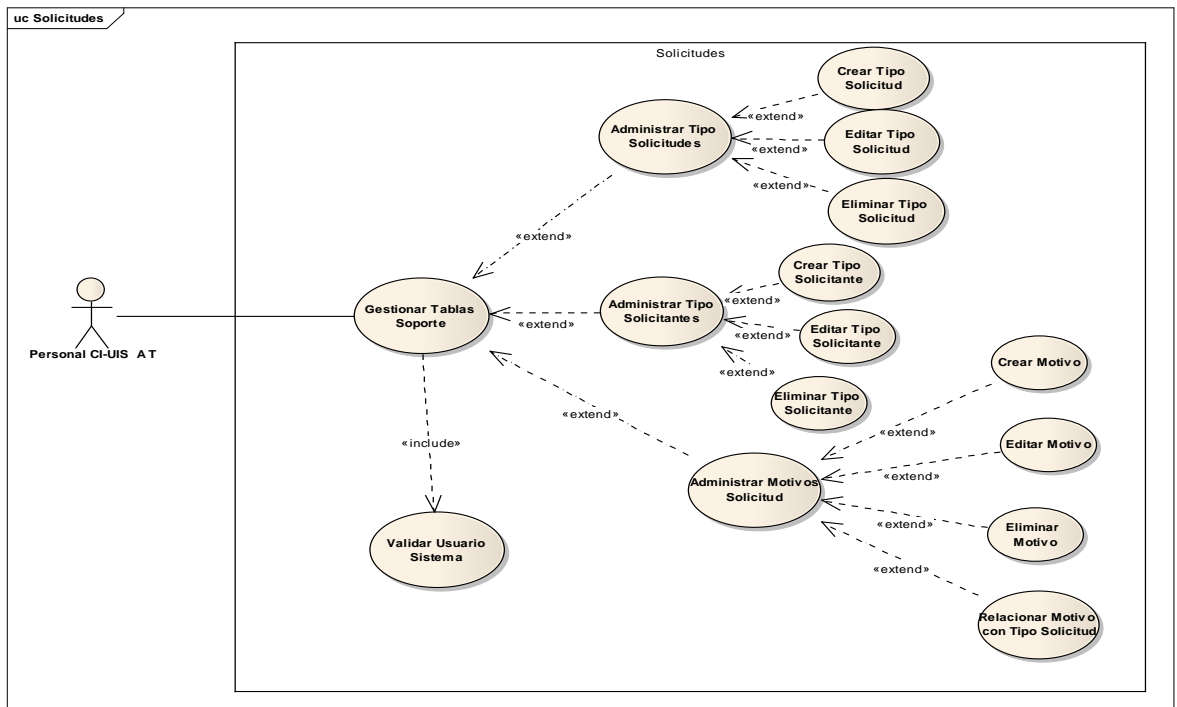


Figura 20. Caso de Uso. Gestionar tabla Soporte

CASO DE USO	GESTIONAR TABLAS SOPORTE	
Objetivo	Mantener actualizada la información de las tablas de la base de datos que sirven de soporte al sistema, y darle flexibilidad al usuario al poder tener control sobre la funcionalidad del sistema	
Actores	Personal CI-UIS AT	
Pre condiciones	Validarse como Administrador del Sistema, Personal CI-UIS AT	
Post Condiciones	Tablas Soporte Actualizadas	
Pasos	Acción del Actor	Respuesta del Sistema
	Ingresar al módulo de Administración	
		Le presenta las diferentes tablas soporte que puede gestionar
	Escoge una tabla	
		Le presenta la opción de Crear, Editar, Modificar o crear una relación de un registro según la tabla a soportar
	Escoge la opción que desea realizar, según las presentadas por el sistema	
		Se actualizan los registros en la base de datos y muestra los cambios realizados en pantalla
Variaciones	Se puede cambiar de tabla soporte, y realizar los mismos procedimientos.	
Extensiones	Se accede desde el sistema a la base de datos, para crear, editar o modificar un registro, según la tabla escogida.	

Tabla 3. Descripción de Caso de Uso. Gestionar Tablas Soporte

Caso de Uso: Leer solicitud

En este caso de uso el actor tiene la posibilidad de leer lo manifestado por el ciudadano y decidir que va a hacer con la solicitud, si la redirige a una Unidad Académico Administrativa ó la contesta él mismo.

En caso de que decida responderla, la respuesta es registrada con el código de la Dirección de Control Interno y Evaluación de Gestión

El actor tiene la posibilidad de dejar sin acción la solicitud, poniéndola en estado de “No procede” en el caso que éste estime conveniente.

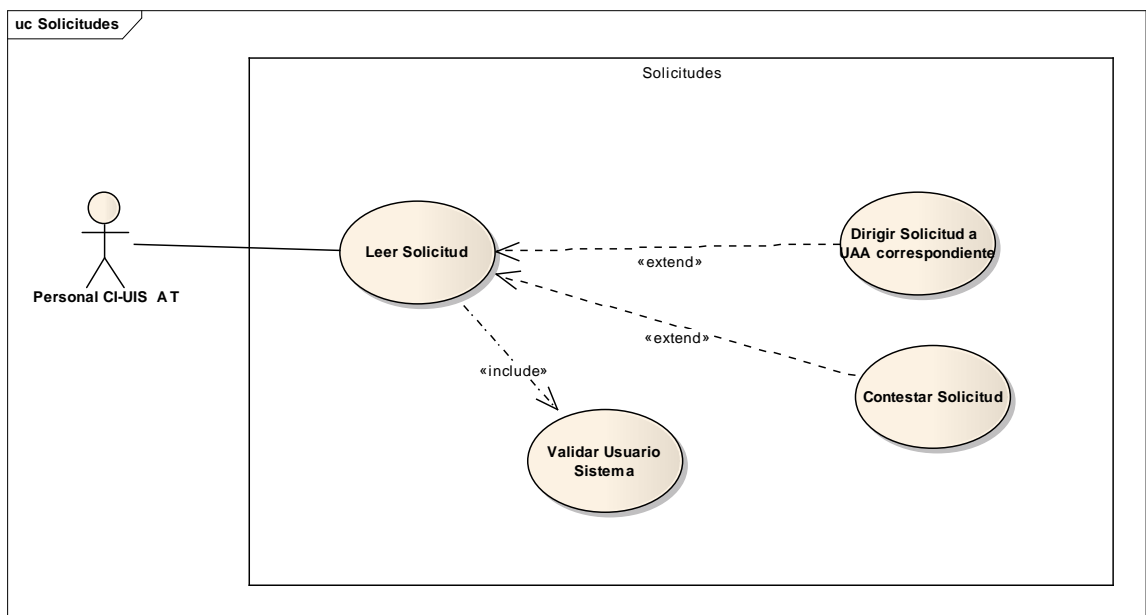


Figura 21. Caso de Uso Leer Solicitud

CASO DE USO	LEER SOLICITUD	
Objetivo	El actor lee la solicitud, registrada por el ciudadano y decide la acción a tomar con ella.	
Actores	Personal CI-UIS AT	
Pre condiciones	Validarse como Administrador del Sistema, Personal CI-UIS AT	
Post Condiciones	Solicitud con respuesta o redirigida a la Unidad académica administrativa que sea conveniente.	
Pasos	Acción del Actor	Respuesta del Sistema
	Ingresar al Sistema en el modulo de las Solicitudes sin Asignar	
		Muestra las solicitudes que no se les ha realizado ninguna acción (pendientes o confirmadas)
	Escoge una Solicitud	
		Muestra el detalle de la solicitud enviada por el solicitante, le da la opcion de Enviarla a la UAA o de responderla directamente
	Enviar Solicitud a UAA	
		Muestra la Fecha y la hora en la que se ha redirigido la solicitud a la UAA
	Responder Solicitud	
	Envía la respuesta al Solicitante al email y publica la respuesta.	
Variaciones	El usuario puede optar por hacer las acciones en una ocasión posterior.	
Extensiones	El sistema Accede a la base de datos para ingresar el registro de: La respuesta de la Solicitud en caso de que sea respondida, y el envió a la UAA en caso de que ésta sea redirigida.	

Tabla 4. Descripción Caso de Uso: Leer Solicitud

Caso de Uso: Evaluar Respuesta UAA

En este caso de uso el actor, decidirá la pertinencia de la respuesta que envía la UAA, y podrá:

- Devolver la respuesta a la UAA para verificarla
- Editar la respuesta que fue enviada.
- Publicar la respuesta, tal cual como llegó de la UAA.

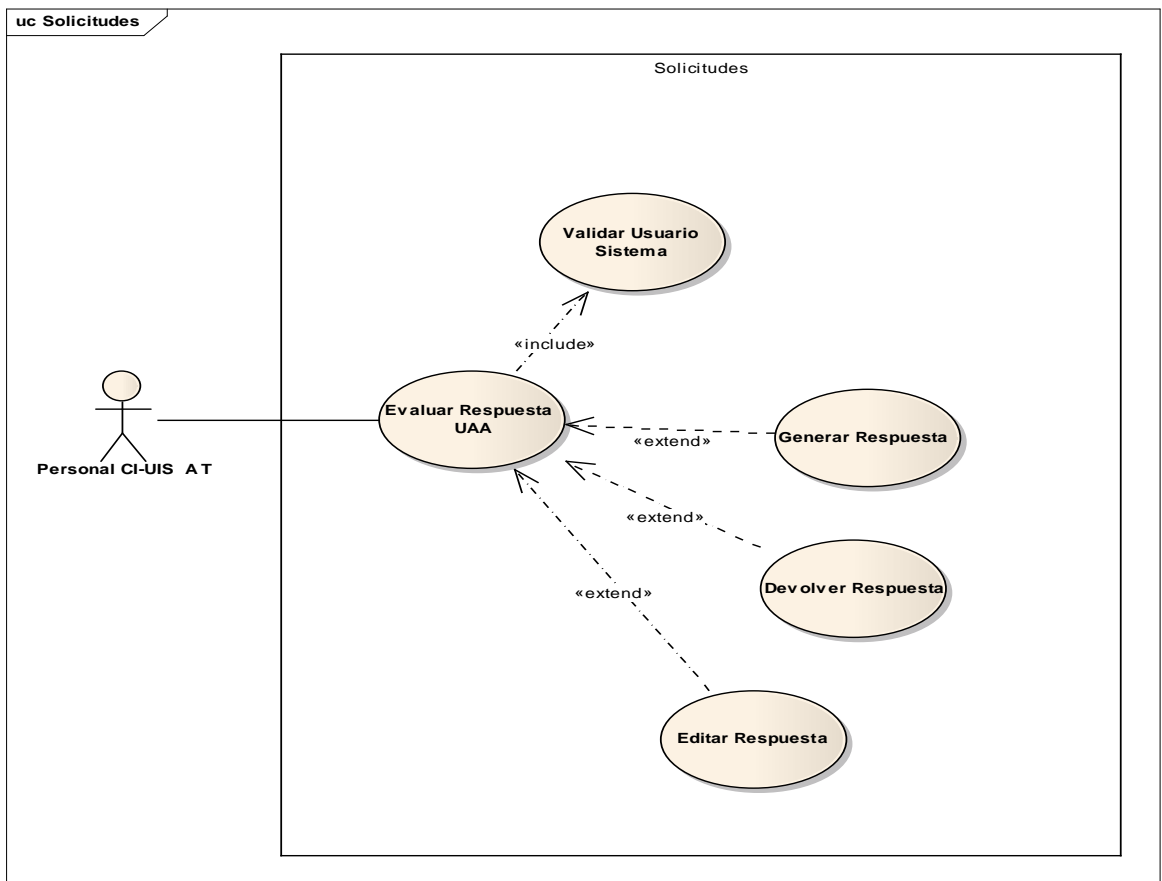


Figura 22. Caso de Uso. Evaluar Respuesta UAA

CASO DE USO	EVALUAR RESPUESTA UAA	
Objetivo	Decidir la pertinencia de la respuesta enviada por la UAA	
Actores	Personal CI-UIS AT	
Pre condiciones	Validarse como Administrador del Sistema, Personal CI-UIS AT	
Post Condiciones	Respuesta Publicada y enviada al solicitante ó la respuesta devuelta en la UAA para verificaciones	
Pasos	Acción del Actor	Respuesta del Sistema
	Ingresar al Sistema	
		Muestra las solicitudes que ya han sido respondidas por las UAA
	Publicar respuesta	
		Envía la respuesta al email del solicitante y regresa al listado de solicitudes respondidas
	Devolver Respuesta a UAA	
		Se genera un pantallazo donde se expresa la fecha y hora en que se ha devuelto la Solicitud
	Editar Respuesta	El Sistema da la opción de editar la respuesta que llego para publicarla
Variaciones		
Extensiones	El sistema accede a la base de datos para ingresar el registro de: la devolución a la UAA en caso de que ésta sea devuelta	

Tabla 5. Descripción Caso de Uso: Evaluar Respuesta UAA.

La documentación completa, de los casos de uso en los que interviene éste actor se encuentra en el CD anexo, que forma parte integral del presente documento en la carpeta: \Sistema PQRS\Documentación UML\Diagramas Casos de Uso\Personal CI-UIS AT\

Actor: Personal control interno, permisos restringidos. (Personal CI-UIS AR)

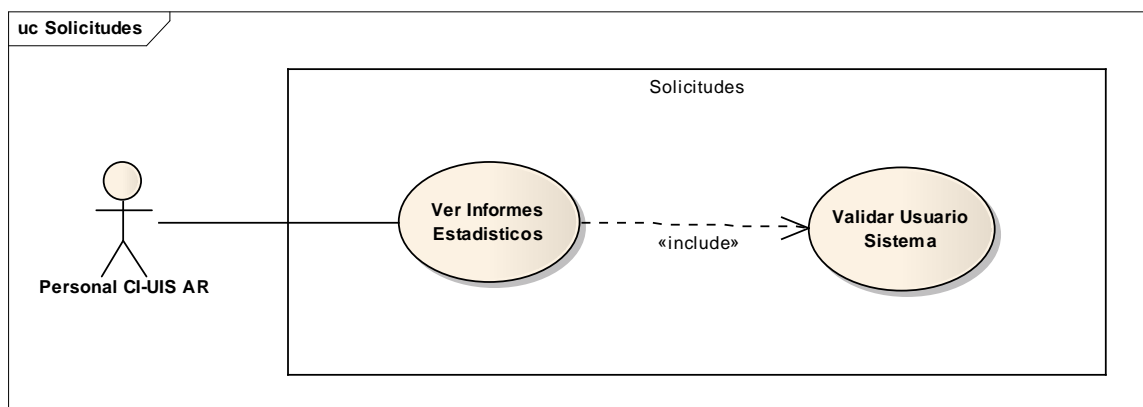


Figura 23. Caso de Uso: Ver Informes Estadísticos. Personal CI-UIS AR

Para éste actor identificamos el caso de uso:

- Ver Informes Estadísticos.

En este caso de uso el actor tendrá la posibilidad de consultar y generar los informes estadísticos que genera el sistema de información.

Podrá filtrar los datos por varios tipos de criterio, pero en ningún caso este actor tendrá la posibilidad de hacer consultas del historial de registro de solicitudes, ni hacer ningún tipo de acción con cada una de ellas.

CASO DE USO	VER INFORMES ESTADISTICOS	
Objetivo	Ver e Imprimir los informes estadísticos que genera el sistema	
Actores	Personal CI-UIS AR	
Pre condiciones	Validarse como Personal de Control interno con acceso restringido, Personal CI-UIS AR	
Post Condiciones	Generar Informes	
Pasos	Acción del Actor	Respuesta del Sistema
	Ingresa al Sistema	
		Muestra Pantalla inicial de las estadísticas sin filtrar
	Filtra datos	
		Genera nuevamente las estadísticas de acuerdo a los filtros dados por el actor
	Generar Informes	
		Se generar los archivos de informes de acuerdo con los datos filtrados que ingresó el actor.
Variaciones	El usuario puede escoger y cambiar a su gusto los filtros que le imponga al sistema	
Extensiones	El sistema accede a la base de datos para contabilizar los datos que se piden y generar los reportes estadísticos	

Tabla 6. Descripción Caso de Uso. Ver Informes Estadísticos

La documentación completa, de los casos de uso en los que interviene éste actor se encuentra en el CD anexo, que forma parte integral del presente documento en

la carpeta: \Sistema PQRS\Documentación UML\Diagramas Casos de Uso\Personal CI-UIS AR\

Actor: Personal UAA

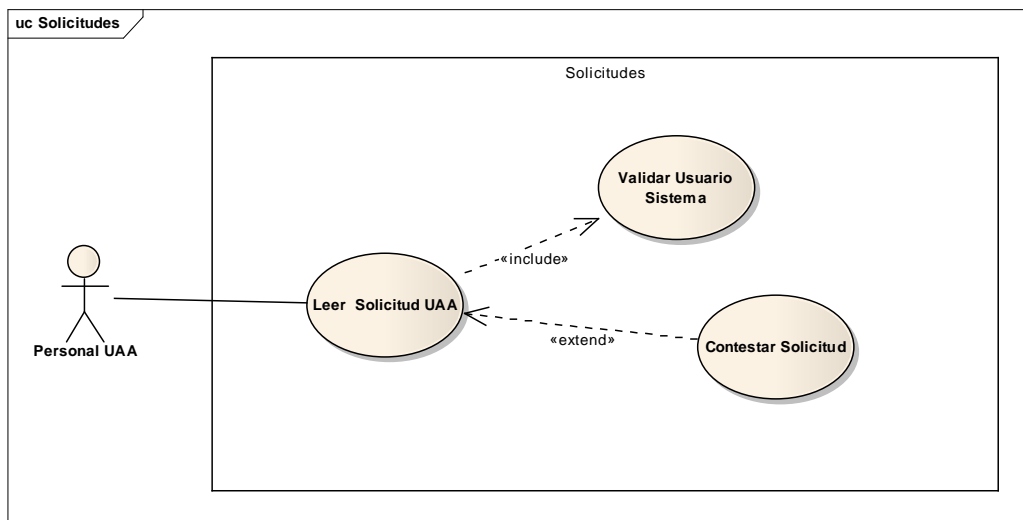


Figura 24. Casos de Uso Personal UAA.

A éste actor se le puede identificar el caso de uso:

- Leer Solicitud UAA

En este caso de uso el actor tiene la posibilidad de leer lo manifestado por el ciudadano, su única opción es responderla de manera inmediata o dejar para responderla en otro momento, se le da la posibilidad de adjuntar archivos como soporte a la respuesta.

Las solicitudes que están pendientes por respuesta, puede ser por dos motivos, uno es que haya sido redirigida una nueva solicitud a ésta unidad o una respuesta

dada con anterioridad por la unidad, no fue satisfactoria y ha sido devuelta para su revisión.

CASO DE USO	LEER SOLICITUD UAA	
Objetivo	El usuario de la Unidad Académico administrativa, lee la solicitud, que permitirá dar una posible respuesta a la misma	
Actores	Personal UAA	
Pre condiciones	Validarse en el sistema como Jefe de Unidad Académico administrativa	
Post Condiciones	Haber permitido leer la Solicitud, en su defecto enviar respuesta	
Pasos	Acción del Actor	Respuesta del Sistema
	Ingresa al Sistema	
		Muestra la pantalla inicial, donde se muestran las solicitudes pendientes por leer y las que han sido devueltas por control interno para revisión
	Escoge una solicitud	
		Muestra en pantalla el detalle de la solicitud que se escogió, permite la opción de responderla
	Responder la solicitud	
		Se envía la respuesta a Control Interno para ser analizada
Variaciones	El usuario puede No responder la solicitud que se le ha dirigido en ese instante	
Extensiones	El sistema accede a la base de datos para traer las solicitudes que le corresponden a esa unidad académico administrativa	

Tabla 7. Caso de Uso. Leer Solicitud UAA.

La documentación completa, de los casos de uso en los que interviene éste actor se encuentra en el CD anexo, que forma parte integral del presente documento en la carpeta: \Sistema PQRS\Documentación UML\Diagramas Casos de Uso\Personal UAA\

4.3.2 Diagramas de Secuencias

A continuación se presentaran los diagramas de secuencia, que amplían de manera considerable la visión general sobre el funcionamiento e interacción del sistema con el usuario ya sea éste ciudadano, personal de control interno o personal de las unidades académico administrativas.

Se elaboraron los diagramas de secuencia de los casos de uso, considerados importantes, con el fin de ampliar la información que éstos brindan y lograr abstraer mejor la idea de cada elemento del sistema.

Aquí se presentan algunos de ellos, la documentación completa sobre los diagramas de secuencia elaborados se encuentra en el CD anexo que forma parte del presente documento en la carpeta: \Sistema PQRS\Documentación UML\Diagramas Secuencia\

Para el Caso de Uso. Registrar Solicitud.

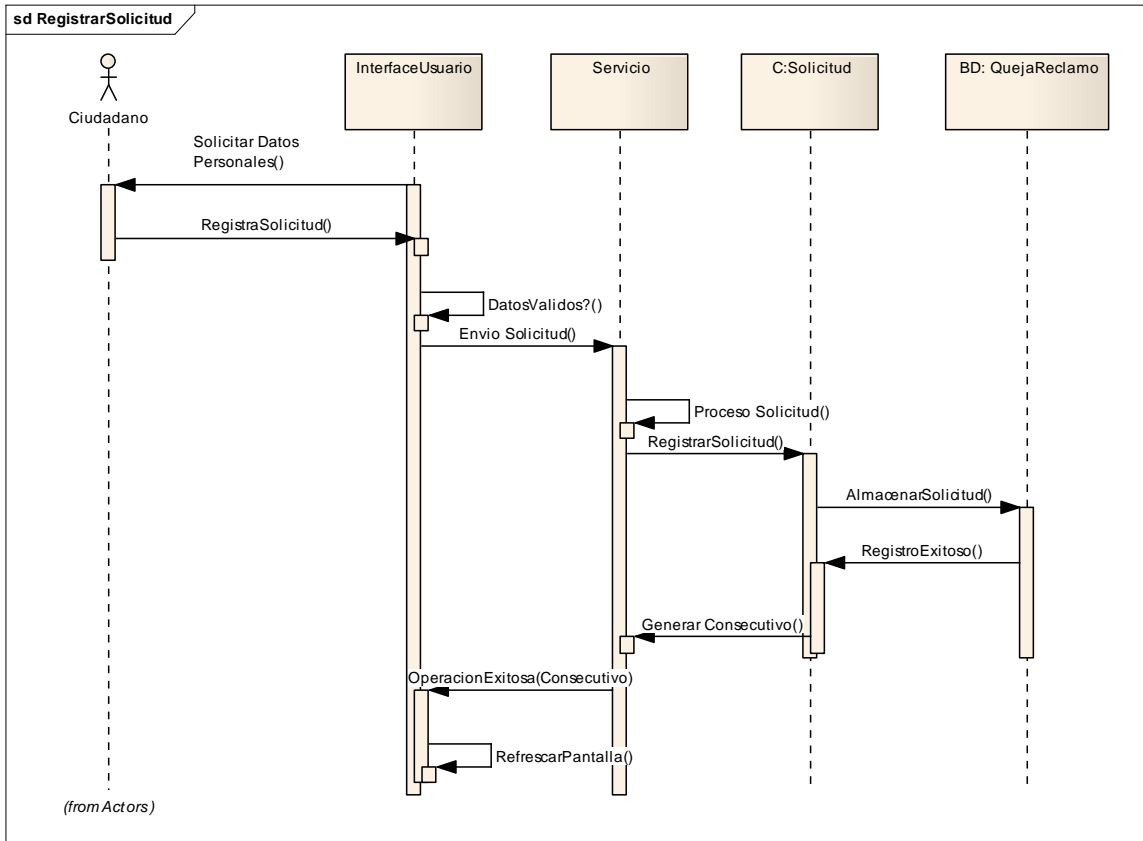


Figura 25. Diagrama de Secuencia para el caso de uso. Registrar Solicitud.

Para el Caso de Uso: Leer Solicitud CI.

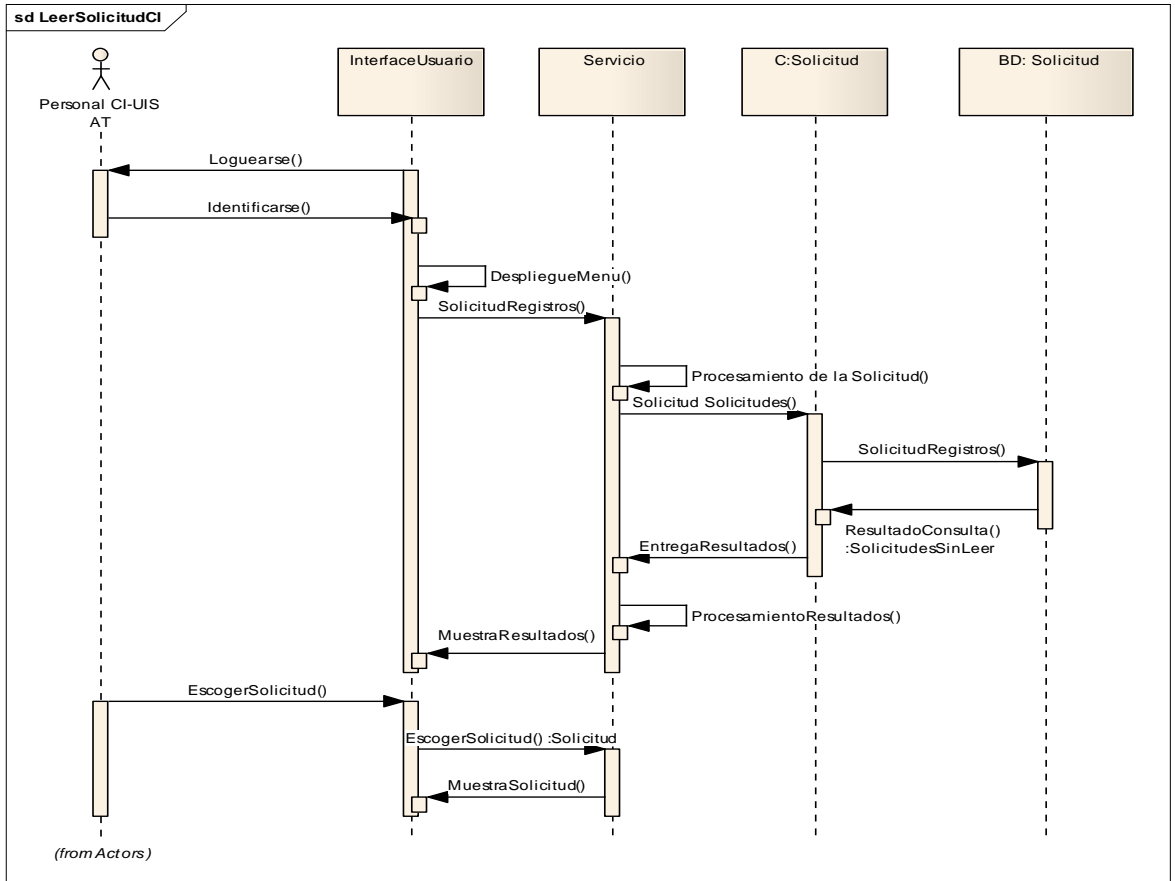


Figura 26. Diagrama de Secuencia. Caso de uso Leer Solicitud CI.

Para el Caso de Uso: Leer Solicitud UAA.

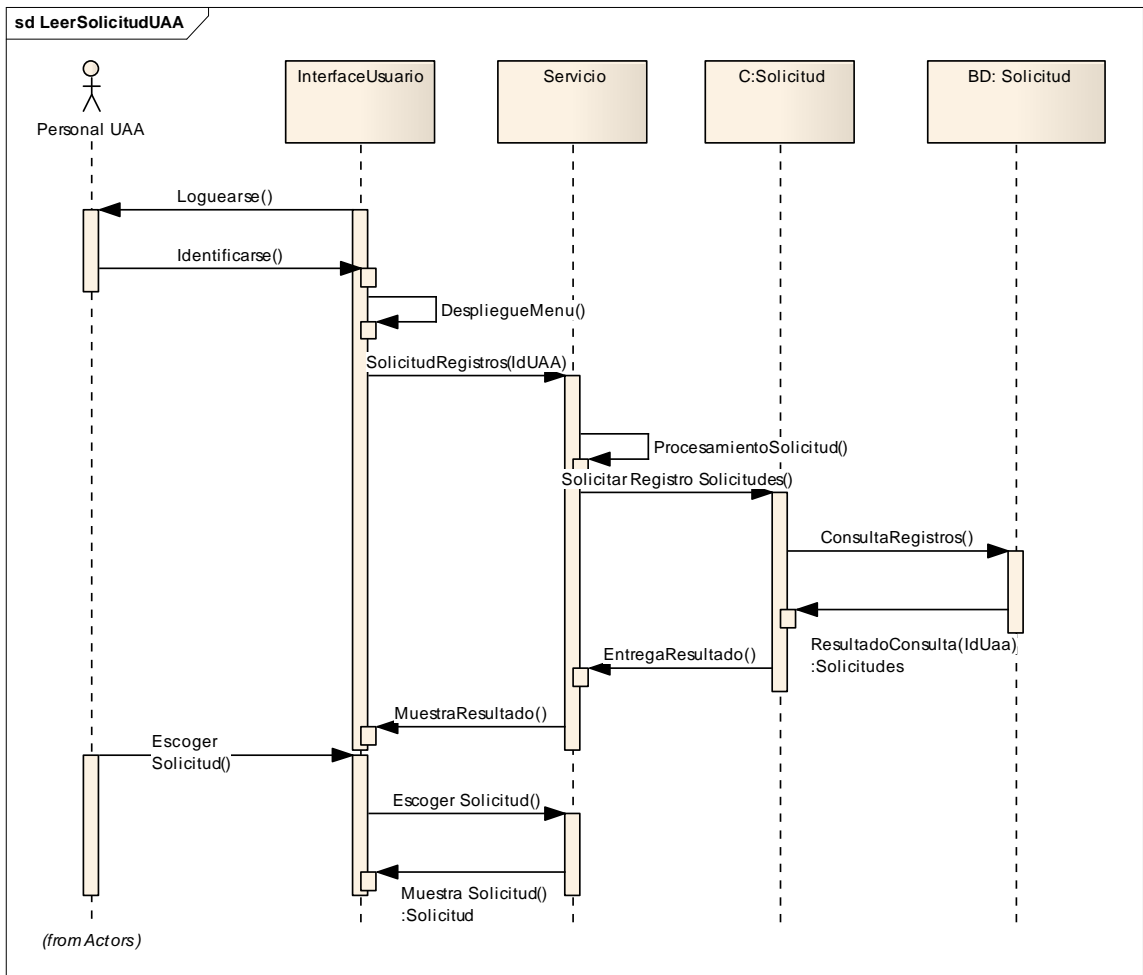


Figura 27. Diagrama de Secuencia para el caso de Uso. Leer Solicitud UAA.

4.3.3 Diagrama de Clases

Aquí se presenta el esquema general del diagrama de clases, resultante en la etapa de análisis del proyecto.

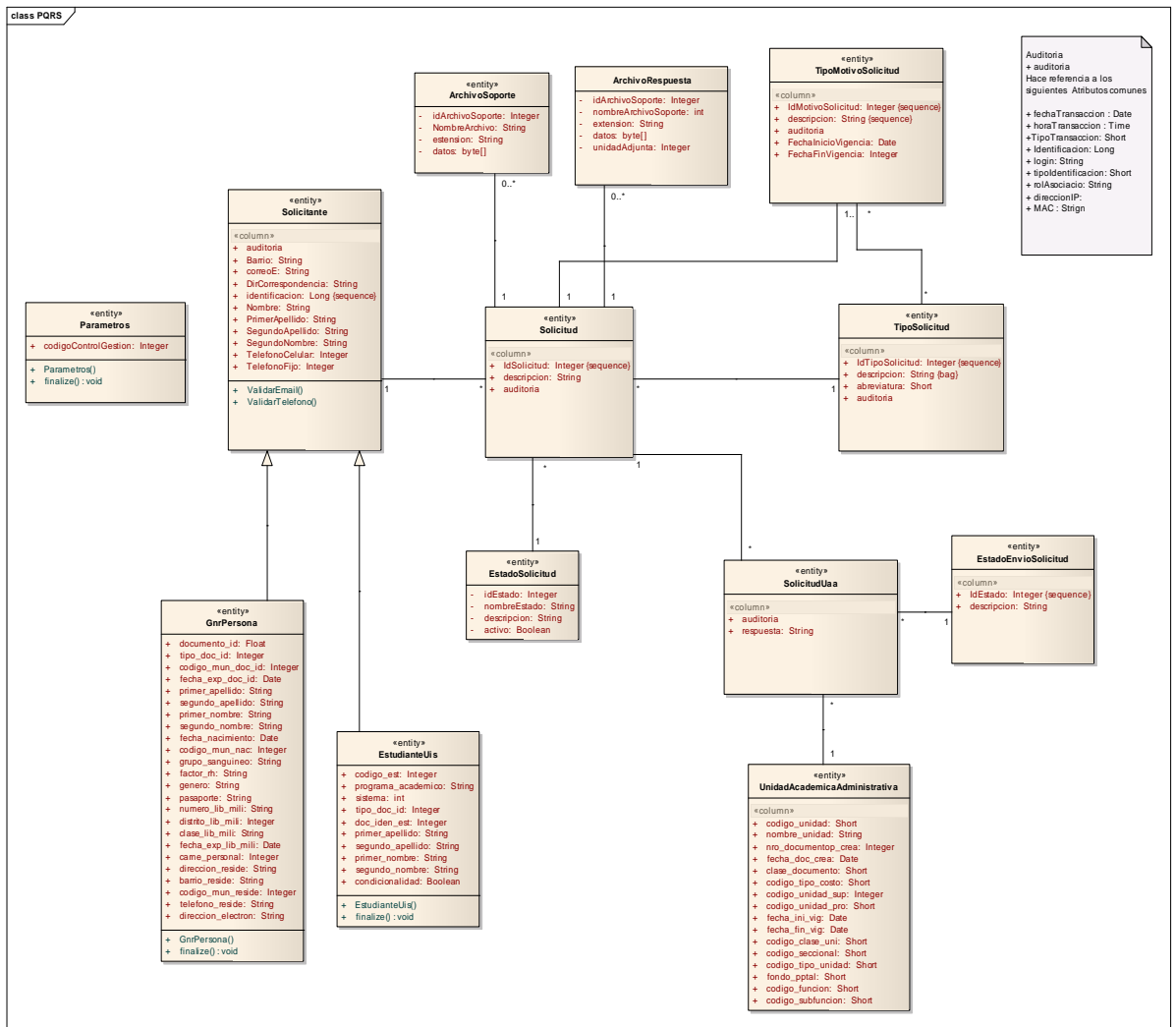


Figura 28. Diagrama de Clases sistema PQRS.

La documentación completa de este diagrama nombres de las clases, atributos de ellas y características se encuentran documentadas en el CD anexo a este documento en la carpeta Sistema PQRS\ Documentación UML\Diagrama de Clases\.

- archivo_respuesta
- solicitante
- solicitud
- estado_solicitud
- solicitud_uaa (relación entre solicitud y las uaa, envío de solicitud)
- estado_envio_solicitud
- tipo_solicitante
- tipo_solicitud
- tipo_motivo_solicitud
- motivo_tipo (relación entre los tipos de motivo de solicitud y los tipos de solicitud)
- parámetros

Las entidades que identificamos en rojo, son aquellas que no son propias del sistema pero se utilizan para el funcionamiento, hacen parte de otros sistemas de información de la Universidad Industrial de Santander, éstas son:

- datos_personal_per
- unidad_academico_administrativa
- estudiantes_uis
- municipios
- departamentos
- tipos_doc_identidad

La documentación completa de este diagrama nombres de las clases, atributos de ellas y características se encuentran documentadas en el CD anexo a este documento en la carpeta Sistema PQRS\ Documentación UML\Diagrama de Datos\.

4.3.5 Diagrama de Actividades

Se definieron diagramas de actividades como soporte a los casos de uso, considerados fundamentales en el desarrollo del sistema.

Con el fin de ampliar la información del procedimiento que debe desarrollar esos casos de uso, aquí se especifican cada uno de ellos.

La documentación completa de cada diagrama se encuentra en el CD anexo a éste documento en Sistema PQRS\ Documentación UML\Diagrama de Datos\.:

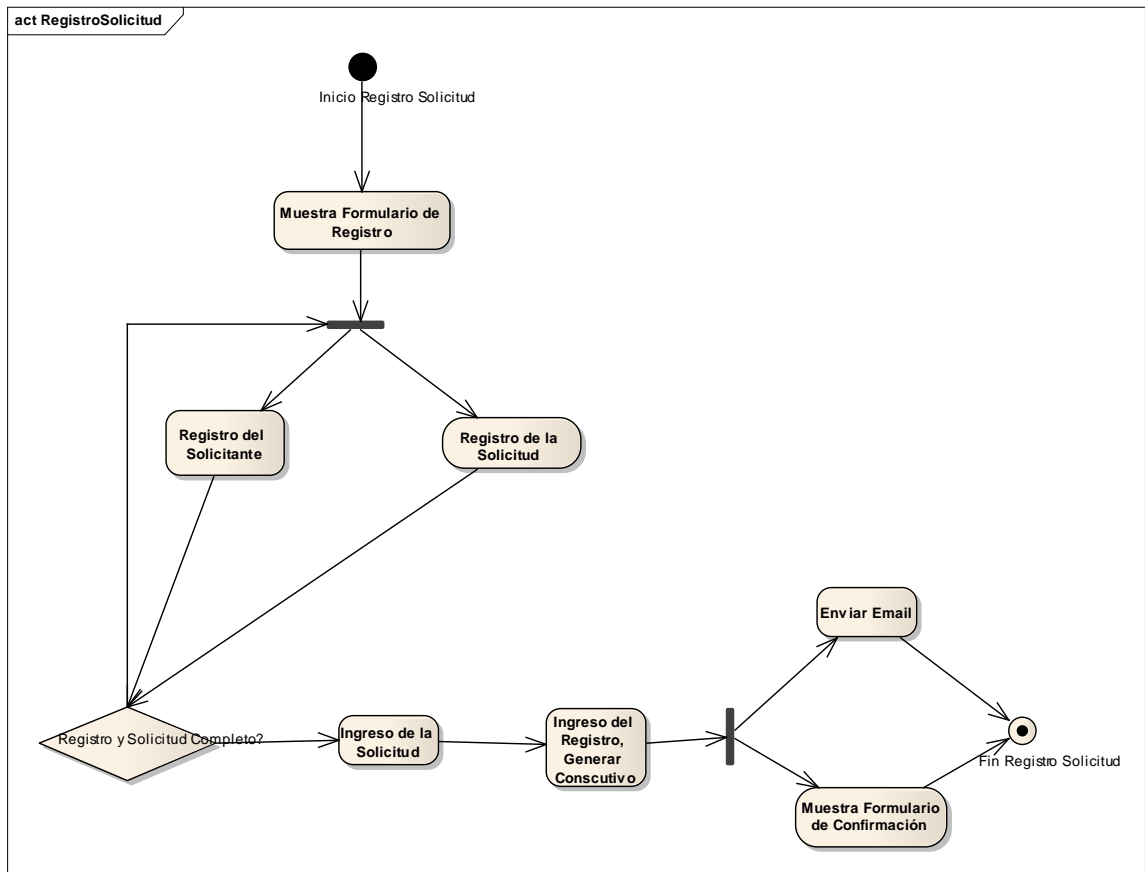


Figura 30. Diagrama de Actividades Registro de la Solicitud.

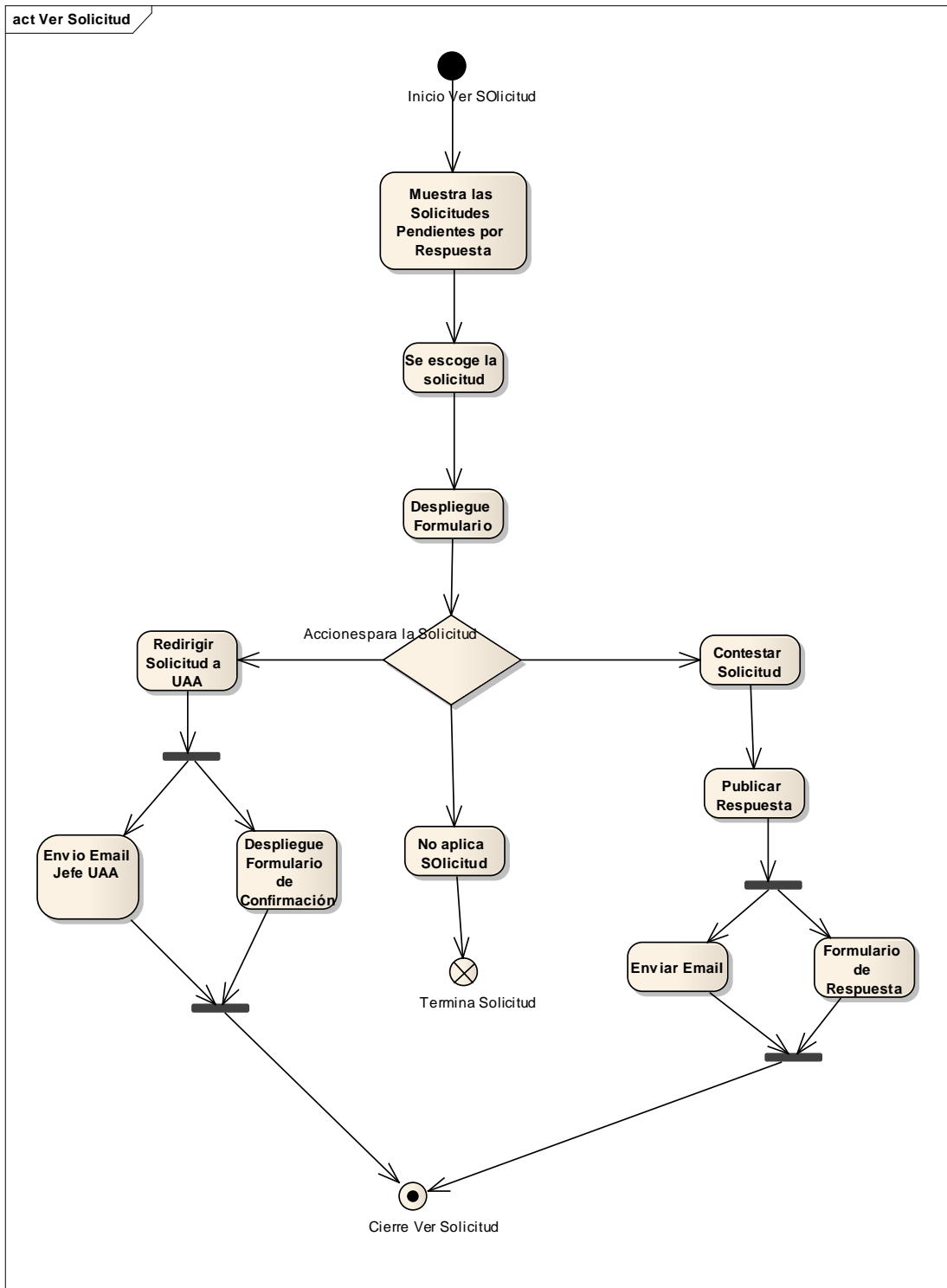


Figura 31. Diagrama Actividades Ver Solicitud.

4.4 PROTOTIPO INICIAL

4.4.1 Generalidades del prototipo inicial

El proceso de construcción del prototipo inicial comenzó desde el momento en manifestar el interés de desarrollar el proyecto y continuó con una serie de encuentros periódicos, con el Ingeniero Jorge Vidal Rodríguez, profesional adscrito a la Dirección de Control Interno y Evaluación de Gestión de la Universidad Industrial de Santander, quien será el directo responsable de la gestión del nuevo sistema de información, y quien además actuó como codirector de éste proyecto.

En éstos encuentros se escuchó y analizó cada uno de los requerimientos deseados para el sistema y se fue estructurando un prototipo no funcional que serviría como carta de navegación en el transcurso del desarrollo del sistema.

El prototipo no funcional se creó como una página web básica sin ningún tipo de programación de fondo, que mostraría gráficamente el esquema de navegación del sistema, de ésta manera se definieron detalles de alta importancia tales como la información que el usuario quiere ver, la que desea ocultar, el contenido de cada una de las vistas y los datos que se le van a solicitar a cada uno de los usuarios que interactuarían con el sistema.

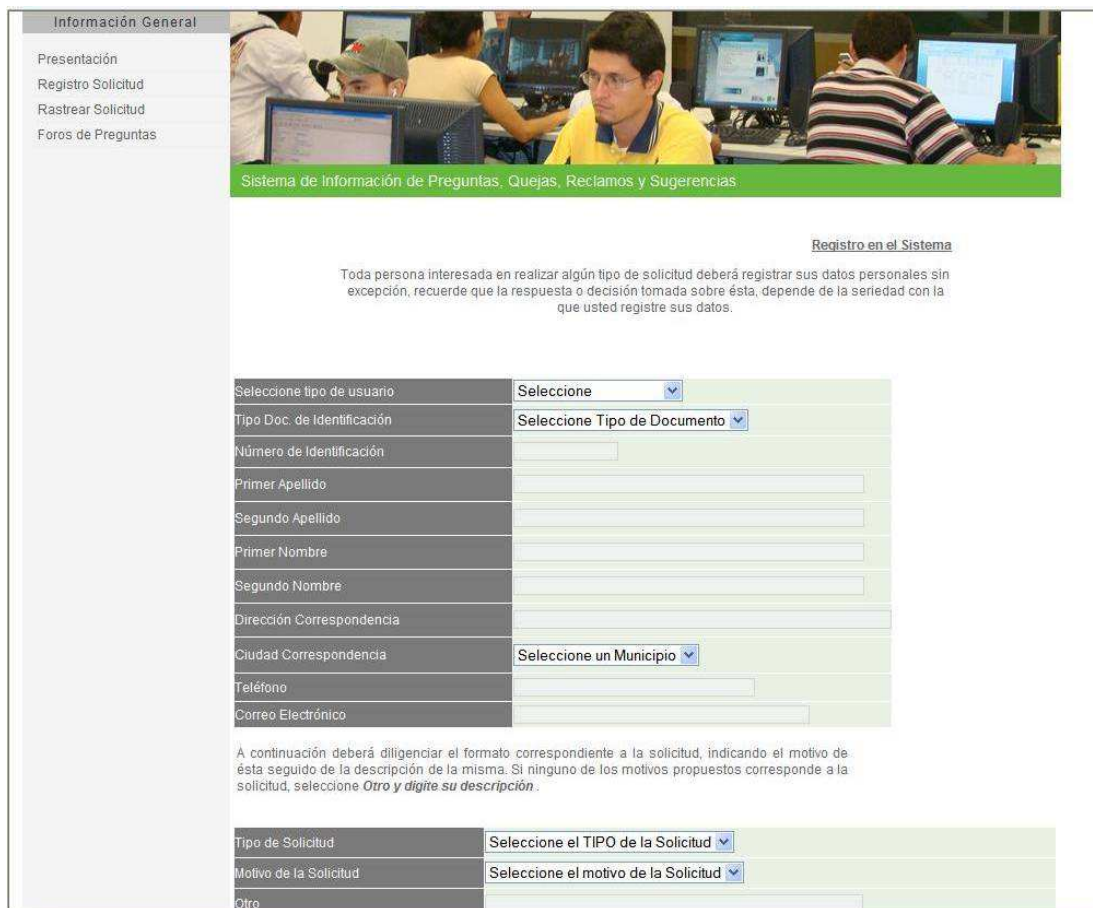
Se manejo un esquema cíclico de perfeccionamiento y refinamiento de tal manera que se llegara a un nivel de acercamiento detallado de lo que quiere alcanzar el usuario con el sistema, de esta manera se logra que el proceso de refinamiento del prototipo funcional sea mínimo agilizando de esta manera el ciclo de vida del proyecto.

El tiempo destinado a esta etapa fue bastante amplio, ya que del resultado obtenido dependería la eficiencia del desarrollo y programación del sistema.

Durante el transcurso de la construcción del prototipo se definió todo el diseño integrado por los diagramas de casos de uso, de clases, de datos, de secuencia y de actividades.

4.4.2 Interfaz resultado del prototipo inicial

Se presenta a continuación apartes de la interfaz resultante de la construcción del prototipo inicial (no funcional).



The screenshot shows a web interface for a system titled "Sistema de Información de Preguntas, Quejas, Reclamos y Sugerencias". On the left, there is a navigation menu with the following items: "Información General", "Presentación", "Registro Solicitud", "Rastrear Solicitud", and "Foros de Preguntas". The main content area features a header image of people at computers, followed by a green banner with the system name. Below this is a link for "Registro en el Sistema" and a paragraph of text explaining the registration process. The registration form consists of several fields: "Seleccione tipo de usuario" (dropdown), "Tipo Doc. de Identificación" (dropdown), "Número de Identificación" (text), "Primer Apellido" (text), "Segundo Apellido" (text), "Primer Nombre" (text), "Segundo Nombre" (text), "Dirección Correspondencia" (text), "Ciudad Correspondencia" (dropdown), "Teléfono" (text), and "Correo Electrónico" (text). Below the form, there is a paragraph of text and two more dropdown menus: "Tipo de Solicitud" and "Motivo de la Solicitud".

Información General

- Presentación
- Registro Solicitud
- Rastrear Solicitud
- Foros de Preguntas

Sistema de Información de Preguntas, Quejas, Reclamos y Sugerencias

[Registro en el Sistema](#)

Toda persona interesada en realizar algún tipo de solicitud deberá registrar sus datos personales sin excepción, recuerde que la respuesta o decisión tomada sobre ésta, depende de la seriedad con la que usted registre sus datos.

Seleccione tipo de usuario: Seleccione

Tipo Doc. de Identificación: Seleccione Tipo de Documento

Número de Identificación: [input]

Primer Apellido: [input]

Segundo Apellido: [input]

Primer Nombre: [input]

Segundo Nombre: [input]

Dirección Correspondencia: [input]

Ciudad Correspondencia: Seleccione un Municipio

Teléfono: [input]

Correo Electrónico: [input]

A continuación deberá diligenciar el formato correspondiente a la solicitud, indicando el motivo de ésta seguido de la descripción de la misma. Si ninguno de los motivos propuestos corresponde a la solicitud, seleccione *Otro* y *digite su descripción*.

Tipo de Solicitud: Seleccione el TIPO de la Solicitud

Motivo de la Solicitud: Seleccione el motivo de la Solicitud

Otro: [input]

Figura 32. Vista de Registro de Solicitud. Prototipo Inicial

Vista de la Dirección de Control Interno, respecto a las nuevas solicitudes que han ingresado al sistema.

Sistema de Información de Preguntas, Quejas, Reclamos y Sugerencias

UNIDAD ACADEMICO ADMINISTRATIVA: CONTROL INTERNO Y EVALUACION DE GESTION.

Radicado N°	Nombre	Tipo Solicitud	Motivo	Fecha - Hora Registro	Estado	Accion
000000001	Christian Fernando Pacheco Jaimes	Q - R	Motivo	2010-07-06 05:41 p.m.	Sin Accion	Ver Soporte Ver Seguimiento
000000002	Eduardo Estevez Mendoza	P	Admisiones	2010-07-06 05:41 p.m.	Dirigida	Ver Soporte Ver Seguimiento
000000003	Claudia Victoria Correa Pugliese	S	Asesoría Contable	2010-07-06 05:41 p.m.	Sin Accion	Ver Soporte Ver Seguimiento

Salir

Figura 33. Vista entrada control Interno. Prototipo Inicial

Vista del detalle de la solicitud, por parte de personal de Control Interno con permisos totales.

Información General

- Inicio
- Revisar nuevas solicitudes
- Revisar Respuestas Unidades Solicitud
- Foros de Preguntas
- Estadísticas

Sistema de Información de Preguntas, Quejas, Reclamos y Sugerencias

Administración Solicitud

En éste módulo usted puede responder o dirigir la solicitud a la unidad academico administrativa

Número de identificación	Documento identificación
Nombre del Ciudadano	Nombre Ciudadano ... Aaaaaa Bbbbb Cocco
Dirección Correspondencia	Direccion
Ciudad Correspondencia	Ciudad nnnn
Teléfono	0000000000
Correo Electrónico	email@email.com
Tipo de Solicitud	(Queja - Reclamo , Pregunta o Sugerencia)
Numero Radicado	1234456677

Descripción de la Solicitud .

Tipo de Solicitud	(QUEJA- RECLAMO , PREGUNTA o SUGERENCIA)
Numero Radicado	
Motivo de la Solicitud	(DEPENDE DE SI ES QUEJA- RECLAMO , PREGUNTA o SUGERENCIA)

Descripción de la Solicitud

2000 caracteres

Figura 34. Vista de la Solicitud por parte de Control Interno. Prototipo Inicial.

Vista de las respuestas enviadas por las unidades académico administrativas.
Opciones para Control Interno.

Número de Identificación	Documento identificación
Nombre del Ciudadano	Nombre Ciudadano ... Aaaaaa Bbbbb Ccccc
Dirección Correspondencia	Direccion
Ciudad Correspondencia	Ciudad nnnn
Teléfono	0000000000
Correo Electrónico	email@email.com
Tipo de Solicitud	(Queja - Reclamo , Pregunta o Sugerencia)
Numero Radicado	1234456677

Descripcion de la Solicitud

Tipo de Solicitud	(QUEJA- RECLAMO , PREGUNTA o SUGERENCIA)
Numero Radicado	
Motivo de la Solicitud	(DEPENDE DE SI ES QUEJA- RECLAMO , PREGUNTA o SUGERENCIA)

Descripcion de la Solicitud	

2000 caracteres

Respuesta de la Unidad: Unidad Encargada

Fecha y Hora de Respuesta: Fecha y Hora

--	--

2000 caracteres

Figura 35. Vista Respuesta enviada por UAA. Prototipo Inicial.

El prototipo inicial completo puede ser encontrado y consultado en el CD anexo al presente documento en la carpeta \Sistema PQRS\Prototipo Inicial\.

4.5 PROTOTIPO FINAL

4.5.1 Generalidades del Prototipo Final

Al término del desarrollo del sistema de información de Preguntas, Quejas, Reclamos y Sugerencias, se obtuvo un prototipo que se adapta plenamente a las características que se han definido desde el comienzo del desarrollo del proyecto.

Tomando como punto de partida el prototipo no funcional, se logró consolidar el producto definitivo, que con un proceso mínimo de refinamiento por parte del cliente, después de un esquema de transición, reemplazará al sistema actual en la administración y trámite de las solicitudes que los diferentes ciudadanos deseen plantearle a las distintas dependencias de la Universidad Industrial de Santander.

El producto cumple todos y cada uno de los parámetros que se definieron en el documento de requerimientos que se menciona en el apartado 4.1 de este documento, y después de un proceso de evaluación de parte del cliente, ha sido aceptado por éste, manifestando su plena conformidad con el resultado.

A continuación se describirá de forma general las funcionalidades más importantes, clasificadas por módulos, propias del sistema de información de Preguntas, Quejas, Reclamos y Sugerencias de la Universidad Industrial de Santander

Modulo de registro de la solicitud

- El sistema permite el registro de la solicitud por parte de: Estudiantes, Empleados UIS, Proveedores y Ciudadanía en general, en caso de estar vinculado de manera directa a la universidad (Estudiantes , Empleados y Proveedores UIS) se confronta ésta información con las bases de datos internas de la UIS.

- Se le permite al solicitante un espacio de hasta 2000 caracteres para la descripción de la solicitud, y como novedad se le da la facilidad de adjuntar archivos que den soporte a lo descrito en la solicitud.

Registro Solicitud

Toda persona interesada en realizar algún tipo de solicitud deberá registrar sus datos personales sin excepción, recuerde que la respuesta o decisión tomada sobre ésta, depende de la seriedad con la que usted registre sus datos.

Seleccione el tipo de usuario

Primer nombre <input type="text"/>	Segundo nombre <input type="text"/>
Primer apellido <input type="text"/>	Segundo apellido <input type="text"/>
Teléfono Fijo <input type="text"/>	Dirección de Correspondencia <input type="text"/>
Municipio de Residencia <input type="text"/>	Barrio <input type="text"/>
Correo Electrónico <input type="text"/>	

Solicitud

A continuación deberá diligenciar el formato correspondiente a la Solicitud, seleccionando el motivo de la misma, seguido de la descripción respectiva. Si ninguno de los motivos propuestos corresponde, seleccione Otro y digite su descripción.

Seleccione el Tipo Solicitud

Motivo de la Solicitud

B I U ABC ↻ ↺ ↻

Adjuntar soporte (opcional)

Figura 36. Vista Registro Solicitud. Prototipo Final.

- Al correo electrónico del solicitante le llega un acuse de recibo de la solicitud con el número de radicado, con el cual posteriormente podrá hacer seguimiento al estado en que se encuentre la solicitud.

The screenshot shows a web interface with a blue header bar containing the text "Registro Solicitud Exitosa". Below the header, there is a yellow box with a message: "Esta informacion ha sido enviada al Correo electronico suministrado, por favor verifique en su bandeja de entrada que haya sido recibido satisfactoriamente. Con su documento de identificacion y numero de radicado, aqui proporcionados podra hacerle seguimiento a su solicitud." Below this message is a white box containing a list of details for the registration. The details are as follows:

Numero de Radicado	115
Número de Identificación	10986325
Nombre	CHRISTIAN FERNANDO PACHECO JAIMES
Correo Electrónico	cpachecojaimes@gmail.com
Tipo Solicitud	QUEJA - RECLAMO
Motivo de la Solicitud	ASESORIA CONFIABLE
Estado	Pendiente

At the bottom of the white box, there is a button labeled "Volver a Registro Solicitud".

Figura 37. Vista de la confirmación del recibo de la Solicitud. Prototipo Final.

- El solicitante podrá rastrear, en cualquier momento, el estado actual de la solicitud registrada, esto se debe hacer con el número de radicado asignado. Al ingresar al estado de la solicitud, el solicitante también tendrá la opción de visualizar el historial de solicitudes que fueron realizadas anteriormente por él.

Rastrear mi Solicitud

Ingrese su Identificación y Numero de Radicado, que le fueron asignadas al momento del registro, para verificar el estado de su solicitud.

Numero de Radicado Solicitud

Identificación Usuario

Figura 38. Vista Formulario Rastreo Solicitud. Prototipo Final.

Estado de la Solicitud

Numero de Radicado	110
Número de Identificación	10986325
Nombre	CHRISTIAN FERNANDO PACHECO JAIMES
Correo Electrónico	cpachecojaimes@gmail.com
Tipo Solicitud	SUGERENCIA
Motivo de la Solicitud	ASESORIA CONFIABLE
Estado	En Tramite
Descripcion Estado	Se encuentra en proceso de tramite en una o varias unidades academico administrativas

Figura 39. Vista Rastreo de la Solicitud. Prototipo Final

Modulo de Control Interno y Evaluación de Gestión

- El usuario de control Interno con permisos totales, podrá visualizar las nuevas solicitudes que no tengan ninguna acción realizada, clasificadas en: “Solicitudes sin Revisar” y “Solicitudes leídas y sin Asignar”.



The screenshot shows a web interface with two tabs: "Solicitudes sin Revisar" (selected) and "Solicitudes leídas y sin Asignar". Below the tabs is a table titled "Solicitudes sin Revisar". The table has seven columns: "Numero radicado", "Nombre", "Tipo Solicitud", "Motivo", "Fecha - Hora Registro", "Estado", and "Acciones". There are two rows of data, both for "CHRISTIAN FERNANDO PACHECO JAIMES" with "Q - R" type and "ASESORIA CONFIABLE" motive. The first row has a registration time of 9:23:43 AM and the second of 10:19:11 AM, both with a "Pendiente" status. Each row has an "Acciones" column with a small icon. Below the table is a pagination bar and the text "Total Registros: 2".

Numero radicado	Nombre	Tipo Solicitud	Motivo	Fecha - Hora Registro	Estado	Acciones
115	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	ASESORIA CONFIABLE	2010-10-26 9:23:43 AM	Pendiente	
116	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	ASESORIA CONFIABLE	2010-10-26 10:19:11 AM	Pendiente	

Total Registros: 2

Figura 40. Vista Solicitudes Nuevas Control Interno. Prototipo Final

- El usuario podrá escoger cualquier solicitud, ver su descripción completa, decidir si es pertinente enviarla a alguna unidad académico administrativa. Si es así, se enviará un correo electrónico al Jefe de la Unidad respectiva informándole la novedad. Control Interno, según su criterio, podrá dar respuesta definitiva a la solicitud, en caso de no proceder la solicitud, Control Interno tendrá la opción “No procede” para cerrar el trámite de la solicitud sin ningún otro tipo de acción.

Ver Detalle de la Solicitud

Número de Radicado	115
Número de Identificación	10986325
Nombre	CHRISTIAN FERNANDO PACHECO JAIMES
Dirección de Correspondencia	CALLE 8 N 2-87
Municipio de Residencia	
Barrio	LA CANTERA
Teléfono Fijo	6560083
Correo Electrónico	cpachecojaimes@gmail.com

Tipo Solicitud	QUEJA - RECLAMO
Motivo de la Solicitud	ASESORIA CONFIABLE
Fecha y hora de registro	2010-10-26 , 9:23:43 AM

Descripción

Esta es mi solicitud de queja- reclamo, por asesoria confiable.

B I U ABC ↶ ↷ ↻ ↺

Soportes Adjuntos		
Nombre	Descripción	Acción
115-PLAN DE PROYECTO PQRS.doc	PLAN DE PROYECTO PQRS.doc	Ver

Redirigir Solicitud
Responder Solicitud
No Aplica
Cancelar

Figura 41. Vista Detalle de la Solicitud. Prototipo Final.

- El usuario de Control Interno con permisos totales, podrá visualizar las respuestas que enviaron las unidades académico administrativas y como novedad tendrá acceso al historial del recorrido que ha tenido esa solicitud, ver cada una de las respuestas que se han generado (en caso de ser más de una) y decidir la pertinencia de la respuesta actual. En caso de no serlo podrá devolver la respuesta a la unidad respectiva, adjuntándole un comentario a la misma.
- Si la respuesta es pertinente, pero hay alguna corrección de menor trascendencia, podrá editarla y publicarla.

Solicitudes con Respuesta de Uaa						
Numero radicado ↕	Nombre ↕	Tipo Solicitud ↕	Motivo ↕	Fecha - Hora Registro ↕	Estado ↕	Acciones
105	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	ASESORIA CONFIABLE	2010-10-12 9:21:18 AM	En Tramite	
109	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	COMUNICACION	2010-10-13 1:15:17 PM	En Tramite	
110	CHRISTIAN FERNANDO PACHECO JAIMES	S	ASESORIA CONFIABLE	2010-10-14 12:50:38 PM	En Tramite	
114	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	ATENCION AL BENEFICIARIO	2010-10-15 3:59:31 PM	En Tramite	

Total Registros: 4

Figura 42. Vista Lista Solicitudes con Respuesta de UAA. Prototipo Final.

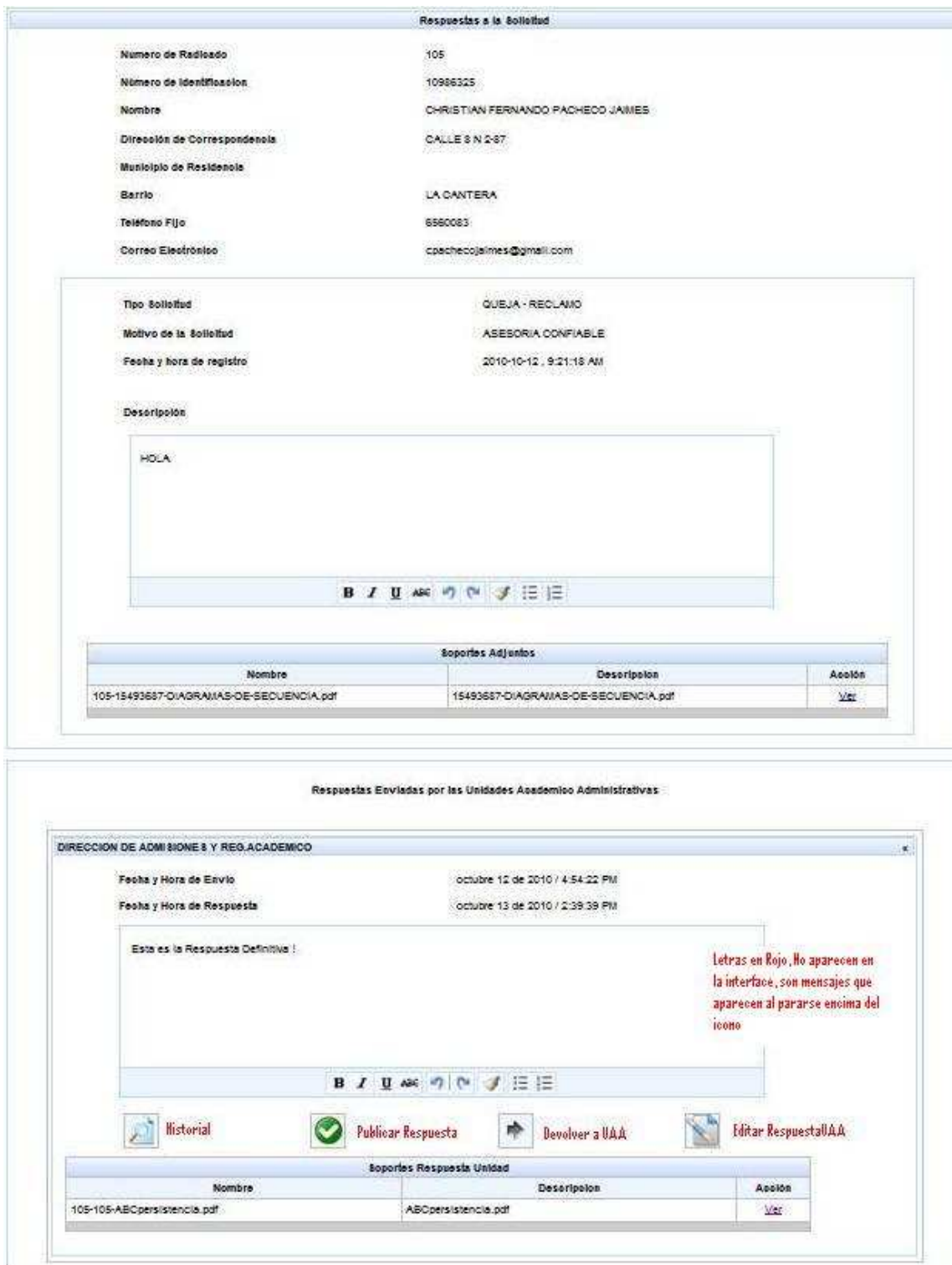


Figura 43. Vista Respuesta enviada por la UAA. Prototipo Final.

- El usuario de Control Interno con permisos totales, podrá consultar y filtrar todo el historial de solicitudes que han sido registradas, de una manera dinámica y por diversos campos.
- El usuario de Control Interno, con permisos totales y restringidos podrá ver e imprimir el reporte estadístico que genera el sistema.

Solicitudes en Trámite Solicitudes Cerradas

Algunos filtros de búsqueda

Solicitudes en Trámite

Radicado	Solicitante	Tipo Solicitud	Motivo	Fecha Registro	Unidades	Acciones
85	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	COMUNICACION	2010-10-08	<ul style="list-style-type: none"> DIRECCION DE ADMISIONES Y REG. DIVISION DE SERVICIOS DE INFORM. ESCUELA DE BIOLOGIA 	
86	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	ATENCION AL BENEFICIARIO	2010-10-09	<ul style="list-style-type: none"> DIRECCION DE ADMISIONES Y REG. DIVISION DE PLANTA FISICA 	
92	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	DISPONIBILIDAD DE RECURSOS	2010-10-10	<ul style="list-style-type: none"> DIRECCION DE ADMISIONES Y REG. 	
93	CHRISTIAN FERNANDO PACHECO JAIMES	S	COMUNICACION	2010-10-11	<ul style="list-style-type: none"> DIRECCION DE CONTROL INTERNO DIRECCION DE ADMISIONES Y REG. 	
94	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	COMUNICACION	2010-10-11	<ul style="list-style-type: none"> DIRECCION DE CONTROL INTERNO DIRECCION DE ADMISIONES Y REG. 	
95	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	INFORMACION VERAZ	2010-10-11	<ul style="list-style-type: none"> DIRECCION DE CONTROL INTERNO DIRECCION DE ADMISIONES Y REG. 	
96	CHRISTIAN FERNANDO PACHECO JAIMES	S	ATENCION AL BENEFICIARIO	2010-10-11	<ul style="list-style-type: none"> DIRECCION DE CONTROL INTERNO DIRECCION DE ADMISIONES Y REG. 	

Figura 44. Vista consulta de solicitudes. Prototipo Final.

Módulo de la Unidad Académico Administrativa

- Los jefes de las unidades académico administrativas, podrán ver y responder cada una de las solicitudes que lleguen a su unidad. Para cada una, se verá el historial de las respuestas dadas anteriormente (en caso de tenerlas).

A las respuestas dadas por cualquier unidad se le podrán adjuntar archivos como complemento a las mismas.

Unidad Académico Administrativa :
DIVISION DE SERVICIOS DE INFORMACION

Nuevas Solicitudes Solicitudes Devueltas

Solicitudes Devueltas						
Numero radicado ↕	Nombre ↕	Tipo Solicitud ↕	Motivo ↕	Fecha - Hora Registro ↕	Fecha - Hora Envio ↕	Acciones
85	CHRISTIAN FERNANDO PACHECO JAIMES	Q - R	COMUNICACION	2010-10-08 10:38:22 AM	2010-10-11 1:55:12 PM	
99	CHRISTIAN FERNANDO PACHECO JAIMES	S	ASESORIA CONFIABLE	2010-10-11 3:03:04 PM	2010-10-11 3:54:46 PM	
100	CHRISTIAN FERNANDO PACHECO JAIMES	S	INFORMACION VERAZ	2010-10-11 3:57:11 PM	2010-10-11 6:03:28 PM	

Total Registros: 3

Figura 45. Vista Lista de Solicitudes UAA. Prototipo Final.

Fecha - Hora Asignacion : octubre 11 de 2010 - 3:58:47 PM Fecha - Hora Respuesta : octubre 11 de 2010 - 3:59:41 PM



Comentario de Control Interno a la Respuesta Dada

Respuesta debe ser Revisada

Respuesta de la DIVISION DE SERVICIOS DE INFORMACION

B *I* U ABC ↶ ↷ 📎 ☰ ☱

Fecha - Hora Asignacion : octubre 11 de 2010 - 6:03:28 PM

Adjuntar Soporte (opcional)  

[Enviar Respuesta](#)

Soportes a la Respuesta, por parte de la Unidad		
Nombre	Descripcion	Acción
100-ABCpersistencia.pdf	ABCpersistencia.pdf	Ver

Figura 46. Vista Respuesta de la Solicitud por la UAA. Prototipo Final.

- Los jefes de las unidades académico administrativas podrán consultar y filtrar todo el historial de solicitudes que han sido remitidos a su unidad, y ver e imprimir el reporte estadístico que genera el sistema de la unidad a su cargo.

4.5.3 Proyecto enmarcado en el esquema de Seguridad de la UIS.

Para este proyecto se utiliza el esquema de seguridad definido por la División de Servicios de Información para los diferentes sistemas de información que apoyan la gestión de la Universidad Industrial de Santander, el cual está basado en la estructura de roles – usuarios.

Los roles se establecen en cada una de las unidades académico administrativas, UAA, responsables de cada sistema, de acuerdo a las actividades que realizan. A cada uno de los roles definidos se le asocian los usuarios de acuerdo a las funciones que desempeñen.

4.5.3.1 Estructura de la Base de Datos soporte

La base de datos que soporta el esquema de seguridad contempla básicamente las siguientes tablas:

Sistema: Contiene información de los sistemas de información de la universidad. Para cada sistema se especifica: Nombre, descripción del sistema, fecha y hora de creación en la base de datos, fecha y hora de inicio de vigencia del sistema, fecha y hora de cierre de vigencia del sistema.

Rol: contiene información de los diferentes roles definidos para cada sistema de información, como: Nombre asignado al rol, descripción del rol, fecha y hora de creación, fecha y hora de inicio de vigencia del rol, fecha y hora de cierre de vigencia del rol.

Usuario: Contiene información de los posibles usuarios de los sistemas de información. Entre esta información está: tipo y número de documento de identidad del usuario, fecha y hora de creación del usuario, fecha y hora de inicio de vigencia del usuario, fecha y hora de cierre de vigencia del usuario.

Sistema–rol: Contiene los roles definidos para cada uno de los sistemas de información, indicando: rol, sistema, fecha y hora de creación del rol – sistema, fecha y hora de inicio de vigencia del rol en el sistema, fecha y hora de cierre de vigencia del rol en el sistema.

Rol–usuario: Contempla los usuarios asociados a cada uno de los roles definidos, considerando: Rol, usuario, fecha y hora de creación del rol – usuario, fecha y hora de inicio de vigencia del usuario en el rol, fecha y hora de cierre de vigencia del usuario en el rol.

Menú–rol–sistema: Contiene los menús asociados a los roles en los distintos sistema de información, contemplando: Sistema de información, nombre del menú, descripción del menú, fecha y hora de creación del menú, fecha y hora de inicio de vigencia del menú asociado al rol, fecha y hora de cierre de vigencia del menú asociado al rol.

Opción–menú–rol: Contempla las opciones definidas para cada una de los posibles menús establecidos para cada sistema de información. Contiene: Nombre de la opción, descripción de la opción, nombre del menú superior, nombre del menú que contiene la opción, nombre del programa a ejecutar cuando la opción es la de más bajo nivel, fecha y hora de creación de la opción del menú, fecha y hora de inicio de vigencia de la opción, fecha y hora de cierre de la opción.

Tabla–sistema: Contiene información de las tablas que conforman la base de datos que soporta cada uno de los sistemas de información. Considera: Sistema de información, nombre de la tabla, descripción de la tabla.

Tipo–permiso: Establece para cada tabla de un sistema de información, los roles que tienen permisos para incluir registros, para modificar registros o para eliminar registros en ella. Contiene: Sistema de información, nombre de la tabla, clase de permiso (inclusión, modificación, eliminación de registros), fecha y hora de

creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.

Acceso–tabla: Define para las tablas de un sistema de información si un rol tiene permiso sobre toda la información de la tabla o sobre una parte de esta. Considera: Sistema, nombre de la tabla, clase de acceso (total, parcial), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.

Atributo–tabla: Establece los atributos sobre los cuales se debe controlar el acceso a una tabla, cuando a un rol se le concede permiso para hacer uso parcial de la información existente en una tabla. Contiene: Sistema de información, nombre de la tabla, nombre del atributo sobre el cual se controla el acceso a la tabla, descripción del atributo, fecha y hora de creación del atributo, fecha y hora de inicio de vigencia del atributo, fecha y hora de fin de vigencia del atributo.

Valor–atributo-proceso: Contiene los valores que deben tener los atributos definidos en cada tabla en la tabla atributo – tabla que permiten el acceso a la información asociada a estos valores. Específica: Sistema de información, nombre de la tabla, nombre del atributo, valor del atributo, descripción, fecha y hora de creación del valor del atributo, fecha y hora de inicio de vigencia del valor del atributo, fecha y hora de fin de vigencia del valor del atributo.

Acceso-sistema: Contempla el histórico de acceso que un usuario ha realizado a un sistema, identificando las opciones que ha seleccionado. Contiene: Login de usuario, rol, identificación de la sesión, sistema, opción seleccionada, fecha y hora de ingreso, fecha y hora de salida.

4.5.3.2 Entorno de Navegación

Para cada sistema de información, la UAA responsable define los roles necesarios para el adecuado uso del sistema de información de acuerdo a las funciones que realice y establece los usuarios asociados a cada uno de ellos.

Para cada rol se define el menú de inicio, el cual le permite a cada usuario que hace parte de este rol, empezar la navegación por las distintas opciones que le ofrece el sistema, hasta llegar al nivel más bajo en el cual se ejecuta el proceso que soporta la actividad que desea realizar.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, menú-rol, opción-menú rol, descritas en “ESTRUCTURA DE LA BASE DE DATOS SOPORTE”.

4.5.3.3 Entorno de Control de Datos

Para los roles definidos en cada uno de los sistemas de información se especifican las tablas a las cuales puede acceder, el tipo de transacción que puede realizar sobre estas tablas (inclusión, modificación o eliminación de registros), si tiene acceso total o parcial a la información que contiene la tabla.

Para el acceso a la información de la tabla de manera parcial, se debe establecer el atributo o atributos seleccionados, los valores que estos atributos deben tener para autorizar el acceso solicitado.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, tabla-sistema, tipo-permiso, acceso-tabla, atributo-tabla, valor atributo proceso, descritas en “ESTRUCTURA DE LA BASE DE DATOS SOPORTE”.

4.5.3.4 Auditoría

Todas las tablas que conforman la base de datos soporte del esquema de seguridad tienen el historial de las transacciones realizadas sobre cada una de ellas.

El historial de las transacciones de cada tabla contiene información de los registros incluidos en la tabla, de los registros modificados y de los registros eliminados. Adicionalmente, en cada transacción se especifica: Fecha de la transacción, hora de la transacción, tipo de transacción (I/U/D), tipo y número de documento de identidad del usuario que realizó la transacción, login, rol asociado, dirección IP y MAC del equipo desde el cual llevó a cabo la transacción.

4.5.4 Organización de Directorios.

Consideraciones iniciales

Para cada uno de los sistemas principales se debe crear un proyecto que siga los estándares para la estructura de directorios y nombres. En estos proyectos van las entidades propias de cada sistema, así como los componentes comunes a varias aplicaciones.

Las entidades van, en cada sistema, empaquetadas en un .jar, de tal manera que el desarrollo de una nueva aplicación no implica la implementación de éstas.

En el paquete general van los elementos que son utilizados por todos los módulos.

Las entidades van a estar todas en `co.edu.uis.[Sistema].entidades`.

Los servicios van a estar todos en `co.edu.uis.[Sistema].servicios`.

Para la creación de las entidades se toma la estructura de la base de datos. Por lo tanto se hace indispensable comenzar por este paso, seguido de la creación de los componentes para posteriormente desarrollar las demás aplicaciones.

Carpetas

Deben comenzar con minúscula y están basados en la estructura de directorios que se genera mediante “Seam-Gen” cuando se crea un nuevo proyecto. Las carpetas que se crean son las siguientes:

- bootstrap
- classes
- dist
- exploded-archives
- lib
- nbproject
- resources
- src
 - hot
 - main
 - test
- test-build
- view

A continuación se hace una breve descripción de las carpetas con las que el desarrollador tiene contacto directo.

Carpeta src

En esta carpeta se guardan los archivos fuentes de las clases del sistema distribuidos en dos subcarpetas de la siguiente manera:

Carpeta main

Contiene las clases de apoyo al proceso. No incluye a las entidades, pues estas estarán disponibles en otro lugar que se describe más adelante en este documento.

Carpeta hot

Contiene los casos de uso reflejados en los EJBs y sus interfaces.

Carpeta resources

Contiene los archivos de configuración de la aplicación.

Carpeta view

Contiene las páginas, imágenes, estilos y todo lo referente a aspecto de la aplicación.

Carpeta dist

Contiene los archivos jar, war y ear de la aplicación. Estos se generan automáticamente.

Carpeta lib

Contiene las librerías necesarias para la ejecución de la aplicación.

Nombres de archivos

Se sigue el mismo estándar dado para el nombre de variables descrito en el apartado 4.2.3 de éste documento.

Organización del código fuente dentro de la estructura de directorios.

Paquetes

La composición del nombre de los paquetes se sigue teniendo en cuenta la siguiente sintaxis:

co.edu.uis.[sistema].[aplicación].[módulo].[caso de uso]

Ejemplo:

co.uis.edu.co. sipqrs.administracion.mantenimiento.TipoSolicitud

En el caso de los proyectos principales (los que empaquetan entidades y componentes para cada sistema) los paquetes serán los siguientes:

Administración (src/action): co.edu.uis.[Sistema].[Aplicación].administracion

Clases generales (src/action): co.edu.uis.[Sistema].[Aplicación].general

Paginas (carpeta view)

Dentro de esta carpeta se tendrán la siguiente estructura:

- Plantillas : Plantillas del sitio
- imágenes
- estilos
- scripts (Si son necesarios. No es obligatorio y se deben evitar en la medida de lo posible).
- Administración : Todo lo referente a la administración
- generales : Páginas de uso global en la aplicación
- ayudas
- módulos

4.5.5 Documentación de programas fuente¹⁶.

Nombre de archivos, variables, constantes, atributos, métodos y parámetros.

Los nombres dentro del código fuente siguen los parámetros establecidos en el estándar general de nombres, con las siguientes particularidades.

- Los nombre de los parámetros de los métodos empiezan con guión de piso, el resto del nombre sigue el estándar para nombres de variables. Una vez dentro del código fuente, se deben asignar a una nueva variable con el mismo nombre pero sin el guión de piso.
- Los nombres de constantes o variables finally, se escriben en mayúscula sostenida separando las palabras por guiones de piso.

Comentarios Java:

Los programas en Java pueden tener dos tipos de comentarios: comentarios de implementación y comentarios de documentación. Los comentarios de implementación están delimitados por `/*.. */` cuando se trata de varias líneas y `//` cuando se trata de una línea. Los comentarios de documentación (conocidos como "comentarios JavaDoc") son específicos de Java y están delimitados por :

`/** ... */`. Los comentarios de documentación se pueden extraer a ficheros HTML usando la herramienta javadoc.

Los comentarios de implementación están destinados a comentar el código o para comentarios sobre la implementación en particular, buscan dar una orientación y hacer aclaraciones sobre la implementación a quien observa el código fuente. Los comentarios de documentación están destinados a describir la especificación del

¹⁶ Fuente: Estándares de la División de Servicios de Información de la Universidad Industrial de Santander

código, desde una perspectiva independiente de la implementación, están hechos para ser leídos por desarrolladores que pueden no tener necesariamente el código fuente a mano.

Los comentarios se deben usar para dar una visión general del código y para proporcionar información adicional que no esté disponible fácilmente en el propio código.

Orden dentro de los archivos de código fuente:

Cada archivo de código fuente Java debe contener una única clase o interfaz pública y deben seguir el siguiente orden:

- Sentencias package.
- Sentencias import.
- Comentario de documentación de la clase/interfaz.
- Declaraciones de clase e interfaz.
- Comentario de la implementación de la clase/interfaz si fuera necesario.
- Declaración de constantes (solo se declaran dentro de interfaces).
- Declaración de atributos. (la declaración se debe hacer uno por línea y al frente debe ir un comentario de implementación de ser necesario.)
- Declaración de variables. (Según los estándares no se deben declarar variables públicas, para ello se definen los métodos gets() y sets()). Las variables aquí declaradas serán privadas y se utilizarán como indicadores de estado de la clase. Las variables se deben declarar en el siguiente orden:
 - Variables estáticas: organizadas por ámbito o accesibilidad de la siguiente manera: públicas, protegidas, de paquete (sin modificador) privadas.
 - Variables de instancia: organizadas de la misma manera que las estáticas.
 - Al igual que los atributos se debe declarar una por línea para facilitar los comentarios de implementación frente a ellas en caso de necesitarse.
 - Comentario de documentación sobre cada uno de los constructores (no aplica a entidades).

- Declaración de constructores. (Siempre se declarará el constructor sin parámetros, se requiera o no una acción dentro de este. No aplica a entidades).
- Comentario de documentación sobre cada uno de los métodos que lo requieran.
- Declaración de métodos de la lógica del negocio: estos deben ir agrupados por funcionalidad en lugar de por ámbito, siendo el objetivo de esta organización el hacer el código de mas fácil lectura y comprensión.
- Declaración de métodos gets() , sets() e is().
- Sobre escritura del método equals().
- Sobre escritura del método hashCode().
- Sobre escritura del método compare().
- Sobre escritura del método compareTo().

Especificaciones sobre el formato del código fuente:

Tabulación:

La unidad de tabulación será de 2 espacios.

Longitud de línea

Se deben evitar líneas de 80 o más caracteres ya que algunas herramientas no las manejan bien, además que líneas demasiado extensas hacen difícil la lectura del código.

Ruptura de líneas

- Cuando una expresión no cabe en una única línea, se debe romper de acuerdo a estos principios generales:
 - Romper después de una coma.
 - Romper antes de un operador.

- Preferir las rupturas de alto nivel a las de bajo nivel. (por ejemplo no romper por operador dentro de paréntesis).
- Alinear la nueva línea con el principio de la expresión al mismo nivel que la línea anterior y dar cuatro tabulaciones.

Declaraciones y sentencias

Sentencias package e imports:

Deben seguir las siguientes reglas de formato.

- Estas sentencias no van tabuladas.
- Se debe declarar una por línea.
- No utilizar comodines.

Variables locales:

Todas las variables locales deben inicializarse en el sitio en donde se declaran.

Las variables deben declararse al inicio de cada método y no esperar a declararlas hasta su primer uso. La única excepción son las variables de bucles que en Java se pueden declarar dentro de la sentencia for. A propósito de estas variables, se utilizarán las letras de la i a la z en la medida que se vayan necesitando.

Clases, interfaces y métodos:

En las declaraciones se deben seguir las siguientes reglas de formato:

- Ningún espacio entre el nombre del método y el paréntesis "(" que abre su lista de parámetros.
- La llave de apertura "{" aparece al final de la misma línea que la sentencia de declaración.
- La llave de cierre "}" comienza una línea nueva tabulada para coincidir con su sentencia de apertura correspondiente, excepto cuando es un bloque vacío

que la llave de cierre "}" debe aparecer inmediatamente después de la de apertura "{".

- Los métodos están separados por una línea en blanco.
- Las clases e interfaces principales no van tabuladas
- Los bloques de código que pertenecen a una clase o método van tabulados.

Sentencias simples:

Se debe escribir solo una sentencia por línea, y no escribir más de una separadas por punto y coma en la misma línea sin importar lo cortas que puedan ser.

Sentencias compuestas:

Las sentencias compuestas son sentencias que contienen una lista de sentencias encerradas entre llaves "{" y "}" y deben seguir las siguientes reglas de formato.

- Las sentencias internas deben estar tabuladas un nivel más que la sentencia compuesta.
- La llave de apertura debe estar al final de la línea que comienza la sentencia compuesta; la llave de cierre debe estar en una nueva línea y estar tabulada al nivel del principio de la sentencia compuesta.
- Las llaves se usan en todas las sentencias compuestas, incluidas las sentencias únicas, cuando forman parte de una estructura de control, como una sentencia if-else o un bucle for. Esto hace más fácil introducir nuevas sentencias sin provocar errores accidentales al olvidarse añadir las llaves.

Líneas en blanco

Las líneas en blanco mejoran la legibilidad resaltando secciones de código que están relacionadas lógicamente.

- En las siguientes circunstancias, siempre se deben usar dos líneas en blanco:
- Entre secciones de un archivo de código fuente.

- Entre definiciones de clases e interfaces.
- En las siguientes circunstancias, siempre se debería usar una línea en blanco:
- Entre métodos.
- Entre las variables locales de un método y su primera sentencia.
- Antes de un comentario de bloque o de una sola línea.
- Entre las secciones lógicas de un método, para mejorar la legibilidad.

Espacios en blanco:

Los espacios en blanco deberían usarse en las siguientes circunstancias:

- Una palabra reservada seguida por un paréntesis Por ejemplo:

```
while (true) {
    sentencias;
}
```

- En las listas de argumentos, debe haber un espacio después de cada coma.
- Todos los operadores binarios, excepto el operador punto (.) deben estar separados de sus operandos por espacios. Los operadores unarios (incremento ++, decremento --, negativo -) nunca deben estar separados de sus operandos. Por ejemplo:
- Las expresiones de una sentencia for deben estar separadas por espacios en blanco. Por ejemplo:

```
for (expr1; expr2; expr3);
```

- Las conversiones de tipo (cast) deberían estar seguidas de un espacio en blanco. Por ejemplo:

```
variableEntera = (int) variableCadena;
```

Ejemplo de documentación básica de un EJB de Sesión.

```
package co.edu.uis.sipqrs.administracion.mantenimiento.TipoSolicitud;

import java.util.ArrayList;
import java.util.List;
import javax.ejb.Stateful;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.Scope;
import org.jboss.seam.annotations.datamodel.DataModel;
import org.jboss.seam.international.StatusMessages;

import co.edu.uis.excepciones.DSIException;
import co.edu.uis.sipqrs.entidades.TipoSolicitud;
import javax.ejb.Remove;
import javax.persistence.EntityManager;
import javax.persistence.Query;

import org.hibernate.Criteria;
import org.hibernate.criterion.Restrictions;
import org.jboss.seam.ScopeType;
import org.jboss.seam.annotations.Destroy;

import org.jboss.seam.annotations.Out;

/**
 * @autor Christian Pacheco Este EJB Administra la Gestion( Crear ,
Modificar
 * eliminar ) de los Tipos de Solicitud
 */
@Stateful
@Scope(ScopeType.CONVERSATION)
@Name("AdministrarTipoSolicitud")
public class AdministrarTipoSolicitudEJB implements
AdministrarTipoSolicitud {

    // Consulta de la base de datos el tipos de solicitudes existentes
    @DataModel
    private List<TipoSolicitud> tipoSolicitudes;

    @In
    private EntityManager em;

    @In
    private StatusMessages statusMessages;

    @In(required = false)
    @Out(required = false)
    private TipoSolicitud tipoSolicitud;

    // 1: editando | 2: viendo | 3: eliminando
    private int estado;
```

```

// define el exito o fallo de la consulta

private Boolean exito;

/**
 * Este metodo se encarga de dar espacio de memoria en el momento
 * de la carga de la pagina que me va a permitir realizar la
 * Administración de los tipos de solicitud
 */
public void asignarTipoSolicitud() throws DSIEException {
    this.tipoSolicitud = new TipoSolicitud();
    try {
        consultarTipoSolicitudes();
    } catch (Exception e) {
        throw new DSIEException(e);
    }
}

/**
 * Se consultan los tipos de solicitudes que ya estan registradas
 * en la Base de datos
 */
public void consultarTipoSolicitudes() throws DSIEException {
    tipoSolicitudes = new ArrayList<TipoSolicitud>();
    try {
        String jpql = "FROM TipoSolicitud";
        Query query = em.createQuery(jpql);
        this.tipoSolicitudes = query.getResultList();
    } catch (Exception e) {
        throw new DSIEException(e);
    }
}

/**
 * Con este metodo, registramos en la Base de Datos un Nuevo tipo de
 * solicitud.
 */
public void crearTipoSolicitud() throws DSIEException {
    TipoSolicitud aTipoSolicitud = null;
    // verificar si existe en BD
    org.hibernate.Session sesion = (org.hibernate.Session)
this.em.getDelegate();
    Criteria criterios =
sesion.createCriteria(TipoSolicitud.class);
    criterios.add(Restrictions.eq("nombreTipoSolicitud",
tipoSolicitud.getNombreTipoSolicitud()));
    aTipoSolicitud = (TipoSolicitud) criterios.uniqueResult();

    // fin verificar
    try {
        if (aTipoSolicitud == null) {

```

```

        em.persist(this.tipoSolicitud);
        em.flush();
        statusMessages.addToControl("mensajesValidacion",
            "Registro Exitoso");
    } else {
        statusMessages.addToControl("mensajesValidacion",
            "El nombre del tipo de Solicitud ya
            existe en la BD");
    }
} catch (Exception e) {
    throw new DSException(e.getMessage());
}
}
/**
 * Edita un tipo de solicitud seleccionado, antes verifica que no
 * se este* creando uno ya existente
 */
public void editarTipoSolicitud() throws DSException {
    List<TipoSolicitud> consulta;

    String jpql = "SELECT p FROM TipoSolicitud p "
        + " WHERE p.idTipoSolicitud<>:idTipoSolicitud"
        + " AND
p.nombreTipoSolicitud=:nombreTipoSolicitud";
    Query query =
em.createQuery(jpql).setParameter("idTipoSolicitud",

this.tipoSolicitud.getIdTipoSolicitud()).setParameter(
    "nombreTipoSolicitud",
    this.tipoSolicitud.getNombreTipoSolicitud());
    consulta = query.getResultList();

    this.exito = true;
    try {
        if (consulta.isEmpty()) {
            this.em.merge(tipoSolicitud);
            this.em.flush();
            statusMessages.addToControl("mensajesValidacion",
                "Edición Exitosa");
        } else {
            this.exito = false;
            statusMessages.addToControl("mensajesValidacion",
                "El nombre del Tipo Solicitud ya existe en la Base de
                Datos");
            tipoSolicitudes = new ArrayList<TipoSolicitud>();
        }
    } catch (Exception e) {
        this.exito = false;
        throw new DSException(e);
    }
}
/**
 * El metodo selecciona un tipo de solicitud, lo guarda en memoria
 * para poderlo visualizar Escoge el Tipo de solicitud, y con el
 * estado define si lo que está haciendo es Editando o eliminando

```

```

*
* @param aTipoSolicitud
* @param estado
*/
public void modificarTipoSolicitud(TipoSolicitud aTipoSolicitud,
                                   int estado)
    throws DSIEException {
    this.tipoSolicitud = aTipoSolicitud;
    setEstado(estado);
    if (estado == 3) {
        statusMessages.addToControl("mensajesValidacion",
        "Esta seguro que desea eliminar el Tipo de Solicitud?");
    }
}

/**
 * Elimina el tipo de Solicitud que se selecciono
 */
public void eliminarTipoSolicitud() throws DSIEException {
    try {
        this.em.remove(this.em.merge(tipoSolicitud));
        this.em.flush();
        statusMessages.addToControl("mensajesValidacion",
        "Registro Eliminado con Éxito");
    } catch (Exception e) {
        throw new DSIEException(e);
    }
}

/**
 * Cancela la accion y libera el espacio de memoria
 */
public void cancelarTipoSolicitud() throws DSIEException {
    this.tipoSolicitud = new TipoSolicitud();
}

/**
 * @return Tipos de Sololicudes
 */
public List<TipoSolicitud> getTipoSolicitudes() {
    return tipoSolicitudes;
}

/**
 * @param tipoSolicitudes
 *         los tipos de solicitudes a asignar
 */
public void setTipoSolicitudes(List<TipoSolicitud> tipoSolicitudes)
{
    this.tipoSolicitudes = tipoSolicitudes;
}

/**
 * @return el em

```

```

    */
    public EntityManager getEm() {
        return em;
    }

    /**
     * @param em
     *         the em to set
     */
    public void setEm(EntityManager em) {
        this.em = em;
    }

    /**
     * @return la tipoSolicitud
     */
    public TipoSolicitud getTipoSolicitud() {
        return tipoSolicitud;
    }

    /**
     * @param tipoSolicitud
     *         the tipoSolicitud to set
     */
    public void setTipoSolicitud(TipoSolicitud tipoSolicitud) {
        this.tipoSolicitud = tipoSolicitud;
    }

    public void setEstado(int estado) {
        this.estado = estado;
    }

    public int getEstado() {
        return estado;
    }

    /**
     * @param exito
     *         the exito to set
     */
    public void setExito(Boolean exito) {
        this.exito = exito;
    }

    /**
     * @return the exito
     */
    public Boolean getExito() {
        return exito;
    }

    @Remove
    @Destroy
    public void destroy() { }
}

```

5. CONCLUSIONES

- La nueva versión del sistema de información de Preguntas, Quejas, Reclamos y Sugerencias de la Universidad Industrial de Santander se convierte en la solución al problema de funcionalidad y practicidad para la recepción y trámite de las solicitudes, que hacen los ciudadanos externos, a la universidad.
- El reporte de información útil y oportuna es fundamental dentro de los objetivos organizacionales de la Universidad Industrial de Santander.
- La nueva versión del sistema de información de Preguntas, Quejas, Reclamos y Sugerencias llena las expectativas de la Dirección de Control Interno y Evaluación de Gestión en cuanto a oportunidad, calidad de la información y facilidad de uso.
- Los sistemas de información de cualquier tipo, deben volverse fáciles de usar, flexibles y amigables para el usuario de tal forma que la dependencia con el desarrollador sea reducida al máximo.
- La utilización de herramientas libres como Java, aparte de reducir costos, permite la realización de aplicaciones seguras, robustas y de tipo empresarial, debido al permanente aporte que hacen usuarios de todo el mundo y la facilidad de capacitación online en foros de discusión.
- El proceso de construcción de prototipos no funcionales, permite un ahorro de tiempo debido a que el usuario interactúa con las entradas y salidas que

va a tener el sistema y su esquema de navegación y hace las correcciones y sugerencias, sin desgastes de tiempo de programación.

- La implementación de un esquema rol – usuario permite crear un esquema de seguridad que me garantice que la accesibilidad del sistema solo será permitida a usuarios autorizados.

6. RECOMENDACIONES

- Construir un sistema de información que permita la formulación, seguimiento y toma de decisiones de las acciones de mejora que se generen de las quejas, reclamos y sugerencias dadas por las comunidad.
- Dar a conocer, más ampliamente, a la comunidad académica y a la comunidad en general este espacio de participación ciudadana, para que los objetivos que se quieren alcanzar con el sistema tengan mayor impacto en la vida universitaria.
- Dada la gran experiencia adquirida, desarrollando el proyecto de grado con la División de Servicios de Información, se recomienda crear un semillero entre esta unidad y la Escuela de Ingeniería de Sistemas e Informática, para posteriores experiencias académicas que permitan mejorar la calidad académica y profesional de los futuros ingenieros de sistemas egresados de la escuela.

7. BIBLIOGRAFÍA

PRESSMAN, Roger. Ingeniería del Software. Un enfoque práctico. Quinta Edición. McGraw Hill. España. 2005.

COBOS, Carlos Alberto; MENDOZA, Martha Eliana. Manual De Informix – SQL . Universidad Industrial de Santander, 1998.

MANUAL DE CALIDAD. Código. MDI 01. Universidad Industrial de Santander.

LÓPEZ GUSTAVO, Blanco Ignacio. Tesis en Ingeniería Informática

GROFF, James R. – WEINBERG, Paul N. APLIQUE SQL. OsBORNE-McGrawHill, 1991.

FERRÉ GRAU, Xavier - SÁNCHEZ S. María Isabel. Desarrollo Orientado a Objetos con UML. Facultad de Informática - UPM

SITIOS WEB

- www.wikipedia.org
- <http://www.sparxsystems.com.ar/>
- http://download.oracle.com/docs/cd/E17802_01/j2ee/javaee/jaserverfaces/2.0/docs/pdl/docs/facelets/index.html
- <http://livedemo.exadel.com/richfaces-demo/richfaces/comboBox.jsf?tab=usage&cid=170149>

ANEXO A

FORMATO DE RECEPCION DE QUEJAS Y RECLAMOS


	PROCESO SEGUIMIENTO INSTITUCIONAL		Código : FSE.06		
	RECEPCIÓN DE QUEJAS Y RECLAMOS		Version 03		
Solicitud No.		FECHA	DD	MM	AA
Nombre*:					
Identificación*: CC NIT TI		No.	Tipo de Solicitante: Funcionario: _____ Estudiante : _____ Ciudadano: _____ Proveedor: _____		

TIPO DE QUEJA O RECLAMO	
Percepción del Servicio: _____	Infraestructura: _____
Oportunidad en el servicio: _____	Información veraz: _____
Atención al beneficiario _____	Disponibilidad de recursos: _____
Comunicación: _____	Asesoría Confiable: _____
Manejo de recursos: _____	Otro: _____
Dependencia de la Universidad que motiva la queja o reclamo*:	
DESCRIPCIÓN DE LA QUEJA Y/O RECLAMO	

ANEXOS (Relación de soportes que complementen la queja o reclamo, en caso que existan)	
Nombre y cargo del funcionario que recibe la queja y/o reclamo	

ANEXO B

FORMATO DE RECEPCION DE SUGERENCIAS

	PROCESO SEGUIMIENTO INSTITUCIONAL		Código : FSE.13		
	RECEPCIÓN DE SUGERENCIAS		Versión: 03		
Solicitud No.		FECHA	DD	MM	AA
Nombre*:					
Identificación*: CC NIT TI		No.	Tipo de Solicitante: Funcionario: _____ Estudiante : _____ Ciudadano: _____ Proveedor: _____		

TIPO DE SUGERENCIA	
Percepción del Servicio: _____	Infraestructura: _____
Oportunidad en el servicio: _____	Información veraz: _____
Atención al beneficiario _____	Disponibilidad de recursos: _____
Comunicación: _____	Asesoría Confiable: _____
Manejo de recursos: _____	Otro: _____
Dependencia de la Universidad que motiva la sugerencia*:	
DESCRIPCIÓN DE LA SUGERENCIA	

Nombre y cargo del funcionario que recibe		
la sugerencia		