

**DISEÑO, IMPLEMENTACIÓN Y PUESTA EN FUNCIONAMIENTO DEL
MÓDULO DE PLANIFICACIÓN: GESTIÓN DE RECURSOS HUMANOS,
COMPRAS Y ADQUISICIONES DEL SISTEMA DE GESTIÓN DE PROYECTOS
UIS (SGPUIIS)**

**EDWING ANTONIO GARCIA MALDONADO
YENNY LISSETH TOBO NIÑO**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2011**

**DISEÑO, IMPLEMENTACIÓN Y PUESTA EN FUNCIONAMIENTO DEL
MÓDULO DE PLANIFICACIÓN: GESTIÓN DE RECURSOS HUMANOS,
COMPRAS Y ADQUISICIONES DEL SISTEMA DE GESTIÓN DE PROYECTOS
UIS (SGPUIS)**

**EDWING ANTONIO GARCIA MALDONADO
YENNY LISSETH TOBO NIÑO**

**Trabajo de Investigación para optar por el Título de
Ingeniero de Sistemas**

Director

**ENRIQUE TORRES LÓPEZ
Ingeniero de Sistemas**

Codirector

**GIOBANI SERRANO DURÁN
Ingeniero Industrial**

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

2011

A mis amados padres Antonio y Nubia, por siempre haber estado a mi lado, por su apoyo incondicional y sobre todo por la paciencia mostrada durante todo este tiempo.

A mis queridos hermanos Diana, Omaira y Jose Fernando, también a mi sobrinita Camila por apoyarme y entenderme en el transcurso de esta etapa de mi vida.

A mi compañera de proyecto, Yenny Lisseth Tobo Niño, por haber compartido esta experiencia conmigo y haberme brindado un gran apoyo, para así poder alcanzar esta importante meta.

A mis amigos, por haber compartido conmigo y brindarme su apoyo.

Edwing

*A Dios y la Virgen por todas las oportunidades que
me han dado en la vida.*

*A mis papis Rosa y Hernán por su esfuerzo y
sacrificio y por ser mi fortaleza en todo momento.*

*A mis Hermanos Yezid, Giselle y Camilo por
brindarme su cariño, los quiero mucho.*

*A mis abuelitos María Eva y Antonio por sus
oraciones al santísimo y por sus sabios consejos;
Carlos Antonio y Placidia por ser mi luz guía y
protectora.*

*A mis amigos Omar y Edwing por su gran amistad
y por los buenos momentos en su compañía
agradable y sincera.*

*A Carlos Andrés, por su cariño, apoyo y
comprensión, muchas Gracias por todos los
momentos tan maravillosos que juntos compartimos.*

*Y a todas aquellas personas que de alguna manera
colaboraron en la realización de este sueño.*

Yenny

AGRADECIMIENTOS

A Dios, por brindarnos tanta sabiduría para culminar esta meta tan anhelada y por darnos la Fortaleza de nunca desfallecer antes los problemas.

A la Universidad Industrial de Santander por la formación integral a nivel Profesional y como seres humanos brindado a lo largo de nuestra carrera.

A los Ingenieros Enrique Torres López y Giobani Serrano Duran directores de este proyecto, por toda su colaboración y apoyo incondicional a lo largo del desarrollo de este mismo.

A los Ingenieros Humberto Ruiz y Emilio Cárcamo, por el continuo apoyo y las enseñanzas aportadas durante la realización del proyecto.

A nuestros compañeros Marco Antonio Ruiz, Mario Jerez, Laura Rojas y Pedro Otero, por su colaboración y acompañamiento en el desarrollo de este proyecto.

A la División Servicios de Información, por todo el apoyo y su colaboración intelectual, para el desarrollo de este proyecto.

A todos nuestros amigos que con su ánimo, apoyo y contribuciones personales y profesionales hicieron que este proyecto llegara a su meta.

Y a todas aquellas personas que de una u otra forma colaboraron o participaron en la culminación de esta meta hoy alcanzada.

CONTENIDO

	pág.
1. CONTEXTO GENERAL	27
1.1 DEFINICIÓN DEL PROBLEMA	27
1.1.1 ENTIDAD INTERESADA EN EL PROYECTO.	27
1.2 OBJETIVOS	30
1.2.1 OBJETIVO GENERAL.	30
1.2.2 OBJETIVOS ESPECÍFICOS.	30
1.3 JUSTIFICACIÓN	31
1.4 VIABILIDAD	33
2. MARCO TEÓRICO	35
2.1 ESTADO DEL ARTE	35
2.2 GENERALIDADES	36
2.2.1 PROYECTO	36
2.2.1.1 Características del proyecto	37
2.2.1.2 Proyectos y planificación estratégica	38
2.2.2 DIRECCIÓN DE PROYECTOS- ÁREAS DE CONOCIMIENTO DE LA DIRECCIÓN DE PROYECTOS	39
2.2.2.1 La estructura de la gerencia de proyectos	40
2.2.2.2 Áreas de conocimiento en la dirección de proyectos	40
2.3 GESTIÓN DE LA INVERSIÓN EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER	42
2.4 BANCO DE PROGRAMAS Y PROYECTOS DE INVERSIÓN NACIONAL– BPIN	43
2.4.1 COMPONENTE DE METODOLOGÍA	43
2.4.2 COMPONENTE DE CAPACITACIÓN	44
2.4.3 COMPONENTE DE SISTEMAS DE INFORMACIÓN	44
2.5 BANCO DE PROGRAMAS Y PROYECTOS DE INVERSIÓN DE LA UIS (BPPIUIS)	44
2.6 ETAPAS DE UN PROYECTO	46
2.7 ESQUEMA DE DESARROLLO	49
2.7.1 DIAGRAMACIÓN UML	49
2.7.1.1 Diagramas de caso de uso	51
2.7.1.2 Relaciones de Casos de Uso	52
2.7.1.3 Diagrama de Actividades	54
2.7.1.4 Diagrama de secuencia	55
2.7.2 JAVA ENTERPRISE EDITION 5	56
2.7.2.1 Generalidades acerca de JAVA EE 5	56
2.7.2.2 Características principales de JAVA EE 5.	60
2.7.2.3 API de Java EE 5	60
2.7.3 JAVA SERVER FACES.	66
2.7.4 HIBERNATE Y ORM	67

2.7.5	EJB 3.0	69
2.7.6	JPA	69
2.7.7	JPQL	70
2.7.8	SERVIDOR DE APLICACIONES – JBOSS	70
2.7.9	ENTERPRISE ARCHITECT	71
3.	<u>METODOLOGÍA DE DESARROLLO</u>	72
<hr/>		
3.1	CICLO DE VIDA DEL PROYECTO	72
3.1.1	ANÁLISIS DE REQUERIMIENTOS	73
3.1.2	DISEÑO	73
3.1.3	IMPLEMENTACIÓN DE LA APLICACIÓN	75
3.1.4	PRUEBAS DE SOFTWARE	76
3.1.5	AJUSTES	76
3.2	METODOLOGÍA DE DESARROLLO	77
3.2.1	MODELO DE CONSTRUCCIÓN POR PROTOTIPOS.	77
3.2.1.1	Estructura	78
3.3	APLICACIÓN DE LA METODOLOGÍA	79
3.3.1	ESTÁNDARES DSI.	79
3.3.1.1	Reglas de sintaxis generales	79
3.3.2	DIAGRAMAS UML	84
3.3.2.1	Diagramas de Casos de Uso	84
	CASO DE USO: ADMINISTRAR COMPETENCIAS	87
3.3.2.2	Diagramas de Clases	93
3.3.2.3	Diagrama de Secuencias	98
3.4	PRIMER PROTOPIPO	102
	FIGURA 19. FORMATO DETALLE, EDICIÓN Y ELIMINACIÓN ROL PROYECTO.	104
3.5	SEGUNDO PROTOTIPO	104
	FIGURA 20. FORMATO DE EDICIÓN DE ROL PROYECTO Y ASOCIACIÓN DE COMPETENCIAS	105
3.6	PROTOTIPO FINAL	106
	FIGURA 21. FORMATO FINAL DE CREACIÓN DE ROL PROYECTO	107
	FIGURA 22. FORMATO FINAL DE EDICIÓN DE ROL PROYECTO Y ASOCIACIÓN DE COMPETENCIAS	108
	FIGURA 23. FORMATO DE EDICIÓN Y ELIMINACIÓN DE COMPETENCIAS	109
3.6.1	EL PROYECTO ENMARCADO EN EL ESQUEMA SEGURIDAD Y EL SGPUIS	109
3.6.1.1	Estructura de la base de datos de soporte	109
3.6.1.2	Entorno de navegación	112
3.6.1.3	Entorno de control de datos	113
3.6.1.4	Auditoría	113
3.6.2	ORGANIZACIÓN DE DIRECTORIOS	114
3.6.2.1	Organización del código fuente dentro de la estructura de directorios.	116
4.	<u>CONCLUSIONES</u>	118
<hr/>		
5.	<u>RECOMENDACIONES</u>	120

6.	<u>BIBLIOGRAFÍA</u>	<u>121</u>
7.	<u>ANEXOS</u>	<u>123</u>

LISTA DE TABLAS

	pág.
<u>TABLA 1. CASOS DE USO: CREAR ROLES Y RESPONSABILIDADES</u>	<u>86</u>
<u>TABLA 2. CASOS DE USO: ADMINISTRAR COMPETENCIAS</u>	<u>87</u>
<u>TABLA 3. CASOS DE USO: REGISTRAR POLÉMICAS</u>	<u>89</u>
<u>TABLA 4. CASOS DE USO: PLANIFICAR CONTRATOS</u>	<u>91</u>
<u>TABLA 5. CASOS DE USO: SELECCIONAR ELEMENTOS ORDEN DE COMPRA</u>	<u>92</u>
<u>TABLA 6. ATRIBUTOS PRINCIPALES DE LA CLASE PROYECTO</u>	<u>96</u>
<u>TABLA 7. ATRIBUTOS PRINCIPALES DE LA CLASE ROLPROYECTO</u>	<u>96</u>
<u>TABLA 8. ATRIBUTOS PRINCIPALES DE LA CLASE COMPETENCIAROL</u>	<u>97</u>
<u>TABLA 9. ATRIBUTOS PRINCIPALES DE LA CLASE PERSONALPROYECTO</u>	<u>98</u>
<u>TABLA 10. ATRIBUTOS PRINCIPALES DE LA CLASE MODALIDADCONTRATACIÓN</u>	<u>98</u>

LISTA DE FIGURAS

	pág.
<u>FIGURA 1. DESCRIPCIÓN GENERAL DE LAS ÁREAS DE CONOCIMIENTO DE LA GESTIÓN DE PROYECTOS</u>	<u>41</u>
<u>FIGURA 2. SISTEMA DE PLANEACIÓN.</u>	<u>45</u>
<u>FIGURA 3. CICLO DE VIDA DEL TRABAJO.</u>	<u>47</u>
<u>FIGURA 4. CICLO BÁSICO DE UN PROYECTO.</u>	<u>47</u>
<u>FIGURA 5. COMPONENTES DE UN DIAGRAMA DE CASOS DE USO.</u>	<u>52</u>
<u>FIGURA 6. DIAGRAMA DE SECUENCIA</u>	<u>55</u>
<u>FIGURA 7. APLICACIONES DISTRIBUIDAS EN MULTICAPA</u>	<u>59</u>
<u>FIGURA 8. API DE LA PLATAFORMA JAVA EE.</u>	<u>61</u>
<u>FIGURA 9. ESQUEMA DEL MODELO BASADO EN PROTOTIPOS</u>	<u>78</u>
<u>FIGURA 10. ESTRUCTURA DEL MODELO BASADO EN PROTOTIPOS</u>	<u>78</u>
<u>FIGURA 11. DIAGRAMA DE CASOS DE USO – PLANIFICAR RECURSOS HUMANOS</u>	<u>85</u>
<u>FIGURA 12. DIAGRAMA DE CASOS DE USO – ADMINISTRAR EQUIPO DEL PROYECTO</u>	<u>88</u>
<u>FIGURA 13. DIAGRAMA DE CASOS DE USO – ADMINISTRAR COMPRAS Y ADQUISICIONES</u>	<u>90</u>
<u>FIGURA 14. DIAGRAMA DE CLASES – SGPUIS, GESTIÓN DE RECURSOS HUMANOS, COMPRAS Y ADQUISICIONES.</u>	<u>94</u>
<u>FIGURA 15. DIAGRAMA DE SECUENCIA PARA EL CASO DE USO CREAR ROLES Y RESPONSABILIDADES.</u>	<u>99</u>

<u>FIGURA 16. DIAGRAMA DE SECUENCIA PARA EL CASO DE USO ADMINISTRAR COMPETENCIAS.</u>	<u>100</u>
<u>FIGURA 17. DIAGRAMA DE SECUENCIA PARA EL CASO DE USO ADMINISTRAR COMPETENCIAS.</u>	<u>101</u>
<u>FIGURA 18. FORMATO CREAR ROL PROYECTO.</u>	<u>103</u>
<u>FIGURA 19. FORMATO DETALLE, EDICIÓN Y ELIMINACIÓN ROL PROYECTO</u>	<u>104</u>
<u>FIGURA 20. FORMATO DE EDICIÓN DE ROL PROYECTO Y ASOCIACIÓN DE COMPETENCIAS</u>	<u>105</u>
<u>FIGURA 21. FORMATO FINAL DE CREACIÓN DE ROL PROYECTO</u>	<u>107</u>
<u>FIGURA 22. FORMATO FINAL DE EDICIÓN DE ROL PROYECTO Y ASOCIACIÓN DE COMPETENCIAS</u>	<u>108</u>
<u>FIGURA 23. FORMATO DE EDICIÓN Y ELIMINACIÓN DE COMPETENCIAS</u>	<u>109</u>

LISTA DE ANEXOS

	pág.
<u>ANEXO A.</u>	<u>123</u>
<u>ANEXO B.</u>	<u>123</u>

GLOSARIO TÉCNICO

API: (Application Programming Interface). Interfaz de Programación de Aplicaciones). Grupo de rutinas (conformando una interfaz) que provee un sistema operativo, una aplicación o una biblioteca, que definen cómo invocar desde un programa un servicio que éstos prestan. En otras palabras, una API representa un interfaz de comunicación entre componentes software.

BASE DE DATOS: Se define como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. Una base de datos es un sistema de archivos electrónico.

CÓDIGO FUENTE: (Source code, code base). Texto escrito en un lenguaje de programación específico y que puede ser leído por un programador.

COMPILACIÓN: Proceso de traducción de un código fuente (escrito en un lenguaje de programación de alto nivel) a lenguaje máquina (código objeto) para que pueda ser ejecutado por la computadora.

CONTRASEÑA: (Clave o Password) Conjunto finito de caracteres limitados que forman una palabra secreta que sirve a uno o más usuarios para acceder a un determinado recurso. Las claves suelen tener limitaciones en sus caracteres (no aceptan algunos) y su longitud.

DBA: (Data Base Administrator). Administrador de la Base de Datos, quien se encarga del control general del Sistema de Base de Datos.

DBMS: (Data Base Management System). Sistemas de Gestión de Bases de Datos. Es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

EJB: Son componentes del contexto de servidor que cubren la necesidad de intermediar entre la capa web y diversos sistemas empresariales. Son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE (ahora JEE 5.0) de Oracle Corporation (inicialmente desarrollado por Sun Microsystems).

FRAMEWORK: Se refiere a “ambiente de trabajo, y ejecución”. En general los framework son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

HOME PAGE: (Start page, front page, main web). Página de inicio o principal de un sitio web. Suele tener el nombre: index.htm(l), aunque también se usan index.php, index.asp, etc.

HTML: (HyperText Markup Language). Lenguaje de Marcado de Hipertexto, es el lenguaje de programación predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, además de complementar el texto con objetos tales como imágenes.

INFORMIX: Es uno de los cuatro grandes plataformas o manejadores de la bases de datos (DBMS) junto con DB2 de IBM, SQL Server de Microsoft y Oracle.

INTERFAZ: En software, parte de un programa que permite el flujo de información entre un usuario y la aplicación, o entre la aplicación y otros programas o periféricos. Esa parte de un programa está constituida por un conjunto de comandos y métodos que permiten estas intercomunicaciones.

INTERNET: Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

INTRANET: Es una red de ordenadores privada basada en los estándares de Internet. Las Intranets utilizan tecnologías de Internet para enlazar los recursos informativos de una organización, desde documentos de texto a documentos multimedia, desde bases de datos legales a sistemas de gestión de documentos.

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.

JAVA EE: (Java Platform Enterprise Edition). Anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4, es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JBOSS DEVELOPER STUDIO (JBDS): Es un entorno de desarrollo comercial creado y actualmente desarrollado por JBoss (una división de Red Hat) y Exadel, Inc.

JBOSS EAP: (JBoss Enterprise Application Platform). Es la plataforma líder en el mercado para las aplicaciones Java innovadora y escalable. Integrada, simplificada y entregada por el líder en software empresarial open source, incluye tecnologías open source líderes para la construcción, implementación y hosting de servicios y aplicaciones empresariales Java.

JBOSS HIBERNATE: Hibernate es un potente servicio de mapeo objeto/relacional de alto desempeño para consulta y persistencia. Le permite desarrollar clases

persistentes siguiendo el lenguaje orientado a objetos, incluyendo la asociación, herencia, polimorfismo, composición y colecciones.

JBOSS SEAM: Es un framework que integra y unifica los distintos estándares de la plataforma Java EE 5.0, pudiendo trabajar con todos ellos siguiendo el mismo modelo de programación. Ha sido diseñado intentado simplificar al máximo el desarrollo de aplicaciones, basando el diseño en Plain Old Java Objects (POJOs) con anotaciones. Estos componentes se usan desde la capa de persistencia hasta la de presentación, poniendo todas las capas en comunicación directa.

JDK: (Java Development Kit). Es un grupo de herramientas para el desarrollo de software provisto por Sun Microsystems Inc. Incluye las herramientas necesarias para escribir, testear, y depurar aplicaciones y applets de Java.

JPA: (Java Persistence API). Es un lenguaje de programación Java que permite a los desarrolladores gestionar datos relacionales en aplicaciones que usan Java Platform Standard Edition y Java Platform Enterprise Edition.

JPQL: (Java Persistence Query Language). Es un lenguaje de consulta orientado a objetos, definida como parte de la especificación Java Persistence API. JPQL se utiliza para hacer consultas a las entidades almacenados en una base de datos relacional. Está fuertemente inspirado en SQL y sus preguntas se asemejan a las consultas SQL en la sintaxis, pero operan contra los objetos entidad JPA y no directamente con las tablas de base de datos.

JVM: (Java Virtual Machine). Máquina virtual Java. Es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el Java bytecode), el cual es generado por el compilador del lenguaje Java.

LOGIN: (autenticación). Es el proceso de intento de verificar la identidad digital del remitente de una comunicación como una petición para conectarse. El remitente

siendo autenticado puede ser una persona que usa un ordenador, un ordenador por sí mismo o un programa del ordenador.

MAPEO OBJETO RELACIONAL: (Object-Relational Mapping - ORM). Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

PÁGINA: Documento o archivo codificado en un lenguaje de marcado, usualmente HTML, que suele contener textos con hipervínculos, imágenes, sonidos, etc., que en la mayoría de los casos reside en un sitio o servidor web, y puede ser consultado empleando un navegador web.

PERSISTENCIA: Se refiere a la propiedad de los datos para que estos sobrevivan de alguna manera. De forma sencilla puede entenderse que los datos tienen una duración efímera, desde el momento en que estos cambian de valor se considera que no hay persistencia de los mismos.

REGISTRO: Es un tipo de dato estructurado formado por la unión de varios elementos bajo una misma estructura. Estos elementos pueden ser, o bien datos elementales (entero, real, carácter, etc.), o bien otras estructuras de datos. A cada uno de esos elementos se le llama campo.

SERVIDOR: Es una computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.

SISTEMA DE INFORMACIÓN: Es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Un sistema de

información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

TRIGGER O DISPARADOR: Se define así a una subrutina que es ejecutada de manera automática cuando se produce algún tipo de transacción (inserción, borrado o actualización) en la tabla de una base de datos.

UML: (Unified Modeling Language). Lenguaje Unificado de Modelado. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

USUARIO: Es la persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público, privado, empresarial o profesional.

RESUMEN

TITULO: Diseño, implementación y puesta en funcionamiento del módulo de planificación: gestión de recursos humanos, compras y adquisiciones del Sistema de Gestión de Proyectos UIS (SGPUIIS).¹

AUTORES: YENNY LISSETH TOBO NIÑO²
EDWING ANTONIO GARCIA MALDONADO³

PALABAS CLAVES:

Project Management Institute (PMI), PMBOK, División de Servicios de Información, JAVA EE5, JBoss Developer Studio.

DESCRIPCIÓN: La Universidad Industrial de Santander (UIS), ha venido consolidando políticas para el manejo de sus recursos de inversión y proporcionando elementos a través del Banco de Programas y Proyectos de Inversión, fomentando una cultura de trabajo más estructurada y orientada a proyectos. Este trabajo de grado fue desarrollado en conjunto con la División de Servicios de Información, unidad encargada de administrar y desarrollar el software que soporta el funcionamiento de la UIS, y la Oficina de Planeación de la Universidad Industrial de Santander encargada de coordinar y supervisar la formulación, ejecución, seguimiento y control de los proyectos que se enmarcan dentro del plan de desarrollo institucional.

La cantidad de información manejada en el Banco de Programas y Proyectos de inversión, hace necesaria la elaboración de un sistema de información moderno que agilice los procesos. En la gestión de proyectos el manejo del recurso humano, el proceso de compras y adquisiciones, juegan un papel importante para alcanzar los objetivos propuestos en un proyecto, razón que motivó el desarrollo de este software como soporte a estos procesos. En la gestión del recurso humano la nueva herramienta sistematiza los procesos del manejo de personal asociado al proyecto, ya que está debidamente integrado con el sistema de Recursos Humanos de la Universidad Industrial de Santander.

¹ Proyecto de grado en la modalidad de Investigación.

^{2,3} Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director: Ing. Enrique Torres López. Codirector: Ing. Giobanni Serrano Durán.

ABSTRAC

TITLE: Design, implementation and operation planning module: Human resources, procurement and acquisition management of Project Management System UIS (SGPUIS).⁴

AUTHORS: YENNY LISSETH TOBO NIÑO⁵
EDWING ANTONIO GARCIA MALDONADO⁶

KEY WORDS:

Project Management Institute (PMI), PMBOK, Information Services Division, JAVA EE5, JBoss Developer Studio.

DESCRIPTION: The Industrial University of Santander (UIS) has been consolidating policies for the management of their investment resources and providing elements through the Bank of Programs and Investment Projects, promoting a work culture more structured and projects-oriented. This work was developed jointly with the Information Services Division, a unit responsible for managing and developing the software that supports the operation of UIS, and the Office of Planning of the Industrial University of Santander who coordinates and supervises the formulation, execution, monitoring and control of the projects that fall within the institutional development plan.

The amount of information handled in the bank of programs and investment projects, requires the elaboration of a modern information system that speeds up the processes. In the project management, the human resource management and the process of purchase and acquisition, have an important role in achieving the objectives in a project, making it the reason behind the development of this software to support these processes. In the human resource management, the new tool systematize the process of personnel management associated to the project, because it is properly integrated with the system of Human Resources in the Industrial University of Santander.

⁴ Degree Project in the modality Research

⁵ ⁶ Faculty of Physical-Mechanical Engineering. Systems and Computer Engineering.
Director: Ing. Enrique Torres López. Codirector: Ing. Giovanni Serrano Durán.

INTRODUCCIÓN

Actualmente la velocidad con la que se crean nuevas tecnologías crece exponencialmente, así como la aparición de nuevas y mejores metodologías para el desarrollo de proyectos informáticos, acompañado con el auge de las aplicaciones orientadas a la Web, con las que las organizaciones tienen mayor control y manejo de su información, la cual se convierten en un valor agregado de suma importancia, pues de acuerdo a la forma como esta es utilizada, se ve reflejado en el desempeño económico y competitivo de la organización.

La Universidad Industrial de Santander y particularmente la Oficina de Planeación con el Banco de Proyectos (enfocado en la formalización y organización de la inversión en la Universidad), no es ajena a esta situación, ya que debe satisfacer todos los requerimientos de la Universidad para cumplir las metas de desarrollo institucional contempladas en el Plan de Desarrollo.

Con el cambio de las tecnologías usadas en la Universidad y las modificaciones que se realizaron a los modelos para la gestión de proyectos, se exigía el desarrollo de una herramienta más moderna que se adaptara a la naturaleza dinámica de la universidad y los proyectos que ella gestiona.

Con base en las motivaciones expuestas, se desarrolla el módulo de planificación: gestión de recursos humanos, compras y adquisiciones del Sistema de Gestión de Proyectos Institucional (SGPUI) en conjunto con la División de Servicios de información como parte de un proyecto de grado.

Como resultado de este proyecto se diseñó y desarrolló una herramienta moderna, dinámica, con mayores facilidades de uso y que maneja las políticas actuales; todo esto envuelto en un proyecto macro que comprende otros proyectos enfocados a satisfacer las necesidades de áreas críticas de la gestión. Dicho

módulo se realizó con los estándares de programación definidos en la División de Sistemas de Información y bajo las herramientas allí utilizadas para que futuros cambios que se requieran, se hagan de una forma rápida y sencilla.

Este documento contiene el soporte teórico, metodológico y técnico del desarrollo Web del módulo de planificación: gestión de recursos humanos, compras y adquisiciones del Sistema de Gestión de Proyectos (SGPUI) de la Universidad Industrial de Santander.

1. CONTEXTO GENERAL

1.1 DEFINICIÓN DEL PROBLEMA

1.1.1 Entidad interesada en el proyecto.

Nombre: UNIVERSIDAD INDUSTRIAL DE SANTANDER⁷.

Misión:

La Universidad Industrial de Santander es una organización que tiene como propósito la formación de personas de alta calidad ética, política y profesional; la generación y adecuación de conocimientos; la conservación y reinterpretación de la cultura y la participación activa liderando procesos de cambio por el progreso y mejor calidad de vida de la comunidad.

Orientan su misión los principios democráticos, la reflexión crítica, el ejercicio libre de la cátedra, el trabajo interdisciplinario y la relación con el mundo externo.

Sustenta su trabajo en las cualidades humanas de las personas que la integran, en la capacidad laboral de sus empleados, en la excelencia académica de sus profesores y en el compromiso de la comunidad universitaria con los propósitos institucionales y la construcción de una cultura de vida.

Visión:

La Universidad Industrial de Santander es una institución de educación superior estatal y autónoma, financiada por el Estado, comprometida con la defensa de un estado social y democrático de derecho y de derechos humanos y la proposición

⁷ La Misión, Visión, fueron tomados del Proyecto Institucional, Universidad Industrial de Santander

de políticas públicas que garanticen el acceso de la población a condiciones de vida digna.

La UIS es actor principal del desarrollo económico, social y cultural de la región y ejemplo de democracia, convivencia, autonomía y libertad responsable. Es lugar de consulta sobre las tendencias y desarrollos en el campo de las ciencias, los avances tecnológicos, las necesidades y oportunidades del mundo del trabajo y los deseos de bienestar de la comunidad.

La vigencia social de la universidad se manifiesta en su participación activa en Organismos de planificación local, regional y nacional, en agrupaciones de participación ciudadana para la proposición y el seguimiento de políticas y programas de desarrollo social, económico y cultural. En el fortalecimiento de sus relaciones con los sectores políticos, sociales y generadores de bienes y servicios que propendan por el bien común, en el marco de la conveniencia institucional. En la integralidad de todos los miembros de la comunidad universitaria, los cuales están formados en el espíritu científico. En la apropiación y el ejercicio de los derechos humanos universales y los derechos políticos, económicos, sociales y culturales correspondientes a la práctica de la ciudadanía y en el ejercicio de una conducta profesional solidaria con la construcción de la nación colombiana.

Es sitio obligado de referencia y consulta para proponer o evaluar las alternativas de solución a los problemas prioritarios de la comunidad, y su contribución es ampliamente valorada como insumo crítico para continuar avanzando en la construcción de una sociedad en donde la equidad, la justicia, la solidaridad y el respeto por los derechos humanos y la naturaleza, sean los pilares del desarrollo humano sostenible en el marco de una cultura de paz.

Es líder del desarrollo científico en bio-ingeniería, fuentes alternativas de energía, petroquímica y carboquímica, nuevas opciones para uso de combustibles,

nuevos materiales y tecnologías de materiales compuestos, aprovechamiento y uso sostenible de la biodiversidad, promoción de la salud, prevención y control de las enfermedades de mayor ocurrencia, estímulo y acompañamiento a procesos de organización comunitaria orientados al desarrollo social y cultural, y mejoramiento de la calidad de la educación en todos sus niveles. Mantiene como líneas transversales la investigación en electrónica, telecomunicaciones, informática y ciencia y tecnología del medio ambiente. Promueve el desarrollo de la literatura y las artes. En todas sus Escuelas, Centros e Institutos, los miembros de la comunidad universitaria actúan como docentes-investigadores y se mantienen interconectados con grupos de pares académicos que cooperan local, nacional e internacionalmente.

Ofrece, desde la región nororiental al país, formación permanente de alta calidad y pertinencia social, propendiendo por la equidad en el acceso, con fundamento en el mérito académico. Sostiene intercambios y pasantías de profesores y estudiantes con Universidades extranjeras de alta calidad y presenta una amplia oferta de programas presenciales e interactivos mediante tecnologías para la educación virtual.

Es una organización inteligente capaz de adaptarse con eficacia a la velocidad de los cambios y a las necesidades emanadas del entorno. Recibe del Estado los recursos suficientes para adelantar sus funciones de investigación, formación y proyección social, en reconocimiento a su calidad, a los resultados presentados anualmente ante la sociedad y a sus políticas de eficiencia en la utilización de los recursos. Invierte sus rentas propias para fortalecer su posición de excelencia en el medio universitario.

1.2 OBJETIVOS

1.2.1 Objetivo general.

Realizar el diseño, implementación y puesta en funcionamiento del módulo de Planificación del Sistema de Gestión de Proyectos UIS, específicamente en la gestión de recursos humanos, compras y adquisiciones; mediante el uso de últimas tecnologías de programación web.

1.2.2 Objetivos específicos.

- Servir de instrumento para la planificación de los recursos humanos específicamente para:
 - Facilitar el registro, actualización y consulta: Del organigrama del proyecto, de los roles establecidos para la gestión del proyecto con sus competencias y responsabilidades.
 - Permitir el registro, actualización, consulta: Del equipo humano seleccionado para la ejecución del proyecto, de las polémicas que se presenten durante el desarrollo del proyecto y resultados de la evaluación de sus competencias. Este módulo estará integrado con el sistema de Talento Humano de la universidad
- Identificar las necesidades del proyecto mediante la planificación de Compras y Adquisiciones, este proceso implica:
 - Hacer seguimiento y generar alarmas del plan de compras y adquisiciones establecido para el proyecto.
 - Facilitar el proceso de selección de las compras y adquisiciones para establecer si se hacen bajo orden de compra, convocatoria o licitación.
 - Generar la orden de compra debidamente integrada con el sistema financiero.

- Generar los archivos base para la elaboración de los pliegos de la convocatoria o de la licitación con información de los insumos y/o recurso humano a ser contratados.
 - Generar las órdenes de compra y/o contratos derivados de las convocatorias y licitaciones realizadas durante la ejecución del proyecto.
 - Facilitar el seguimiento de las compras y adquisiciones realizadas durante el proyecto.
 - Generar variaciones de valores de las compras y/o adquisiciones realizadas con relación al presupuesto inicial.
- Seguir las políticas de seguridad de la universidad Industrial de Santander teniendo en cuenta:
 - Establecer roles para los usuarios con el fin de definir los permisos de acceso al sistema.
 - Definir los alcances y funciones que puede desempeñar cada usuario en el sistema, buscando así mantener la integridad y la seguridad del mismo.
 - Servir de Herramienta de apoyo para la realización de un proceso confiable de auditorías, fijando las pistas de auditoría necesarias como parte de la estructura de datos establecida.

1.3 JUSTIFICACIÓN

La Universidad Industrial de Santander, es una entidad de educación superior en continuo cambio: el aumento del número de estudiantes, la creación permanente de nuevos programas académicos, avances en proyectos de investigación, entre otros aspectos, hace que el trabajo de las Unidades académico administrativas (UAA) de apoyo a la gestión institucional sea más complejo y difícil de realizar. Su estructura tradicional, aunque funciona en pro de las necesidades del entorno académico – administrativo, se ve limitada en eficiencia y velocidad en los

procesos de soporte académico y de investigación, dado que los tiempos de respuesta a las solicitudes y a las necesidades de implementación de las políticas no es el mejor. En ese sentido, la gestión de proyectos se convierte en una herramienta, que hace más dinámica la aplicabilidad de las políticas de desarrollo institucional.

La aplicación para la formulación y seguimiento de proyectos, que utiliza actualmente la UIS, no satisface las expectativas actuales, dado que no se adapta a la naturaleza dinámica de la institución y no permite un óptimo control de los proyectos en cada una de las fases de su ciclo de vida; de ahí, que, el nuevo sistema de Gestión de Proyectos de la UIS (SGPUI) surge como alternativa para gestionar de manera adecuada los proyectos identificados en el marco del Plan de Desarrollo Institucional, el cual tiene una cobertura hasta el 2018 y cuenta con el respaldo económico de Colciencias, Recursos de Estampilla ProUIS, y recursos propios orientados a la generación de proyectos.

Este proyecto está siendo coordinado por la Oficina de Planeación con el apoyo de la División de Servicios de Información y contará con la participación de algunos actores potenciales que harán uso de esta herramienta, entre ellos, la Dirección de Contratación, División Financiera, División de Planta Física, División de Mantenimiento Tecnológico, Vicerrectoría Académica, Vicerrectoría Administrativa, Vicerrectoría de Investigación y Extensión, las cuales tienen bajo su responsabilidad la formulación y ejecución de la mayor parte de proyectos que adelanta la institución.

Los beneficios en los que podría contribuir esta investigación son los siguientes:

- Brindar a la Dirección de la Universidad, una herramienta gerencial que le permita orientar los recursos de inversión hacia las estrategias de desarrollo plasmadas en el Plan de Desarrollo Institucional PDI.

- Facilitar el seguimiento y control del desarrollo de un proyecto durante todas sus fases.
- Aportar información para valorar los resultados de las inversiones realizadas versus los avances académicos e institucionales proyectados en los planes de desarrollo de la Universidad.
- Dotar a las UAA de la Universidad de los elementos procedimentales para el desarrollo de los proyectos, con el fin de garantizar la coherencia en la maduración y desarrollo de las propuestas.
- Contribuir con mejoras operativas, minimización de procedimientos y confiabilidad de la información, mediante la integración del SGPUIS con los sistemas de Recursos Humanos, Financiero y Académico.
- Permitir la Administración de roles y responsabilidades en la gestión de un Proyecto, para contar con un proceso confiable de auditorías y de gestión del personal.
- Controlar de forma eficaz la ejecución de compras y adquisiciones de un proyecto, así como el uso de los recursos disponibles.

1.4 VIABILIDAD

- Técnica

Para el desarrollo del proyecto, se cuenta con el recurso humano suficiente, apoyado por profesionales de la División de Servicios de Información con alta experiencia en la dirección de proyectos y en el manejo de las herramientas de modelado y desarrollo de aplicaciones. Adicionalmente, existe una infraestructura tecnológica conformada por software, equipos de desarrollo,

servidores de base de datos, prueba y producción que garantizan la óptima ejecución del proyecto.

- Económica

Como se ha mencionado, la División de Servicios de Información cuenta con todas las herramientas para la elaboración del proyecto, entre ellas se destacan las licencias de software de modelado y programación, los recursos necesarios para la capacitación y el personal de apoyo, que estarán disponibles a lo largo del desarrollo del proyecto.

- Social

El proyecto satisface las necesidades planteadas por la División de Servicios de Información y la oficina de Planeación, referentes a la gestión de los proyectos de la universidad. Se le considera de alto impacto porque beneficia a gran parte de la comunidad universitaria:

- A la Dirección de la Universidad le permitirá orientar los recursos de inversión hacia las estrategias de desarrollo plasmadas en el Plan de Desarrollo Institucional PDI.

A las unidades Académico-Administrativas le permitirá contar con una herramienta y elementos procedimentales para el desarrollo de los proyectos, que garanticen la coherencia en la maduración y desarrollo de sus propuestas.

2. MARCO TEÓRICO

2.1 ESTADO DEL ARTE

La herramienta que existe actualmente para el Banco Proyectos de la Universidad Industrial de Santander, fue desarrollada en el año 2001 por parte de los estudiantes Silvia Andrea Suarez Barón y Edwin Hernández Ramírez, como proyecto de Grado para obtener el Título de Ingenieros de Sistemas.

El sistema está implantado en la intranet de la Universidad Industrial de Santander y fue implementado en entorno web, utilizando ASP como plataforma de desarrollo; La División de Servicios de Información es la encargada de dar el respectivo soporte técnico, mantenimiento y continuo desarrollo del Sistema.

Actualmente el sistema que soporta al Banco de Proyectos de la UIS, no cuenta con facilidades de evaluación de viabilidad específicos, no permite el seguimiento de los proyectos durante su ejecución, ni tampoco su liquidación, no se encuentra integrado con los sistemas de Recursos Humanos, Financiero y Académico, criterios indispensables para hacer eficiente la gestión, seguimiento y evaluación de los Proyectos que se ejecutan en la institución.

Lo anterior, refleja las debilidades administrativas y tecnológicas que presenta el sistema de información que soporta el Banco de Programas y Proyectos, que hacen necesario desarrollar una solución que supla estas debilidades y que permita contar con una alternativa que llene las expectativas de la Oficina de Planeación, unidad de la Universidad Industrial de Santander encargada de administrar el banco de proyectos de la institución. La solución propuesta está enmarcada dentro del proyecto Sistema de Gestión de Proyectos Institucional (SGPUIIS).

El sistema de Gestión de Proyectos Institucional (SGPUIIS) tiene como objetivo gestionar de manera adecuada los proyectos identificados en el marco del Plan de Desarrollo Institucional: desde ideas sencillas como proyectos de mantenimiento, hasta proyectos de mayor impacto, como la creación de un nuevo programa académico.

Para servir de soporte a todos los proyectos que adelanta la Universidad, el SGPUIIS utiliza como base la metodología de gestión de proyectos, propuesta por el Project Management Institute (PMI®)⁸ y la metodología general aplicada de Planeación Nacional (MGA).

2.2 GENERALIDADES

2.2.1 Proyecto

Un proyecto es una planificación que consiste en un conjunto de actividades y medidas institucionales que se encuentran interrelacionadas y coordinadas; la razón de un proyecto es alcanzar objetivos específicos dentro de los límites que imponen un presupuesto, y que es desarrollado en un periodo determinado. Surge como respuesta a una necesidad, acorde con la visión de la organización, aunque ésta puede desviarse en función del interés. El proyecto finaliza cuando se obtiene el resultado deseado, desaparece la necesidad inicial, o se agotan los recursos disponibles [1].

⁸ El Project Management Institute (PMI®) es una organización internacional sin fines de lucro que asocia a profesionales para la gestión de proyectos. Sus principales objetivos son: 1) Formular estándares profesionales, 2) Generar conocimiento a través de la investigación y 3) Promover la Gestión de Proyectos como profesión a través de sus programas de certificación.

Project Management Institute, A guide to project management body of knowledge: PMBOK guide. 4rd ed., Project Management Institute, 2008.

La gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer los requisitos del proyecto [2].

2.2.1.1 Características del proyecto

Temporal

Temporal significa que cada proyecto tiene un comienzo definido y un final definido. El final se alcanza cuando se han logrado los objetivos del proyecto o cuando queda claro que los objetivos del proyecto no serán o no podrán ser alcanzados, o cuando la necesidad del proyecto ya no exista y el proyecto sea cancelado. Temporal no necesariamente significa de corta duración; muchos proyectos duran varios años. En cada caso, sin embargo, la duración de un proyecto es limitada. Los proyectos no son esfuerzos continuos.

Productos, servicios o resultados únicos

Un proyecto crea productos entregables únicos. Productos entregables son productos, servicios o resultados. Los proyectos pueden crear:

- Un producto o artículo producido, que es cuantificable, y que puede ser un elemento terminado o un componente.
- La capacidad de prestar un servicio como, por ejemplo, las funciones del negocio que respaldan la producción o la distribución.
- Un resultado como salidas o documentos. Por ejemplo, de un proyecto de investigación se obtienen conocimientos que pueden usarse para determinar si existe o no una tendencia o si un nuevo proceso beneficiará a la sociedad. La singularidad es una característica importante de los productos entregables de un proyecto. Por ejemplo, se han construido muchos miles de edificios de oficinas, pero cada edificio individual es único: diferente propietario, diferente

diseño, diferente ubicación, diferente contratista, etc. La presencia de elementos repetitivos no cambia la condición fundamental de único del trabajo de un proyecto.

Elaboración gradual

La elaboración gradual es una característica de los proyectos que acompaña a los conceptos de temporal y único. “Elaboración gradual” significa desarrollar en pasos e ir avanzando mediante incrementos. Por ejemplo, el alcance de un proyecto se define de forma general al comienzo del proyecto, y se hace más explícito y detallado a medida que el equipo del proyecto desarrolla un mejor y más completo entendimiento de los objetivos y de los productos entregables. La elaboración gradual no debe confundirse con la corrupción del alcance.

2.2.1.2 Proyectos y planificación estratégica

Los proyectos se convierten en una manera de organizar actividades que no pueden ser tratadas dentro de los límites operativos normales de la organización. Por tanto, los proyectos son empleados a menudo como un medio de lograr el plan estratégico de la organización. Generalmente, los proyectos son autorizados como resultado de una o más de las siguientes consideraciones estratégicas:

- Demanda del mercado.
- Necesidad de la organización.
- Solicitud de un cliente.
- Avance tecnológico.
- Requisito legal.

2.2.2 Dirección de Proyectos- Áreas de conocimiento de la Dirección de Proyectos

La dirección de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer las necesidades del mismo. La dirección de proyectos se logra por medio de la aplicación e integración de los procesos de dirección de proyectos de inicio, planificación, ejecución, seguimiento y control, y cierre.

El término Dirección de Proyectos se utiliza a veces para describir un enfoque organizacional para el manejo o administración de operaciones continuas. Este enfoque, más correctamente llamado gestión por proyectos, trata los diversos aspectos de las operaciones continuas como proyectos a los cuales se les aplican las técnicas de gestión de proyectos.

La dirección de un proyecto incluye:

- Identificar los requisitos.
- Establecer unos objetivos claros y posibles de realizar.
- Equilibrar las demandas concurrentes.
- Adaptar las especificaciones, planes y enfoque a las distintas inquietudes y expectativas de los interesados.

El director o líder del proyecto (El Project Manager (**PM**)) es la persona encargada de alcanzar los objetivos del proyecto, además es clave en todo este proceso. Un **PM** debe tener capacidad de liderazgo “de hecho”, es decir, que los integrantes del grupo le reconozcan un poder de mando concreto. Además, requiere una alta capacidad de comunicación, de rápida resolución de los problemas y de influencia sobre la organización. Por último, debe disponer de pensamiento sistémico o global, que le permita ver las cosas desde una perspectiva amplia y general.

2.2.2.1 La estructura de la gerencia de proyectos

Establece una estructura básica para la comprensión de la Dirección de Proyectos.

El Contexto de la Gestión de Proyectos, describe el ambiente en que operan los proyectos. El equipo de gestión de proyectos debe entender este contexto más ampliamente –la gestión de las actividades diarias del proyecto es algo necesario para el éxito, pero no suficiente.

Los Procesos de la Gestión de Proyectos, describe una visión generalizada de cómo interactúan comúnmente los distintos procesos de la gestión de proyectos.

Las Áreas de Conocimiento de la Gestión de Proyectos, describen el conocimiento y la práctica de la gestión de proyectos en términos de sus procesos integrados.

2.2.2.2 Áreas de conocimiento en la dirección de proyectos

Organiza los 44 procesos de dirección de proyectos de los Grupos de Procesos de la Dirección de Proyectos en nueve áreas de conocimiento:

- Gestión de Integración de Proyectos
- Gestión del Alcance del Proyecto
- Gestión de Duración del Proyecto
- Gestión de Costos del Proyecto
- Gestión de Calidad del Proyecto
- Gestión de Recursos Humanos del Proyecto
- Gestión de Comunicaciones del Proyecto
- Gestión de Riesgos del Proyecto
- Gestión de las Adquisiciones del Proyecto

Cada uno de los procesos de dirección de proyectos requeridos se muestra en el Grupo de Procesos en el cual se lleva a cabo la mayor parte de la actividad.

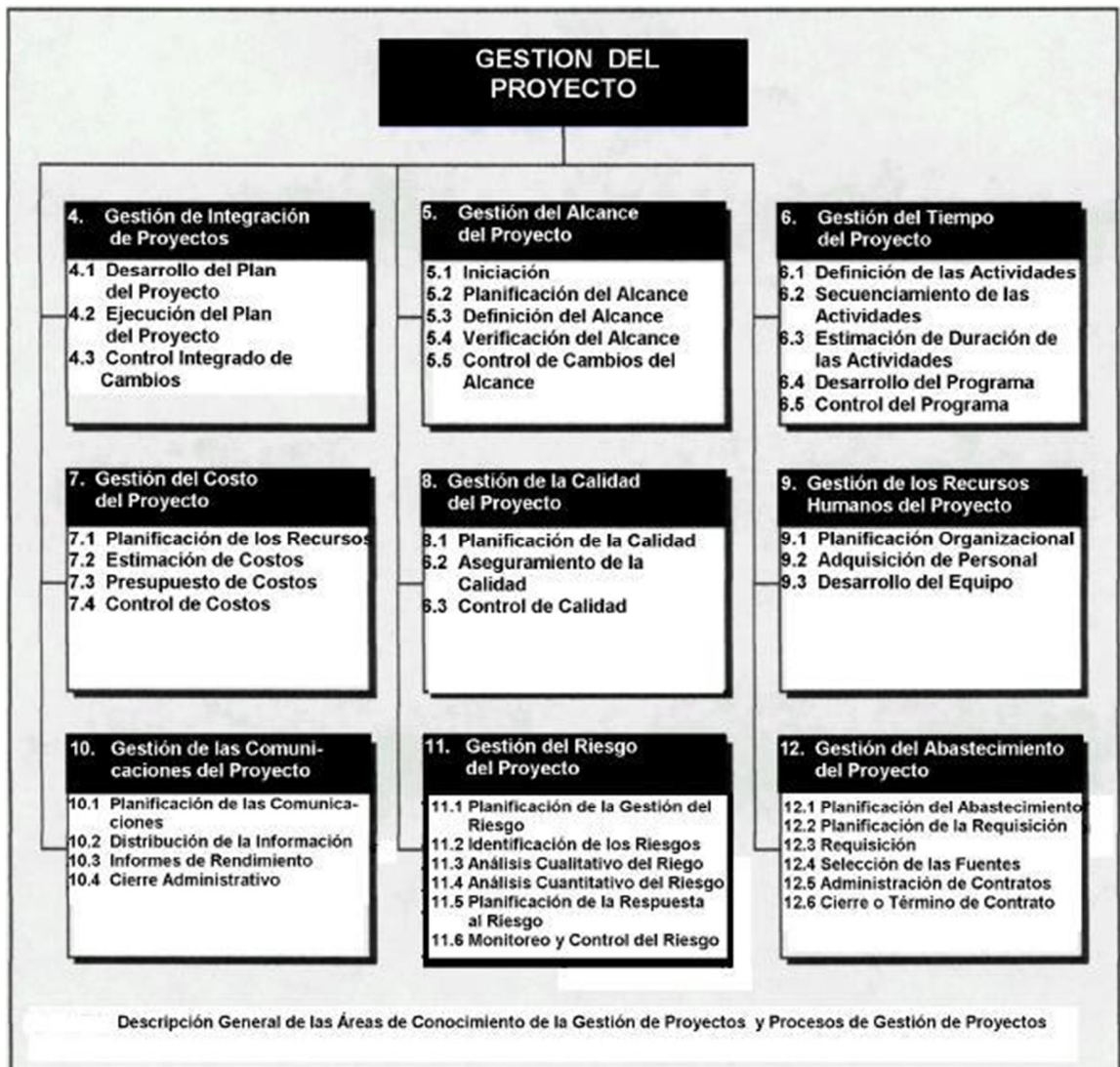


Figura 1. Descripción General de las Áreas de conocimiento de la Gestión de Proyectos

Especialmente el presente proyecto se centra en lograr completar los procesos de la Gestión de los Recursos Humanos del Proyecto y la Gestión de las Adquisiciones del Proyecto.

2.3 GESTIÓN DE LA INVERSIÓN EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

Mediante el acuerdo N° 032 del año 2002 expedido por el Consejo Superior de la Universidad Industrial de Santander se aprobó la reglamentación de la inversión institucional de dicha entidad.

El motivo por el cual se implementó esta regulación fue “brindar las herramientas normativas, técnicas y administrativas para la asignación de recursos a las diferentes Unidades Académico Administrativas de la Universidad, teniendo como base criterios de claridad, transparencia, conforme a los lineamientos de desarrollo de la institución y mediante la aplicación del Sistema de Planificación Institucional de la Universidad”⁹.

Con este acuerdo, se logró conferir solidez a la estructura de todos los procedimientos involucrados en las diferentes responsabilidades de las Directivas de la Institución.

Los nuevos lineamientos del Banco de Proyectos están constituidos por una variedad de componentes que hacen de la gestión de proyectos un eslabón indispensable para cumplir con “las políticas de desarrollo institucional, en la aplicación del proyecto educativo institucional (PEI) y en todas aquellas iniciativas que apuntaran al crecimiento de la Universidad”¹⁰. Por tal razón, con el tiempo fue siendo pieza fundamental para el Sistema de Planificación Institucional, junto con el Sistema de Inversión Institucional y el Programa de Gestión Anual, para darle eficiencia y eficacia a la labor de la Dirección de la entidad educativa.

⁹ SERRANO, Giobani. Plan de proyecto: propuesta de modelo para la Gestión de Proyectos de Inversión en una Institución Pública de Educación Superior en Colombia: una perspectiva desde el Pensamiento Sistémico. Universidad Industrial de Santander. Bucaramanga. 2009. P. 23.

¹⁰ *Ibíd.* P 23.

2.4 BANCO DE PROGRAMAS Y PROYECTOS DE INVERSIÓN NACIONAL– BPIN

El BPIN, Banco de Programas y Proyectos de Inversión Nacional nace con la ley 38 de 1989, como producto de un importante convenio de cooperación con el Banco Interamericano de Desarrollo (BID) y el Instituto Latinoamericano de Planificación Económica y Social (ILPES).

EL BPIN se constituye en el organismo encargado del montaje y consolidación de la red nacional de Bancos de Proyectos y para el cumplimiento de de su misión requiere la ejecución de las siguientes funciones.

- Establecer políticas estratégicas de desarrollo y operación de los bancos de programas y proyectos de inversión pública en Colombia.
- Consolidar información acerca de la inversión pública nacional.
- Promover acciones para consolidar la cultura de la preinversión.
- Apoyar la planeación, presupuestación, seguimiento, control y evaluación a partir de programas y proyectos.
- Dirigir procesos de capacitación y asistencia técnica en teoría de proyectos, metodologías y marco conceptual.

El BPIN además de contar con un soporte legal e institucional, incluía tres componentes básicos desde su diseño:

2.4.1 Componente de Metodología

Con este elemento se busca dar una nueva presentación y proposición para la ejecución de los recursos de inversión del Presupuesto General de la Nación; formulando ordenadamente las propuestas a través de proyectos, los cuales deben cumplir unas condiciones mínimas y procedimientos impuestos por el Departamento Nacional de Planeación (DNP), para todas las entidades del ámbito público.

2.4.2 Componente de Capacitación

La capacitación tiene el objetivo de fomentar el uso de las Metodologías sancionadas por el DNP y también, la divulgación del uso de técnicas concernientes a la evaluación de proyectos para finalmente lograr una cultura de proyectos en todo el territorio nacional.

2.4.3 Componente de Sistemas de Información

Con la creciente adopción de los Sistemas de Información como pieza clave para la toma de decisiones y principal responsable del almacenamiento de gran cantidad de información, el DNP no descarta el desarrollo de una herramienta computacional con el fin de darle funcionalidad al BPIN en la identificación de proyectos viables para su posterior aprobación o rechazo.

2.5 BANCO DE PROGRAMAS Y PROYECTOS DE INVERSIÓN DE LA UIS (BPPIUIS)

A partir del Acuerdo N° 32 de 2002 del Consejo Superior se instaura el Banco de Programas y Proyectos de la Universidad Industrial de Santander cuyo funcionamiento, está directamente involucrado en varios de los procesos que tienden a optimizar el empleo de los recursos destinados a la inversión adoptando el Programa Operativo Anual de Inversiones POAI¹¹. En este sentido el Banco de Programas y Proyectos es un instrumento invaluable para los procesos de planeación, presupuestación, programación, ejecución, evaluación y gestión, los cuales deben tener como objetivo final, que el plan de Desarrollo Institucional se haga realidad y se perciban sus beneficios para la comunidad Universitaria.

¹¹ Acuerdo N° 064 de 2008 del Consejo Superior – Universidad Industrial de Santander



Figura 2. Sistema de Planeación.

Este banco, está adscrito a Planeación y tiene como principales funciones:

- Garantizar la eficiente y oportuna información y programación de inversiones en la Universidad.
- Fortalecer la capacidad de gestión de la Universidad en la definición y manejo de las inversiones para lograr así mayor eficiencia y transparencia en la asignación de recursos.
- Articular los procesos de planeación con la programación y el uso de los recursos de inversión de la Universidad.
- Fortalecer y consolidar la cultura de proyectos en la Universidad.
- Elevar la oferta de programas y proyectos de inversión bien identificados y preparados como antecedente fundamental para mejorar el proceso de toma de decisiones para la asignación de recursos.
- Mejorar la eficiencia y la eficacia en las etapas de planeación, presupuestación, programación, ejecución, evaluación y gestión de la inversión.
- Establecer un sistema de información que facilite la articulación del presupuesto de inversión y del programa de gestión.

- Conocer el estado de avance y las causas de los atrasos de los proyectos de inversión institucional.

Es necesario destacar que el sistema presupuestal guarda coherencia con el sistema de planeación y está constituido por el Programa Operativo Anual de Gestión, el Plan Financiero, de Inversiones y el Presupuesto General de la Universidad.

2.6 ETAPAS DE UN PROYECTO

Para poder administrar un proyecto en un contexto de calidad, éste deberá pasar por varias fases, al final de las cuales deberán definirse los acontecimientos importantes. Cada etapa se relaciona con una prestación y una validación basadas en un documento específico. Esto permite supervisar los productos finales para que cumplan con los requisitos definidos y asegurar el cumplimiento de los costos pactados y del tiempo establecido.

Estas etapas de validación, que representan una de las tareas de la gestión de proyectos, permiten una rápida identificación de los distintos problemas y la adaptación a nuevas restricciones que surgen debido a circunstancias no previstas inicialmente. Es esencial realizar un seguimiento del tiempo destinado a cada tarea e indispensable analizar los riesgos. De hecho, la puesta en marcha de un proyecto genera mucha incertidumbre, ya que sus características todavía no se han formalizado. Esto presenta tantos riesgos para el proyecto, que es necesario intentar identificarlos y anticiparlos. Así, deben preverse medidas tanto para prevenir como para corregir la aparición de este tipo de riesgos en caso de ser necesario. En cada etapa, es posible que el cliente decida detener el proyecto si considera que no podrán lograrse los objetivos.

La cadena de etapas y validaciones entre el surgimiento de la necesidad y la entrega del producto se denomina "ciclo de vida del producto". El "ciclo de vida del trabajo" corresponde a las etapas y a los productos necesarios para la realización de los trabajos.

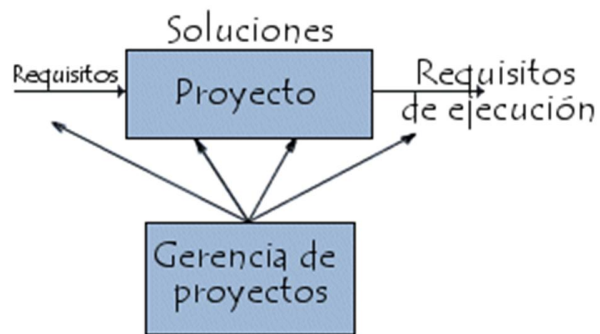


Figura 3. Ciclo de vida del trabajo.

El proyecto se puede delinear básicamente de la siguiente manera:



Figura 4. Ciclo Básico de un Proyecto.

- **La idea de proyecto o Perfil:** consiste en establecer la necesidad u oportunidad a partir de la cual es posible iniciar el diseño del proyecto. La idea de proyecto puede iniciarse debido a alguna de las siguientes razones [2]:

- Porque existen necesidades insatisfechas actuales o se prevé que existirán en el futuro si no se toma medidas al respecto.
 - Porque existen potencialidades o recursos sub-aprovechados que pueden optimizarse y mejorar las condiciones actuales.
 - Porque es necesario complementar o reforzar otras actividades o proyectos que se producen en el mismo lugar y con los mismos involucrados.
- **Diseño o Planificación:** Es la etapa de un proyecto en la que se valoran las opciones, tácticas y estrategias a seguir teniendo como indicador principal el objetivo a lograr. En esta etapa se produce la aprobación del proyecto, que se suele hacer luego de la revisión del perfil de proyecto y/o de los estudios de pre-factibilidad, o incluso de factibilidad. Una vez dada la aprobación, se realiza la planificación operativa, un proceso relevante que consiste en prever los diferentes recursos y los plazos de tiempo necesarios para alcanzar los fines del proyecto, asimismo establece la asignación o requerimiento de personal respectivo.

Especialmente el presente proyecto contempla las siguientes actividades de esta etapa:

- Identificación y Documentación de Roles del Proyecto, y creación del Plan de Gestión.
- Fortalecimiento de Competencias y Relaciones entre los miembros del equipo del proyecto.
- Dirección del Grupo del Proyecto.

Ejecución: En esta etapa se llevan a cabo las actividades necesarias y programadas para llevar a cabo el plan de gestión, con el propósito de cumplir satisfactoriamente con los requisitos planteados en el proyecto. Los procesos realizados son:

- Dirigir y gestionar la ejecución del proyecto
 - Realizar aseguramiento de la calidad
 - Gestión de adquisiciones
 - Adquirir y desarrollar el Equipo del Proyecto
 - Distribución de la Información
- **Evaluación.** La etapa final de un proyecto en la que éste es revisado, y se llevan a cabo las valoraciones pertinentes sobre lo planeado y lo ejecutado, así como sus resultados, en consideración al logro de los objetivos planteados.

2.7 ESQUEMA DE DESARROLLO

2.7.1 Diagramación UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso

Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

En UML hay trece (13) tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente. Los Diagramas de Estructura enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos
- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue
- Diagrama de paquetes

Los *Diagramas de Comportamiento* enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

Los *Diagramas de Interacción* son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x)
- Diagrama de tiempos (UML 2.0)
- Diagrama global de interacciones o Diagrama de vista de interacción (UML 2.0)

2.7.1.1 Diagramas de caso de uso

El estándar de Lenguaje de Modelado Unificado de OMG define una notación gráfica para realizar diagramas de casos de uso, pero no el formato para describir casos de uso. Mucha gente sufre la equivocación pensando que un caso de uso es una notación gráfica (o es su descripción). Mientras la notación gráfica y las descripciones son importantes, ellos forman parte de la documentación de un caso de uso --un propósito para el que el actor puede usar el sistema. El valor verdadero de un caso de uso reposa en dos áreas:

- La descripción escrita del comportamiento del sistema al afrontar una tarea de negocio o un requisito de negocio. Esta descripción se enfoca en el valor suministrado por el sistema a entidades externas tales como usuarios humanos u otros sistemas.
- La posición o contexto del caso de uso entre otros casos de uso. Dado que es un mecanismo de organización, un conjunto de casos de uso coherente, consistente promueve una imagen fácil del comportamiento del sistema, un entendimiento común entre el cliente/propietario/usuario y el equipo de desarrollo.

Es práctica común crear especificaciones suplementarias para capturar detalles de requisitos que caen fuera del ámbito de las descripciones de los casos de uso. Ejemplos de esos temas incluyen rendimiento, temas de escalabilidad/gestión, o cumplimiento de estándares.

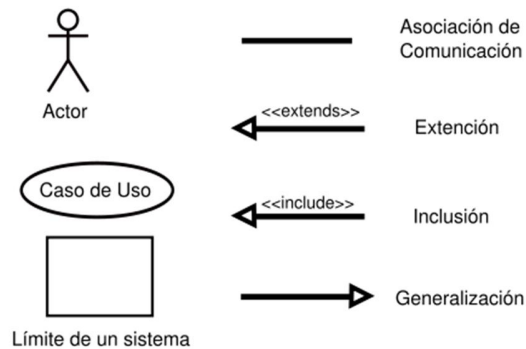


Figura 5. Componentes de un diagrama de Casos de Uso.

2.7.1.2 Relaciones de Casos de Uso

Las tres relaciones principales entre los casos de uso son soportadas por el estándar UML, el cual describe notación gráfica para esas relaciones.

- *Inclusión (include o use)*

Es una forma de interacción, un caso de uso dado puede "incluir" otro. El primer caso de uso a menudo depende del resultado del caso de uso incluido. Esto es útil para extraer comportamientos verdaderamente comunes desde múltiples casos de uso a una descripción individual, desde el caso de uso que lo incluye hasta el caso de uso incluido, con la etiqueta "<<include>>". Este uso se asemeja a una expansión de una macro, donde el comportamiento del caso incluido es colocado dentro del comportamiento del caso de uso base. No hay parámetros o valores de retorno.

- *Extensión (Extend)*

Es otra forma de interacción, un caso de uso dado, (la extensión) puede *extender* a otro. Esta relación indica que el comportamiento del caso de uso extensión puede ser insertado en el caso de uso extendido bajo ciertas condiciones. La notación, es una flecha de punta abierta con línea discontinua, desde el caso de uso extensión al caso de uso extendido, con la etiqueta «extend». Esto puede ser útil para lidiar con casos especiales, o para acomodar nuevos requisitos durante el mantenimiento del sistema y su extensión. "La extensión, es el conjunto de objetos a los que se aplica un concepto. Los objetos de la extensión son los ejemplos o instancias de los conceptos."

- *Generalización*

En la tercera forma de relaciones entre casos de uso, existe una relación generalización/especialización. Un caso de uso dado puede estar en una forma especializada de un caso de uso existente. La notación es una línea sólida terminada en un triángulo dibujado desde el caso de uso especializado al caso de uso general. Esto se asemeja al concepto orientado a objetos de sub-clases, en la práctica puede ser útil factorizar comportamientos comunes, restricciones al caso de uso general, describirlos una vez, y enfrentarse a los detalles excepcionales en los casos de uso especializados. "Entonces la Generalización es la actividad de identificar elementos en común entre conceptos y definir las relaciones de una superclase (concepto general) y subclase (concepto especializado). Es una manera de construir clasificaciones taxonómicas entre conceptos que entonces se representan en jerarquías de clases. Las subclases conceptuales son conformes con las superclases conceptuales en cuanto a la intensión y extensión."

2.7.1.3 Diagrama de Actividades

En el Lenguaje de Modelado Unificado, un *diagrama de actividades* representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema. Un Diagrama de Actividades muestra el flujo de control general.

En SysML el diagrama de Actividades ha sido extendido para indicar flujos entre pasos que mueven elementos físicos (e.g., gasolina) o energía (e.g., presión). Los cambios adicionales permiten al diagrama soportar mejor flujos de comportamiento y datos continuos.

Descripción

En UML 1.x, un diagrama de Actividades es una variación del Diagrama de estados UML donde los "estados" representan operaciones, y las transiciones representan las actividades que ocurren cuando la operación es completa.

El diagrama de Actividades UML 2.0, mientras que es similar en aspecto al diagrama de Actividades UML 1.x, ahora tiene semánticas basadas en redes de Petri. En UML 2.0, el diagrama general de Interacción está basado en el diagrama de Actividades.

Diagrama de actividad. Es una forma especial de diagrama de estado usado para modelar una secuencia de acciones y condiciones tomadas dentro de un proceso.

La especificación del Lenguaje de Modelado Unificado OMG define un diagrama de actividad como: "... una variación de una máquina estados, lo cual los estados representan el rendimiento de las acciones o subactividades y las transiciones se provocan por la realización de las acciones o subactividades."

El propósito del diagrama de actividad es modelar un proceso de flujo de trabajo (workflow) y/o modelar operaciones. Una Operación es un servicio proporcionado

por un objeto, que está disponible a través de una interfaz. Una Interfaz es un grupo de operaciones relacionadas con la semántica.

2.7.1.4 Diagrama de secuencia

El *diagrama de secuencia* es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un *diagrama de secuencia* muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista *business* del escenario, el *diagrama de secuencia* contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos. Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se tiene modelada la descripción de cada caso de uso como una secuencia de varios pasos, entonces puedes "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un *diagrama de secuencia* muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

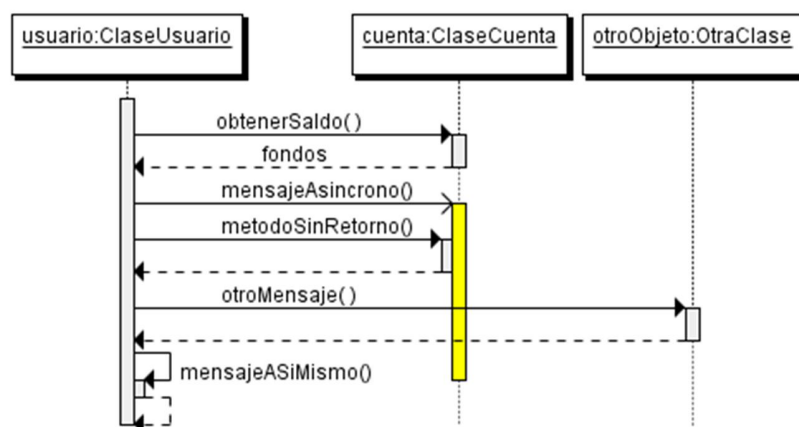


Figura 6. Diagrama de Secuencia

Existen dos tipos de mensajes: síncronos y asíncronos. Los mensajes síncronos se corresponden con llamadas a métodos del objeto que recibe el mensaje. El objeto que envía el mensaje queda bloqueado hasta que termina la llamada. Este tipo de mensajes se representan con flechas con la cabeza llena. Los mensajes asíncronos terminan inmediatamente, y crean un nuevo hilo de ejecución dentro de la secuencia. Se representan con flechas con la cabeza abierta.

También se representa la respuesta a un mensaje con una flecha discontinua. Los mensajes se dibujan cronológicamente desde la parte superior del diagrama a la parte inferior; la distribución horizontal de los objetos es arbitraria. Durante el análisis inicial, el modelador típicamente coloca el nombre 'business' de un mensaje en la línea del mensaje. Más tarde, durante el diseño, el nombre 'business' es reemplazado con el nombre del método que está siendo llamado por un objeto en el otro. El método llamado o invocado, pertenece a la definición de la clase instanciada por el objeto en la recepción final del mensaje.

2.7.2 JAVA Enterprise Edition 5

2.7.2.1 Generalidades acerca de JAVA EE 5

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), componentes de transacción (EJB 3.0), persistencia de bases de datos (Java Persistence) 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.

La plataforma Java EE 5 introduce un modelo de programación simplificado. Con la tecnología de Java EE 5, los descriptores de despliegue XML son opcionales.

Un desarrollador puede simplemente introducir la información directamente de una anotación en un fichero fuente de Java, así Java EE y el servidor pueden configurar el componente en el despliegue y la ejecución. Estas anotaciones se utilizan generalmente para incrustar en un programa datos que, de otro modo se suministren en un descriptor de despliegue.

Con un enfoque primario, la plataforma Java EE 5 ofrece a los desarrolladores fácil acceso a una red segura, portable y a una plataforma escalable para sus aplicaciones empresariales. También, ayuda a empresas y proveedores de servicio a reducir su coste total de propiedad, acelerar el tiempo al mercado, y aumentar la productividad.

En la plataforma Java EE, la inyección de dependencia puede ser aplicada a todos los recursos que un componente necesite, ocultando de manera efectiva la creación y búsqueda de recursos del código de aplicación. La Inyección de dependencia puede ser utilizada en los contenedores de EJB, los contenedores Web, y la aplicación cliente. También permite que el contenedor Java EE inserte automáticamente las referencias a otros componentes necesarios utilizando los recursos o las anotaciones.

El Java Persistence API proporciona un mapeo de tipo objeto/relacional para la gestión de los datos relacionales, componentes Web, y aplicación de los clientes que son manejados en la empresa. También se puede utilizar en aplicaciones Java SE, fuera del entorno Java EE.

Enterprise JavaBeans (EJB) 3.0, incluye dos nuevos conjuntos de API: el primero es una simplificación de la API EJB que promueve la facilidad del desarrollo, y el segundo es una nueva API, para la gestión de la persistencia y el mapeo objeto/relacional.

Las anotaciones Java ahora pueden ser utilizadas para especificar el mapeo de objetos Java a las empresas en una base de datos relacional. La persistencia en

la aplicación del servidor de aplicaciones es compatible con la generación de esquemas de bases de datos a objetos Java y el mapeo de los actuales esquemas de base de datos a objetos Java. Además de utilizar la persistencia en el contenedor EJB, puede usar el API de persistencia con Java SE. Esto permite la máxima flexibilidad en la aplicación de necesidades de persistencia, con una solución común para ambos entornos de cliente y servidor.

Java EE 5 SDK incrementa la productividad con la aplicación de la nueva API Java EE 5 y las anotaciones que reducen significativamente la cantidad de código a las necesidades de los desarrolladores para escribir.

La plataforma Java EE utiliza una aplicación distribuida en multicapa para la aplicación de los modelos empresariales. La lógica de aplicación se divide en componentes según la función, y los diversos componentes de aplicación que conforman una aplicación Java EE se han instalado en diferentes máquinas en función del nivel en el ambiente multicapa de Java EE al cual pertenece el componente de la aplicación.

En la Figura 2. Aplicaciones Distribuidas en Multicapa, veremos cómo se distribuyen las aplicaciones que están compuestas por varias capas. A continuación se mencionan de manera corta, las capas de la figura.

- Los componentes de la capa del cliente se ejecutan en la máquina cliente.
- Los componentes de la capa Web se ejecutan en el servidor de Java EE.
- Los componentes de la capa de Negocios se ejecutan en el servidor de Java EE.
- La capa del Sistema de Información Empresarial (SIE) de software se ejecuta en el servidor EIS.

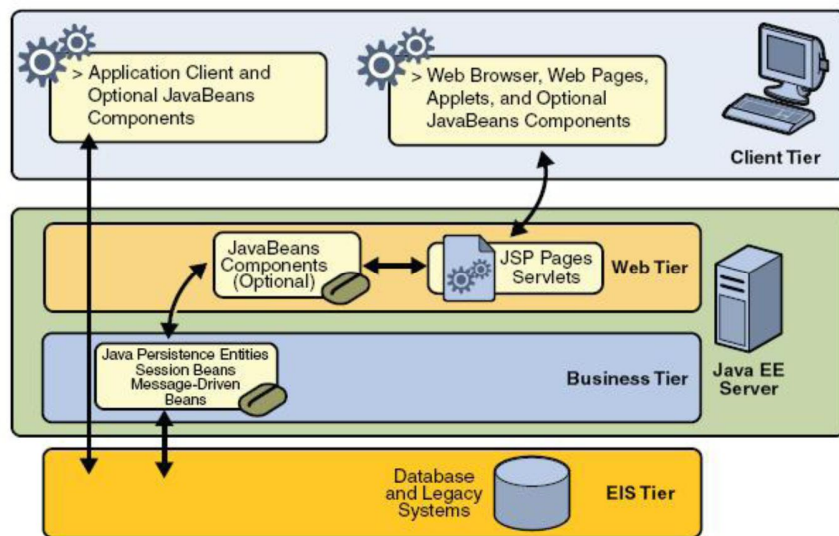


Figura 7. Aplicaciones Distribuidas en Multicapa

A pesar de que una aplicación Java EE puede consistir en las tres o cuatro capas como se muestra en la figura 2, las aplicaciones multicapa de Java EE son generalmente consideradas tres capas, porque las aplicaciones se distribuyen en tres lugares: los equipos-cliente, el servidor Java EE, y la base de datos. Las aplicaciones de tres capas son una ampliación de la aplicación de dos capas del modelo de cliente-servidor colocando un servidor de una aplicación multicapa entre el cliente y la aplicación de almacenamiento al final.

La capa de Presentación (Cliente), es la responsable de manejar la interfaz gráfica del usuario y las entradas del usuario.

La capa de la Lógica del Negocio, es el corazón de la aplicación contiene el flujo de trabajo y procesamiento de la lógica. También recupera los datos que se han guardado de la Base de Datos utilizando la capa de persistencia.

2.7.2.2 Características principales de JAVA EE 5.

A continuación se mencionarán algunas de las características más importantes de la tecnología JAVA EE 5.

- Es una aplicación robusta, comercial, gratis para el desarrollo, despliegue, y la redistribución.
- Contiene EJB 3.0 que apoyan a los POJOs, (Plain Old Java Objects), lo que significa que hay menor código a escribir, lo cual conlleva a un mantenimiento más ágil. EJB es una plataforma para la construcción portátil, reutilizable y escalable de aplicaciones de negocio utilizando el lenguaje de programación JAVA.
- La nueva Java Persistence (API) hace que el mapeo objeto-relacional sea más claro y más fácil.
- Tiene un mayor rendimiento, el tiempo de respuesta es más rápido, y la mejora de las características de administración para simplificar el despliegue es más significativa.
- Su tiempo de inicio es un treinta por ciento más rápido, con un treinta por ciento menos de memoria, respecto a versiones anteriores.

2.7.2.3 API de Java EE 5

La figura ilustra las API de Java EE5 para cada tipo de contenedor Java EE. La siguiente sección nos muestra un resumen breve de las tecnologías requeridas por la plataforma Java EE y la API utilizada en las aplicaciones Java EE.

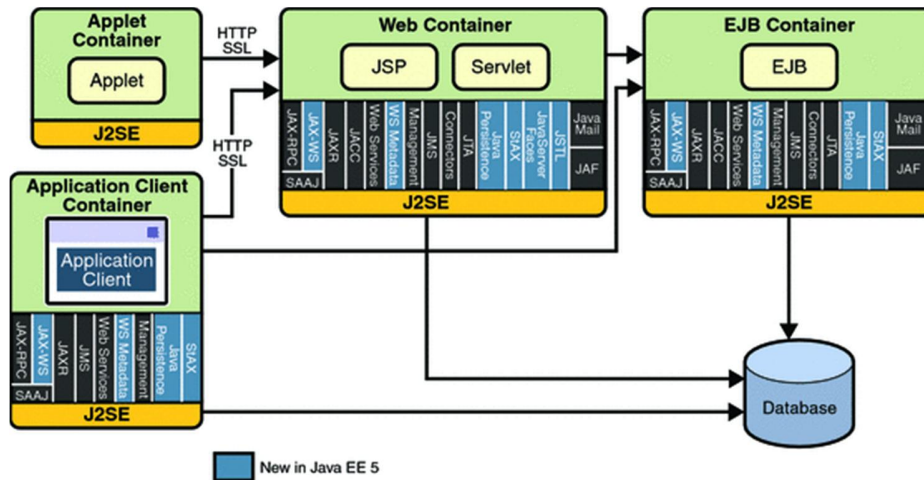


Figura 8. API de la plataforma Java EE.

- **Tecnología de JavaBeans empresarial.**

Un componente JavaBean empresarial (EJB), es un cuerpo de código que tiene campos y métodos para implementar módulos de lógica de negocio. Se puede pensar en un bean empresarial como un componente escrito en lenguaje Java encapsulado, en el lado del servidor, que puede ser utilizado solo con otros beans empresariales para ejecutar lógica de negocio en el servidor Java EE.

Hay dos tipos de beans empresariales: los beans de sesión y los beans manejadores de mensajes.

- ⊙ Un bean de sesión representa una conversación transitoria con un cliente. Cuando el cliente termina de ejecutarse, el bean de sesión y sus datos desaparecen.
- ⊙ Un bean manejador de mensajes combina las características de un bean de sesión y uno de escucha de mensajes permitiendo a un componente de negocio recibir mensajes de forma asincrónica. Comúnmente, estos son mensajes JMS (Java MessageService).

En Java EE5, los beans de entidad han sido reemplazados por entidades de la API de persistencia de Java (JPA). Una entidad representa datos persistentes almacenados en una fila de una tabla de la base de datos. Si el cliente termina o el servidor se apaga, el manejador de persistencia asegura que los datos de la entidad sean guardados.

- ***Tecnología Java servlet***

La tecnología servlet de Java permite definir una clase servlet específica para HTTP. Una clase servlet extiende las habilidades de los servidores que hospedan aplicaciones que son accedidas por de la forma del modelo de programación solicitud respuesta. A pesar de que los servlets pueden responder a cualquier tipo de solicitud, comúnmente son utilizados para extender las aplicaciones hospedadas por servidores web.

- ***Tecnología de JavaServerPages.***

La tecnología de JavaServerPages (JSP) nos permite colocar partes del código servlet directamente en un documento de texto. Una página JSP es un documento de texto que contiene dos tipos de texto: datos estáticos y elementos JSP, que determinan como la página construye el contenido dinámico.

- ***Librería estándar de etiquetas de páginas JavaServer***

La librería estándar de etiquetas de páginas JavaServer (JSRL) encapsula la funcionalidad principal común a muchas aplicaciones JSP. En lugar de etiquetas mezcladas de varios vendedores en sus aplicaciones JSP, se emplea un solo, grupo estándar de etiquetas. Esta estandarización permite, desplegar sus aplicaciones en cualquier contenedor que soporte JSTL y lo hace más deseable que la implementación de etiquetas optimizada.

- ***JavaServer 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 solo 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, aun si tales componentes provienen de terceros.

- ***API Java de servicio de mensajes***

La API del servicio de mensajes de Java (JMS) es un estándar de mensajería que permite a componentes de aplicación Java EE crear, enviar, recibir y leer mensajes. Para habilitar comunicación distribuida con bajo acoplamiento, confiable y asincrónica.

- ***API Java de transacciones***

La API Java de transacciones (JTA) proporciona una interface estándar para distinguir transacciones. La arquitectura Java EE proporciona un commit automático para manejar commit y rollbacks. Un auto commit significa que otras aplicaciones que están viendo estos datos verán los datos actualizados luego de que cada base de datos realice la operación de lectura o escritura.

Sin embargo, si su aplicación realiza dos operaciones de acceso a base de datos que dependen una de otra se deseará utilizar la API JTA para distinguir donde la

transacción completa incluyendo ambas operaciones, comienza, se regresa o se completa.

- ***API JavaMail***

Las aplicaciones Java EE utilizan la API JavaMail para enviar correo. La API de JavaMail tiene dos partes: una interface a nivel de aplicación utilizada por los componentes de la aplicación para enviar correo y una interface de proveedor de servicio. La plataforma Java EE incluye JavaMail con un proveedor de servicio que permite a los componentes de aplicación enviar correo por la Internet.

- ***Marco de trabajo de activación de JavaBeans***

El marco de trabajo de activación JavaBeans (JAF) está incluido porque JavaMail lo utiliza. JAF proporciona servicios estándares para determinar el tipo de una porción arbitraria de datos, acceso encapsulado al descubrimiento de operaciones disponibles y creación de los componentes JavaBeans apropiados para realizar estas operaciones.

- ***API Java para procesamiento XML***

La API de Java para procesamiento XML (JASP), parte de la plataforma Java SE, soporta el procesamiento de documentos SML utilizando el Modelo de objetos del documento (DOM), API simple para SML (SAX) y Transformaciones del Lenguaje Extensible de Hojas de estilo (XSLT). JAXP habilita a las aplicaciones a analizar la sintaxis y transformar documentos SML independientemente de una implantación de procesado XML particular.

- ***API de SOAP con adjuntos para Java***

La API de SOAP con adjuntos para Java (SAAJ) es una API de bajo nivel de la cual dependen JAX-WS y JAXR. SAAJ habilita la producción y el consumo de

mensajes que cumplen con la especificación de SOAP1.1 y notas SOAP con adjuntos. Muchos desarrolladores no utilizan la API SAAJ, en lugar de esto utilizan la API de alto nivel JAX-WS.

- ***Arquitectura del conector J2EE***

La arquitectura del conector J2EE es utilizada por los vendedores de herramientas e integradores de sistemas para crear adaptadores que soporten el acceso a sistemas de información empresarial que puedan ser conectados en cualquier producto Java EE. Un adaptador de recurso es un componente de software que permite a los componentes de aplicación Java EE acceder e interactuar con el manejador de recursos de capas bajas del EIS. Dado que un adaptador de recurso es específico de su manejador de recursos, típicamente hay un adaptador de recursos diferente para cada tipo de base de datos o sistema de información empresarial.

- ***API JAVA de conectividad a bases de datos***

La API Java de conectividad a base de datos (JDBC) permite invocar comandos SQL desde métodos del lenguaje de programación Java. Se utiliza la API JDBC en un bean empresarial cuando se tiene un bean de sesión accediendo a la base de datos. Se puede utilizar también la API JDBC desde un servlet o una página JSP para acceder a la base de datos directamente sin pasar a través de un bean empresarial.

La API JDBC tiene dos partes: una interface a nivel de aplicación usado por los componentes de aplicación para acceder a la base de datos y una interface de proveedor de servicio para anexas un controlador JDBC a la plataforma Java EE.

- ***API Java de persistencia***

La API Java de persistencia es una solución Java basada en estándares para persistencia. La persistencia utiliza una estrategia de "mapeo" objeto relacional para unir la brecha entre el modelo orientado a objetos y la base de datos relacional. La persistencia de Java consiste en tres áreas:

- ⊙ La API Java de persistencia
- ⊙ El lenguaje de consultas
- ⊙ Los datos de alto nivel con el mapeo objeto relacional

2.7.3 Java Server Faces.

Java Server Faces (JSF), es un estándar para frameworks de interfaz de componentes de usuarios del lado del servidor para las aplicaciones Web basadas en la tecnología JAVA.

Java Server Faces proporciona:

- Una clara separación entre vista y modelo.
- Un desarrollo basado en componente, no en peticiones.
- Las acciones del usuario se ligan muy fácilmente al código en el servidor.
- Creación de familias de componentes visuales para acelerar el desarrollo.
- Ofrece múltiples posibilidades de elección entre distintos desarrollos.

Dentro de los componentes que constituyen los Java Server Faces, encontramos un API y una implementación de referencia con la cual se pueden representar componentes de la Interfaz del Usuario y manejar su estado; También permite el manejo de eventos, validación del lado del servidor y conversión de datos, la definición de la navegación entre páginas. Así como una librería de etiquetas Java Server Pages (JSP) personalizadas para dibujar componentes de la Interfaz del Usuario dentro de una página Java Server Pages.

Una de las grandes ventajas de la tecnología Java Server Faces es que ofrece una clara separación entre el comportamiento y la presentación. Las aplicaciones Web construidas con tecnología Java Server Pages conseguían parcialmente esta separación. No obstante, una aplicación Java Server Pages no puede mapear peticiones HTTP al manejo de eventos específicos de los componentes o manejar elementos UI como objetos con estado en el servidor. Así mismo, Java Server Faces se hace fácil de usar al aislar al desarrollador del API de Servlet.

Aunque la tecnología Java Server Faces incluye una librería de etiquetas Java Server Pages personalizadas para representar componentes en una página Java Server Pages, los APIs de la tecnología Java Server Faces se han creado directamente sobre el API JavaServlet. Esto permite hacer algunas cosas: usar otra tecnología de presentación junto a Java Server Pages, crear componentes propios personalizados directamente desde las clases de componentes, y generar salida para diferentes dispositivos cliente. Así, se podrán encapsular otras tecnologías como Ajax en componentes Java Server Faces, haciendo su uso más fácil y productivo, al aislar al programador de ellas.

La tecnología Java Server Faces proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos. Sin embargo, no todo es perfecto el uso de Java Server Faces también tiene sus desventajas, entre ellas tenemos que, su naturaleza como estándar hace que la evolución de Java Server Faces no sea tan rápida como pueda ser la de otros entornos como WebWork, Wicket, Spring , etc.

2.7.4 Hibernate y ORM

A pesar de la innegable popularidad de los lenguajes orientados a objetos, las bases de datos relacionales han dominado el mercado por mucho tiempo. Esto ha dado lugar a un desfase entre las tecnologías y herramientas que procesan la información de un sistema y las que la almacenan. Se han intentado crear

iniciativas de bases de datos orientadas a objetos pero no han tenido resultados afortunados. Entonces se ha optado por una alternativa diferente: El software de mapeo objeto-relacional (ORM por Object-relational mapping).

Un software de ORM permite crear una correspondencia entre la información en una base de datos y objetos del lenguaje de programación en que se desee desarrollar una aplicación. Así, se crea la ilusión de una base de datos en memoria que el desarrollador puede manipular mediante técnicas y estructuras de datos típicas de la POO: Clases, métodos, casting, etc.

Los ORM también minimizan el impacto de diferencias radicales entre los objetos y las entidades de una base de datos relacional. Por ejemplo, un objeto es una estructura de datos que no sólo almacena valores, sino que se comporta diferente de acuerdo a los valores que tenga o reciba en un método. El ORM también se encarga de hacer las validaciones y conversiones necesarias entre tipos de datos de la base de datos y la máquina que ejecuta la aplicación, considerando que los tipos de datos primitivos de datos son de diferentes tamaños en diversos sistemas operativos y que las máquinas que albergan la aplicación y la base de datos pueden ser diferentes. El software ORM se vale de las ventajas de la encapsulación y los métodos en POO para ofrecer tal robustez.

Para Java, existe la librería Hibernate, software libre distribuido bajo la Licencia pública general reducida de GNU. Hibernate permite mapear las entidades de una base de datos a clases Java. También maneja todo tipo de cardinalidad de relaciones entre entidades, incluso relaciones reflexivas. Gracias a Hibernate, el código de lenguaje de consultas a introducir se simplifica en cantidad y complejidad de manera considerable, y facilita interactuar con la base de datos como si fuera esta tan sólo un objeto más en memoria. Además, como todo ORM, Hibernate se encarga del flujo de datos de la aplicación a la base de datos, efectuando todas las operaciones necesarias para que las claves, los datos, sus

tipos y tamaños correspondan a las reglas de la base de datos establecidas en el mapeo, garantizando el éxito de las transacciones.

Hibernate permite el mapeo de entidades mediante archivos descriptores XML o mediante JPA (Java Persistence API). La segunda es la usada en el presente trabajo de grado.

2.7.5 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), y facilita la implementación de la persistencia de una nueva manera por medio del api JPA.

2.7.6 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.

Una característica fascinante de JPA, es que permite que se hagan cambios al diseño de la base de datos sin tener que reescribir enteramente las aplicaciones. Para esto JPA introduce:

2.7.7 JPQL

Java Persistence Query Language el cual es el lenguaje de consultas que se usará dentro de la aplicación. Con él, el desarrollador jamás se refiere a las entidades directamente al momento de hacer las consultas, sino a los objetos mismos que fueron mapeados. Si el diseño de la BD cambiara, sólo habría que modificar las anotaciones de las entidades. El resto del código quedaría intacto y seguiría funcionando tal y como antes. Esta facilidad permite optimizar las bases de datos cuando se considere necesario, migrar a bases diferentes, o sencillamente corregir diseños defectuosos con un esfuerzo mínimo.

2.7.8 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).

2.7.9 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.

3. METODOLOGÍA DE DESARROLLO

Con el fin de poder desarrollar exitosamente un proyecto software es preciso establecer desde un principio un enfoque disciplinado y sistemático, mediante la implementación de la metodología para el desarrollo del proyecto, la cual se construye a partir del esquema que se define por los ciclos de vida establecidos para el proyecto. Esta metodología se puede definir como, un conjunto de procedimientos, técnicas, herramientas que cuenta con un soporte documental que colabora a los desarrolladores a lograr el desarrollo del software.

También se deben especificar de una manera concisa y clara las fases en las cuales se divide el proyecto, así como, las tareas que se deben realizar en cada fase. Igualmente se especifican las restricciones que se deben tener en cuenta en todo momento, y en términos generales, brindar esencial apoyo en los procesos de gestión y control.

Es por esta razón, que todo el proceso de desarrollo del sistema, se basó en una metodología elaborada a partir de los marcos definidos por el modelo de construcción por prototipos, debido a las condiciones iniciales sobre las cuales se basaba el proyecto, sin apartarnos claro, de los estándares y metodología de la División de Servicios de Información.

3.1 CICLO DE VIDA DEL PROYECTO

De acuerdo a la metodología implementada para el desarrollo de este proyecto, encontramos que esta se divide en las siguientes cinco fases:

Fase 1. Análisis de Requerimientos.

Fase 2. Diseño.

Fase 3. Implementación de la Aplicación

Fase 4. Pruebas de Software.

Fase 5. Ajustes

3.1.1 Análisis de Requerimientos

En esta fase se realiza el estudio sobre la situación actual del Banco de Programas y Proyectos de la Universidad Industrial de Santander, los recursos existentes, los intereses y necesidades, el sistema en el que se va a trabajar, las personas que estarán vinculadas al proyecto (usuarios y desarrolladores) y cualquier otro dato que resulte de interés, con el fin de identificar los objetivos generales del proyecto.

El análisis de estos datos nos informará de las necesidades e intereses que existen, así como del nivel de importancia de los mismos. También nos da una idea de la situación deseada y comenzarán a surgir las herramientas necesarias para ir logrando el desarrollo del proyecto.

Se especificará, junto con los funcionarios de la Oficina de Planeación, la función y comportamiento que deberán tener los programas a desarrollarse en el transcurso del proyecto, se indicará la interfaz con otros elementos del sistema y se establecerán los estándares de diseño que debe cumplir el sistema.

3.1.2 Diseño

En esta fase se define más concretamente lo que se va a hacer con el proyecto. Primero que todo debemos identificar varias cosas que se van a servir como apoyo durante el desarrollo del proyecto.

La identificación de los actores es muy importante ya que ellos son las personas que van a estar interactuando con el software y son las personas que nos van a brindar los datos importantes para poder lograr el avance del proyecto.

El usuario puede ser diferentes actores dependiendo del caso de uso en el que se esté desempeñando. Identificar los casos de uso, es de gran importancia ya que en base a ellos es que se logra tener un buen desarrollo, es decir, con los casos de uso bien hechos el diseño del software es mejor y con pocas posibilidades de tener la necesidad de ir cambiándolos a medida que se va avanzando con el proyecto.

El caso de uso es algo que un actor (usuario) quiere hacer con el sistema, es decir, cualquier acción que el actor realice con el software se puede representar como un caso de uso. Lo que conoce como caso de uso, en otras palabras es como tener una lista de las posibles acciones que el actor quiere realizar con el sistema, deben ser cosas grandes, no detalles ya que esos son los que hacen parte del caso de uso.

Al detallar los casos de uso vamos poniendo varios puntos, en donde ponemos la narración de lo que hace el actor, es decir, se explica de manera clara y desde el punto de vista del usuario la iteración (comunicación) que hay entre el actor y el sistema.

En el caso de uso se narra el Camino ideal, es decir, sin fallos ni situaciones anormales. Sin embargo, este tipo de situaciones anormales se narran en una pequeña lista aparte en donde se contemplan todos los escenarios que tengan fallos. En cuanto al diseño lo que podemos decir es que con la ayuda de los diagramas que se deben realizar, debemos conjuntamente ir pensando en las interfaces y en cómo se verán estas. Se deben diseñar de manera tal que sean amigables y fáciles de interpretar para el usuario, tratar de no usar lenguaje técnico a menos que el usuario sea una persona capacitada para entenderlo. Esto es teniendo en cuenta que el diseño que obtenemos de los casos es funcional y no uno orientado a objetos.

Debemos tratar de pensar en objetos y almacenarlos juntos en la misma librería cuando estén muy relacionados entre sí y no en funciones. Por ello es buena idea tratar de agrupar las clases del diagrama de clases del negocio en paquetes y tratar de desarrollar la arquitectura a partir de ellas.

Así mismo, se debe elaborar un documento con las especificaciones del diseño que se tiene en cuenta para el proyecto. Este documento tiene que ser aprobado por ambas partes y una vez aprobado sobre él es que se evalúa el cumplimiento por parte de los desarrolladores cuando sea entregado el software.

3.1.3 Implementación de la Aplicación

En esta etapa se procede a generar el software que se ha diseñado 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 la Arquitectura de Desarrollo de Aplicaciones de JAVA EE5 y la plataforma Informix como motor de base de datos.

Algunas de las tareas que se deben realizar en esta fase se mencionarán a continuación:

Estudio detallado de las especificaciones del diseño, esta actividad se debe hacer conjuntamente, es decir, los desarrolladores con los usuarios ya que la comunicación entre ellos es primordial para la definición de las especificaciones que se van a plantear para luego definir el diseño y desarrollo del software.

Escritura y depuración de programas que forman parte del sistema, esta actividad consiste en la realización y ejecución de los programas que se van a utilizar en el software.

3.1.4 Pruebas de Software

Todo sistema debe ser probado mediante su ejecución controlada antes de ser entregado al cliente, las cuales constituyen en método para poder verificar y validar que el software cumple con todas las especificaciones definidas en las fases anteriores, además de dar garantía de la eficiencia y confiabilidad del mismo.

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 los resultados esperados.

Las pruebas serán de carácter permanente a lo largo del desarrollo del proyecto por parte del equipo de trabajo, y habrá un periodo de tiempo para que los usuarios finales interactúen con la aplicación y detecten posibles ajustes.

3.1.5 Ajustes

Después de las Pruebas, cada uno de los errores detectados o las observaciones hechas por los usuarios, deben ser tenidos en cuenta para ajustar el sistema, de tal manera que se adapte plenamente a las necesidades de los mismos.

También estos se harán permanentemente a lo largo del desarrollo del proyecto a la par con las pruebas, en la medida en que se detecten errores o inconsistencias en el sistema que se ha desarrollado.

3.2 METODOLOGÍA DE DESARROLLO

Teniendo como base las actividades anteriormente descritas, se propone como metodología de desarrollo de éste proyecto el *MODELO DE CONSTRUCCION POR PROTOTIPOS*.

Se eligió esta metodología debido a que es muy frecuente que los usuarios que están solicitando el sistema, en este caso *la Oficina de Planeación y la División de Servicios de Información* 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 por 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 funcionales permitirán la familiarización del usuario con el sistema que se está desarrollando.

3.2.1 Modelo de construcción por prototipos.

Se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de Prototipos y se presenta el modelado (en forma de un diseño rápido).

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final (por ejemplo, la configuración de la interfaz con el usuario). El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para

una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

A continuación se observa la estructura del Modelo:

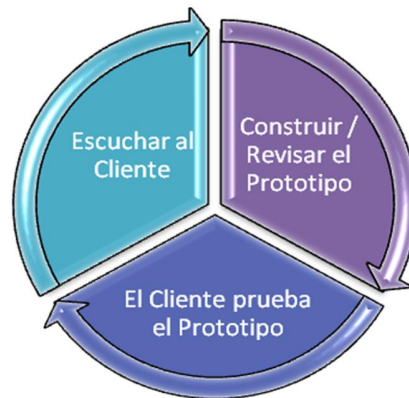


Figura 9. Esquema del modelo basado en prototipos

3.2.1.1 Estructura



Figura 10. Estructura del modelo basado en prototipos

Esta metodología es viable para el desarrollo del proyecto debido a:

- En la creación de los prototipos iniciales se debe trabajar con unas ideas aproximadas de lo que desea *la Oficina de Planeación y la División de Servicios de Información*, para presentar un producto o prototipo inicial, el cual puede evolucionar y refinarse dando como resultado un prototipo más maduro, que cumpla con todos los requerimientos de los usuarios.
- Con el uso del modelo de prototipos se da la facilidad de mejorar de manera temprana los prototipos, teniendo en cuenta las sugerencias del usuario solicitante del proyecto, de tal forma que se cubran a cabalidad sus requerimientos.

En este sistema de desarrollo se debe validar la versión actual del prototipo, para proceder a generar una nueva versión que contemple los nuevos requerimientos, con el fin de evitar retrocesos en el proceso de desarrollo.

3.3 APLICACIÓN DE LA METODOLOGÍA

3.3.1 Estándares DSI ¹².

3.3.1.1 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.

¹² Fuente: Estándares de la División de Servicios de Información de la Universidad Industrial de Santander

- 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, hacerMantenimientoAsignaturasCntxto.

De las 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. 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`.
- El nombre de los EJB utilizados por el patrón `EAO` está conformado por un verbo autorizado. Además debe incluir al final las letras "EAOImp". Por ejemplo: `RegistrarMatriculaEstudianteEAOImp`.
- La interfaz asociada al `EAO` debe llevar el mismo nombre del EJB sin el sufijo "Imp". Ejemplo: `RegistrarMatriculaEstudianteEAO`.
- 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.

De los 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.

De los 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

De las 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.

De las 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.

3.3.2 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.

Los diagramas de casos de uso como el diagrama de clases se mostraran de forma general, pero de allí se extraerán un ejemplo de los casos de usos y de las clases, para mostrarlo como ejemplo del diseño elaborado para el proyecto.

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

3.3.2.1 *Diagramas de Casos de Uso*

Caso de uso: Planificar Recursos Humanos

Facilita el registro, actualización y consulta: Del organigrama del proyecto, de los roles establecidos para la gestión del proyecto con sus competencias y responsabilidades.

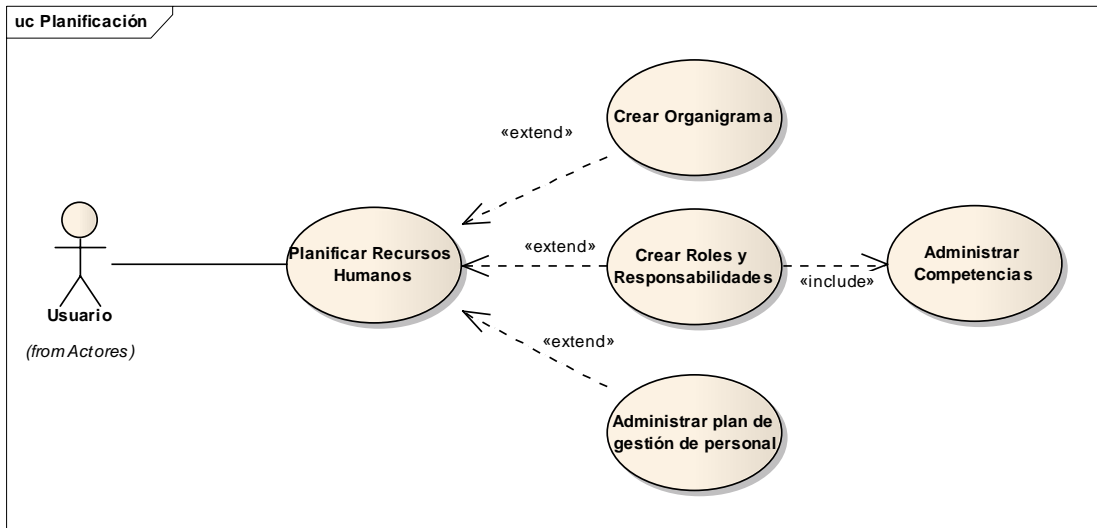


Figura 11. Diagrama de casos de uso – Planificar Recursos Humanos

Se identifican los casos de uso que extienden de Planificar Recursos Humanos:

- Crear Organigrama
- Crear Roles y Responsabilidades
- Administrar Plan de gestión de personal

Aquí se documenta la información del caso de uso: Crear Roles y Responsabilidades, los demás casos de uso se pueden encontrar documentados en el CD que hace parte de éste documento.

Caso de uso: Crear Roles y Responsabilidades

Facilita el registro, actualización y consulta de cada uno de los roles del proyecto.

CASO DE USO		Crear Roles y Responsabilidades	
Objetivo	Definir las características de cada uno de los integrantes del proyecto y el papel que juega en cada una de las etapas del mismo.		
Actores	Usuario		
Pre condiciones	Ninguna		
Flujo Principal	Acción del Actor	Respuesta del Sistema	
	Ingresar al Sistema SGPUIS al módulo Planificación – Mantenimiento - opción Rol Proyecto		
		Muestra la pantalla de creación de un Rol del proyecto. Los campos necesarios para definir el Rol son: <ul style="list-style-type: none"> - Nombre del Rol - Descripción - Función 	
	Define los valores para cada campo y oprime el Botón Guardar		
		Valida que los datos suministrados sean válidos, los graba en la base de datos y muestra en pantalla un mensaje de confirmación exitosa. Muestra el Rol creado en la Tabla de Roles Creados, con las opciones Ver, Modificar y Eliminar.	
	Oprime el botón Cancelar		
	Muestra nuevamente la pantalla de creación de un Rol del Proyecto.		
Fuente:	Autores Proyecto		

Tabla 1. Casos de uso: Crear Roles y Responsabilidades

Caso de uso: Administrar competencias

Facilita el registro, actualización y consulta de las competencias asociadas a cada rol del proyecto.

CASO DE USO		Administrar Competencias
Objetivo	Definir las habilidades y capacidades necesarias de los integrantes del proyecto para completar las actividades de proyecto.	
Actores	Usuario	
Pre condiciones	Debe existir un Rol para poder asociarle las competencias.	
Flujo Principal	Acción del Actor	Respuesta del Sistema
	Desde la pantalla de edición de un Rol del Proyecto, selecciona la opción Asociar Competencias	
		Muestra la pantalla de creación de competencias asociadas al Rol. Los campos necesarios para definir la competencia son: <ul style="list-style-type: none"> - Conocimiento - Experiencia - Habilidad - Otros
	Define los valores para cada campo y oprime el Botón Guardar	
		Valida que los datos suministrados sean validos, los graba en la base de datos y muestra en pantalla un mensaje de confirmación exitosa. Muestra la competencia creada en la Tabla de Competencias del Rol, con las opciones de Modificar y Eliminar.
	Oprime el botón Cancelar	
	Muestra nuevamente la pantalla de creación de un Rol del Proyecto.	
Fuente: Autores Proyecto		

Tabla 2. Casos de uso: Administrar Competencias

Caso de uso: Administrar Equipo del Proyecto

Implica hacer un seguimiento del rendimiento de los miembros del equipo, proporcionar retroalimentación, resolver polémicas y coordinar cambios a fin de mejorar el rendimiento del proyecto.

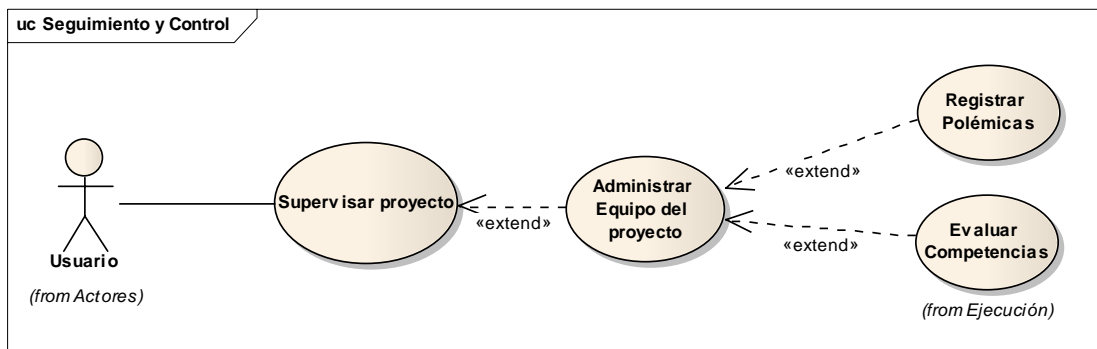


Figura 12. Diagrama de casos de uso – Administrar Equipo del Proyecto

Se identifican los casos de uso que extienden de Administrar Equipo del Proyecto:

- Registrar Polémicas
- Evaluar Competencias

Caso de uso: Registrar Polémicas

Facilita el registro, actualización y consulta: de las personas responsables de resolver polémicas específicas antes de una fecha objetivo.

CASO DE USO		Registrar Polémicas
Objetivo	Registrar el equipo del proyecto responsable de supervisar las polémicas hasta el cierre.	
Actores	Usuario	
Pre condiciones	Debe estar establecido el equipo del proyecto.	
Flujo Principal	Acción del Actor	Respuesta del Sistema
	Ingresa al Sistema SGPUIS al modulo Ejecución – opción Registro de Polémicas	
		Muestra la pantalla de registro de la Polémica Los campos necesarios para definir la polémica son: - Descripción polémica - Fecha Polémica - Involucrado - Responsable
	Define los valores para cada campo y oprime el Botón Guardar	
		Valida que los datos suministrados sean validos, los graba en la base de datos y muestra en pantalla un mensaje de confirmación exitosa. Muestra el registro de la polémica en la Tabla de Polémicas.
	Oprime el botón Cancelar	
	Muestra nuevamente la pantalla de registro de la Polémica.	
Fuente:	Autores Proyecto	

Tabla 3. Casos de uso: Registrar Polémicas

Caso de uso: Administrar Compras y Adquisiciones

Implica determinar qué comprar o adquirir, cuándo y cómo hacerlo, teniendo en cuenta que sirve para efectuar el seguimiento, control y evaluación de la inversión.

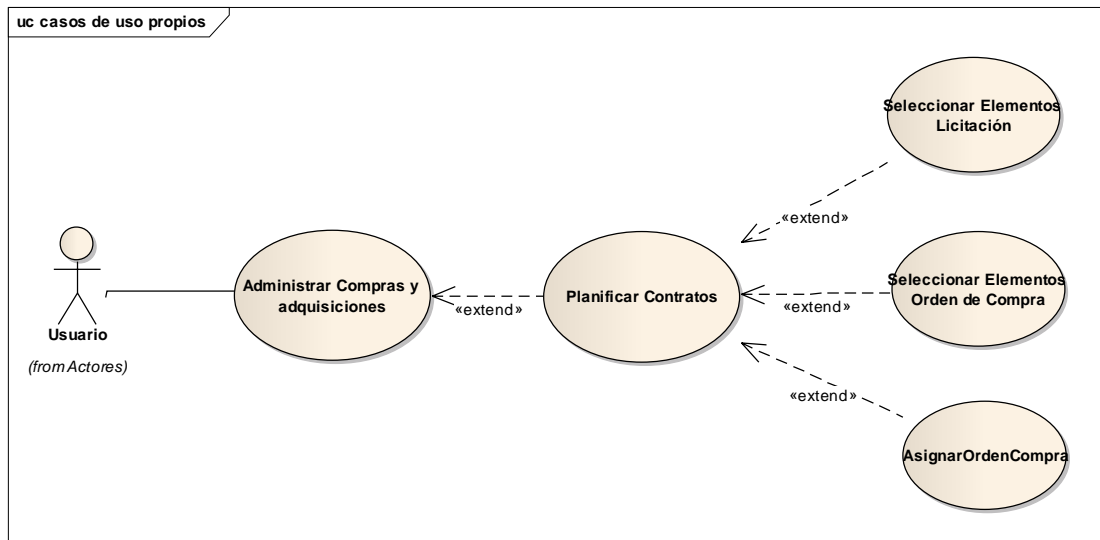


Figura 13. Diagrama de casos de uso – Administrar Compras y Adquisiciones

Se identifican los casos de uso que extienden de Administrar Compras y Adquisiciones:

- Planificar Contratos.

El cual a su vez incluye los casos de uso:

- Seleccionar Elementos Orden de Compra
- Seleccionar Elementos Licitación
- Asignar Orden de Compra

Caso de uso: Planificar Contratos

Facilita el registro, actualización y consulta: de los documentos de la adquisición (invitación a licitación, solicitud de propuesta, aviso de oferta, invitación a la negociación y respuesta inicial del contratista), de los criterios de evaluación para calificar propuestas y actualización del enunciado del trabajo del contrato.

CASO DE USO	Planificar Contratos	
Objetivo	Generar los elementos que van a ser parte de las órdenes de compra y/o contratos derivados de las convocatorias y licitaciones realizadas durante la ejecución del proyecto.	
Actores	Usuario	
Pre condiciones	Ninguna	
Flujo Principal	Acción del Actor	Respuesta del Sistema
	Ingresa al Sistema SGPUIS al modulo Ejecución	
		Muestra el menú las opciones: -Elemento Orden de Compra - Licitación
	Selecciona una de las opciones.	
		Dependiendo de la opción escogida: - Si selecciono la opción <i>Elemento Orden de Compra</i> , procederá a generar solicitud de Orden de Compra a un Proyecto en ejecución. - Si selecciono la opción <i>Elemento Licitación</i> , procederá a generar solicitud de Licitación a un Proyecto en ejecución.
Fuente:	Autores Proyecto	

Tabla 4. Casos de uso: Planificar Contratos

Caso de uso: Seleccionar Elementos Orden de Compra

Facilita el registro y consulta de la solicitud de orden de Compra asociadas a un Proyecto en estado Ejecución. Describiendo la cantidad de cada elemento planificado para la Fase, actividad y Rubro.

CASO DE USO	Seleccionar Elementos Orden de Compra	
Objetivo	Generar las solicitudes de orden de compra del proyecto. Fijando la cantidad necesaria de cada elemento planificado en el Proyecto.	
Actores	Usuario	
Pre condiciones	Ninguna	
Flujo Principal	Acción del Actor	Respuesta del Sistema
	Ingresa al Sistema SGPUIS al modulo Ejecución – Elemento Orden de Compra opción Seleccionar	
		Muestra la pantalla de búsqueda de los proyectos en ejecución existentes.
	Selecciona un Proyecto de la tabla de resultados	
		Muestra la pantalla con la descripción al detalle de los elementos planificados en el Proyecto mostrando la Fase, Actividad y Rubro al que pertenece el Elemento. El Campo necesario para definir la solicitud de Orden de compra es: -Cantidad
	Oprime el botón Generar	
		Muestra una pantalla de confirmación indicando el número de la solicitud y el valor con la que se genero la solicitud de la Orden de Compra.
	Oprime el botón Confirmar	
		Valida que los datos suministrados sean validos, los graba en la base de datos y muestra en pantalla un mensaje de confirmación exitosa.
	Oprime el botón Cancelar	
	Muestra nuevamente la pantalla de búsqueda de Proyectos en ejecución	
Extensiones	El usuario registra el numero de orden de Compra en el sistema, el cual se valida y se compara que los elementos, cantidades y valor de la orden de compra coincida con lo existente en la solicitud.	
Fuente:	Autores Proyecto	

Tabla 5. Casos de uso: Seleccionar Elementos Orden de Compra

3.3.2.2 *Diagramas de Clases*

El diagrama de clases, corresponde al módulo completo del Sistema de Gestión de Proyectos UIS, Gestión de Recursos Humano, Compras y Adquisiciones, dicho diagrama se aprecia en la figura 14.

La documentación del ejemplo seleccionado corresponde a la gestión de Recursos Humanos, pero la documentación completa puede ser consultada en el CD adjunto al documento.

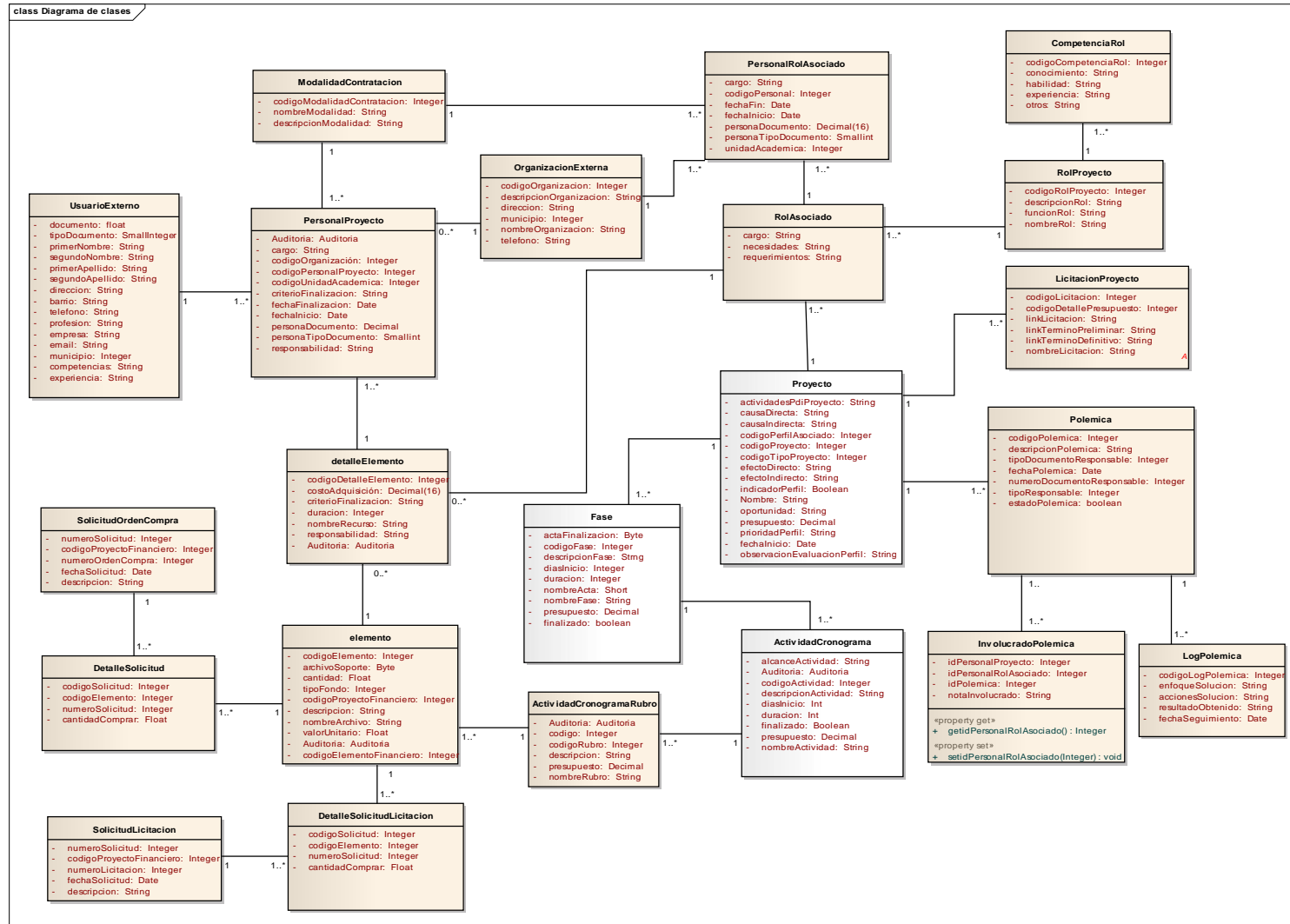


Figura 14. Diagrama de clases – SGPUIS, gestión de recursos humanos, compras y adquisiciones.

- ⊙ **Clase Proyecto:** Entidad principal donde se establecen todas las especificaciones del Proyecto que se lleva a cabo para crear un producto, servicio o resultado único.

El proyecto tiene un comienzo definido y un final definido. El final se alcanza cuando se han logrado los objetivos del proyecto o cuando queda claro que los objetivos del proyecto no serán o no podrán ser alcanzados, o cuando la necesidad del proyecto ya no exista y el proyecto sea cancelado.

Nombre	Tipo de datos	Descripción
actividadesPdiProyecto	String	Descripción de actividades del proyecto asociadas al plan de desarrollo Institucional.
causaDirecta	String	Detalla la motivación directa del proyecto.
causaIndirecta	String	Describe la causa indirecta del proyecto.
codigoPerfilAsociado	Integer	Código del perfil asociado al proyecto trabajado. campo no obligatorio
codigoProyecto	Integer	Identificador del Proyecto.
codigoTipoProyecto	Integer	Identificador del tipo de proyecto a gestionar.
efectoDirecto	String	Almacena el impacto directo que genera el proyecto.
efectoIndirecto	String	Contiene el efecto indirecto generado por el proyecto.
fechaInicio	Date	Fecha de inicio del proyecto para la ejecución (fecha cero).
indicadorPerfil	Boolean	Variable que indica si el proyecto es un perfil, o es un proyecto con todas las de la ley
Nombre	String	Nombre del Proyecto.
Oportunidad	String	Oportunidades alcanzadas con el desarrollo del proyecto.
Presupuesto	Decimal	Indica el presupuesto del proyecto
prioridadPerfil	String	Usado para el perfil de proyecto. Indica la prioridad, para la elección del perfil, este

		puede tomar los siguientes valores: A=alto, M=medio y B=bajo
observacionEvaluacionPerfil	String	Observaciones que el administrador le hace al perfil de proyecto
Fuente: Autores Proyecto		

Tabla 6. Atributos principales de la clase Proyecto

- ⊙ **Case RolProyecto:** Entidad donde se definen los actores que hacen uso del sistema. Están definidos en: Patrocinador, Director del proyecto, Unidad académica promotora, Interesados, entre otros.

Este rol no indica los permisos de una persona sobre unas tablas, se utiliza a nivel de gestión de PMBOOK.

Nombre	Tipo de datos	Descripción
codigoRolProyecto	Integer	Identificador del Rol asociado al Proyecto.
descripcionRol	String	Descripción del Rol.
funcionRol	String	Función que va desempeñar el Rol en el Proyecto.
nombreRol	String	Nombre del Rol.
Fuente: Autores Proyecto		

Tabla 7. Atributos principales de la clase RolProyecto

- ⊙ **Clase competenciaRol:** Entidad donde se establecen las habilidades y capacidades necesarias para completar las actividades de proyecto. Dependiendo de la naturaleza del proyecto, van cambiando las competencias de cada rol

Nombre	Tipo de datos	Descripción
codigoCompetenciaRol	Integer	Identificador de la competencia asociada a un Rol específico.

conocimiento	String	Temas, materias, o especialidades debe conocer, manejar o dominar el Rol.
habilidad	String	Habilidades específicas que debe poseer el Rol y en qué grado.
experiencia	String	Experiencia que debe tener el Rol, sobre qué temas o situaciones, y de qué nivel.
otros	String	Otros requisitos especiales tales como género, edad, nacionalidad, estado de salud, condiciones físicas, etc.
Fuente: Autores Proyecto		

Tabla 8. Atributos principales de la clase competenciaRol

- ⊙ **Clase PersonalProyecto:** Entidad que define los recursos humanos necesarios para realizar el proyecto, es decir el Equipo del Proyecto.

Nombre	Tipo de datos	Descripción
Auditoria	Auditoria	Campos de Auditoria
cargo	String	Cargo que desempeña la persona.
codigoOrganización	Integer	Código de la Organización asociada al Proyecto.
codigoPersonalProyeto	Integer	Identificador de la entidad
codigoUnidadAcademica	Integer	En caso de que la persona trabaje en una unidad Académica de la UIS, se registra el identificador de dicha Unidad Académico Administrativa asociada al Proyecto.
criteroFinalizacion	String	Criterio por el que se dio la finalización
fechaFinalizacion	Date	Fecha en que se finaliza la participación en el Proyecto.
fechalnicio	Date	Fecha en que se inicia la participación en el Proyecto.
personaDocumento	Decimal	Número del documento de identificación del personal asociado al proyecto

personaTipoDocumento	SmallInt	Tipo Documento de identificación de la persona
responsabilidad	String	Descripción de las responsabilidades del personal vinculado con el proyecto
Fuente: Autores Proyecto		

Tabla 9. Atributos principales de la clase PersonalProyecto

- ⊙ **Clase ModalidadContratación:** Entidad donde se indica las diferentes modalidades de contratación para asignar el personal del proyecto.

Nombre	Tipo de datos	Descripción
codigoModalidadContratación	Integer	Identificador de la modalidad de contratación de los servicios.
nombreModalidad	String	Nombre de la modalidad de contratación
descripcionModalidad	String	Descripción de la modalidad de contratación del personal del Proyecto
Fuente: Autores Proyecto		

Tabla 10. Atributos principales de la clase ModalidadContratación

3.3.2.3 Diagrama 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.

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

La documentación sobre los diagramas de secuencia elaborados se encuentra en el CD anexo que forma parte del presente documento en la carpeta: Documentación UML

Para el caso de uso: Crear Roles y Responsabilidades

El diagrama de clases que se presenta a continuación da una visión mas amplia acerca del funcionamiento y la interacción del sistema con el usuario, en el caso de uso Crear roles y responsabilidades.

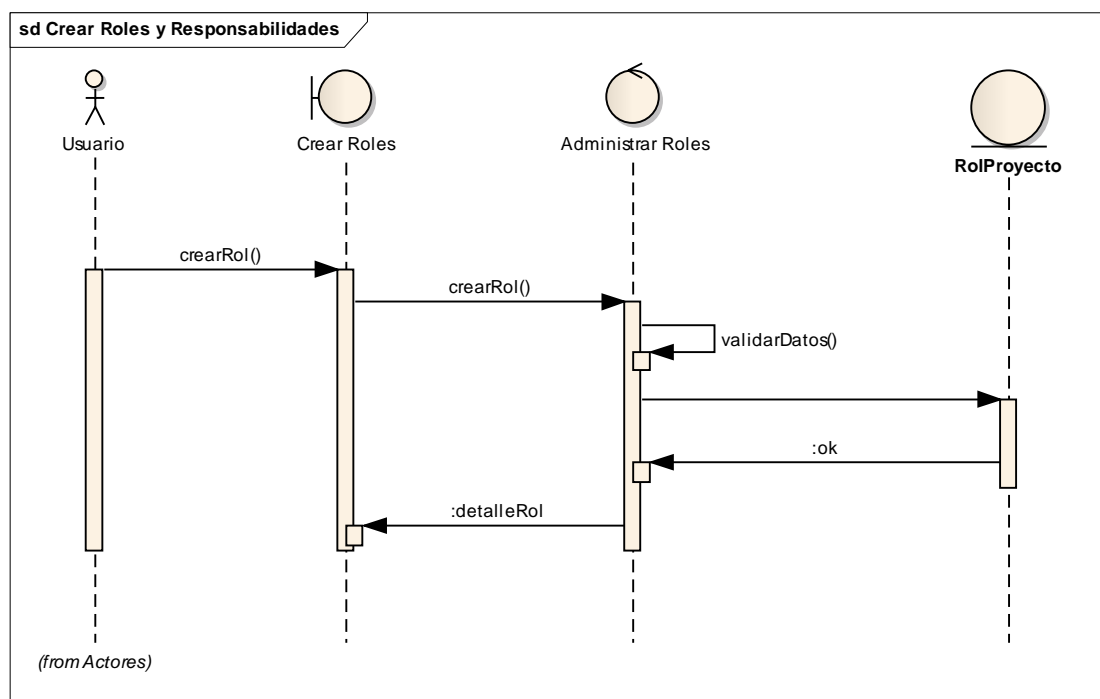


Figura 15. Diagrama de Secuencia para el caso de uso Crear Roles y Responsabilidades.

Para el caso de uso: Administrar Competencias

El diagrama muestra el funcionamiento y la interacción del sistema con el usuario, para definir las habilidades y capacidades necesarias de los integrantes del proyecto.

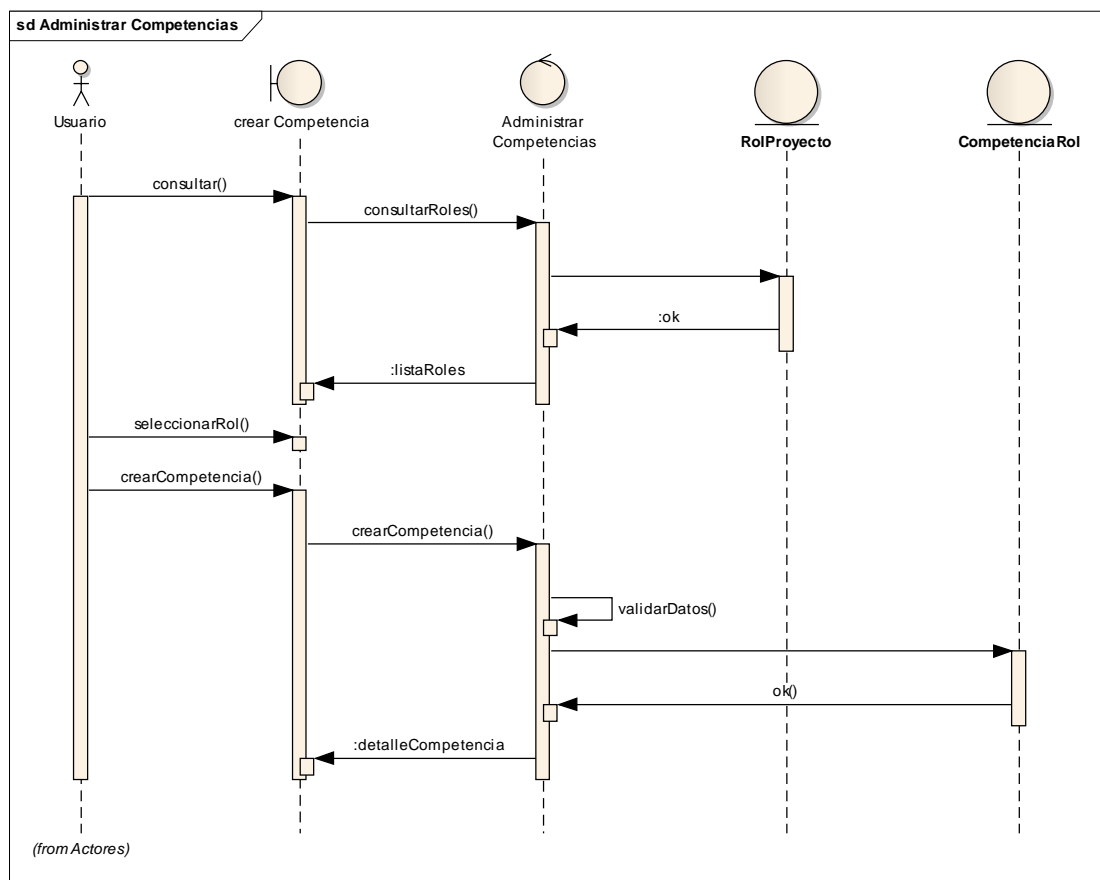


Figura 16. Diagrama de Secuencia para el caso de uso Administrar Competencias.

Para el caso de uso: Crear Organigrama

El diagrama muestra el funcionamiento y la interacción del sistema con el usuario, para definir los miembros del equipo del proyecto y su forma de contratación.

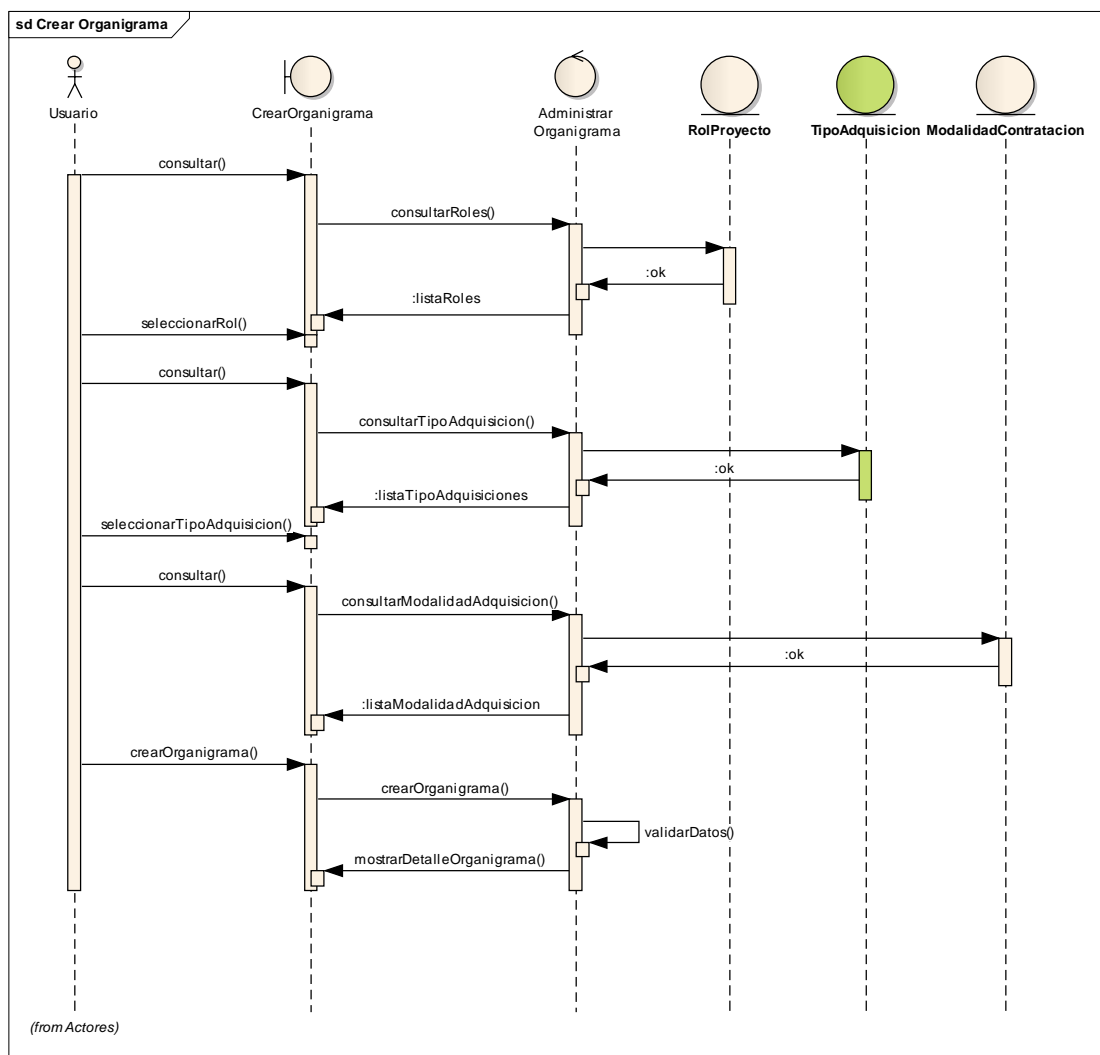


Figura 17. Diagrama de Secuencia para el caso de uso Administrar Competencias.

3.4 PRIMER PROTOPIPO

Para el módulo de planificación: Gestión de recursos humanos, compras y adquisiciones de SGPUIS, se desarrollo un prototipo inicial con funciones básicas como crear, modificar y eliminar registros. El prototipo consistía en una serie de formularios maestro-detalle que manipulaban las clases, rolProyecto, como se puede apreciar en el diagrama.

Funcionalidades:

- Crear registros de los roles del proyecto.
- Ver información detallada de los registros en una tabla paginada.
- Modificar los registros creados.
- Eliminar un registro seleccionado, previa confirmación.
- Capturar excepciones transaccionales (manipulación de la base de datos).

A continuación se muestra la interfaz, que permite la creación de los roles de proyecto y su visualización en la tabla:

Roles del Proyecto

Crear Roles del Proyecto

Nombre Rol

Descripción

Funciones

Guardar Cancelar

Listado Roles del Proyecto			
Nombre ↕	Descripción	Funciones	Acciones
PROJECT MANAGER	descripcion	funcion	
BENEFICIARIO			
INTERESADO			

Figura 18. Formato Crear Rol Proyecto.

La siguiente imagen pertenece al formato de detalle, modificación o eliminación de los roles del proyecto:

The screenshot displays a web form for editing project roles. The title bar reads 'Eliminacion Roles del Proyecto'. Below it, a section titled 'Detalles del Rol' contains three main input areas: 'Nombre Rol' with the text 'PROJECT MANAGER', 'Descripción' with the text 'descripcion', and 'Funciones' with the text 'funcion'. The 'Funciones' field is equipped with a rich text editor toolbar featuring icons for bold, italic, underline, text color, background color, bulleted list, and numbered list. At the bottom of the form are two buttons: 'Eliminar' and 'Cancelar'.

Figura 19. Formato Detalle, Edición y Eliminación Rol Proyecto.

El prototipo presentaba falencias, no se encargaba de otras validaciones de datos. No se determinaba que los valores de los datos fueran lógicos, no se podían hacer directamente asociaciones con las competencias.

3.5 SEGUNDO PROTOTIPO

El siguiente prototipo del módulo de planificación: Gestión de recursos humanos, compras y adquisiciones de SGPUIS, sufrió cambios tanto en la capa de la lógica de negocio, como en la capa de presentación.

Se permitió directamente la asociación de las competencias, a cada uno de los diferentes roles del proyecto, en el momento de realizar la modificación.

Funcionalidades:

- Crear registros de las competencias asociadas a cada uno de los diferentes roles del proyecto.

- Ver información detallada de los registros en una tabla paginada.
- Modificar los registros de las competencias creadas.
- Eliminar un registro seleccionado, previa confirmación.
- Capturar excepciones transaccionales (manipulación de la base de datos).

La siguiente imagen pertenece al formato de modificación de los roles del proyecto, donde se pueden asociar las competencias:

Edición Roles del Proyecto

Detalles del Rol

Nombre Rol

Descripción

Funciones

funcion

B / U ABC ↺ ↻ ✓ ☰ ☷

Agregar Competencia

Habilidad

Conocimiento

Experiencia

Otros

Listado de Competencias

Conocimiento	Habilidad	Experiencia	Otros	Acciones
conocimiento	habilidad	experiencia	otros	✎ ✖
Total registros: 1				

Figura 20. Formato de Edición de Rol Proyecto y Asociación de Competencias

3.6 PROTOTIPO FINAL

El prototipo final debe tener la calidad apropiada, también es necesario que el producto resultante sea de fácil mantenimiento del código fuente. En los prototipos anteriores se observaron detalles, que podrían dificultar este mantenimiento en el futuro, entonces se le dio un mejor manejo al código fuente, documentando los métodos, manejando paginas indexadas desde la capa de presentación e implementando los estándares de desarrollo definidos por la División de Servicios de información.

Se debe tener en cuenta la utilización de AJAX, mediante RichFaces generando así más rapidez en las páginas.

Se incluyeron validaciones tales como:

- Verificar la repetición de nombres.
- Insertar en la capa de presentación mensajes descriptivos acerca del error, que no permitió completar la transacción.
- Impedir eliminar registros, que tuvieran asociaciones con otras tablas.
- Por ejemplo: impedir la eliminación de un rol proyecto que tuviera competencias asociadas. Esto no es permitido, con el fin de mantener la integridad referencial de las bases de datos. Debido a que al hacer esto se puede perder una gran cantidad de información, dependiendo de si la configuración permite eliminar todos los atributos asociados, así la eliminación en cascada traería graves consecuencias al sistema.

En la capa de presentación se mejoro el aspecto de la pantalla eliminando algunos encabezados y colocando iconos de ayuda, para hacer la aplicación más entendible al usuario final.

En seguida se puede observar la interfaz para la creación de un rol proyecto:

Crear Roles del Proyecto

Nombre Rol

Descripción

Funciones

B I U ABC ↺ ↻ 📄 ☰ ☷

Listado Roles del Proyecto			
Nombre ↕	Descripción	Funciones	Acciones
PROJECT MANAGER	descripcion	funcion	
BENEFICIARIO			
INTERESADO			

Total registros: 3

Figura 21. Formato Final de Creación de Rol Proyecto

A continuación se puede observar la interfaz para la edición de un rol proyecto y al mismo tiempo la creación de las competencias asociadas a este:

• El Rol del Proyecto tiene competencias Asociadas

Modificar los Roles del Proyecto

Nombre Rol

Descripción

Funciones

funcion

Agregar Competencia

Habilidad

Conocimiento

Experiencia

Otros

Listado de Competencias				
Conocimiento	Habilidad	Experiencia	Otros	Acciones
conocimiento	habilidad	experiencia	otros	
Total registros: 1				

Figura 22. Formato Final de Edición de Rol Proyecto y Asociación de Competencias

A continuación se puede observar la interfaz a través de un modal panel para la edición y eliminación de una competencia asociada a un rol proyecto:

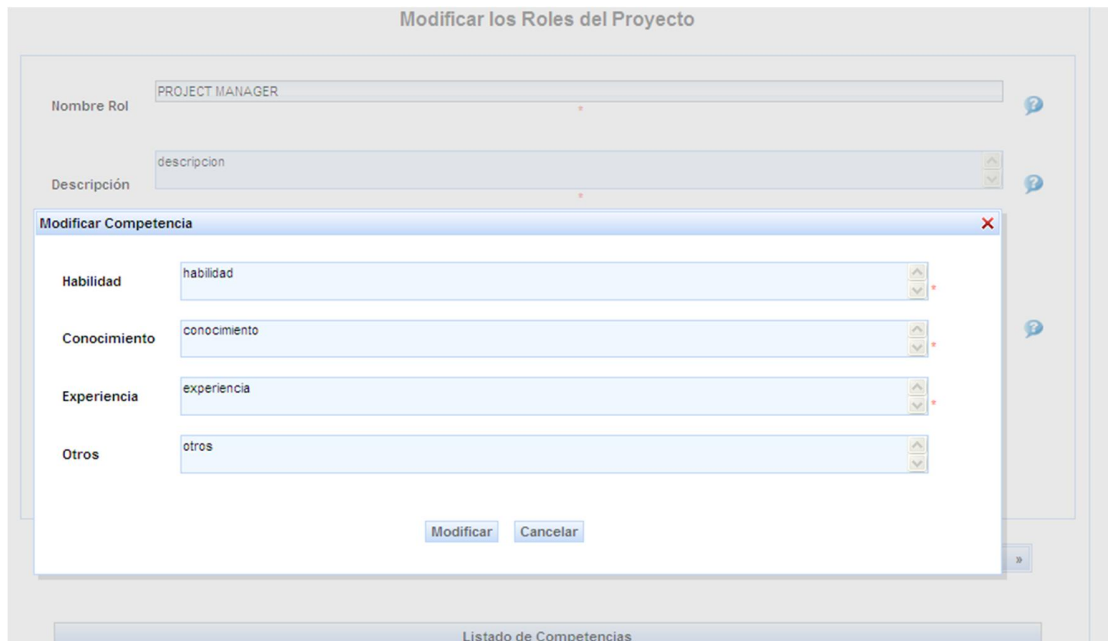


Figura 23. Formato de Edición y Eliminación de Competencias

3.6.1 El Proyecto enmarcado en el Esquema seguridad y el SGPUIS

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.

3.6.1.1 Estructura de la base de datos de 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 sistemas 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.

3.6.1.2 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-menu rol, descritas en “ESTRUCTURA DE LA BASE DE DATOS SOPORTE”.

3.6.1.3 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”.

3.6.1.4 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.

3.6.2 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.

3.6.2.1 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. sgpais.administracion.mantenimiento.RolProyecto

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. CONCLUSIONES

- El Nuevo sistema de información realizado para el sistema de Gestión de Proyectos de la Universidad Industrial de Santander, cumple con las expectativas de gestión de Proyectos contempladas en la metodología propuesta por el Project Management Institute (PMI®), brindando una planificación coordinada de sus recursos, para la correcta ejecución y evaluación de los proyectos.
- En el antiguo Sistema, la manipulación de la información del Recurso Humano asociado al Proyecto, se llevaba a cabo de una manera manual soportada por papeles y documentación archivada. El desarrollo del módulo brinda la confiabilidad y eficacia suficiente para sistematizar los procesos del manejo de personal asociado al proyecto, ya que está debidamente integrado con el sistema de Recursos Humanos de la Universidad Industrial de Santander.
- A pesar que no se logro la integración total con el sistema de Financiero de la Universidad Industrial de Santander por decisiones administrativas, el modulo de compras y adquisiciones se aproxima de manera considerable a lo planteado en los objetivos de este proyecto.
- Los insumos que se establecen en las solicitudes de las órdenes de compra, corresponden a los existentes en financiero y esto asegura la calidad de la información y facilita el registro de las órdenes de compra.
- Con la realización del módulo, pudimos comprobar que es muy importante para el desarrollo del software que cumpla con las expectativas adecuadas, la etapa de análisis de requerimientos iniciales, de modo que se contemplen todas las posibles necesidades del cliente, pues si no se

especifican desde un principio, pueden surgir cambios en el desarrollo del proyecto, que afectan el avance del mismo.

- El proyecto fue desarrollado con la implementación del esquema de seguridad, auditoría, estándares de diseño y desarrollo instaurada por la División de Servicios de Información de la UIS.
- Es importante destacar el trabajo en equipo, ya que el SGPUIS en general fue desarrollado por varios grupos; resaltando el aprendizaje obtenido de una tecnología nueva para nosotros. Por esto se hicieron necesarias diferentes capacitaciones en Java estándar y Java EE5. además de la continua investigación para poder llevar a cabo el buen desarrollo del sistema.
- Gracias a la exhaustiva revisión de la herramienta por parte de la DSI, se generó un código fuente ordenado y con buena documentación, que ayuda al fácil mantenimiento del mismo.
- Este sistema ha tenido una buena aceptación por parte de las directivas de la universidad, con quienes nos hemos reunido en diferentes ocasiones.

5. RECOMENDACIONES

- Es recomendable que el módulo se integre totalmente con el sistema de Financiero de la Universidad Industrial de Santander, con el fin de lograr registrar las órdenes de compra automáticamente en dicho sistema y evitar operaciones manuales que lleven a confusiones u olvido por parte del usuario al momento del registro del número de la orden de compra asociada con cada solicitud elaborada previamente.
- Es necesario que la División de servicios de Información junto con la oficina de Planeación, capacite adecuadamente a las personas encargadas de la administración general del módulo, de esta forma se garantizará un buen manejo del sistema con respecto a los resultados esperados.
- Con el desarrollo de este proyecto, se confirmó que el uso del framework RichFaces, hace más fácil el desarrollo ya que es posible crear componentes propios y ponerlos en práctica mediante la funcionalidad AJAX, sin embargo, esto podría hacer que se tomara demasiado tiempo en el desarrollo. Es por esto que se debe incentivar el uso de estas tecnologías para desarrollos web de este tipo.
- Es importante que la Escuela de Ingeniería de Sistemas impulse el aprendizaje y utilización de estas nuevas tecnologías.

6. BIBLIOGRAFÍA

[1] HERNÁNDEZ, Edwin; SUAREZ, Silvia. Sistema de información para el banco de programas y proyectos de inversión de la Universidad Industrial de Santander Volumen 1. Proyecto de pregrado Universidad Industrial de Santander. Bucaramanga, 2001.

[2] Project Management Institute, Guía de los fundamentos de gestión de proyectos -PMBOK, Tercera Edición.

[3] SERRANO, Giobani. Plan de proyecto: propuesta de modelo para la Gestión de Proyectos de Inversión en una Institución Pública de Educación Superior en Colombia: una perspectiva desde el Pensamiento Sistémico. Universidad Industrial de Santander. Bucaramanga. 2009. P. 23.

[4] CADENA RUIZ, Ana María. Antecedentes BPIN. {En línea}. {08 Febrero de 2010}. Disponible en:
(http://www.dnp.gov.co/PortalWeb/Portals/0/archivos/documentos/DIFP/Presupuesto/Antecedentes_Bpin.pdf)

[5] De Amescua Seco, Antonio. Cuadrado Gallego, Juan José. Ernica Lafuente, Emilio “Análisis Y Diseño Estructurado Y Orientado A Objetos De Sistemas Informáticos”. Madrid: Editorial Mc Graw Hill, 2003.

[6] Villalobos, Jorge Alberto. Casallas, Rubby. “Fundamentos de programación – Aprendizaje Activo Basado en Casos: Un Enfoque Moderno Usando Java, Uml, Objetos Y Eclipse”. México: Pearson Educación, 2006.

[7] Marchioni, Francesco. "JBoss AS 5 Development". Birmingham, Reino Unido: Packt Publishing, 2009.

[8] Dudley, Bill. Lehr, Jonathan. Willis, Bill. Mattingly, LeRoy. "Mastering JavaServer Faces". Indianapolis, Estados Unidos: Wiley Publishing, 2004.

[9] Universidad Industrial de Santander, Acuerdo N° 064 de 2008

[10] Factor 10 Recursos financieros, Autoevaluación Institucional disponible en:
<https://www.uis.edu.co/webUIS/es/administracion/rectoria/acreditacionInstitucional/autoevaluacionInstitucional/informe/factor10.pdf>

7. ANEXOS

Anexo A.

Diagramas UML. En el medio magnético adjunto pueden ser consultados en su totalidad los diagramas UML correspondientes a este proyecto. Se encuentran ubicados en la carpeta /Proyecto SGPUIS/DIAGRAMAS_UML, se puede acceder a ellos mediante el archivo index.html.

Anexo B.

Estándares DSI. Consta de un documento en el que pueden ser consultados los prototipos de estándares de desarrollo para la División de Servicios de Información de la Universidad Industrial de Santander. Puede ser consultado en el medio magnético adjunto, su nombre es:

- Estándares DSI.pdf