

SISTEMA DE INFORMACIÓN WEB PARA EL TRÁMITE DE SOLICITUDES
DEL ÁREA DE RECURSOS HUMANOS DE LA UNIVERSIDAD INDUSTRIAL
DE SANTANDER

EDGAR ANDRES GIRALDO GUTIERREZ
HERNÁN DARÍO ROJAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2004

SISTEMA DE INFORMACIÓN WEB PARA EL TRÁMITE DE SOLICITUDES
DEL ÁREA DE RECURSOS HUMANOS DE LA UNIVERSIDAD INDUSTRIAL
DE SANTANDER

EDGAR ANDRES GIRALDO GUTIERREZ
HERNAN DARIO ROJAS

Proyecto de grado presentado como requisito parcial
para optar al título de Ingeniero de Sistemas

Director
OLGA GAMBOA PORRAS
Profesional División de Servicios de Información

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2004

AGRADECIMIENTOS

Cuando se termina una etapa en nuestras vidas y vemos hacia atrás para evaluar nuestros triunfos y logros obtenidos, nos detenemos un poco y agradecemos a todas aquellas personas que fueron partícipes de la meta que un día anhelamos y que gracias al esfuerzo, la dedicación y el apoyo incondicional hoy se hace realidad.

En nosotros siempre existirá el recuerdo grato de todos y cada uno de los momentos que hoy nos brindan la seguridad y confianza para seguir soñando y construyendo las metas que luego serán parte de nuestra vida.

Queremos antes que nada, agradecer a Dios por ser nuestro guía y luz durante esta etapa y durante toda nuestra vida.

A la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, por la formación académica brindada.

A la División de Servicios de Información de la Universidad Industrial de Santander, por compartir el conocimiento e involucrarnos en el desarrollo de nuestra institución, brindando así, crecimiento a nuestra vida profesional.

A los ingenieros Enrique Torres López, Olga Gamboa Porras, por la confianza depositada, el apoyo y credibilidad en nuestras capacidades.

A todos los compañeros de la División de Servicios de Información, en especial a Rocío Carolina y Carmen Imelda por su valiosa colaboración como profesionales y amigas nuestras.

A nuestros amigos, Walter Calderón, Diego Pinzón, Andrés García, Federico García, Olga Villamizar, Laura Anaya, Jeovany Lopez, Leonel Ariza y todas aquellas personas que en algún momento nos brindaron su apoyo.

A Dios, por ser la fuente y esencia de nuestras vidas.

A mi madre Elssy Gutiérrez Cuellar y a mi padre Jesús Hernán Giraldo Viatela, pilares fundamentales de mi vida, quienes con dedicación y amor sembraron todos aquellos valores que configuran mi ser; en quienes siempre encuentro amor, amistad, consejo y comprensión, sea en tiempos difíciles como en momentos de felicidad.

Gracias, los quiero mucho.

A mis hermanos, Mauricio, Carlos y David, quienes creyeron en mí y me brindaron una voz de aliento en aquellos momentos en que la necesité. Los quiero mucho.

A Johanna, quien brindó alegría a mi vida y estuvo a mi lado en los momentos en que necesité seguridad y tranquilidad. Gracias Joha.

A mis amigos “9711” con quienes inicié el recorrido hacia esta meta, y a quienes hoy agradezco por su sinceridad y apoyo. Creímos en algo y luchamos por ello.

Edgar Andrés Giraldo Gutiérrez

A Dios por ser mi luz y mi guía durante mi vida

A mi madre Rosa Tulia Rojas, su apoyo, su esfuerzo, su confianza, su seguridad, con ella las cosas parecen ser mucho más fáciles. La deuda nunca se saldrá y la palabras de agradecimiento siempre se quedarán cortas. Gracias Madre.

A mis otras dos marías: mi tía María y mi abuela María, a ellas les debo todos los años de la vida que me han entregado, mis infinitos agradecimientos.

A toda mi familia, mis tíos y primos de los cuales obtengo la seguridad en momentos de flaqueza.

A todos mis maestros y profesores, desde mi maestra de Kinder la señora Mariela, hasta mis últimos en la Escuela de Ing. De Sistemas y la División de Servicios de Información. Pero en especial a aquellos que no me ofrecieron cálculo o álgebra, sino de los que aprendí de la asignatura más importante de todas: la vida.

A los amigos sinceros, los que siempre estuvieron ahí, para brindar su apoyo y por los cuales uno puede decir que no está solo.

A mis compañeros de Universidad, en especial a los de mi carrera, los “9711”, con los cuales no sólo compartí las aulas de clases sino parte de sus vidas.

Hernán D. Rojas

CONTENIDO

	pág.
INTRODUCCIÓN	1
1. PRESENTACIÓN	3
1.1 DESCRIPCIÓN DEL DOCUMENTO	3
1.2 ANTECEDENTES	4
1.2.1 Sistema De Información De Administración De Recursos Humanos Para La Universidad Industrial De Santander	5
1.3 DESCRIPCIÓN DEL PROBLEMA	11
1.4 OBJETIVOS	12
1.4.1 Objetivo General	12
1.4.2 Objetivos Específicos	13
1.5 IMPACTO Y VIABILIDAD	15
1.5.1 Impacto	15
1.5.2 Viabilidad	16
1.6 DESARROLLO DEL SISTEMA	17
1.6.1 Hardware	17
1.6.2 Software	17

2. FUNDAMENTACIÓN TEÓRICA	19
2.1 GENERALIDADES ACERCA DE INTERNET	19
2.1.1 Protocolo HTTP	20
2.2 ARQUITECTURA DE INFORMACIÓN	21
2.2.1 Modelo Cliente/Servidor	22
2.2.2 Arquitectura de la Aplicación	23
2.3 LENGUAJES DE PROGRAMACIÓN UTILIZADOS	26
2.3.1 HTML: HyperText Markup Language	26
2.3.2 Lenguajes de Script	26
2.3.3 Active Server Pages	28
2.3.4 Visual Basic y Dll's	29
2.4 BASES DE DATOS	29
2.4.1 Aspectos Generales de Bases de Datos	29
2.4.2 Ventajas de las Bases de Datos	30
2.4.3 Gestor de Bases de Datos	31
2.4.4 Motor de Bases de Datos	32
2.5 ADO (MICROSOFT ACTIVEX DATA OBJECTS)	32
2.6 SERVIDOR DE SERVICIOS DE INTERNET: INTERNET INFORMATION SERVER	34

2.7 SISTEMAS DE INFORMACIÓN	35
2.8 METODOLOGÍA DE DESARROLLO	36
2.8.1 El Proceso Unificado de Desarrollo de Software	36
2.9 UML: EL LENGUAJE UNIFICADO DE MODELADO	40
2.9.1 Objetivos del UML	40
2.9.2 Diagramas de UML	41
3. FASE DE INICIO	44
3.1 FLUJO DE REQUISITOS	44
3.1.1 Listado de Funcionalidades	47
3.1.2 Listado de Actores	48
3.2 FLUJO DE ANÁLISIS	50
3.2.1 Análisis de Procedimientos	50
3.2.2 Análisis de la Arquitectura	56
3.2.3 Dependencia entre Paquetes de Análisis y Capas	58
3.2.4 Análisis de Casos de Uso	58
3.3 CONCLUSIONES DE LA FASE DE INICIO	60
4. FASE DE ELABORACIÓN	63
4.1 FLUJO DE REQUISITOS	64

4.1.1	Listado de Procedimientos	64
4.1.2	Listado de Funcionalidades	66
4.1.3	Actores	67
4.1.4	Clasificación de los Casos de Uso	71
4.2	FLUJO DE ANÁLISIS	87
4.2.1	Análisis de la Arquitectura	87
4.2.2	Análisis de Casos de Uso	90
4.2.3	Análisis de Clases	110
4.3	FLUJO DE DISEÑO	103
4.3.1	Diseño de la Arquitectura	103
4.3.2	Diseño de los Casos de Uso	105
4.3.3	Diseño de las Clases	121
4.4	FLUJO DE IMPLEMENTACIÓN	122
4.4.1	Interfaces	123
4.4.2	Clases	125
4.4.3	Modelo de Datos	127
4.5	CONCLUSIONES DE LA FASE DE ELABORACIÓN	134
5.	FASE DE CONSTRUCCIÓN	137

5.1 FLUJO DE REQUISITOS	137
5.1.1 Casos de Uso Detallados	139
5.2 FLUJO DE ANÁLISIS	148
5.2.1 Análisis de la Arquitectura	148
5.2.2 Análisis de Clases	152
5.3 FLUJO DE DISEÑO	153
5.3.1 Diseño de los Casos de Uso	155
5.3.2 Diseño de las Clases	161
5.4 FLUJO DE IMPLEMENTACIÓN	161
5.4.1 Interfaces	162
5.4.2 Clases	162
5.4.3 Modelo de Datos	163
5.5 FLUJO DE PRUEBAS	168
5.6 CONCLUSIONES DE LA FASE DE CONSTRUCCIÓN	174
6. ESTRUCTURA DEL SISTEMA	176
6.1 DISTRIBUCIÓN DE PÁGINAS EN EL SERVIDOR	176
6.2 ESTRUCTURA DE PÁGINAS ASP DEL SISTEMA	181
6.3 SEGURIDAD DEL SISTEMA	182

6.4 LIBRERÍAS DINÁMICAS (DLL) – SOLDRH.DLL	185
6.4.1 Clases y Definición de Clases	185
6.4.2 Definiciones Relacionadas con el Concepto de Clases	186
6.4.3 Estructura de las Clases de la Librería Dinámica	187
6.4.4 Clase de Ejemplo “persona.cls”	187
6.4.5 Definición de las Clases Principales	191
6.5 ESTRUCTURA DEL INBOX	192
7. CONCLUSIONES	195
8. RECOMENDACIONES	197
BIBLIOGRAFIA	199
BIBLIOGRAFÍA EN INTERNET	201

LISTADO DE FIGURAS

	pag.
Figura 1. Comunicación entre cliente y servidor	20
Figura 2. Modelo Cliente/Servidor	22
Figura 3. Capa de presentación	24
Figura 4. Capa de acceso a datos	24
Figura 5. Capa del negocio	25
Figura 6. Arquitectura de tres capas	25
Figura 7. Solicitud y respuesta de una página ASP	28
Figura 8. Objetos que conforman ADO	33
Figura 9. Accesos a bases de datos con IIS	34
Figura 10. Conexión a bases de datos con IIS	35
Figura 11. Diagrama de Sistema de Información	36
Figura 12. Flujos y fases del Proceso Unificado de Desarrollo de Software	39
Figura 13. Diagrama de actividades para la generación de solicitudes	46
Figura 14. Diagrama de actividades para la evaluación de solicitudes	47
Figura 15. Diagramas de casos de uso – Fase de inicio	49
Figura 16. Diagrama de estados para el procedimiento de permisos	50
Figura 17. Diagrama de estados para el procedimiento de Licencia no Remunerada	51
Figura 18. Ejemplo de estados para una Comisión de servicios	52
Figura 19. Ejemplo de estados para una Comisión de estudios	53
Figura 20. Diagrama de estado para el procedimiento de Periodo sabático	54
Figura 21. Diagrama de estados para el procedimiento de Acumulación de Vacaciones	54
Figura 22. Diagrama de estados para el procedimiento de Disfrute de vacaciones	55
Figura 23. Diagrama de estados para el procedimiento de Informes	55
Figura 24. Diagrama de paquetes – Fase de inicio	57

Figura 25. Paquete de servicio	58
Figura 26. Dependencia entre paquetes – Fase de inicio	58
Figura 27. Representación gráfica de las clases de análisis	59
Figura 28. Diagrama de colaboración – Verificar Ingreso	60
Figura 29. Relación entre los actores del sistema	68
Figura 30. Diagrama de casos de uso – Fase de Elaboración	69
Figura 31. Diagrama de estados – Solicitar Permiso	73
Figura 32. Diagrama de estados – Ver Estado	76
Figura 33. Diagrama de estados – Cancelar Solicitud	76
Figura 34. Diagrama de estados – Aprobar/Visar Solicitud	78
Figura 35. Diagrama de estados – Rechazar Solicitud	79
Figura 36. Diagrama de estados – Ver Pendientes	80
Figura 37. Diagrama de estados – Registrar Reintegro	81
Figura 38. Diagrama de estados – Consultar Información	82
Figura 39. Diagrama de estados – Cambiar Rol	83
Figura 40. Diagrama de estados – Cruzar Tiempo	84
Figura 41. Diagrama de estados – Registrar Grupales	85
Figura 42. Diagrama de estados – Registrar Causados	87
Figura 43. Paquetes de Análisis relacionados con casos de uso	89
Figura 44. Paquetes de servicio	90
Figura 45. Dependencia entre paquetes	90
Figura 46. Diagrama de colaboración – Solicitar Permiso	91
Figura 47. Diagrama de colaboración – Ver Estado	92
Figura 48. Diagrama de colaboración – Cancelar Solicitud	93
Figura 49. Diagrama de colaboración – Aprobar/Visar Solicitud	93
Figura 50. Diagrama de colaboración – Rechazar solicitud	94
Figura 51. Diagrama de colaboración – Ver pendientes	95
Figura 52. Diagrama de colaboración – Registrar Reintegro	95
Figura 53. Diagrama de colaboración – Consultar Información	96
Figura 54. Diagrama de colaboración – Cambiar Rol	97
Figura 55. Diagrama de colaboración – Registrar Causados	97
Figura 56. Diagrama de colaboración – Registrar Grupales	98

Figura 57. Diagrama de colaboración – Cruzar Tiempo	99
Figura 58. Modelo de despliegue	103
Figura 59. Subsistemas intermedios y de software del sistema	104
Figura 60. Dependencias entre subsistemas	105
Figura 61. Diagrama de secuencia – Solicitar Permiso	107
Figura 62. Diagrama de secuencia - Ver Estado	109
Figura 63. Diagrama de secuencia – Cancelar Solicitud	110
Figura 64. Diagrama de secuencia – Aprobar/visar Solicitud	111
Figura 65. Diagrama de secuencia – Cambiar Rol	113
Figura 66. Diagrama de secuencia – Consultar Información	114
Figura 67. Diagrama de secuencia – Registrar Causados	116
Figura 68. Diagrama de secuencia – Registrar Grupales	117
Figura 69. Diagrama de secuencia – Registrar Reintegro	118
Figura 70. Diagrama de secuencia – Ver Pendientes	120
Figura 71. Diagrama de secuencia – Cruzar Tempo	121
Figura 72. Interfaz del caso de uso Solicitar Permiso	123
Figura 73. Interfaz PaginaAsistentes caso de uso Registrar Grupales	124
Figura 74. Interfaz del caso de uso Ver Estado	125
Figura 75. Diagrama de clases – Fase de Elaboración	126
Figura 76. Diagrama de clases – Fase de Elaboración	128
Figura 77. Modelo de datos para actividades	129
Figura 78. Modelo de datos para la solicitud de Permiso	130
Figura 79. Modelo de datos para tipos de permiso	131
Figura 80. Modelo de datos para permisos compensatorios	132
Figura 81. Modelo de datos para la solicitud de cancelación de permisos	133
Figura 82. Modelo de casos de uso – Fase de Construcción	139
Figura 83. Diagrama de estados – Enviar Informes	143
Figura 84. Diagrama de estados – Registrar concepto informes	144
Figura 85. Diagrama de estados – Modificar Solicitud.	145
Figura 86. Diagrama de Estados – Registrar Información	147
Figura 87. Diagrama de colaboración – Enviar Informes	149
Figura 88. Diagrama de colaboración – Registrar concepto informes	150

Figura 89. Diagrama de Colaboración – Modificar solicitud	151
Figura 90. Diagrama de Colaboración – Registrar Información	152
Figura 91. Diagrama de secuencia – Enviar informes	156
Figura 92. Diagrama de secuencia – Registrar Concepto Informes	157
Figura 93. Diagrama de secuencia – Registrar Concepto Informes (opcional)	158
Figura 94. Diagrama de secuencia – Modificar Solicitud	159
Figura 95. Diagrama de secuencia – Registrar información	161
Figura 96. Diagrama de clases – Fase de Construcción	163
Figura 97. Modelo de datos para las solicitudes	165
Figura 98. Modelo de datos para las solicitudes de cancelación	166
Figura 99. Modelo de datos para las solicitudes de prórroga	168
Figura 100. Modelo de interfaz de evaluación	173
Figura 101. Niveles jerárquicos en la distribución de archivos en el sistema	176
Figura 102. Distribución de Páginas y archivos en el servidor IIS	177
Figura 103. Esquema de pagina ASP	181
Figura 104. Flujo de navegación en el sistema	183
Figura 105. Componentes de una clase	186
Figura 106. Estructura de una clase	187
Figura 107. Lista de listas	190
Figura 108. INBOX del sistema	192
Figura 109. Baner del Sistema de Recursos Humanos	192
Figura 110. Combo para cambio de rol	193
Figura 111. Menú dinámico del sistema	193
Figura 112. Cuerpo principal de INBOX	194

LISTADO DE TABLAS

	pag.
Tabla 1. Diagramas presentes en UML	41
Tabla 2. Aprobación de comisiones de acuerdo a la duración	51
Tabla 3. Casos de uso, actores y prioridad	72
Tabla 4. Descripción de clases de análisis	100
Tabla 5. Clases de diseño – Solicitar Permiso	106
Tabla 6. Clases de diseño – Ver Estado	108
Tabla 7. Clases de diseño – Cancelar Solicitud	109
Tabla 8. Clases de diseño – Aprobar/Visar Solicitud	111
Tabla 9. Clases de diseño – Cambiar Rol	112
Tabla 10. Clases de diseño – Consultar Información	113
Tabla 11. Clases de diseño – Registrar Causados	115
Tabla 12. Clases de diseño – Registrar Grupales	116
Tabla 13. Clases de diseño – Registrar Reintegro	118
Tabla 14. Clases de diseño – Ver Pendientes	119
Tabla 15. Clases de diseño – Cruzar Tiempo	120
Tabla 16. Resumen de actividades de la Fase de Elaboración	135
Tabla 17. Cuadro comparativo de los casos de uso de solicitud	148
Tabla 18. Descripción de clases de análisis – Fase de Construcción	152
Tabla 19. Relación entre clases de análisis y diseño	154
Tabla 20. Clases de diseño – Enviar Informes	155
Tabla 21. Clases de diseño – Registrar Concepto Informes	157
Tabla 22. Clases de diseño – Modificar Solicitud	159
Tabla 23. Clases de diseño – Registrar Información	160
Tabla 24. Relación entre las solicitudes y tablas del modelo de datos	164
Tabla 25. Relación entre tipos de solicitud y tablas de manejo	165
Tabla 26. Relación entre tipos de solicitud y tablas de manejo de Cancelaciones	167
Tabla 27. Identificadores de tipos de datos	189

1. TITULO SISTEMA DE INFORMACION WEB PARA EL TRAMITE DE SOLICITUDES DEL AREA DE RECURSOS HUMANOS DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.
2. AUTORES GIRALDO GUTIERREZ EDGAR ANDRES, ROJAS HERNAN DARIO. **
3. PALABRAS CLAVES Sitio Web, Sistema de Información, Active Server Pages, División de Recursos Humanos, Gestión de Solicitudes, Trámite de solicitudes, Situaciones administrativas.
4. DESCRIPCIÓN

La Universidad Industrial de Santander consciente de la continua demanda de servicios que contribuyan al eficiente manejo de la información y apoyen su proceso de modernización y expansión, cuenta con un Sistema de Información para la Gestión del Recurso Humano con el cual realiza el soporte informático a la División de Recursos Humanos de la Universidad, atendiendo diferentes áreas de la administración del personal.

El presente proyecto se desarrolló con el propósito de apoyar a la División de Recursos Humanos de la Universidad, en la gestión de situaciones docente-administrativas y proporcionar una alternativa en el trámite de solicitudes del personal de planta de la Universidad.

Debido al enfoque Web del sistema de información, la Universidad alcanza con éste una mayor cobertura de servicios, la cual a su vez mejora la calidad de vida laboral del personal planta docente y administrativo.

El proyecto fue desarrollado bajo la metodología del Proceso Unificado de Desarrollo de Software, la cual permitió llevar a cabo de una manera clara el análisis, diseño, elaboración y construcción del mismo. El Lenguaje Unificado de Modelado (UML) se utilizó como herramienta de documentación para las actividades descritas por la metodología.

El Sistema de Información ofrece un primer módulo para el trámite de las siguientes situaciones administrativas: permisos, licencias no remuneradas, comisiones de estudios y servicios, período sabático, prórroga de solicitudes, disfrute y acumulación de vacaciones y registro y evaluación de informes. Un segundo módulo contempla el siguiente conjunto de funcionalidades, complementando el trámite de las situaciones administrativas mencionadas: cancelación de solicitudes, registro de reintegros y modificación de solicitudes, entre otros.

* Proyecto de Grado

** Facultad de Ciencias Físico-Mecánicas, Ingeniería de Sistemas, Dir. Ing. Olga Gamboa Porras

1. TITLE WEB INFORMATION SYSTEM FOR THE PROCEEDING OF THE HUMAN RESOURCES AREA REQUESTS OF THE INDUSTRIAL UNIVERSITY OF SANTANDER.
2. AUTHORS GIRALDO GUTIERREZ EDGAR ANDRES, ROJAS HERNAN DARIO.**
3. KEY WORDS Website, Information System, Active Server Pages, Department of Human Resources, Management of Requests, Proceeding of Requests, Administrative Situations.
4. DESCRIPTION

The Industrial University of Santander conscious of the continuous demand of services that contribute to the efficient information management and it help its process of modernization and expansion, the university has an information System for the management of human resource to make the informatic support to the University Department of Human Resource, taking care of different personal management areas.

The present project was developed with the purpose to support to the University Department of Human Resources in the management of the educational and administrative situations and provide an alternative in the proceeding of request of University base personal.

Due to the Web Information System approach, the University reaches with this a great services cover, which it improves the quality of life as well of the base teacher and administrative personal.

The project was developed following the Unified Process of Software Develop methodology, which allow us to obtain of a clear way the analysis, design, elaboration and construction of the project. The Unified Modeling Language (UML) was used as documentation tool to the described methodology's activities.

The Information System offers a first module to the proceeding of the following administrative situations: permissions, license nonremunerated, commissions of studies and services, sabbatic period, proroguing of requests, enjoy and accumulation of vacations and registry and evaluation of reports. A second module has the following set of functionalities, complementing the proceeding of the administrative situations mentioned: requests' cancellation, registry of the personal reenter and requests' modification, among others.

* Project.

** Physique-Mechanics Sciences Department, Computer Science Engineering, Olga Gamboa Porras.

GLOSARIO

ASP: Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para el web.

ATRIBUTO: Valor de un dato que es almacenado en los objetos de una clase. Cada atributo tiene un valor para una instancia de un objeto.

BROWSER: Un browser o navegador es una aplicación cliente de software para Internet que sirve como interface para navegar a través del mundo de información en la web. Existen muchos navegadores, pero todos tienen más similitudes que diferencias y son muy sencillos de usar; el mercado está prácticamente dividido entre dos de ellos: Internet Explorer de Microsoft ® y Navigator de Netscape ®.

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

CLIENTE: En la arquitectura cliente – servidor, el cliente es el sistema o persona que a través de su computador solicita un servicio a una servidor, el cual está en capacidad de dar respuesta a dichas solicitud.

DRIVER: para el caso de las conexiones a un DBMS, un driver es un archivo que se activa desde la aplicación (software) para poder acceder y operar sobre las bases de datos que se encuentren en el manejador.

HTTP: (Protocolo de transferencia de Hipertexto) Protocolo que permite la transmisión de documentos de hipertexto entre el cliente que lo solicita y el servidor que lo suministra.

INSTANCIA: Es un objeto que es creado a partir de una clase determinada, el cual tiene atributos y operaciones propias.

INTERFAZ: La idea fundamental en el concepto de interfaz es el de mediación. La interfaz es lo que "media", lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza, típicamente el ser humano y una máquina como el computador.

MANEJADOR DE BASES DE DATOS: El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. Las funciones principales de un DBMS son crear y organizar la base de datos, establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accedidos rápidamente, manejar los datos de acuerdo a las peticiones de los usuarios y registrar el uso de las bases de datos.

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

PROTOCOLO: Un protocolo es un conjunto de reglas y estándares que permiten a los equipos intercambiar información.

SCRIPT: Un script es una secuencia de ordenes, en un determinado lenguaje, que puede ser ejecutado por un cliente Web desde su navegador y visualizar el documento en que está contenido. Actualmente los dos lenguajes de script mas usados en paginas Web son JavaScript y VBScript. La inclusión de scripts en los documentos HTML hace que éstos sean más inteligentes. El contenido se genera en forma dinámica, mientras que los valores introducidos en los formularios pueden comprobar localmente, sin necesidad de contar con un servidor y emplear un cierto tiempo en ello.

SERVIDOR: En la arquitectura cliente servidor es el sistema que recibe solicitudes por parte de los clientes, las procesa y les regresa respuestas como resultados a las solicitudes.

SERVIDOR WEB: Software que tiene como objetivo recibir las solicitudes que un cliente le hace a través de Internet, para procesarlas y generar los documentos o páginas Web como resultado al cliente.

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

SQL: (*Structured Query Language*) Lenguaje de cuarta generación que permite ejecutar operaciones como consultas, y actualizaciones sobre las entidades existentes en bases de datos relacionales.

URL: Uniform Resource Locator. Localizador Uniforme de Recursos. Dirección electrónica que representa o apunta a una página o fichero incluido en la WEB de Internet.

VALIDACIÓN: Es la actividad de comprobar que los atributos de un objeto pertenezcan al rango de valores admisibles por un sistema.

INTRODUCCIÓN

Durante los últimos años se ha vivido un creciente proceso de modernización en la vida social general y en particular en lo que se refiere a los medios de comunicación. La llegada de la televisión por cable y su creciente masificación, la digitalización de las líneas telefónicas, la difusión de la telefonía celular, el uso del fax, crecimiento de los computadores personales, el uso del correo electrónico y la implementación de Internet en Universidades, empresas e incluso hogares.

La difusión de muchas tecnologías que brindan la posibilidad de estar a la vanguardia en el área de las telecomunicaciones tanto a nivel local como global, a pesar de no ser usadas por la totalidad de la población y que sus verdaderas capacidades no sean realmente aprovechadas en nuestro medio, como se puede apreciar en el uso diario de la Internet y todas la aplicaciones Web desarrolladas hasta el momento, hacen de las comunicaciones uno de los campos más interesantes de estos tiempos.

Internet es un sistema de redes que conecta computadores en todo el mundo mediante satélites y redes. La importancia y popularidad de este sistema ha ido creciendo con fuerza en nuestro medio y cada vez es más utilizado por investigadores, docentes y estudiantes de diferentes disciplinas; razón por la cual se ha incrementado considerablemente el desarrollo de nuevas aplicaciones que hacen uso de la Internet y se ha obligado al mejoramiento de las ya existentes.

Las aplicaciones para el Web o sistemas de información en línea son la mayor de las tecnologías en crecimiento; desde los últimos diez (10) años la evolución de estos sistemas en los diferentes sectores de la sociedad ha sido arrolladora y al parecer vienen para quedarse como pilar fundamental en el éxito futuro de las organizaciones. Las ventajas que ofrecen los sistemas de información en línea son inmensas, pero aquellas que los hacen diferentes y necesarios para las organizaciones son la continua interacción de sus usuarios ubicados en cualquier

parte del mundo y la publicidad que puede dar a su negocio frente a todos los clientes de esta supercarretera de la información, como es denominada Internet.

Es por esta razón que cada vez más organizaciones buscan un alto desarrollo y la utilización de los sistemas de información en línea que incrementan su cobertura y mejoran los servicios que pueden ofrecer a la comunidad en general.

La Universidad Industrial de Santander como protagonista y pilar fundamental en el desarrollo de la educación superior del nororiente colombiano y el país, conciente de la continua demanda de servicios que contribuyan al eficiente manejo de la información, que mejoren la calidad del servicio prestado y apoyen su proceso de modernización y expansión, da inicio a la implementación del presente proyecto con el fin de presentar otra alternativa dinámica y funcional para la atención y trámite de solicitudes del Recurso Humano como elemento clave en el funcionamiento de toda entidad educativa.

El desarrollo del sistema de información para el Area de Recursos Humanos se emprende a través de la División de Servicios de Información con el ánimo de brindar tanto a docentes como administrativos un instrumento de apoyo en la gestión de situaciones administrativas que asuma con agilidad las políticas determinadas por las directivas de la Universidad para dicha gestión.

Debido al enfoque Web del sistema de información, la Universidad logrará una mayor cobertura de servicios, la cual a su vez mejorará la calidad de vida laboral del personal docente y administrativo vinculado.

Teniendo en cuenta, la existencia del recurso humano altamente calificado y disponible en los estudiantes de la Escuela de Ingeniería de Sistemas e Informática y la División de Servicios de Información de la Universidad, resultó viable la implementación del sistema de información bajo la modalidad de práctica empresarial para suplir la necesidad de crear sistemas que utilicen las tecnologías vanguardistas con que cuenta la Universidad.

1. PRESENTACIÓN

1.1 DESCRIPCIÓN DEL DOCUMENTO

Aunque a nivel general, no se le dé gran importancia a esta sección del documento porque no se considera como una parte relevante en el desarrollo del proyecto y en el cumplimiento de los objetivos planteados, esta sección sirve de guía al lector al momento de querer estudiar el proyecto paso a paso y a la vez facilita el entendimiento y comprensión del mismo.

La intención del documento es ser comprensible a cualquier tipo de lector, sin importar su grado de preparación en el tema, por lo cual, basándose en una implantación clara y concisa, muestra el proceso que se llevó a cabo en el desarrollo del proyecto dejando ver la fundamentación teórica utilizada como referencia para la implantación del sistema y el cumplimiento de los objetivos iniciales.

La organización por capítulos del presente documento, lleva un orden secuencial para facilitar el entendimiento de las fases llevadas a cabo en la elaboración del sistema de información.

En el capítulo 1 se presentan los antecedentes para la realización del proyecto, el objetivo general y los objetivos específicos en el plan de proyecto.

El capítulo 2 presenta la fundamentación teórica básica para el desarrollo del sistema de información.

El capítulo 3 describe la fase de inicio en el desarrollo del sistema en donde se presenta la concepción inicial de sistema.

El capítulo 4 describe la fase de elaboración.

El capítulo 5 presenta la fase de construcción de cada uno de los módulos que conforman el sistema por medio del desarrollo de los casos de uso correspondientes.

El capítulo 6 describe la estructura del sistema en lo que refiere a la disposición de las páginas Web.

El capítulo 7 contiene las conclusiones y recomendaciones para tener en cuenta en la continuidad del sistema de información.

1.2 ANTECEDENTES

Hacia el año 1994 los lineamientos de las Instituciones públicas y las políticas de modernización de ese entonces de la Universidad fueron la base para proyectar nuevos desarrollos de sistemas de información utilizando tecnologías mas recientes en relación a los sistemas en uso. De esta manera, se desarrolló la primera fase del proyecto OSIRIS (Optimización de Servicios Informáticos de las Redes de Información como Soporte de la Gestión Universitaria), proyecto que tuvo como uno de sus objetivos dar soporte informático a las actividades académicas y administrativas de la Institución. De esta fase se obtuvo los procedimientos y procesos que soportan la gestión de la Universidad.

Concientes del papel que cumplen los sistemas de información en el funcionamiento y desarrollo de múltiples labores de la Institución, las directivas de la Universidad Industrial de Santander aprobaron en Diciembre de 1999 las prácticas empresariales con el fin de aprovechar el recurso humano altamente calificado de los estudiantes de la Escuela de Ingeniería de Sistemas, utilizando este potencial en el desarrollo de sus sistemas de información, indispensables para el control y desarrollo de la gestión académica y administrativa.

Con el objetivo de ofrecer un soporte informático adecuado para la División de Recursos Humanos, la Universidad, en Octubre de 1998 inició el desarrollo del

Sistema de Información de la Gestión de Recursos Humanos con la definición del modelo de datos requerido para el sistema de información.

Inicialmente se construyeron los módulos de hoja de vida, de administración de personal y de administración de liquidación de pagos. El desarrollo fue orientado especialmente hacia el registro y consulta de información necesaria para los procesos de liquidación de pagos de las diferentes nóminas, procesos seleccionados con alta prioridad por acercarse el cambio de milenio y requerir una renovación en los sistemas existentes.

En la segunda fase, se trabajó en procesos especiales y en la generación de reportes de los módulos de administración de personal, administración de hoja de vida y administración de pagos y se desarrollaron los módulos de actividad académica y programas de desarrollo.

1.2.1 Sistema de Información de Administración de Recursos Humanos para la Universidad Industrial de Santander. Actualmente la Gestión del Recurso Humano cuenta con un sistema de información que atiende diferentes áreas de la administración del personal, con el cual se mantiene la información requerida para el personal vinculado, se realiza la ejecución de procesos complejos que involucran a todo el personal, se facilita el control de situaciones específicas referentes a docentes y administrativos y se simplifica el suministro de información a las diferentes unidades de la Universidad y organismos externos.

El Sistema de información permite mantener para el personal vinculado con la Institución, el registro histórico en línea de la información referente a la hoja de vida, información laboral, información de pagos, información de capacitación, dotación y registro de actividad docente. Los tipos de vinculación existente son:

- Planta de Personal
- Jubilados y Sustitutos

- Docentes de Cátedra
- Servicios Prestados
- Aprendices
- Auxiliares
- Becas de Sostenimiento

1.2.1.1 Generalidades. El sistema está diseñado con un enfoque parametrizado, el cual permite al usuario registrar la información necesaria para aplicar los cambios del entorno que afectan los procesos implementados.

El sistema dispone de un módulo de consulta que permite buscar información de los funcionarios por los conceptos existentes en la base de datos de acuerdo a los criterios de búsqueda suministrados, genera reportes y permite observarlos por pantalla y/o enviarlos a impresión.

El sistema cuenta con cinco módulos principales los cuales corresponden a la clasificación de las áreas de información más importante registradas para los funcionarios.

El módulo de Hoja de Vida presenta las siguientes características:

- Permite registrar información de datos personales, nacionalidades, idiomas, datos de familiares, estado civil y estudios.
- Registra información detallada de las situaciones administrativas de los funcionarios como: licencias, permisos, suspensiones, comisiones de estudio, comisiones de servicio, vacaciones y períodos sabáticos.

- Realiza el chequeo de vencimiento de reintegro de situaciones administrativas y de vencimiento en la presentación de informes a una fecha.
- Apoya los procesos de vacaciones colectivas, vencimiento y acumulación de vacaciones.
- Registra información de los procesos disciplinarios y evaluaciones de desempeño.
- Permite llevar el registro histórico de los diferentes puntajes asignados al personal de la Institución por los conceptos correspondientes para los regímenes vigentes.
- Permite el registro de los cambios de categoría profesional, ascensos y renovación de tenencia en el escalafón docente y permite realizar el chequeo de vencimiento de tenencia. También permite el registro de las promociones en la escala salarial para el personal no profesional.
- Dispone de reportes generales para personal con criterios para selección y con variado contenido de información; también dispone de reportes estadísticos a una fecha determinada y permite crear archivos planos para suministrar información a diferentes unidades
- Cuenta con un módulo especial diseñado para procesar la evaluación docente en el cual se generan los reportes comparativos detallados por docente.

El módulo de Administración de Personal presenta las siguientes características:

- Permite llevar el registro histórico de las vinculaciones de una persona en la Institución y los cargos desempeñados.

- Brinda control en línea en la asignación de puestos de trabajo por considerar la planta de personal con asignación individual.
- Mantiene el registro de información de reconocimiento de jubilados y sustitutos a la institución.
- Permite guardar información de los ofrecimientos generados para servicios prestados, de los contratos para aprendices SENA y las asignaciones de becas de sostenimiento y auxilias estudiantiles en cada semestre.
- Existe un mecanismo para imprimir las cartas de ofrecimiento para contratación de personal por servicios prestados.
- Dispone de un módulo que facilita el trámite de las solicitudes de docentes de cátedra a través de las diferentes unidades académico administrativas involucradas en la aprobación de la contratación y genera los contratos correspondientes.
- Mantiene para docentes de cátedra información sobre las asignaturas contratadas en cada tipo de contratación (tecnológica, salud, INSED, Instituto de lenguas y regionalización).
- Genera reportes detallados y resumidos sobre la planta de personal ejecutada que permite conocer la situación de la misma en una fecha determinada.
- Dispone de un módulo de generación de certificados laborales y constancias de trabajo para personal de planta, por servicios prestado, docentes y jubilados.

El módulo de Pagos presenta las siguientes características:

- Mantiene información básica requerida para la liquidación de pagos como: sueldos, descuentos, afiliaciones obligatorias y voluntarias, régimen salarial y de cesantías y formas de pago según el tipo de vinculación.
- Permite generar un reporte de chequeo de sueldos contra escala salarial o puntajes según el régimen salarial y tipo de empleado.
- Realiza los cálculos de porcentaje de retención en la fuente para personal de planta y de cátedra según el proceso asignado a la Institución.
- Mantiene un histórico de pagos realizados por liquidación y permite el registro de órdenes de pago y reintegros de pagos de los funcionarios, requeridos para generar los acumulados anuales.

El módulo de Liquidación de Pagos presenta las siguientes características:

- Realiza los procesos de liquidación de pagos(mensual, primas, cesantías, liquidación de contratos) según la vinculación de los funcionarios y la información registrada a la fecha del proceso.
- Genera para cada proceso de liquidación los listados y archivos requeridos. También registra la información financiera necesaria para realizar los pagos a empleados y a terceros por tesorería y/o fiducia.
- Genera los certificados de ingreso y retención anuales por cada tipo de nómina y los certificados de liquidación de cesantías anuales.
- Realiza el cálculo de los nuevos sueldos por empleado y calcula los ajustes correspondientes de los conceptos pagados.

- Realiza la proyección de futuros jubilados, calcula los pagos proyectados para planta de personal y jubilados, según los parámetros dados y realiza el registro financiero para la creación del presupuesto anual.

El módulo de Programas de Desarrollo contiene dos submódulos. El submódulo de Dotación presenta las siguientes características:

- Permite definir la dotación asignada teniendo en cuenta el género, los cargos y la planta de personal por tipo de dotación.
- Realiza la asignación de la dotación según el tipo seleccionado y asigna los elementos según condiciones dadas y considerando las tallas de los funcionarios que corresponden.
- Genera los reportes de dotación, para firmas y de costeo requeridos.

El submódulo de Capacitación presenta las siguientes características:

- Permite abrir cursos, programar horarios, reservar cupos para cargos específicos, registrar inscripciones, matricular y registrar la aprobación para personal vinculado.

El módulo de Actividad Académica presenta las siguientes características:

- Permite registrar los empleados representantes a convenios, proyectos, comités y a organismos internos.
- Permite registrar PADS por desempeño en cargos de administración y otras actividades.
- Consolida información de docencia directa y de investigación para control de PADS de cada profesor.

1.3 DESCRIPCIÓN DEL PROBLEMA

Las nuevas tecnologías de la información, han avanzado rápidamente hacia un nuevo marco de procedimientos en el que se da la importancia necesaria y correspondiente a la información, tomando esta no solo como el conglomerado de datos que surgen del desempeño laboral que se lleva a cabo en el interior de toda organización, sino que presenta la misma como esencial para el correcto desarrollo y objeto fundamental del ente organizacional.

Estas tecnologías brindan el soporte necesario y nuevas formas de manejo para cumplir con los objetivos de toda organización, presentándose organizada y actualizada para cualquier interesado en la misma.

La World Wide Web (comúnmente conocida como Web), es un avance significativo de las últimas décadas en el campo de las comunicaciones y en el manejo de información, con beneficios sorprendentes para cualquier organización, como por ejemplo, el mantener su información organizada y oportuna, además de brindar una amplia cobertura, al ser un medio de difusión a nivel mundial. Sin embargo, en muchas organizaciones no encontramos la correcta utilización de esta tecnología, o simplemente se encuentra parcialmente implantada. Es claro, que cada día es de mayor importancia la migración de procedimientos del papeleo tradicional hacia procedimientos digitales.

La Universidad Industrial de Santander ha iniciado este proceso mediante su sitio en la Web y la implantación de algunos procedimientos en línea.

El Área de Recursos Humanos mediante el presente proyecto, brindará soporte en línea al personal de planta vinculado en el trámite de las solicitudes administrativas y funcionalidades relacionadas con estas.

Para la correcta implementación de los procedimientos, la División de Servicios de Información proporcionó las facilidades y garantías necesarias para un desarrollo adecuado prestando la asistencia y ayudas necesarias, además de los recursos

hardware y software requeridos durante la ejecución del proyecto, mostrando así su interés por el beneficio de las dependencias Universitarias, con una visión de cada una como un elemento importante para el correcto funcionamiento de un todo, llamado Universidad Industrial de Santander.

1.4 OBJETIVOS

Mediante del correcto aprovechamiento de las nuevas tecnologías de la información se logrará brindar grandes beneficios en cuanto a comodidad, agilización y reducción de costos en todas las actividades que conciernen al procesamiento de dicha información, cubriendo la generación, almacenamiento, transmisión, manipulación y visualización de información.

La División de Recursos Humanos de la Universidad Industrial de Santander busca dinamizar la estructura organizacional mediante el uso de tecnologías apropiadas y la racionalización continua de sus procesos para el mejor desarrollo de las actividades de gestión docente y administrativa fundamentales de la Universidad.

Para apoyar esta misión, el presente proyecto desarrollará herramientas y servicios que apoyen el proceso correspondiente al manejo y control de solicitudes generadas por el personal docente y administrativo de la Universidad.

Estas herramientas y servicios permitirán manejar las solicitudes a través de la Web, brindando una buena posibilidad para el desarrollo de estos procedimientos, diferente al manejo de los mismos por medio del tradicional material impreso.

1.4.1 Objetivo general

Diseñar e implementar un Sistema de Información bajo tecnología Web, que apoye el trámite de las solicitudes docentes y administrativas propias del Área de Recursos Humanos de la Universidad Industrial de Santander.

1.4.2 Objetivos específicos

- Desarrollar y ofrecer en ambiente web un módulo para el trámite de situaciones administrativas propias de la gestión del Recurso Humano de la Universidad; presentando así, una alternativa que agilice y brinde mayor cobertura a los usuarios. Las situaciones administrativas a desarrollar son las siguientes:
 - 1) Permiso personal.
 - 2) Permiso por cita médica.
 - 3) Permiso por calamidad doméstica.
 - 4) Permiso para compensar.
 - 5) Permiso compensatorio.
 - 6) Permiso grupal.
 - 7) Licencias no remuneradas.
 - 8) Comisiones de estudio.
 - 9) Comisiones de servicio.
 - 10) Período sabático.
 - 11) Prórroga de solicitudes.
 - 12) Cancelación de solicitudes.
 - 13) Solicitud de modificación.

14) Solicitud de acumulación de vacaciones.

15) Solicitud de disfrute de vacaciones.

16) Registro de Informes de comisión de estudio.

17) Registro de Informes de período sabático.

18) Evaluación de informes.

- Diseñar e implementar un módulo de funcionalidades que complementen el trámite de las situaciones administrativas a desarrollar.

Las funcionalidades que se considerarán son las siguientes:

1) Registrar permisos causados.

2) Cancelar solicitudes en trámite de aprobación.

3) Cancelar solicitudes aprobadas.

4) Solicitar el reintegro antes de finalizar la situación aprobada.

5) Modificar los términos de una solicitud (objeto, fecha y hora de inicio y fecha y hora de fin).

6) Consultar información de apoyo para el trámite de cada solicitud.

- Realizar el desarrollo del proyecto siguiendo actividades correspondientes a la metodología de Proceso Unificado de Desarrollo de software.

- Desarrollar los módulos descritos en los objetivos anteriores utilizando: El lenguaje de programación Visual Basic 6.0 para el desarrollo de DLL's (Dynamic Link Library); los lenguajes de programación Web, ASP (Active Server Pages), VisualBasic Script y JavaScript para la creación de páginas Web dinámicas; y la utilización del sistema gestor de bases de datos INFORMIX.

1.5 IMPACTO Y VIABILIDAD

1.5.1 Impacto. La gran red de computadores personales y servidores, Internet, nos da la posibilidad de acceder a la información y servicios en ella contenidos, en cualquier momento, y desde cualquier punto del mundo en el que contemos con un medio de conexión, brindando así facilidades de cobertura y autonomía de espacios físicos que en muchos casos buscamos, o que ha de ser necesaria en algunos otros.

Las empresas de hoy en día buscan soportar sus procesos mediante la utilización de las nuevas tecnologías de la información con el fin de ampliar las posibilidades con que cuentan en cuestión de cobertura y autonomía, en lo que refiere a los procesos de gestión administrativa que surgen en su interior. Estos procesos no se limitarán a la infraestructura física de la entidad y al manejo de medios físicos de registro de información, presentando así una alternativa en el manejo de la información que se genera de sus situaciones administrativas.

En el caso en que un docente que se encuentre en comisión de estudios en el exterior y su fecha de reintegro esté próxima a cumplirse, puede recurrir al trámite de prórroga, vacaciones, licencias u otro tipo de solicitud para finalizar sus estudios. Una buena alternativa de llevar a cabo la solicitud seleccionada sería encender su equipo portátil, conectarse a la Web y navegar hasta el sitio de la organización a la que él esté vinculado, para así, diligenciar y luego en cualquier momento poder monitorear a través de la red, la solicitud para conocer el estado de la misma. Podrá conocer también el resultado del proceso una vez se concluya. De esta manera se agilizará este proceso, se brindará información en línea y oportuna del

trámite, que permita tomar decisiones respecto de la forma de terminar el estudio mencionado.

1.5.2 Viabilidad. El presente proyecto es desarrollado como Práctica Empresarial, una de las cuatro modalidades existentes para optar al título de Ingeniero de Sistemas, de acuerdo al reglamento estudiantil vigente. Dicha práctica es posible gracias al convenio que existe entre la Escuela de Ingeniería de Sistemas e Informática y la División de Servicios de Información. De acuerdo con este convenio, los estudiantes del programa de Ingeniería de Sistemas, pueden realizar su proyecto de grado en esta dependencia, la cual sirve como ente facilitador durante el desarrollo del mismo; proporcionando sus instalaciones, las herramientas computacionales existentes y los profesionales expertos en el manejo de los servicios informáticos institucionales con que cuenta actualmente la Universidad.

Los proyectos que se realizan con la División de Servicios de Información, buscan aprovechar de una mejor manera los recursos con los que la Universidad cuenta, para así colaborar en el mejoramiento de sus procesos, impulsando la innovación tecnológica y facilitando el uso de la información a todos los integrantes de la comunidad universitaria.

El hecho de estar optando por un título profesional, lleva a asumir este paso con firmeza y responsabilidad, aplicando todos los conocimientos adquiridos durante nuestra formación universitaria en la Escuela de Ingeniería de Sistemas e Informática. Estamos seguros de poder corresponder al compromiso adquirido y presentar una alternativa al manejo de ciertos procedimientos docente-administrativos generados al interior de las unidades de la Universidad, en pro de mejorar la gestión de los mismos.

Teniendo en cuenta todo lo expuesto anteriormente, podemos decir que el proyecto cuenta con bases firmes, con lo cual la Universidad asegura que se le entregará un buen producto, optimizando sus recursos y ofreciendo otra visión de gestión de procesos. Al mismo tiempo, permitirá a los autores obtener la experiencia necesaria antes de ingresar al sector profesional.

1.6 DESARROLLO DEL SISTEMA

Para el desarrollo del Sistema de Información Web para el trámite de solicitudes del Área de Recursos Humanos de la Universidad Industrial de Santander se utilizará un equipo de las siguientes características:

1.6.1 Hardware

- Procesador Pentium II 500 MHz o superior.
- Memoria RAM 192 MB o superior.
- Disco Duro 8 GB o superior.
- Tarjeta de Red.
- Monitor a color de 14”.
- Unidad de CD ROM 40X o superior.
- Unidad de Disquete 3 ½.
- Mouse.
- Teclado.
- Punto de Conexión a la red LAN.

1.6.2 Software

- Sistema Operativo Windows 2000 Server.

- Internet Information Server 5.0.
- Microsoft Visual Basic 6.0.
- Microsoft Visual InterDev 6.0.
- Macromedia Dreamweaver Ultadev 4.0.
- Macromedia Fireworks 4.0.
- Power Designer 7.0.0.
- Microsoft Internet Explorer 5.0.

2. FUNDAMENTACIÓN TEÓRICA

Con el fin de cumplir con cada uno de los objetivos descritos en el presente proyecto, se utilizaron diferentes tecnologías y una metodología guía en el desarrollo de software, las cuales son descritas a continuación.

2.1 GENERALIDADES ACERCA DE INTERNET

Internet es una red de computadores comunicados entre sí globalmente con el propósito de transferir información mediante la utilización del conjunto de protocolos¹ conocido como TCP/IP (Transmission Control Protocol / Internet Protocol), los cuales interpretan la información y la describen en paquetes de datos que serán transferidos.

Los protocolos TCP/IP permiten encaminar la información por diferentes rutas de transferencia si se encuentra un malfuncionamiento en uno de sus nodos, brindando así confiabilidad en el envío y recepción de la información empaquetada.

Internet se basa en la arquitectura cliente-servidor, en la cual existen dos clases de computadores: el cliente o el computador que realiza una solicitud; y el servidor o el computador que responde a la solicitud.

El actual crecimiento acelerado de Internet se debe a la rápida acogida de los navegadores (browsers) y a la tecnología World Wide Web (WWW) ampliamente utilizada por los usuarios de Internet. Esta herramienta es un gran conjunto de páginas Web que son visualizadas mediante la utilización del navegador, el cual permite un fácil acceso a la información contenida en la red haciendo el papel de cliente, el cual realiza la comunicación con el servidor mediante la utilización del protocolo HTTP o Hypertext Transfer Protocol.

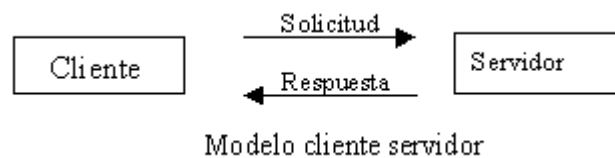
¹ En computación, un protocolo es Un conjunto de reglas formales que describen como transmitir datos, especialmente a través de una red. <http://www.hyperdictionary.com/dictionary/protocol>.

2.1.1 Protocolo HTTP. Los computadores en el Word Wide Web utilizan el protocolo de aplicación HTTP (Hypertext Language Protocol) con el fin de comunicarse entre sí para intercambiar Información (texto, imágenes, sonido, video o cualquier otro tipo de dato multimedia). Para tal fin el protocolo HTTP provee un conjunto de instrucciones las cuales permiten el intercambio preciso de esta información.

Cuando se realiza una conexión a un servidor WEB solicitando una página Web para ser descargada, se hace uso de este protocolo. De este modo, HTTP es el estándar que habilita a cualquier navegador para conectarse a cualquier servidor, en cualquier lugar del mundo.

La comunicación entre el cliente (navegador) y el servidor, implica una solicitud enviada por el cliente a la cual el servidor dará respuesta (solicitud y respuesta).

Figura 1. Comunicación entre cliente y servidor.



Cualquier transacción http que se da entre un cliente y un servidor consta del siguiente esquema básico:

- Conexión: El cliente establece una conexión con el servidor.
- Petición: El cliente envía una petición al servidor.
- Respuesta: El servidor envía al cliente la respuesta.
- Cierre: Ambas partes cierran la conexión.

Esto quiere decir, que cada transacción se realiza bajo la arquitectura cliente-servidor y consiste de tres partes principales:

- Una línea de solicitud o respuesta
- Cabecera de información
- El cuerpo del mensaje (BODY)

La línea de solicitud (request line) se encuentra conformada por tres piezas de información: Un comando http que se conoce como método, la URL (Uniform Resource Location) o Localizador de Recurso Uniforme del archivo que el cliente está solicitando y la versión de http que se está utilizando.

La URL se encuentra conformada por cuatro partes: el servicio o protocolo que se utiliza, el nombre de dominio o la dirección IP del servidor, la ubicación del documento y el nombre del mismo.

Ejemplo:

`http://www.uis.edu.co/web/index.jsp`

Servicio o Protocolo : http
Nombre de dominio : www.uis.edu.co
Ubicación documento : /web/
Nombre documento : index.jsp

Entre los métodos más destacados de una línea de solicitud o respuesta, podemos encontrar el método GET, el cual se encarga de la recuperación de cualquier tipo de información, el método HEAD que únicamente indica información de cabecera HTTP y el método POST el cual indica los datos que harán parte del cuerpo del mensaje HTTP.

2.2 ARQUITECTURA DE INFORMACIÓN

2.2.1 Modelo Cliente / servidor. Este es un modelo estándar para la ejecución de aplicaciones de redes y para sistemas operativos distribuidos. Este modelo, puede verse como entidades lógicas diferentes que desarrollan determinada función sobre una red para llevar a cabo una tarea.

El término servidor se aplica a cualquier programa capaz de ofrecer servicios que pueden ser obtenidos a través de la red. El cliente suele ser un programa ejecutado por un usuario para solicitar a un servidor algún servicio o recurso que se encuentre alojado en una máquina diferente.

Figura 2. Modelo Cliente/Servidor.



Básicamente en el modelo cliente / servidor existe alguien que ofrece algo (servidor) y alguien quien solicita algo (cliente).

Entre las características mas importantes con que cuenta un sistema cliente / servidor se encuentran²:

- *Servicio*: un esquema cliente / servidor puede verse como una relación entre procesos corriendo o ejecutándose en máquinas separadas, el servidor es un proveedor de servicios y el cliente es un consumidor de servicios.
- *Recursos compartidos*: Un servidor puede atender muchos clientes al mismo tiempo y regular el acceso de los mismos a los recursos compartidos.

²Tomado de: <http://dis.eafit.edu.co/cursos/st133/material/introduccion/introduccion-cliente-servidor.html>

- *Mezclados:* El software ideal en el esquema cliente / servidor es independiente de la plataforma de hardware y del sistema operativo utilizado.
- *Basados en el intercambio de mensajes:* un esquema cliente servidor debe estar acoplado e interactuar en un mecanismo de paso de mensajes. Los mensajes son usados para solicitar y recibir un servicio.
- *Escalabilidad:* Los sistemas cliente / servidor deben escalarse tanto horizontal (adicionar o retirar estaciones de trabajo) como verticalmente (migrar a estaciones más grandes y rápidas o sistemas distribuidos).
- *Integridad:* El código y los datos en el servidor están administrados de forma centralizada, lo cual ofrece integridad y seguridad a los datos.

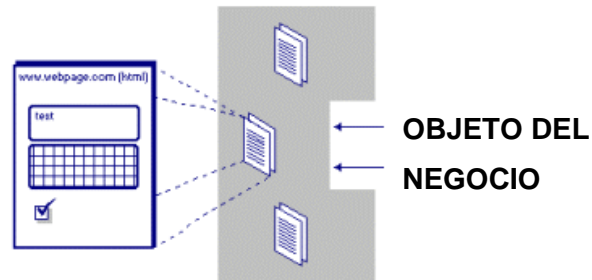
2.2.2 Arquitectura de la aplicación. La arquitectura es un término usado al momento de diseñar aplicaciones, particularmente del tipo Cliente-Servidor. Esta arquitectura se refiere a la manera de diseñar tanto física como lógicamente la aplicación.

En el diseño físico se especifica exactamente donde se encontrarán las piezas de la aplicación (como discos, ejecutables, cable de red y computadoras) y en el diseño lógico o conceptual se especifica la estructura de la aplicación y sus componentes.

2.2.2.1 Arquitectura de tres capas. La arquitectura de tres capas es una arquitectura que separa los componentes de una aplicación en tres capas lógicas.

- **Capa de presentación.** Esta capa es la encargada de presentar datos al usuario y opcionalmente permitir la manipulación de los mismos, como en la entrada de nuevos datos.

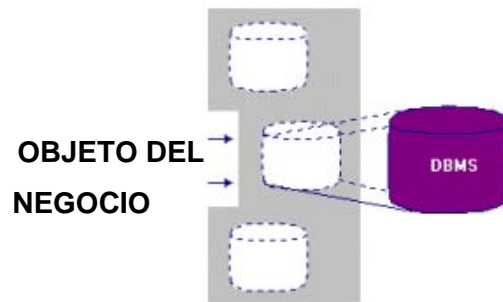
Figura 3. Capa de presentación.



Para esta arquitectura, la capa de presentación se comunica únicamente con la capa lógica del negocio, nunca con la capa de acceso a bases de datos.

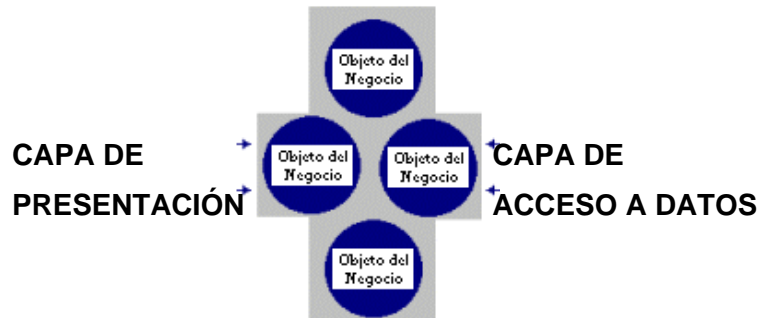
- **Capa de acceso a bases de datos.** Es la capa dedicada a la información del negocio. Es responsable de almacenar, recuperar y mantener los datos, garantizando la consistencia de los mismos en un ambiente distribuido.

Figura 4. Capa de acceso a datos.



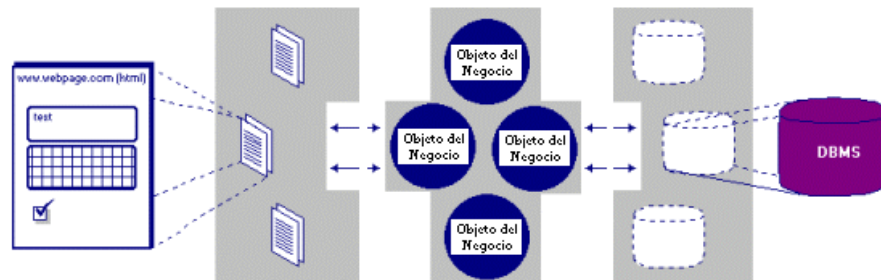
- **Capa lógica del negocio.** Esta capa almacena componentes que encapsulan las reglas del negocio.

Figura 5. Capa del negocio.



Sirve como puente entre la capa de presentación y la capa de acceso a bases de datos aislando al usuario de la interacción directa con la base de datos.

Figura 6. Arquitectura de tres capas.



Ventajas de la arquitectura de tres capas:

- Tratar por separado los componentes lógicos de una aplicación garantiza que la aplicación sea de fácil administración.
- Es sencillo modificar o reemplazar cualquiera de las capas sin afectar las otras.
- Disminuye las grandes cargas de procesamiento para los clientes.
- La información contenida en el servidor de base de datos se puede ocultar a los clientes.

2.3 LENGUAJES DE PROGRAMACIÓN UTILIZADOS

2.3.1 HTML: HyperText Markup Language. El HTML es el lenguaje de interpretación de marcas o etiquetas con el cual se desarrollan los hipertextos de la información contenida en Internet.

Un documento de texto puede ser enriquecido por medio de este lenguaje, añadiendo al mismo:

- Una estructura de presentación para el texto. Por ejemplo, los títulos, los encabezados o listas de elementos.
- Los estilos del texto: enfatizado y citas entre otros.
- Contenido de objetos multimedia: imagen, video y sonido.
- Conexiones realizadas por medio de enlaces hipertextuales o partes del documento por medio de las cuales podemos direccionarnos a otros sitios de la Web.

2.3.2 Lenguajes de scripts. Los lenguajes de scripts son lenguajes de programación que permiten manipular, adaptar y modificar una página Web.

Los lenguajes de scripts poseen características que los diferencian de los lenguajes de programación más normales:

- Funcionan únicamente bajo la aplicación para la que fueron creados (los navegadores Netscape Navigator e Internet Explorer)
- Son lenguajes interpretados, no compilados.

- Tratan de mantener una programación sencilla y cómoda al no poseer un control estricto de tipos para las variables.

Entre los usos más frecuentes que podemos encontrar en el momento de utilizar un lenguaje de scripts podemos encontrar:

- La validación de los datos de una página, en el momento previo de ser enviados.
- Asignar y actualizar información en los campos presentes en los formularios de las páginas.
- Realizar procesos que sean independientes al servidor, es decir, procesos que no requieran de la utilización de información centralizada.

2.3.2.1 JavaScript. Es un lenguaje compacto que se basa en objetos, diseñado para el desarrollo de aplicaciones cliente-servidor que permite la realización de aplicaciones de propósito general a través de Internet.

JavaScript permite realizar efectos sobre las páginas Web para presentar así el contenido de las mismas dinámicamente o ejecutar instrucciones en respuesta a los eventos del usuario.

2.3.2.2 VBScript. Es una pequeña versión del lenguaje de programación Visual Basic de Microsoft. Es un lenguaje para el desarrollo de scripts cuya funcionalidad es la manipulación y modificación de páginas Web.

Utilizando VBScript del lado del servidor no existe la preocupación de cómo el navegador del cliente interpreta el código, pues el servidor se encarga de procesar el código incrustado en cada página antes de ser transmitido al cliente, así el cliente recibe únicamente una página con código HTML. En contraste con esto, al utilizar VBScript del lado del cliente logramos transferir la carga de procesamiento del

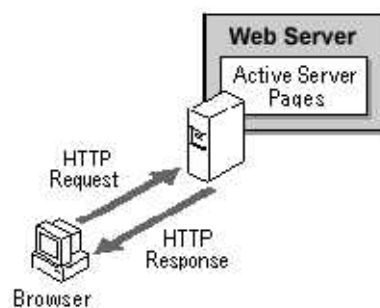
código al navegador del cliente con el fin de reducir esta carga de procesamiento en el servidor.

2.3.3 Active Server Pages. ASP es el acrónimo de Active Server Pages, es decir, Páginas Activas de Servidor. Las páginas ASP son una tecnología propiedad de Microsoft, que sirve para crear y ejecutar aplicaciones Web interactivas, dinámicas y de alto rendimiento del lado del servidor (entorno server-side scripting).

La ejecución de una página ASP comienza cuando el cliente solicita al servidor Web un archivo ASP (de extensión .asp) a través del explorador. El servidor lee el archivo que se ha solicitado, ejecuta las instrucciones VBScript que se encuentran en él, y envía los resultados al cliente en HTML para ser desplegados por el navegador, como se aprecia en la figura 7.

Las secuencias de comandos contenidas en una página ASP son transparentes al usuario, es decir, permanecen ocultas a él, dado que estos solo reciben el resultado de la ejecución en formato HTML. Razón por la cual, el problema de si el cliente puede o no ejecutar sentencias de comandos es solucionado ya que el código HTML es interpretable por cualquier explorador.

Figura 7. Solicitud y respuesta de una página ASP.



2.3.4 Visual Basic Y Dll's

2.3.4.1 Visual Basic. Visual Basic es un lenguaje de programación que posee una interfaz gráfica de fácil manejo y comprensión, con la característica de poder incorporar de una manera sencilla controles visuales, imágenes y multimedia en nuestras aplicaciones.

La base de este lenguaje se encuentra en el lenguaje Basic y en la programación orientada a objetos. La palabra "Visual" hace referencia a la metodología de la que se hace uso para crear la interfaz gráfica de la aplicación que se desarrollará, utilizando los controles ya prefabricados de Visual Basic.

La palabra "Basic" hace referencia al lenguaje de programación Basic (Beginners AllPurpose Symbolic Instruction Code) o Código de Instrucciones Simbólicas de uso General para Principiantes. Visual Basic se presenta como el producto evolutivo de este lenguaje el cual ahora cuenta con un conjunto extenso de instrucciones, funciones y palabras claves.

2.3.4.2 Dll's. Dll (Dynamic Link Library) o "Biblioteca de vínculos dinámicos". Es un componente de software que es utilizado por una aplicación en tiempo de ejecución.

La Dll contiene funciones que se pueden llamar desde aplicaciones u otras Dll's. Estas son utilizadas por los desarrolladores para poder reciclar el código y aislar las diferentes tareas. Las Dll's no pueden ejecutarse directamente, es necesario llamarlas desde un código externo.

2.4 BASES DE DATOS

2.4.1 Aspectos Generales de Bases de Datos. Una base de datos puede ser definida como una colección automatizada de datos operacionales almacenados, que sirven a las necesidades de múltiples usuarios dentro de una o más organizaciones y que varían con el tiempo, tanto en tamaño como en valores.

Una base de datos debe cumplir con las características planteadas según el modelo ACID:

- **Atomicidad:** Una transacción debe ser atómica. Cuando una transacción está compuesta por un número cualquiera de eventos, el sistema los debe considerar como una única operación, la cual puede tener éxito y en tal caso se hacen permanentes los cambios generados por cada evento componente de la transacción; o fracasar, en este caso el sistema queda en el mismo estado, como si la transacción nunca hubiera ocurrido.
- **Consistencia:** Todos los cambios provocados por la transacción deben dejar al sistema en un estado correcto.
- **Aislamiento:** Las transacciones que se ejecutan concurrentemente no se ven afectadas unas con otras. Si una transacción A cambia un sistema de un estado E1 a un estado E2, una transacción B siempre verá al sistema en un estado E1 o E2, pero nunca en un estado intermedio.
- **Durabilidad:** Si una transacción es terminada en forma exitosa los efectos serán permanentes.

2.4.2 Ventajas de las Bases de Datos. La mejor forma de entender la naturaleza y las características de las bases de datos es conocer las ventajas que proporcionan su utilización. Entre otras se tiene³:

- **Reducción de la redundancia:** En los archivos tradicionales, las aplicaciones repiten gran cantidad de datos ya sea al nivel de archivos o de campos, lo cual se puede ver reflejado en el desperdicio de espacio de almacenamiento, inexactitud de datos e inconsistencias de los mismos.

³ CARCAMO SEPÚLVEDA, José. Bases de datos relacionales: un enfoque practico de diseño. 1997.

- **Evitan las inconsistencias:** Consiste en prevenir la desactualización de la información que tiene más de una ocurrencia en la Base de Datos.
- **Compartir datos:** Esto se presenta cuando se tienen datos en diferentes archivos y se hace necesaria una consulta simultánea de estos; es aquí donde la base de datos logra que se puedan compartir los datos fácilmente ahorrando esfuerzos de programación.
- **Seguimiento de estándares:** Mediante un control centralizado de las bases de datos, se puede asegurar que los diferentes estándares de aplicación puedan ser seguidos en la representación de los datos.
- **Flexibilidad:** Ventaja de almacenar los datos en forma más flexible, debido a que está definida de modo externo a los programas que harán uso de ella.
- **Seguridad:** El sistema de administración de la base de datos permite la definición de niveles de acceso y de líneas de autorización para el acceso a los datos.
- **Integridad:** El problema de la integridad está relacionado con la exactitud de la información de la base de datos. Si los datos difieren, producirán resultados inconsistentes.
- **Independencia de los datos:** Una aplicación es independiente de los datos, cuando es posible cambiar la estructura de almacenamiento o el método de acceso a los datos sin afectar drásticamente la aplicación.

2.4.3 Gestor de Bases de Datos. El Sistema Gestor de Bases de Datos (SGBD) es un conjunto de programas, procedimientos y lenguajes que proporcionan a los usuarios las herramientas necesarias para interactuar con una base de datos. Así, el SGBD actúa como un intermediario entre los usuarios y los datos. Debe manipular los datos permitiendo a los usuarios insertar, suprimir, modificar y consultar datos de

la base de datos y por último, debe permitir usar la base de datos, dando una interfaz adecuada a cada tipo de usuario.

2.4.4 Motor de bases de Datos. El motor de base de datos es el encargado de realizar las consultas, modificaciones y procedimientos de forma transparente al usuario.

- **Informix.** El manejador de bases de datos Informix cuenta con una alta disponibilidad de ejecución en el procesamiento de transacciones en línea, manejando recursos como memoria distribuida y disco I/O. Además, cada una de las tablas que contienen los datos pueden extenderse a múltiples discos, liberando al administrador de imponer restricciones por limitación en el almacenamiento de datos.

Informix brinda a sus usuarios las facilidades suficientes para definir y manejar bases de datos, mediante un entorno para el manejo de la información sin necesidad de programación.

2.5 ADO – (MICROSOFT ACTIVEX DATA OBJECTS)

ActiveX Data Objects (ADO) es una tecnología que ha sido diseñada para permitir a los programadores desarrollar contenido activo Web.

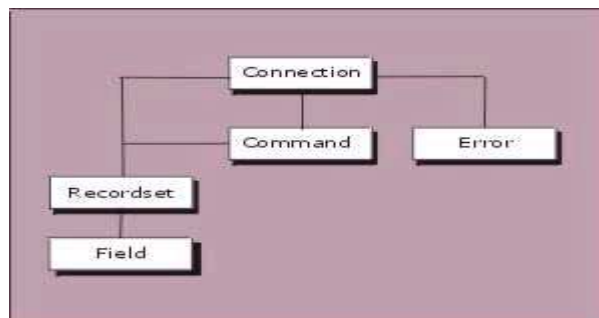
ADO es ampliable y de fácil uso para agregar acceso a bases de datos a las aplicaciones Web y puede ser utilizado para escribir secuencias de comandos compactas y escalables que conecten con bases de datos compatibles con Open Database Connectivity (ODBC, Conectividad abierta de bases de datos) y orígenes de datos compatibles con OLE DB.

ADO es neutral al lenguaje, esto quiere decir que puede ser usado desde Visual Basic, Delphi, Visual C++, Java , Java Script ó VBScript.

ADO usa ODBC para el acceso a bases de datos; esto implica que podemos cambiar la tecnología de la base de datos y si mantenemos la misma estructura de datos, las aplicaciones desarrolladas con ADO pueden seguir funcionando sin cambiar el código.

ADO está formado por varios objetos organizados de forma jerárquica (cada uno de ellos con sus métodos y propiedades específicos).

Figura 8. Objetos que conforman ADO.



- **Connection:** Proporciona una conexión a una base de datos ODBC desde una página ASP. Esta conexión nos permitirá efectuar las operaciones que deseemos sobre la base de datos.
- **Error:** Es una colección en la que se almacenan los posibles errores del objeto Connection.
- **Command:** Representa un comando SQL que se ejecuta contra la base de datos declarada en el objeto Connection. Si el resultado de ese comando es un conjunto de datos, estos se almacenarán en un objeto de tipo Recordset.
- **Recordset:** Representa una tabla o el resultado de una consulta ejecutada contra la base de datos. Va a ser nuestro interface natural contra la base de datos. Los datos se presentarán en filas y columnas.

- **Field:** El objeto Field representa la información relativa a un campo de un Recordset.

2.6 SERVIDOR DE SERVICIOS DE INTERNET : INTERNET INFORMATION SERVER

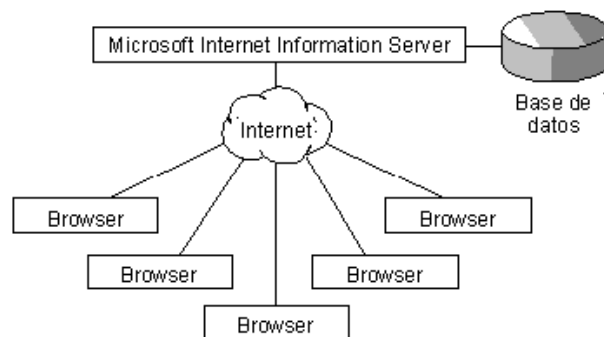
Internet Information Server (IIS) es un servidor web que permite a sus usuarios enviar y recibir información constituyéndose como la base esencial de un sistema de información destinado a la utilización de los servicios de Internet.

IIS abarca un conjunto de herramientas administrativas de fácil uso que permiten realizar el control de sitios Web, FTP, SMTP (correo saliente) y servicio de noticias. Además, dispone del soporte necesario para la creación y mantenimiento de contenidos dinámicos a través de las Active Server Pages (ASP).

IIS realiza el acceso a las bases de datos como se muestra en la figura 9.

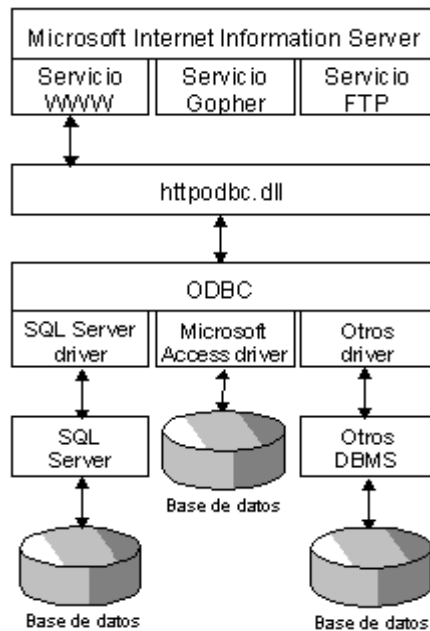
- Los clientes browser envían sus peticiones al servidor Web a través de http.
- El servidor Web responde con un documento en formato HTML.
- El acceso a la base de datos es realizado por el componente IDC (Internet Data Base Conector) del Internet Information Server. El IDC es un programa DLL/ISAPI que utiliza ODBC para el acceso a las bases de datos.

Figura 9. Acceso a bases de datos con IIS.



En la siguiente figura se muestran los componentes para conectarse a bases de datos desde IIS.

Figura 10. Conexión a bases de datos con IIS.



2.7 SISTEMAS DE INFORMACIÓN

Un sistema de información es un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en un entidad o institución.

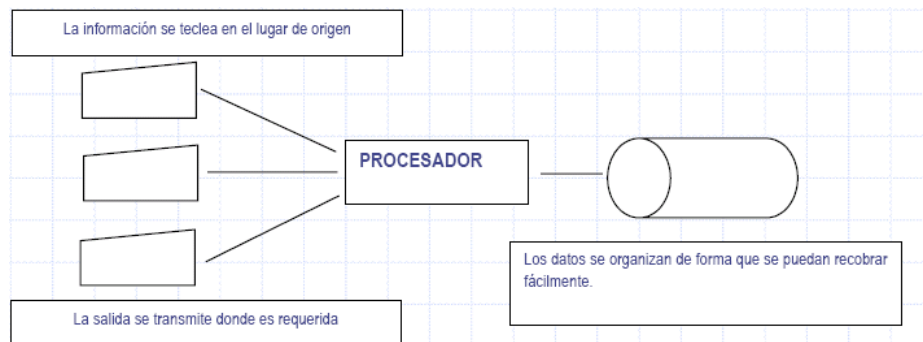
Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

- **Entrada de Información:** Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información.
- **Almacenamiento de información:** El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que

a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos.

- **Procesamiento de Información:** Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida.
- **Salida de Información:** La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior.

Figura 11. Diagrama de Sistema de Información.



2.8 METODOLOGÍA DE DESARROLLO

Una metodología de desarrollo tiene como misión transformar los requisitos de un usuario (personas, compañías, organizaciones u otro elemento software) en un sistema o herramienta, conformada por componentes software interconectados a través de interfaces bien definidas.

2.8.1 El proceso unificado de desarrollo de software⁴. Para el buen desarrollo de un sistema software específico, se necesitan conocer todas las acciones que

⁴ Mayor información: JACOBSON Ivar, BOOCH Grady, RUMBAUGH James. El proceso Unificado de Software.

éste llevará a cabo y que proporcionarán al usuario un resultado importante, lo que en esta metodología se conoce como Casos de Uso (CU). Un diagrama de CU representa cada una de las acciones y funcionalidades que proporciona la herramienta software, brindando un resultado de interés para cada uno de sus usuarios. El conjunto de estos casos de uso constituyen el modelo de casos de uso, que representa al sistema en sí con toda su funcionalidad.

El proceso unificado, se basa en tres grandes características:

- Dirigido por casos de uso.
 - Centrado en la arquitectura.
 - Iterativo e Incremental.
-
- **Dirigido por casos de uso:** Debido a que el sistema se debe centrar en la importancia del desarrollo para el usuario, es decir, cubriendo sus requerimientos y no en términos de cómo el sistema realizará sus funciones.
 - **Centrado en la arquitectura:** Debido a que permite tener una imagen completa antes de comenzar el proceso de construcción, esto se traduce en beneficios como la reducción de tiempo en la corrección de fallas y faltas.
 - **Iterativo e incremental:** El trabajo de desarrollo se divide de manera planeada en partes más pequeñas llamadas iteraciones lo cual genera progresivamente un incremento en el proyecto total, permitiendo una comprensión creciente de los requerimientos pasando a través de todas sus fases:

2.8.1.1 Fase de Inicio. El objetivo de la fase de inicio es desarrollar el análisis del negocio hasta el punto necesario para justificar la puesta en marcha del proyecto. Los objetivos específicos de esta fase son:

- a) Definir el ámbito y los límites del sistema.

- b) Resolver ambigüedades en los requisitos básicos.
- c) Determinar una arquitectura candidata para el sistema.
- d) Mitigar los riesgos críticos del proyecto que afecten su viabilidad.

2.8.1.2 Fase de Elaboración. El objetivo principal de la fase de elaboración es definir una arquitectura estable para guiar el sistema a lo largo de su construcción. Sus objetivos específicos son:

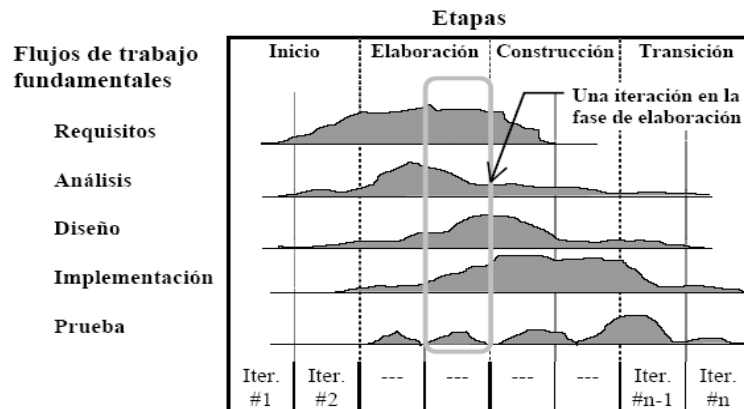
- a) Realizar una descripción de la arquitectura que cubra la funcionalidad del sistema.
- b) Recopilar los casos de uso para aproximadamente el 80 por ciento de los requisitos.
- c) Identificar los riesgos significativos que pueden perturbar los costos y planificaciones de fases posteriores.
- d) Preparar un informe de planificación y costeo del proyecto.

2.8.1.3 Fase de Construcción. El objetivo general de esta fase es alcanzar un producto listo para ser distribuido como versión beta y ser sometido a pruebas. Los objetivos específicos de esta fase son:

- a) Extender la identificación, descripción y realización a todos los casos de uso.
- b) Finalizar la fase de análisis.
- c) Mantener la integridad de la arquitectura, modificándola cuando sea necesario.
- d) Planear preliminarmente la fase de Transición.

2.8.1.4 Fase de transición. Cada una de estas fases concluye con un hito bien definido donde deben tomarse decisiones respecto al proyecto, como la reestructuración del cronograma de trabajo y se divide a su vez en iteraciones.

Figura 12. Flujos y fases del Proceso Unificado de Desarrollo de Software.⁵



Cada iteración sigue la estructura de un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba.

Un cuadro que reúne las actividades y productos de cada una de las fases del Proceso Unificado de Desarrollo de Software es presentado en el libro "Planeación de Proyectos. Un enfoque para Ingeniería de Sistemas. GOMEZ FLOREZ. Luis Carlos".

⁵ Basado en: RUMBAUGH James, BOOCH Grady, JACOBSON Ivar, El Proceso Unificado de Desarrollo de Software. Adison Wesley. Madrid 2000. pag. 11.

2.9 UML: LENGUAJE UNIFICADO DE MODELADO⁶

El Lenguaje Unificado de Modelado (UML: Unified Modeling Language) es un lenguaje gráfico desarrollado para la especificación, visualización, construcción y documentación de cada una de las partes que comprende el desarrollo de software.

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases y los objetos entre otros, hasta la implementación y configuración con los diagramas de despliegue.

2.9.1 Objetivos del UML

- UML es un lenguaje de modelado de propósito general. Está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.

⁶ Mayor información: Sinan Si Alhir. "UML in a Nutshell : A Desktop Quick Reference". O'Reilly & Associates, Inc., 1998. "UML Gota a gota". FOWLER, MARTIN. Addison Wesley Longman de México, S.A. de C.V. México 1999. Primera Edición.

2.9.2 Diagramas de UML. Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar.

Tabla 1. Diagramas presentes en UML⁷.

AREA	VISTA	DIAGRAMAS	CONCEPTOS PRINCIPALES
Estructural	Vista estática	Diagramas de clases	Clase, asociación, generalización, dependencia, relación, interfaz.
	Vista de casos de uso	Diagramas de casos de uso	Caso de uso, actor, asociación, extensión, generalización.
	Vista de implementación	Diagramas de componentes	Componente, interfaz, dependencia, realización.
	Vista de despliegue	Diagramas de despliegue	Nodo, componente, dependencia, localización.
Dinámica	Vista de estados de máquina	Diagramas de estados	Estado, evento, transición, acción.
	Vista de actividad	Diagramas de actividad	Estado, actividad, transición, determinación, división, unión.
	Vista de interacción	Diagramas de secuencia	Interacción, objeto, mensaje, activación.
		Diagramas de colaboración	Colaboración, interacción, rol de colaboración, mensaje.
Administración o gestión de modelo	Vista de gestión de modelo	Diagramas de clases	Paquete, subsistema, modelo.
Extensión de UML	Todas	Todos	Restricción, estereotipo, valores, etiquetados.

⁷ Tomado de: <http://www.creangel.com/uml/diagramas.html>

2.9.2.1 Diagrama de Casos de Uso. Modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.

2.9.2.2 Diagrama de Clases. Muestra las clases que componen el sistema y cómo se relacionan entre sí.

2.9.2.3 Diagrama de Objetos. Muestra una serie de objetos (instancias de las clases) y sus relaciones. A diferencia de los diagramas anteriores, estos diagramas se enfocan en la perspectiva de casos reales o prototipos.

2.9.2.4 Diagrama de Secuencia. Enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.

2.9.2.5 Diagrama de Colaboración. Igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.

2.9.2.6 Diagrama de Estados. Se utiliza para analizar los cambios de estado de los objetos, muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.

2.9.2.7 Diagrama de Actividades. Es un caso especial del diagrama de estados, simplifica el diagrama de estados modelando el comportamiento mediante flujos de actividades y muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

2.9.2.8 Diagrama de Componentes. Muestra la organización y las dependencias entre un conjunto de componentes. Se usan para agrupar clases en componentes o módulos.

2.9.2.9 Diagrama de Despliegue (o implementación). Muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo. Se utiliza para

identificar Sistemas de Cooperación. Durante el proceso de desarrollo el equipo averiguará de qué sistemas dependerá el nuevo sistema y que otros sistemas dependerán de él.

3. FASE DE INICIO

Durante la Fase de Inicio se busca aclarar la situación problema, con el fin de plantear en la siguiente Fase un primer esbozo de la arquitectura del sistema software requerida para abordar la problemática planteada.

La Fase de Inicio se ha desarrollado en una sola iteración; previo a esta se ha hecho una investigación preliminar, con la cual se han identificado los requisitos iniciales para el sistema. Al finalizar la Fase de Inicio se determinó una arquitectura candidata para el sistema a desarrollar.

En el flujo de requisitos se especifica una lista de funcionalidades del sistema, se proponen los posibles actores, se identifica y se presenta un modelo con los casos de uso encontrados en esta Fase, con su respectiva descripción.

Durante el flujo de análisis se identifican los primeros paquetes de análisis a partir de los casos de uso descritos durante el flujo de recopilación de requisitos. Además, se identifican los posibles paquetes de servicio y se establece la dependencia entre dichos paquetes.

3.1. FLUJO DE REQUISITOS

El sistema que se desarrollará tendrá como objetivo principal presentar una alternativa al manejo actual de ciertos procedimientos del personal de planta vinculado con la Universidad. Dicha alternativa se basa en el trámite de solicitudes para estos procedimientos mediante el sistema.

Actualmente, las solicitudes (documentos de solicitud) de los procedimientos relacionados en este proyecto, llegan a la División de Recursos Humanos de la Universidad, en la cual se realiza el proceso de registro en el Sistema de Información. Es decir, que las solicitudes que llegan hasta este punto, ya fueron

aprobadas y por lo tanto no se tiene un control de seguimiento a todo el proceso que generó la aprobación.

Mediante el Sistema de Información propuesto, se busca que desde el mismo momento de la generación de una solicitud, se conozca todo su trámite, incluyendo las observaciones y/o sugerencias realizadas por las personas involucradas en el proceso.

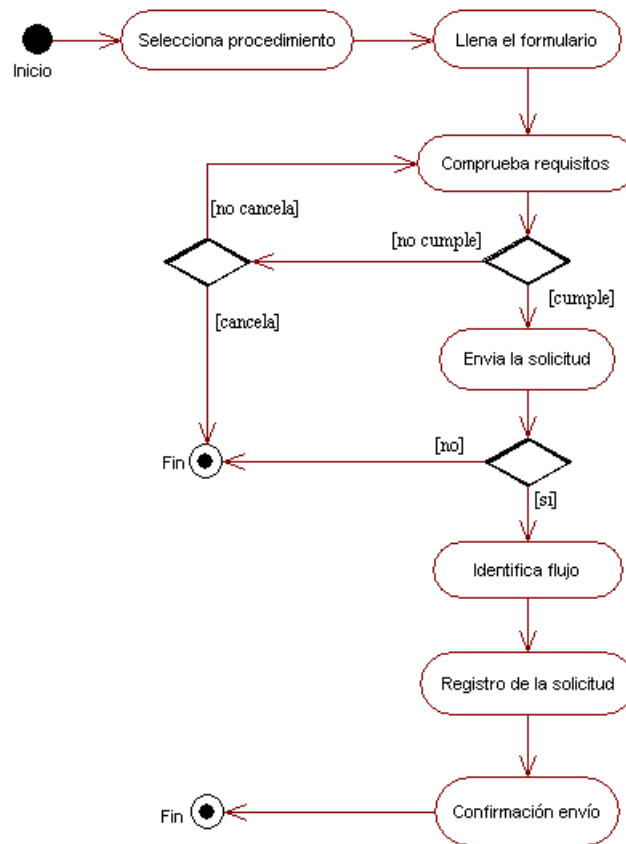
Los procedimientos que se tendrán en cuenta para el desarrollo del sistema son:

- 1) Permisos.
- 2) Licencias No remuneradas.
- 3) Comisiones de Servicio.
- 4) Comisiones de Estudio.
- 5) Período Sabático.
- 6) Vacaciones.
- 7) Informes de Comisiones y Período Sabático.

De manera muy general, las solicitudes que se generan en los seis primeros procedimientos siguen los estados de: Generación, evaluación, aprobación, vigencia y reintegro. Teniendo en cuenta, que el reintegro se puede dar anticipadamente o cuando se haya vencido a la fecha del registro.

A continuación, la figura 13 presenta el diagrama de las actividades que sigue el estado de generación de la solicitud.

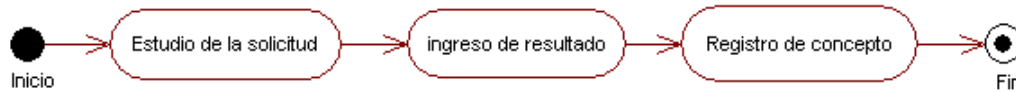
Figura 13. Diagrama de actividades para la generación de solicitudes.



El estado de generación de la solicitud, comienza cuando la persona selecciona un procedimiento. Seguido a esto se llega a una actividad de llenado del formulario correspondiente, la cual puede incluir cierto tipo de validaciones para los campos del mismo. Posteriormente se procede a validar los datos que se ingresaron, así como los requisitos del procedimiento; si no se cumple con éstos, el sistema no deja avanzar en el proceso de generación de la solicitud. Por el contrario, si se cumple con los requisitos, se llega a una actividad de envío, en la cual la persona determina o no el envío de la solicitud. Si se decide el envío de la solicitud, se pasa a la actividad de identificar el flujo de actividades que la solicitud va a tomar para su aprobación; se procede a hacer el registro de la solicitud en el sistema y finalmente se presenta una confirmación de la acción que se llevó a cabo.

Durante el estado de evaluación de una solicitud, se siguen las actividades que muestra la figura 14.

Figura 14. Diagrama de actividades para la evaluación de solicitudes.



Este estado comienza con un estudio de la solicitud por parte de la persona encargada de realizar dicha actividad, la cual determina la conveniencia o no de la misma. Seguidamente, se ingresa el resultado del estudio (que puede ser el rechazo o la aprobación) y opcionalmente las observaciones que se puedan hacer a la solicitud. Además se pasa a la siguiente actividad de acuerdo al flujo inicial establecido para la solicitud. Finalmente, éstos datos son registrados en el sistema.

Cuando la actividad que se realiza es de aprobación, se tiene una connotación diferente; esto debido a que con la aprobación de la solicitud se finaliza el trámite como tal de la misma. Teniendo en cuenta lo anterior, la vigencia de una solicitud se da cuando comienza el período aprobado.

Para dar por terminada una solicitud, se requiere del registro del reintegro de la persona a su lugar de trabajo.

A continuación se presenta un listado de funcionalidades y un diagrama de casos de uso, resultado de la recopilación de información llevada a cabo en la investigación preliminar, previa al comienzo de esta Fase.

3.1.1 Listado de Funcionalidades

- Iniciar cualquiera de los procedimientos mencionados anteriormente.
- Consultar los estados de una solicitud en trámite de aprobación.

- Mostrar todas las solicitudes pendientes para un actor que ingrese al sistema.
- Otorgar visto bueno (Vo.Bo.), aprobar o rechazar una solicitud, según los derechos y los pendientes de cada actor.
- Gestionar la presentación de informes para los procedimientos que lo requieren.
- Registrar el reintegro del personal, según los derechos de los actores y flujo de actividades de la solicitud.

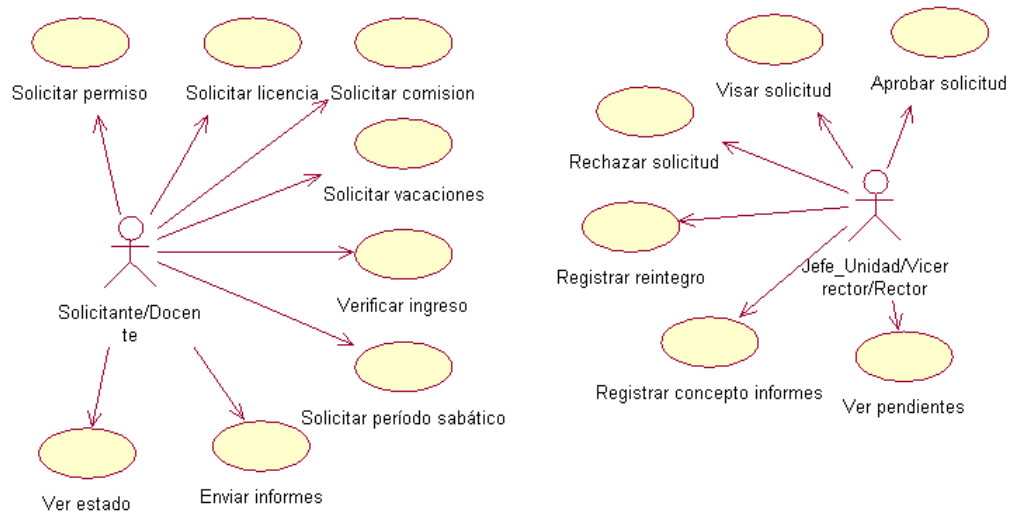
3.1.2 Listado de Actores. Los actores corresponden a los usuarios del sistema. Se han considerado los siguientes:

- Solicitante/Docente
- Jefe Unidad/Vicerrector/Rector

La figura 15, presenta el diagrama de los casos de uso encontrado para el sistema en esta Fase. A continuación se encuentra la descripción de los mismos.

- *Verificar Ingreso:* Representa el control de acceso al sistema.
- *Solicitar:* Corresponde al trámite de la solicitud (Permiso, Licencia, Comisión, entre otros).
- *Enviar Informes:* Le permite al solicitante, una vez se encuentre en vigencia de ciertos procedimientos, enviar informes del desempeño de su labor.

Figura 15. Diagrama de casos de uso – Fase de Inicio.



- *Registrar Concepto Informes:* El jefe de la unidad es el encargado de revisar los informes. Este caso de uso representa la acción de registrar el concepto y observaciones de los informes.
- *Ver Estado:* Este caso de uso representa la acción de consultar el estado en que se encuentra una solicitud junto con sus correspondientes observaciones.
- *Visar Solicitud:* Le permite colocar visto bueno a las solicitudes según corresponda al tipo de solicitud y al flujo de actividades.
- *Aprobar Solicitud:* Corresponde a la actividad de aprobación de la situación presentada.
- *Rechazar Solicitud:* Este caso de uso corresponde a la anulación de una solicitud por parte de uno de los actores que no es el solicitante.
- *Registrar Reintegro:* Permite al jefe inmediato del solicitante, registrar el reintegro de éste, en el momento de regresar a sus labores en el cargo que

desempeña al terminar la situación o antes de terminar el período autorizado.

- *Ver Pendientes:* Corresponde a la acción de consultar las actividades pendientes de atención de trámite de cada usuario.

3.2. FLUJO DE ANÁLISIS

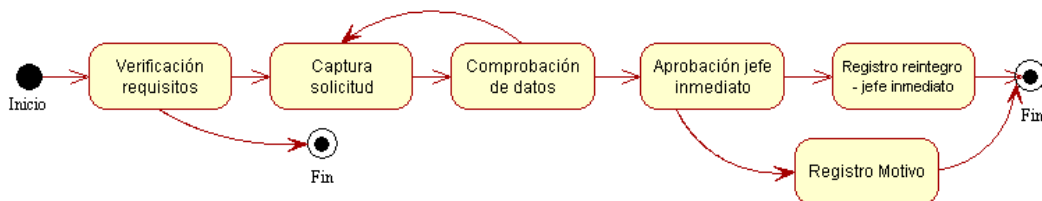
3.2.1 Análisis de procedimientos. Teniendo en cuenta los estados de las solicitudes presentadas en el flujo anterior, se muestra a continuación dichos estados, aplicados a los procedimientos que el sistema maneja.

Para ello se utilizaron diagramas de estados. En las figuras, las flechas curvas indican los posibles saltos que pueden existir entre los estados. El flujo esperado para el procedimiento, esta representado por las flechas rectas entre los estados.

Los estados denominados *Fin*, indican la terminación de la solicitud, ya sea por el registro del reintegro de la persona o por el rechazo de la misma en alguno de los estados donde esta acción se pueda llevar a cabo. El procedimiento, una vez seleccionado, inicia con un estado en el cual se verifican los requisitos que se deben cumplir para el trámite del mismo.

La figura 16, presenta los estados del procedimiento de Permisos.

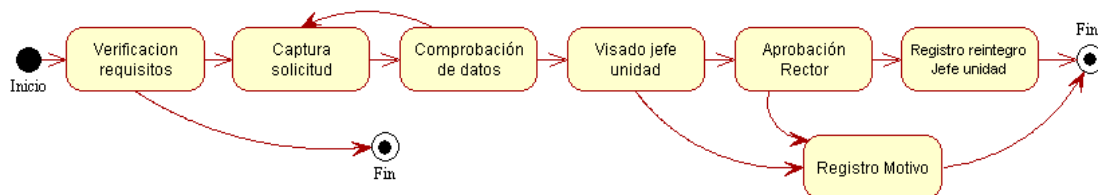
Figura 16. Diagrama de estados para el procedimiento de Permisos.



Este procedimiento presenta una aprobación directa por parte del jefe inmediato del solicitante, actualmente, para permisos de hasta 1 día laboral. Si el período es

mayor la aprobación es realizada por parte del superior al jefe inmediato del solicitante. La figura 17, presenta los estados del procedimiento de Licencias no remuneradas.

Figura 17. Diagrama de estados para el procedimiento de Licencia no remunerada.



Para el caso de las *Licencias no remuneradas* mostrado en la figura 17, la aprobación es realizada por parte del Rector con previo visado del jefe de la unidad a la cual pertenece el solicitante (jefe inmediato). El rechazo de la solicitud se puede producir en el visado de la solicitud o en la aprobación de la misma.

Para el caso de las comisiones, tanto de *Estudios* como de *Servicios*, la persona encargada de otorgar la aprobación de la solicitud depende de la duración de la misma. A continuación, la tabla 2 relaciona las personas encargadas de la aprobación de acuerdo a la duración de la solicitud.

Tabla 2. Aprobación de comisiones de acuerdo a la duración.

DURACIÓN DE LA SOLICITUD (DÍAS)	PERSONA QUE APRUEBA
1-5	Jefe inmediato
6-30	Superior a jefe inmediato
31-180	Vicerrector correspondiente ⁸
Más de 180	Rector

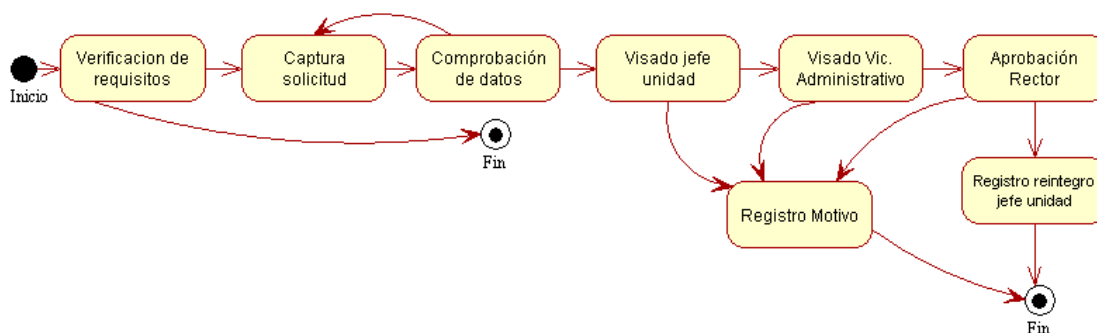
⁸ Para Docentes es el Vicerrector Académico de lo contrario será el Vicerrector Administrativo.

Las solicitudes cuya aprobación es realizada por el Rector o Vicerrector (ya sea Administrativo o Académico), han recibido los visados de todos los superiores desde su jefe inmediato hasta antes de llegar a la instancia correspondiente, según el organigrama vigente.

Las solicitudes aprobadas por el superior al jefe inmediato, al igual que el procedimiento de *Permisos*, requiere de un visado previo por parte del jefe inmediato del solicitante.

A manera de ejemplo la figura 18, muestra el caso de una comisión de servicios solicitada por una persona adscrita a alguna División⁹ (Servicios de Información, Financiera, Recursos Humanos, entre otras) de la Universidad.

Figura 18. Ejemplo de estados para una Comisión de servicios.



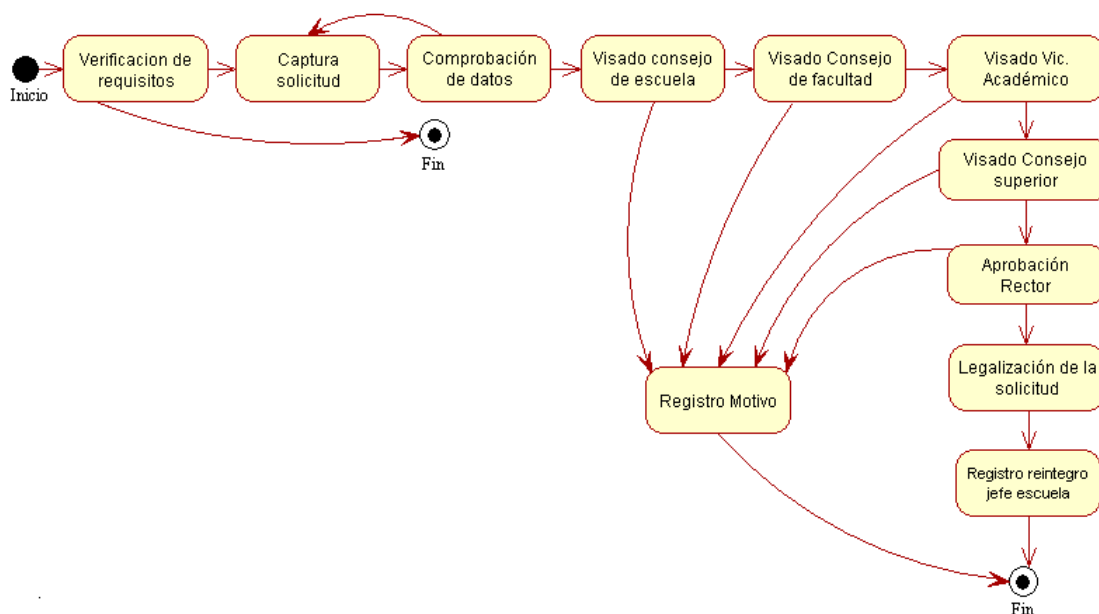
En las *Comisiones de Estudio*, las aprobaciones o visados de instancias diferentes a Rectoría o Vicerrectoría académica son realizadas en consejos, sólo para el caso de los Docentes. Esto indica que a diferencia de las Comisiones de Estudio la decisión no es tomada sólo por el Jefe de una Escuela, por ejemplo, sino por el Consejo de la misma Escuela. Además se pueden generar dos estados adicionales para estas comisiones. Si el solicitante no cumple con el requisito de días de servicio con la Universidad, la solicitud requiere el visado del Consejo Superior al cual llega la solicitud antes de la actividad de aprobación. Otro estado adicional que puede surgir una vez la solicitud se apruebe, es cuando ésta supere los 180 días. Si ello

⁹ Para una completa referencia acerca de las unidades académico-administrativas se puede consultar el Organigrama de la Universidad.

ocurre, la solicitud es enviada a la División de Recursos Humanos para el proceso de legalización correspondiente como se describe en el reglamento de la Universidad¹⁰.

También a manera de ejemplo la figura 19, muestra el flujo que tendría una solicitud de Comisión de Estudios tramitada por un Docente de una Escuela y con una duración mayor a 180 días.

Figura 19. Ejemplo de estados para una solicitud de Comisión de estudios.

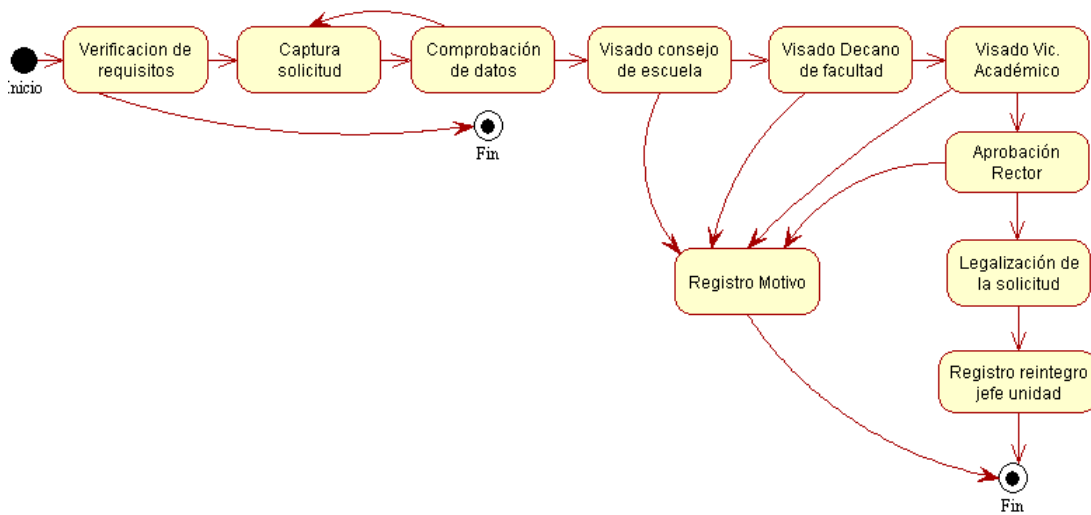


Las *Comisiones de Estudio* aprobadas, requieren de informes periódicos con el fin de llevar un control de las actividades del beneficiario.

Para el caso del *Período Sabático*, el flujo correspondiente es presentado en la figura 20. Este procedimiento es tramitado sólo por los Docentes de la Universidad.

¹⁰ Artículo 92, Reglamento del profesor. Bucaramanga, 1994. Artículo 66, Reglamento del personal administrativo de la Universidad Industrial de Santander.

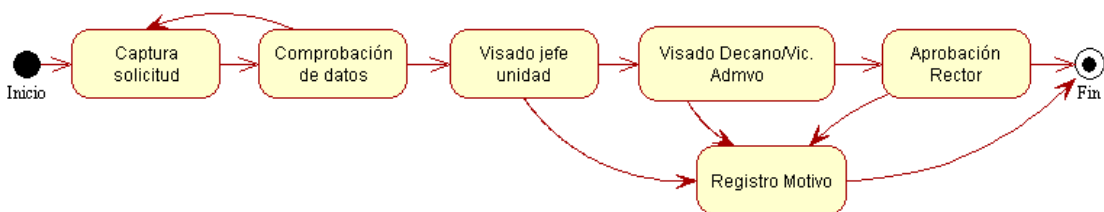
Figura 20. Diagrama de estados para el procedimiento de Período sabático.



De manera similar al procedimiento anterior, las solicitudes aprobadas de Período Sabático presentan el estado de legalización; además, que requieren de informes periódicos para el control de las actividades.

Para el caso de las *Vacaciones*, se distinguen dos solicitudes: las solicitud de *Acumulación* y la solicitud de *Disfrute*. El flujo para el procedimiento de Acumulación se presenta en la figura 21.

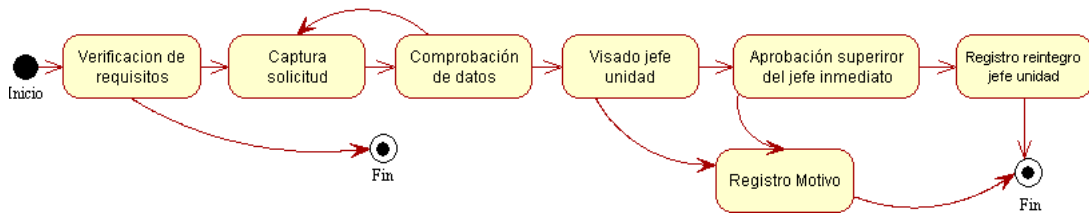
Figura 21. Diagrama de estados para el procedimiento de Acumulación de vacaciones.



El procedimiento de acumulación de vacaciones es aprobado mediante una resolución de la Rectoría, con previo concepto del jefe de la unidad académico-administrativa a la cual pertenece el solicitante y el Decano de la Facultad (para el caso de los docentes) o el Vicerrector Administrativo (en el caso del personal administrativo).

El flujo del procedimiento de *Disfrute de Vacaciones*, se presenta en la figura 22.

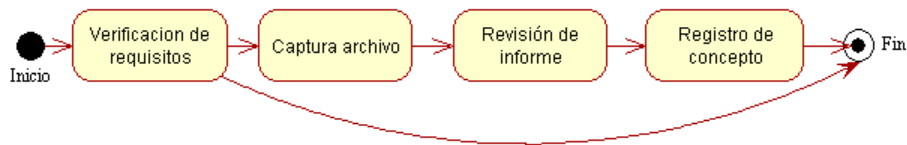
Figura 22. Diagrama de estados para el procedimiento de Disfrute de vacaciones.



Como lo muestra la figura anterior, el trámite de ésta solicitud es muy similar al trámite de la solicitud de *Permiso* mayores a un día.

En el caso de los informes para las solicitudes de *Comisión de Estudios* y *Período Sabático*, el flujo es presentado en la figura 23.

Figura 23. Diagrama de estados para el procedimiento de Informes.



Los informes serán enviados a través del sistema, mediante un módulo el cual tendrá una apariencia similar, a la manera como se envían archivos adjuntos en el correo electrónico. Posteriormente, cuando se haya realizado el proceso de evaluación correspondiente, se registrará en el sistema el concepto acerca del informe.

Las personas encargadas de realizar el proceso de evaluación actualmente, son: para *Comisiones de Estudio*, el Decano de la Facultad correspondiente (para el caso de los Docentes) y el Vicerrector Administrativo (para resto del personal); para *Período Sabático*, el Consejo de Facultad correspondiente al Docente.

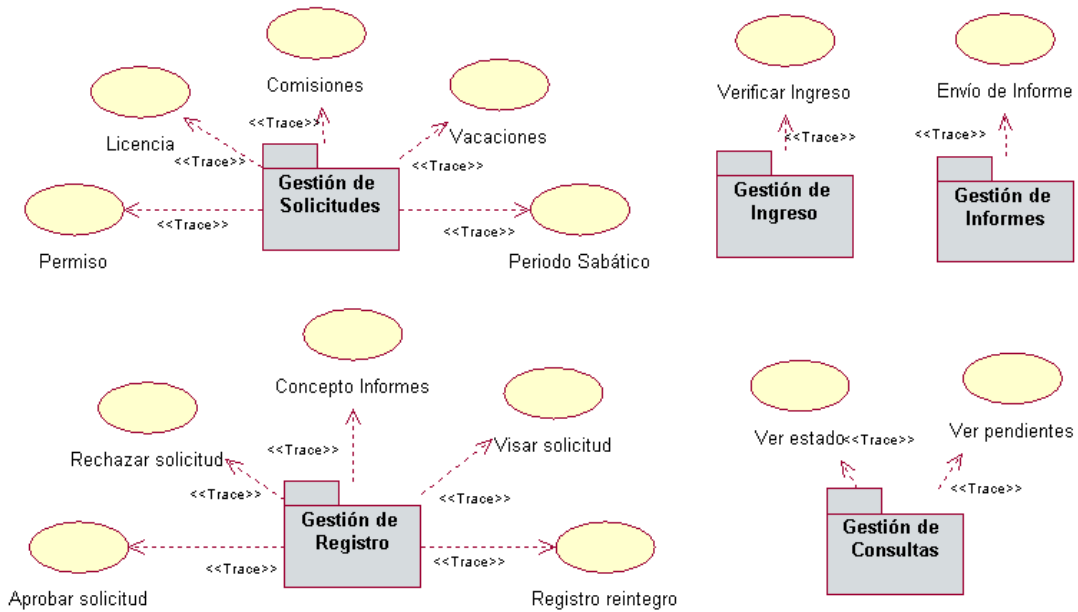
Algunas consideraciones generales para los diagramas presentados, son las siguientes:

- En los estados donde puede ocurrir el rechazo de la solicitud, es necesario especificar el motivo por el cual se realiza esta acción; además, como lo muestran la figuras, la solicitud finaliza su flujo y no continúa en el trámite.
- El estado *Capturar solicitud*, corresponde al llenado del formulario del procedimiento por parte del solicitante.
- El estado *Comprobación de datos*, corresponde a la verificación de los datos capturados en los formularios. Si existe algún error, se regresa e informa acerca del error para que sea corregido y poder continuar con el trámite.
- Como se pudo observar el flujo de las solicitudes que el sistema maneja, contemplan una última actividad llamada *Registro de reintegro*. En esta, como su nombre lo indica, el jefe inmediato de la persona que tramitó la solicitud se encarga de registrar el reintegro de la persona a su puesto de trabajo.

3.2.2 Análisis de la arquitectura. Para el análisis de la arquitectura, se recopilan los casos de uso encontrados en paquetes, cuyo fin es identificar y descomponer la funcionalidad del sistema en subfunciones; lo cual además ayuda a visualizar las posibles dependencias que puedan llegar a tener las clases, en fases más avanzadas del desarrollo del sistema.

En la figura 24, se presenta un diagrama que reúne los paquetes encontrados durante esta Fase.

Figura 24. Diagrama de paquetes – Fase de Inicio.

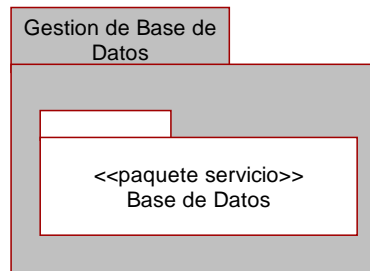


El paquete *Gestión de Solicitudes*, contiene los casos de uso correspondientes a los procedimientos que el sistema maneja. El paquete *Gestión de Consultas*, contiene los casos de uso correspondientes al manejo de consultas directas sobre el sistema.

El paquete *Gestión de Registro*, contiene los casos de uso concernientes a la administración del registro de conceptos en el sistema. *Gestión de Ingreso* y *Gestión Envío Informe*, contienen cada uno los casos de uso Verificar Ingreso y Envío Informe, respectivamente.

Como se puede deducir, todos los paquetes presentados anteriormente hacen uso de la base de datos para soportar su funcionalidad. La figura 25, presenta a *Gestión de Base de Datos*, como un paquete de servicio. Este paquete contiene casos de uso especializados en el manejo directo de la Base de Datos, como realizadores de consultas, métodos de conexión, desconexión, entre otros. Por ello se convierte en un paquete de análisis de aplicación general, el cual es utilizado por los demás.

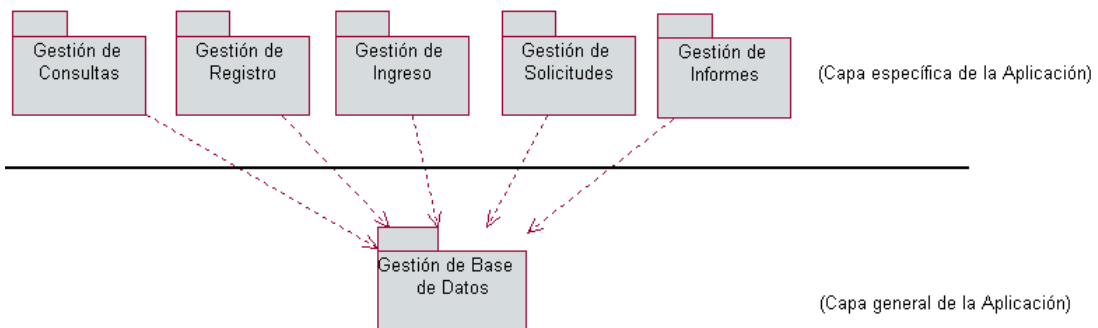
Figura 25. Paquete de servicio.



3.2.3 Dependencia entre paquetes de análisis y capas. Los paquetes hallados en esta Fase, presentan una dependencia entre ellos, una manera de clarificar estas dependencias es establecer una jerarquía de capas, cuyo objetivo es dividir los paquetes en generales y específicos para la aplicación.

La figura 26, presenta estas dependencias:

Figura 26. Dependencia entre paquetes – Fase de Inicio.



Las flechas indican que los paquetes específicos hacen uso del paquete general, que en este caso es *Gestión de Base de Datos*.

3.2.4 Análisis de casos de uso. El análisis se hace mediante diagramas de colaboración. Estos proporcionan una vista interna del desarrollo del caso de uso, de manera análoga a como los diagramas de estados proporcionan una vista

externa. Los diagramas de colaboración utilizan clases denominadas clases de análisis: de interfaz, de control o de entidad¹¹.

- Clases de Interfaz: modelan la interacción entre el sistema y sus actores.
- Clases de Control: representan coordinación, secuenciamiento, transacciones y control de otros objetos.
- Clases de Entidad: modelan la información que tiene una vida larga y que a veces es persistente.

La figura 27, muestra la representación gráfica de éstas clases.

Figura 27. Representación gráfica de las clases de análisis.



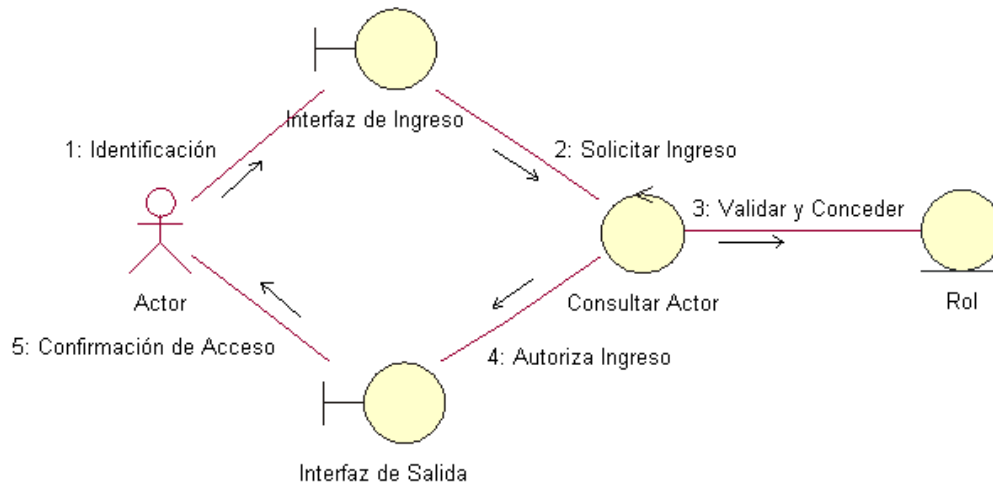
Durante esta Fase se hará el análisis del caso de uso *Verificar Ingreso*, el cual permite el ingreso de los actores al sistema.

- **Verificar Ingreso.** La figura 28 presenta como se desarrolla el caso de uso a través de clases de análisis.

El actor se identifica en el sistema a través del objeto “Interfaz de Ingreso”, proporcionando su correspondiente nombre de usuario y contraseña (1).

¹¹ JACOBSON, Ivar. BOOCH, Grady. RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Addison Wesley. Madrid 1999. Página 173.

Figura 28. Diagrama de colaboración – Verificar ingreso.



La clase “Interfaz de Ingreso” solicita al objeto “Consultar Actor” la búsqueda en la base de datos del actor que trata de ingresar al sistema (2). El objeto “Consultar Actor” comprueba si existe, identifica y asocia el rol al que pertenece este actor y da acceso al sistema; en caso de no existir o no estar vigente, niega el acceso al sistema (3). El objeto “Interfaz de Salida”, recibe de “Consultar Actor”, la orden de visualizar la confirmación de acceso o rechazo al sistema (4). “Interfaz de Salida” da la bienvenida del actor al sistema (5).

3.3. CONCLUSIONES DE LA FASE DE INICIO

Uno de los resultados más importantes durante esta Fase fue la selección de una arquitectura candidata para el sistema, la cual especifica una vista conceptual de la estructura del sistema. La arquitectura candidata propuesta es la arquitectura de tres capas. Esta arquitectura se basa en servicios específicos en cada capa, los cuales se comunican entre sí mediante el uso de componentes¹².

Para las solicitudes, las capas de esta arquitectura estarán representadas mediante:

¹² Una mayor referencia se encuentra en el capítulo 2 Fundamentación Teórica.

- Interfaces de usuario: el sistema se utilizará, como el título del proyecto lo indica, a través de Internet por lo cual las interfaces se desarrollarán en el lenguaje de construcción de páginas Web: HTML.
- Dll's: en las cuales se encapsulará las reglas del negocio.
- Gestión de base de datos: el sistema utilizará el manejador de base de datos Informix.

La Fase de Inicio se desarrolló en una sola iteración, pasando por dos de los cinco flujos de trabajo fundamentales (requisitos, análisis, diseño, implementación y prueba).

Previo al flujo de requisitos hubo una etapa de investigación preliminar donde se recopiló información general la cual fue presentada mediante: un listado de funcionalidades del módulo, un listado de actores y un diagrama inicial de casos de uso. Estas fueron las entradas principales para el flujo de requisitos, en el cual se analizó los procedimientos principales para el módulo de solicitudes. Dicho análisis se llevó a cabo mediante la presentación de los procedimientos en diagramas de estados, en los cuales se muestra el flujo que toman, desde su captura hasta su aprobación.

Para la parte de análisis, se agruparon los casos de uso encontrados en paquetes con el fin de comprender de una mejor manera la relación entre ellos. Seguidamente se presentó la realización del caso de uso *Verificar ingreso*, mediante un diagrama de colaboración.

Durante esta etapa también, se realizó un análisis del modelo de datos que siguen los procedimientos en el Sistema de Información de Recursos Humanos de la

Universidad; análisis que posteriormente servirá para diseñar la arquitectura que éstos seguirán, cuando se implementen las solicitudes¹³.

Paralelamente con la Fase de Inicio, se realizó una capacitación en la herramientas requeridas para el proyecto. Para ello se comenzó, con la adecuación inicial de ciertas interfaces, las cuales serán útiles en la siguiente Fase del desarrollo.

¹³ La referencia sobre la implementación de las solicitudes se encuentran en los capítulos Fase de Elaboración y Fase de Construcción.

4. FASE DE ELABORACIÓN

Durante el desarrollo de esta fase se busca obtener una arquitectura confiable para el sistema. Para ello se divide el trabajo mediante miniproyectos y se identifican un número significativo de los casos de uso encontrados. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a las veces que se lleva a cabo un paso por cada uno de los flujos de trabajo y los incrementos al crecimiento del producto¹⁴, en este caso del sistema.

Para el desarrollo de esta fase se siguieron los flujos de trabajo de: recopilación de requisitos, análisis, diseño e implementación. En el flujo de recopilación de requisitos se complementó la lista de características de la herramienta, se reestructuró la lista de actores propuesta durante la fase de inicio y se identificó, describió y detalló aproximadamente el 80% de los casos de uso.

Al final de la Fase de Elaboración se espera:

- Especificar nuevos requerimientos (mediante casos de uso) encontrados durante los flujos de la fase anterior.
- Reestructurar la lista de funcionalidades, así como el diagrama de casos de uso.
- Describir y detallar aproximadamente el 80% de los casos de uso.
- Complementar la identificación de los paquetes de análisis.
- Analizar mediante clases los casos de uso clasificados como críticos e importantes.

¹⁴ Tomado de: JACOBSON, Ivar. BOOCH, Grady. RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Addison Wesley. Madrid 1999. Sección 1.4.

- Implementar una solicitud que se tomará como base para la arquitectura del sistema.

Esta fase comienza con la especificación de nuevos requerimientos encontrados, tanto en la fase anterior como en el flujo de requisitos. A medida que se avanza en cada iteración se pueden encontrar nuevos requerimientos, eliminar algunos de los considerados o fusionar requerimientos similares.

4.1 FLUJO DE REQUISITOS

4.1.1 Listado de procedimientos. Los siguientes fueron los procedimientos presentados al comienzo de la fase de inicio:

- 1) Permisos.
- 2) Licencias No remuneradas.
- 3) Comisiones de Estudio.
- 4) Comisiones de Servicio.
- 5) Periodo Sabático.
- 6) Informes de Comisiones y Período Sabático.
- 7) Vacaciones.

Al término de esta fase se encontraron que algunos de los procedimientos anteriores sufrieron alguna modificación (ampliación o agrupación) y que además surgieron otros procedimientos.

Un nuevo listado de los procesos es presentado a continuación:

- 1) Solicitud de Permiso.
 - a. Individual.
 - b. Grupal.
 - c. Compensatorios.
 - d. Para compensar

- 2) Solicitud de licencia no remunerada.

- 3) Solicitud de comisión:
 - a. De estudio.
 - b. De servicio.

- 4) Solicitud de período sabático.

- 5) Solicitud de vacaciones:
 - a. Solicitud de acumulación.
 - b. Solicitud de disfrute.

- 6) Solicitud de Prórroga.

- 7) Solicitud de Cancelación.

- 8) Solicitud de Modificación.

- 9) Registro de informes.

- 10) Evaluación de informes.

4.1.2 Listado de funcionalidades. A continuación se presentan las nuevas funcionalidades a las mostradas en la Fase anterior.

- Permitir el ingreso del personal de planta de la Universidad al sistema.
- Cambiar de rol ante el sistema, si se cuenta con más de uno de ellos.
- Colocar visto bueno o dar aprobación a las solicitudes de los procesos nombrados.
- Rechazar solicitudes y registrar el concepto por el cual se rechazan.
- Consultar las observaciones hechas a las solicitudes durante su trámite.
- Validar los datos de las solicitudes contra datos almacenados en la Base de Datos como horarios, restricciones, parámetros, entre otros.
- Cancelar una solicitud en trámite de aprobación o aprobada.
- Permitir el registro de reintegro, con la posibilidad que se pueda realizar antes de la fecha establecida para éste.
- Mostrar las horas adicionales trabajadas, junto con las horas disponibles para iniciar algún proceso de permiso por este concepto.
- Modificar los términos de una solicitud (período, objeto, entre otros).
- Consultar el concepto de informes enviados en ciertos procesos ya aprobados.
- Consultar información general de las solicitudes.

- Consultar información de apoyo al trámite de las solicitudes.
- Permitir el registro de permisos causados.
- Registrar permisos grupales.
- Utilizar el tiempo adicional registrado para pagar solicitudes de permiso aprobadas del tipo *para compensar*.
- Generar y visualizar mensajes de aprobación, cancelación y rechazo de solicitudes.

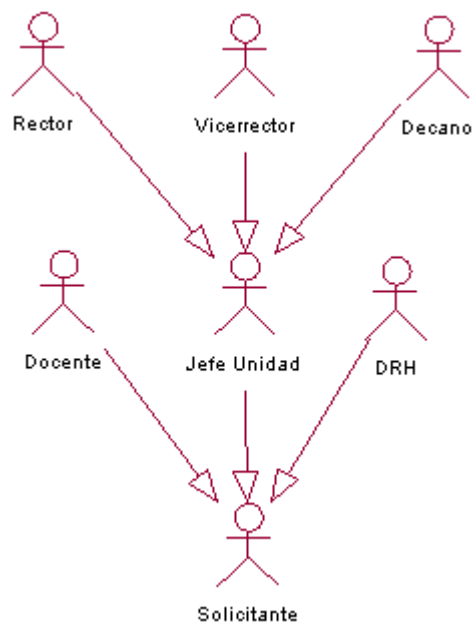
4.1.3 Actores. Los siguientes son los actores del sistema. Incluye los presentados en la Fase anterior y una breve descripción.

- **Solicitante:** Corresponde a los funcionarios administrativos y docentes de planta vinculados con la Universidad.
- **Docente:** Corresponde al personal docente de planta vinculado con la Universidad.
- **Jefe Unidad:** Corresponde a los jefes de las unidades académico-administrativas de la Universidad.
- **División de Recursos Humanos (DRH):** Corresponde a un funcionario de la División de Recursos Humanos de la Universidad, encargado de registrar cierta información.
- **Decano:** Corresponde a los funcionarios que tengan el cargo de Decano de alguna de las facultades de la Universidad.

- **Vicerrector:** Corresponde a los funcionarios que tengan el cargo de Vicerrector en la Universidad, ya sea Académico o Administrativo.
- **Rector:** Corresponde al funcionario cuyo cargo sea el Rector de la Universidad.

La figura No.29 muestra la relación que existe entre los actores del sistema.

Figura 29. Relación entre los actores del sistema.

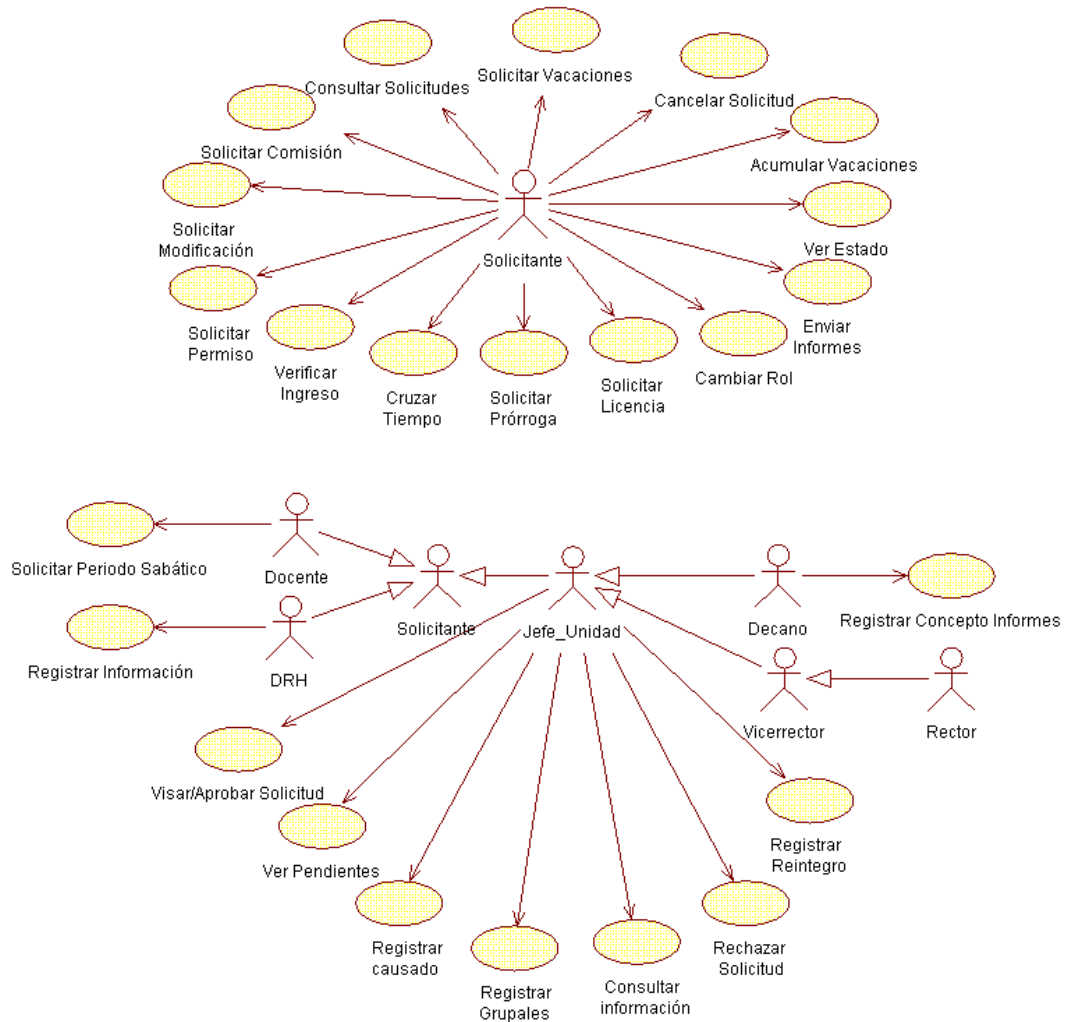


Desde el punto de vista de la programación orientada a objetos (POO) podemos apreciar en el diagrama, una relación de herencia, la cual se ve reflejada en los resultados que le sistema proporcionará a cada uno de sus actores. El actor en el extremo de la flecha indica que éste realiza otros casos de uso sumados a los que realiza el actor de la punta de la flecha. Por ejemplo, el actor *Jefe Unidad* es una especialización del actor *Solicitante*, esto significa que *Jefe Unidad* puede realizar todos los casos de uso a los que el *Solicitante* tiene derecho en el sistema, pero no puede realizar los del *Vicerrector* y por consecuencia los del *Rector*.

La figura 30 muestra las relaciones entre los casos de uso y los actores identificados. Para una mayor comprensión, el diagrama se ha dividido en dos partes:

la primera muestra los casos de uso asociados al actor *Solicitante*, y la segunda muestra las relaciones de herencia entre éste actor y los demás, también muestra los casos de uso asociados al actor *Jefe Unidad*.

Figura 30. Diagrama de casos de uso – Fase de Elaboración.



A continuación se presenta una breve descripción de los casos de uso presentados en la figura 30.

- **Verificar ingreso:** Este caso de uso permite a los actores del sistema obtener ingreso al mismo.

- **Solicitar:** Corresponde al inicio del trámite de una solicitud (Permiso, Licencia, Comisión, Vacaciones, Periodo Sabático), válida para un actor y que ha sido previamente seleccionada de una interfaz (Inbox).
- **Solicitar modificación:** Este caso de uso permite al solicitante, modificar los términos de una solicitud, como por ejemplo: la duración, la fecha de inicio o de fin, entre otros.
- **Cancelar solicitud:** Permite al solicitante tramitar la cancelación de una solicitud, que se encuentre en trámite de aprobación o aprobada que no haya entrado aún en vigencia.
- **Enviar informes:** Le permite al solicitante, una vez se encuentre en vigencia de ciertos procedimientos, enviar informes del desempeño de su labor.
- **Registrar concepto informes:** El actor *Decano* es el encargado de revisar los informes. Este caso de uso representa la acción de registrar el concepto y observaciones a los informes en el sistema.
- **Ver estado:** Este caso de uso le permite a un actor consultar las solicitudes que ha iniciado y observar el estado en el que se encuentra.
- **Visar/Aprobar solicitud:** Le permite a un actor colocar visto bueno o aprobar solicitudes según corresponda al tipo de solicitud y al flujo de actividades.
- **Rechazar solicitud:** Este caso de uso corresponde a la anulación de una solicitud por parte de uno de los actores que no es el solicitante.
- **Registrar reintegro:** Permite al jefe inmediato del solicitante, registrar el reintegro de éste, en el momento de regresar a sus labores en el cargo que desempeña al terminar la situación o antes de terminar el período autorizado.

- **Ver pendientes:** Corresponde a la acción de consultar las actividades pendientes de atención o de trámite para cada usuario.
- **Consultar solicitudes:** Permite consultar las solicitudes propias utilizando criterios como las fechas de solicitud, el estado y el tipo, entre otros.
- **Registrar información:** Le permite al actor DRH realizar el registro de información tal como: Número de resolución, estado de paz y salvo de los solicitantes, entre otros.
- **Consultar información:** Permite realizar consultas de información general de apoyo para el trámite de solicitudes.
- **Cambiar rol:** Permite cambiar el rol con el que un usuario actúa ante el sistema.
- **Registrar Causado:** Permite al actor *Jefe Unidad* realizar el registro de Permisos Causados en el sistema.
- **Registrar Grupales:** Permite al actor *Jefe Unidad* realizar el registro de una solicitud de Permiso Grupal.
- **Cruzar tiempo:** Permite al actor *Solicitante* utilizar registros de tiempo adicional trabajado para compensar cierto tipo de permiso.

4.1.4 Clasificación de los casos de uso. A continuación, la tabla 4 presenta los casos de uso identificados durante los flujos de requisitos de las Fases de Inicio y Elaboración, junto con sus prioridades y los actores asociados a éstos. La información se encuentra ordenada por Prioridad.

4.1.4.1 Casos de Uso Detallados. Para esta Fase se analizarán en detalle los casos de uso denominados con prioridad *Crítica*, esto con el fin de obtener de una mejor manera la línea base de la arquitectura para el sistema.

Tabla 3. Casos de uso, actores y prioridad.

CASO DE USO	ACTORES	PRIORIDAD
Verificar Ingreso	Todos	Crítico
Solicitar Permiso	Solicitante	Crítico
Cancelar Solicitud	Solicitante	Crítico
Ver estado	Todos	Crítico
Aprobar/ Visar Solicitud	Jefe Unidad, Decano, Vicerrector, Rector	Crítico
Rechazar Solicitud	Jefe Unidad, Decano, Vicerrector, Rector	Crítico
Registrar Reintegro	Jefe Unidad	Crítico
Ver Pendientes	Jefe Unidad, Decano, Vicerrector, Rector	Crítico
Consultar Información	Jefe Unidad, Decano, Vicerrector, Rector	Crítico
Cambiar rol	Todos	Crítico
Cruzar tiempo	Solicitante	Crítico
Registrar causado	Jefe Unidad	Crítico
Registrar Grupales	Jefe unidad	Crítico
Solicitar licencia	Solicitante	Importante
Solicitar comisión	Solicitante	Importante
Solicitar vacaciones	Solicitante	Importante
Acumular vacaciones	Solicitante	Importante
Solicitar Periodo Sabático	Solicitante - Docente	Importante
Solicitar Prórroga	Solicitante	Importante
Enviar Informes	Solicitante	Importante
Registrar Concepto Informes	Jefe Unidad	Importante
Consultar solicitudes	Todos	Secundario
Registrar Información	DRH	Secundario
Solicitar Modificación	Solicitante	Secundario

- **Solicitar Permiso**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: Permite al solicitante iniciar el trámite de un permiso.

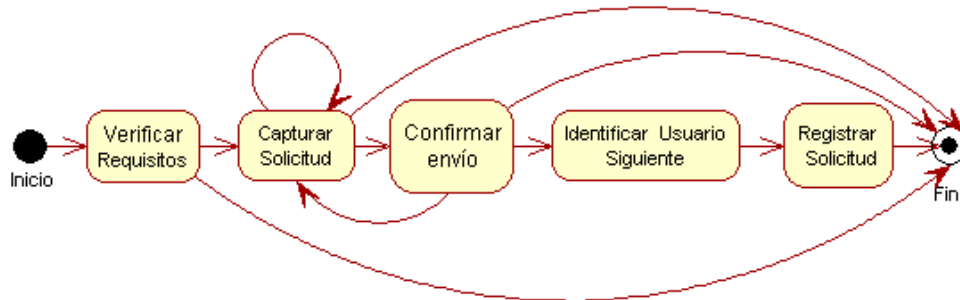
Resumen: El permiso es una situación de carácter remunerado, Se pueden distinguir los siguientes tipos de permisos:

- a. Individual: Incluye Personales, Cita Médica, Capacitación y Calamidad Doméstica.
- b. Grupal: Permiso asignado a las asociaciones de la Universidad (FAVUIS, CAPRUIS, entre otras) con diferentes propósitos, los cuales convocan a un grupo de empleados a reuniones o actividades.
- c. Para compensar: Permiso concedido para pagar con tiempo adicional de trabajo.
- d. Compensatorios: Permiso concedido en compensación por tiempo adicional trabajado.

Precondición: el actor selecciona la opción *Permiso*, del menú de solicitudes desde la página de Inbox correspondiente. Se permite el registro de los tipos: *Individual, Para compensar y Compensatorios*.

Flujo de sucesos: Los flujos identificados son presentados en la figura 31.

Figura 31. Diagrama de estados - Solicitar Permiso.



Camino básico:

Verificar Requisitos: El sistema verifica los requisitos que el solicitante debe tener para tramitar el permiso. Por ejemplo, las horas máximas disponibles de acuerdo a la clase del permiso, entre otros. Seguidamente si el solicitante cumple los requisitos se le presentará la interfaz (formulario) de solicitud.

Nota: Un requisito que es validado para los permisos del tipo *Compensatorio*, es que el solicitante debe tener tiempo adicional trabajado registrado en el Sistema de Información de Recursos Humanos.

Capturar Solicitud: En este estado el sistema presenta el formulario; algunos de los datos del mismo son: el período de tiempo solicitado en forma de fechas y horas (desde y hasta), el motivo y el tipo de permiso. Para los tipos *Compensatorios*, se solicita al usuario seleccionar los registros de tiempo adicional a utilizar, para el permiso. Para los tipos de permiso *Grupales*, además de solicitar el periodo, existe una sección en la cual se puede especificar el número de personas a disfrutar del permiso y la identificación de cada uno de ellos o seleccionar un grupo de personas de acuerdo a algunos criterios de búsqueda disponibles en el sistema, como género, antigüedad, unidad a la que pertenece y cargo desempeñado.

Confirmar Envío: En este estado, el sistema presenta al solicitante la información de la solicitud que va a enviar, con el fin de confirmar el envío. El solicitante, tiene la posibilidad de realizar el envío, modificar la solicitud o cancelar la misma. Además el sistema le puede presentar alertas, las cuales no impiden el envío de la solicitud, porque esas situaciones no afectan su trámite.

Identificar Siguiete Actor: Una vez confirmado el envío de la solicitud, el sistema de acuerdo a la información de la solicitud, se encarga de buscar el proceso asociado con su correspondiente actividad inicial.

Registrar Solicitud: En este estado se registra la solicitud en el sistema y se actualiza el estado, el cual queda en espera de aprobación o de visto bueno (visado).

Fin: La instancia del caso de uso llega a su fin cuando el sistema registra la solicitud en el sistema, luego que el solicitante ha confirmado el envío. El sistema presentará al actor una interfaz de confirmación final, la cual informa si el envío se hizo correctamente.

Caminos alternativos:

Verificar requisitos - Fin: Este camino se presenta cuando el solicitante no cumple con alguno de los requisitos para tramitar la solicitud.

Capturar solicitud: La solicitud permanece en este estado cuando el formulario contenga errores en alguno de sus campos, como por ejemplo fechas y horas fuera del horario de trabajo.

Capturar solicitud – Fin: Este camino se presenta cuando el solicitante decide no realizar el trámite de la solicitud, cancelando el envío del formulario.

Confirmar Envío – Fin: Este camino se presenta cuando el sistema despliega la confirmación de envío para la solicitud y el solicitante decide cancelar el envío.

Confirmar Envío - Capturar Solicitud: Este camino se presenta cuando el sistema despliega la confirmación de envío para la solicitud y el solicitante decide modificar alguno de los datos ingresados.

- **Ver Estado**

Actor: Cualquier actor bajo el rol de *Solicitante*.

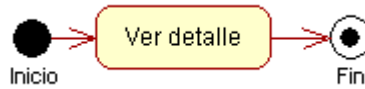
Propósito: Permite ver el estado actual en el que se encuentra una solicitud.

Resumen: Los estados en los cuales una solicitud se puede encontrar son los siguientes: EN TRAMITE, APROBADA, CANCELADA Y RECHAZADA.

Precondición: El actor selecciona una carpeta presente en la página de su Inbox correspondiente, que muestra un listado con las solicitudes cuyo trámite ha sido iniciado por él.

Flujo de sucesos: Los flujos identificados son presentados en la figura 32.

Figura 32. Diagrama de estados - Ver Estado.



Camino básico:

Ver detalle: En este estado se muestran los detalles de una solicitud elegida. Entre estos detalles presentados se encuentran: las observaciones hechas durante el trámite que se ha llevado a cabo para la solicitud, así como la fecha y el actor que la realizó.

Fin: La instancia del caso de uso finaliza cuando el usuario regresa a su Inbox correspondiente.

- **Cancelar Solicitud**

Actor: Cualquier actor bajo el rol de *Solicitante*.

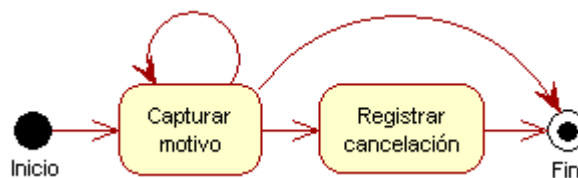
Propósito: Permite cancelar una solicitud.

Resumen: Le permite a un actor cancelar una solicitud que se encuentre en los estados EN TRAMITE o APROBADA, pero que aún no ha entrado en vigencia. Si la solicitud se encuentra en el estado, EN TRAMITE, la cancelación es total; en caso contrario si se encuentra APROBADA, puede ser total o parcial.

Precondición: El actor selecciona de un listado la solicitud que quiere cancelar. El listado contiene las solicitudes que ha iniciado trámite y que tiene en los estados EN TRAMITE o APROBADA, que aún no han entrado en vigencia.

Flujo de sucesos: Los flujos identificados son presentados en la figura 33.

Figura 33. Diagrama de estados - Cancelar Solicitud.



Camino básico:

Capturar motivo: Una vez seleccionada la solicitud del listado se presenta un detalle de la misma, en la cual el actor escribe el motivo por el cual cancela (o solicita cancelar) dicha solicitud. Además de seleccionar el tipos de cancelación: total o parcial.

Registrar cancelación: En este estado se registra la cancelación en el sistema. Dependiendo del estado de la solicitud que se quiere cancelar, se puede generar una solicitud de cancelación o se cancela directamente la solicitud.

Fin: La instancia del caso de uso llega a su fin cuando el sistema registra la cancelación de la solicitud en el sistema. El sistema presentará al actor una interfaz de confirmación de la acción realizada.

Caminos alternativos:

Capturar motivo: El caso de uso permanece en este estado, cuando para una cancelación parcial se ingresa la fecha u hora desde de cancelación en forma errónea.

Capturar motivo – Fin: Esto ocurre cuando el usuario decide no continuar con el proceso de cancelación, una vez el sistema le ha presentado la solicitud.

- **Aprobar/Visar Solicitud**

Actor: Rector, Vicerrector, Decano y Jefe de Unidad.

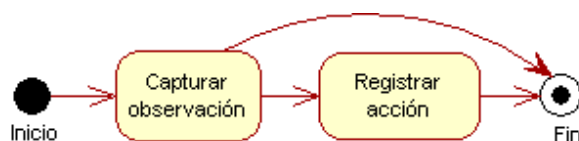
Propósito: Permite dar visto bueno o aprobar una solicitud.

Resumen: Le permite a los actores correspondientes dar Visto Bueno o Aprobar una solicitud, dependiendo del estado en que esta se encuentre.

Precondición: Escoger en la correspondiente página de Inbox, la opción que presenta un listado con las solicitudes pendientes.

Flujo de sucesos: Los flujos identificados son presentados en la figura 34.

Figura 34. Diagrama de estados - Aprobar/Visar Solicitud.



Camino básico:

Capturar observación: Al seleccionar una solicitud del listado, el sistema le presenta todos los datos de ésta. En este estado se captura la aprobación y las observaciones que el actor considera necesarias. Estas observaciones no son requeridas cuando la solicitud es aprobada.

Registrar acción: En este estado se realiza el registro de la acción seleccionada por el actor en el sistema, junto con su correspondiente observación, cuando existe. Durante el registro se actualiza el estado de la solicitud, dependiendo del cual la solicitud puede quedar pendiente para otro actor.

Fin: El caso de uso llega a su fin cuando el sistema presenta una interfaz donde se confirme la acción que se realizó sobre la solicitud.

Caminos alternativos:

Capturar observación – Fin: Ocurre cuando el actor suspende la selección de la acción sobre la solicitud, cuando el sistema le presenta en detalle la solicitud.

- **Rechazar Solicitud**

Actor: Rector, Vicerrector, Decano y Jefe de Unidad.

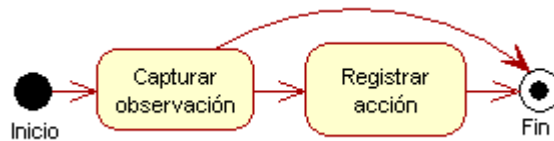
Propósito: Permite rechazar una solicitud.

Resumen: Le permite a los actores descritos rechazar una solicitud, que se encuentra pendiente por registrar acción.

Precondición: Una vez el sistema presenta la solicitud, se debe escoger el rechazo de la misma en el formulario de captura de datos.

Flujo de sucesos: Los flujos identificados son presentados en la figura 35.

Figura 35. Diagrama de estados - Rechazar Solicitud.



Camino básico:

Capturar observación: En este estado el sistema, a través del formulario, se encarga de solicitar las observaciones o el motivo por el cual se rechaza la solicitud.

Registrar acción: Una vez confirmado el rechazo de la solicitud, el sistema actualiza el estado de la misma. La solicitud pasa al estado RECHAZADA.

Fin: El caso de uso llega a su fin cuando el sistema presenta una interfaz donde se confirme la acción de rechazo sobre la solicitud.

Caminos alternativos:

Capturar motivo – Fin: Cuando el sistema le presenta en detalle la solicitud, el actor puede suspender el registro de la acción sobre la misma.

- **Ver Pendientes**

Actor: Cualquier actor.

Propósito: Permite ver la solicitudes pendientes para realizar alguna acción.

Resumen: Le permite a los actores ver cuales solicitudes tiene pendientes para realizar aprobación, dar visto bueno, rechazar, ver el estado en el que se encuentran sus solicitudes o mensajes que el sistema presenta. De acuerdo al rol con el que se haya ingresado al sistema, éste le presenta varios listados:

Pendientes: presenta las solicitudes pendientes por aprobación o visado.

Personales: presenta las solicitudes cuyo trámite ha sido iniciado por el actor que ingresó al sistema.

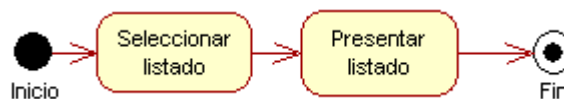
Reintegrar: sólo se encuentra disponible para los actores Rector, Vicerrector, Decano y Jefe de Unidad, y presenta las solicitudes de empleados a su cargo que están pendientes de registrar el reintegro al lugar de trabajo.

Mensajes: presenta los mensajes generados por el sistema ante acciones realizadas sobre solicitudes tramitadas por el rol.

Precondición: Haber ingresado al sistema correctamente y encontrarse en la página de Inbox.

Flujo de sucesos: Los flujos identificados son presentados en la figura 36.

Figura 36. Diagrama de estados - Ver Pendientes.



Camino básico:

Seleccionar listado: Una vez en el Inbox, el actor puede seleccionar uno de los listados nombrados anteriormente. Dependiendo del rol con el cual haya ingresado al sistema, se mostrarán las opciones que permiten ver listados, correspondientes al rol seleccionado.

Presentar listado: En este estado se presenta un listado de solicitudes. Estos listados contienen información diversa para las solicitudes dependiendo del listado seleccionado, debido a que cada uno tiene objetivos diferentes.

Ver pendientes: El caso de uso llega a su fin cuando muestra el listado seleccionado.

- **Registrar Reintegro**

Actor: Rector, Vicerrector, Decano y Jefe de Unidad.

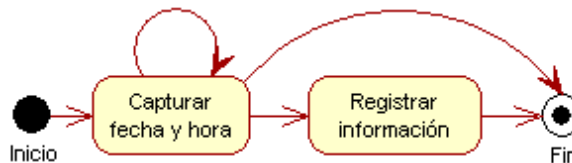
Propósito: Registrar el reintegro de un funcionario.

Resumen: Le permite a los actores relacionados anteriormente, registrar el reintegro de un funcionario, que pertenezca a su unidad, al lugar de trabajo correspondiente.

Precondición: Escoger en la correspondiente página de Inbox, la opción que presenta un listado con las solicitudes que están pendientes de reintegro.

Flujo de sucesos: son presentados en la figura 37.

Figura 37. Diagrama de estados – Registrar Reintegro.



Camino básico:

Capturar fecha y hora: Una vez seleccionada una solicitud para registrar el reintegro del funcionario, se presenta un formulario donde se solicita el ingreso de la fecha y hora de reintegro. Estos datos son validados respecto del período disfrutado y el horario vigente del funcionario a reintegrar.

Registrar información: Confirmado el reintegro de la persona, la fecha y hora de éste es registrada en el sistema. Con esto la solicitud queda terminada.

Fin: La instancia del caso de uso llega a su fin cuando se registra el reintegro del funcionario, actualizando el estado de la solicitud.

Camino alternativo:

Capturar fecha y hora: Cuando los datos ingresados son erróneos el sistema no permite realizar el registro y permanece en este estado. El sistema detecta como la fecha u hora no concuerden con el horario registrado para el solicitante en el sistema, a la fecha de reintegro.

Capturar fecha y hora – Fin: La instancia del caso de uso puede llegar a su fin, cuando el actor cancele el registro del reintegro desde el formulario de captura de la fecha y hora.

- **Consultar Información**

Actor: Cualquier rol.

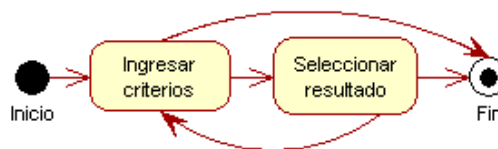
Propósito: Consultar información general de apoyo al trámite de solicitudes.

Resumen: Permite realizar consultas de información general como el número y el tipo de documento de identidad de funcionarios, así como nombre de cargos y unidades académico administrativas; esta información sirve de apoyo al trámite de ciertas solicitudes.

Precondición: Encontrarse en el trámite de alguna solicitud donde se requiera consultar información del tipo descrito anteriormente.

Flujo de sucesos: Los flujos identificados son presentados en la figura 38.

Figura 38. Diagrama de estados - Consultar Información.



Camino básico:

Ingresar criterios: Iniciado el caso de uso se presenta un formulario en el cual se solicitan los criterios con los cuales se realizará la consulta. Seguidamente se presentan los resultados en forma de un listado con datos específicos. Según el tipo

de consulta realizada, la información tendrá un límite de registros a presentar como resultado.

Seleccionar resultado: Una vez presentado el listado con los resultados de la consulta, el usuario selecciona la información requerida. Esta selección se permite en una cantidad mayor a uno, para ciertas consultas.

Fin: La instancia del caso de uso llega a su fin, cuando el actor incluye la información seleccionada en la parte donde es requerida para continuar con el trámite de una solicitud específica.

Caminos alternativos:

Ingresar criterios – Fin: Esto ocurre cuando el actor no ingresa los datos en el formulario y cierra la opción de la consulta.

Seleccionar resultado – Ingresar criterios: Permite redefinir una consulta con el fin de precisar de una mejor manera los criterios de dicha consulta.

- **Cambiar Rol**

Actor: Cualquier rol.

Propósito: Cambiar el rol con el que se actúa en el sistema.

Resumen: Le permite a un usuario que tenga más de un rol válido en el sistema, cambiar en un momento determinado su rol, sin salir del sistema.

Precondición: Tener más de un rol válido para ingresar al sistema y encontrarse en la página de Inbox.

Flujo de sucesos: Los flujos identificados son presentados en la figura 39.

Figura 39. Diagrama de estados – Cambiar Rol.



Camino básico:

Seleccionar rol: En este estado el usuario selecciona el nuevo rol. Los roles son seleccionados de una lista de roles válidos para ingresar al sistema por persona.

Fin: El caso de uso llega a su fin cuando el sistema presenta todas las opciones, funcionalidades e información para el nuevo rol.

- **Cruzar Tiempo**

Actor: Cualquier rol.

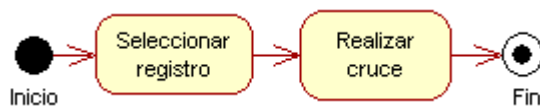
Propósito: Utilizar el tiempo adicional trabajado para compensar tiempo disfrutado mediante permiso.

Resumen: Los permisos que se tienen en cuenta para este caso, son los del tipo *Para Compensar*, cuando una persona disfruta de un permiso de este tipo, queda pendiente de trabajar el tiempo aprobado. Dicho pago se hace utilizando el tiempo adicional trabajado que se encuentre registrado en el sistema.

Precondición: Una vez seleccionada la opción de cruzar tiempos desde el menú, el sistema presentará un listado con las solicitudes disfrutadas de permisos para compensar. Además la persona debe contar con tiempo adicional trabajado registrado en el sistema.

Flujo de sucesos: Los flujos identificados son presentados en la figura 40.

Figura 40. Diagrama de estados – Cruzar Tiempo.



Camino básico:

Seleccionar registro: Una vez presentada la solicitud que se quiere pagar, se selecciona el registro o registros de tiempo adicional trabajado que se utilizarán para

realizar esta acción. En este estado además, se verifica que el registro o registros seleccionados cubran de manera total o parcial el permiso.

Realizar cruce: En este estado se registra en el sistema el tiempo adicional utilizado para cubrir los permisos *Para Compensar* que fueron aprobados.

Fin: La instancia del caso de uso llega a su fin cuando se actualizan los registros de tiempo adicional trabajado.

- **Registrar Grupales**

Actor: Jefe Unidad División de Recursos Humanos.

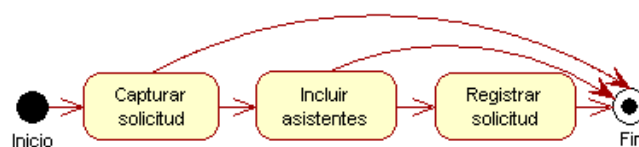
Propósito: Registrar permisos aprobados para miembros de asociaciones de empleados y en general al personal de planta de la Universidad.

Resumen: Este caso de uso le permite al actor correspondiente registrar en el sistema un tipo de permiso dirigido a varias personas, las cuales pueden confirmar su asistencia para disfrutar del permiso correspondiente.

Precondición: Se asume que el permiso fue previamente aprobado para el grupo de personas a las cuales se va a dirigir.

Flujo de sucesos: Los flujos identificados son presentados en la figura 41.

Figura 41. Diagrama de estados – Registrar Grupales.



Camino básico:

Capturar solicitud: Durante este estado el actor correspondiente se encarga de registrar en el sistema el período solicitado, la justificación y selecciona los asistentes de acuerdo al tipo de permiso que seleccione.

Incluir asistentes: Una vez seleccionado el tipo de asistentes, se procede a realizar la inclusión. La inclusión se puede hacer de dos maneras: mediante el documento de identificación de las personas o mediante algunos criterios de selección, como son: el género, la fecha de vinculación, el código de cargo actual, entre otros.

Registrar solicitud: Durante este estado se registra la solicitud en el sistema, la cual queda pendiente de trámite por parte de cada uno de los asistentes seleccionados.

Fin: La instancia del caso de uso llega a su fin cuando mediante una interfaz de confirmación, se comprueba el registro de la misma. Además se actualizan los estados de la solicitud para cada uno de los asistentes seleccionados, quedando pendiente la confirmación de asistencia, en el listado de pendientes de la página de Inbox de cada uno de los mismos.

La confirmación de asistencia es el trámite por medio del cual cada uno de los asistentes al permiso tramita la solicitud ante su jefe inmediato.

Caminos alternativos:

Capturar solicitud – fin, Incluir asistentes – fin: Durante éstos estados la persona puede cancelar el registro de la solicitud.

- **Registrar Causados**

Actor: Jefe de Unidad.

Propósito: Permitir el registro de permisos causados en el sistema.

Resumen: Este caso de uso le permite a los actores *Jefe de Unidad*, registrar un permiso para los empleados a su cargo, cuyo inicio se produjo antes del momento de realizar el registro en el sistema.

Precondición: Se asume que el permiso ya ha comenzado al momento de realizar el registro. Sólo se permiten registrar las situaciones de permisos.

Flujo de sucesos: Los flujos identificados son presentados en la figura 42.

Figura 42. Diagrama de estados – Registrar Causados.



Camino básico:

Capturar solicitud: En este primer estado se captura la solicitud. Entre otros datos que se registran, están el período que se disfrutó, el tipo de la situación de permiso que se dió y la justificación. Además se validan los datos correspondientes al tipo de permiso que se registre. Si el permiso al momento del registro ha terminado, el sistema le permite el registro del reintegro en el mismo formulario.

Confirmar registro: Una vez ingresados los datos, se presenta una confirmación antes de realizar el registro en el sistema. Durante este estado el actor puede cancelar el registro.

Registrar solicitud: Cuando los datos son confirmados, éstos son registrados en el sistema y la solicitud puede quedar en espera de registro de reintegro cuando no se registra esta información en el formulario.

Fin: La instancia del caso de uso llega a su fin cuando se presenta una interfaz que confirme el registro del permiso en el sistema.

Camino alternativo:

Capturar solicitud: Este estado ocurre cuando al capturar los datos del formulario existen errores en alguno de sus campos.

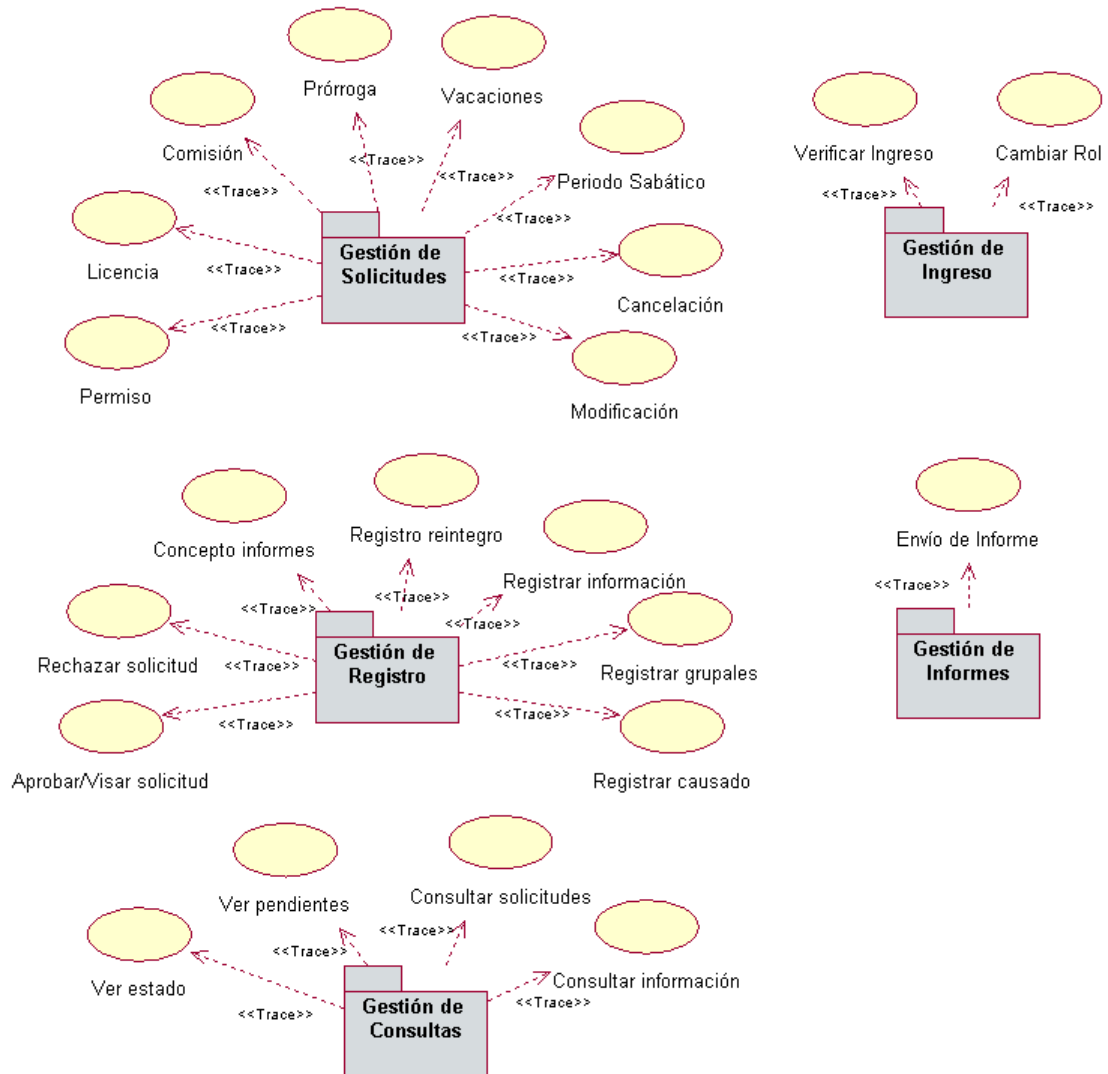
4.2 FLUJO DE ANÁLISIS

4.2.1 Análisis de la arquitectura. En la figura 43, se presentan los casos de uso generales encontrados hasta el momento y la manera en la cual se agrupan en

forma de paquetes. Estos paquetes representan una nueva versión de los presentados durante la Fase de Inicio, adaptado a los nuevos casos de uso.

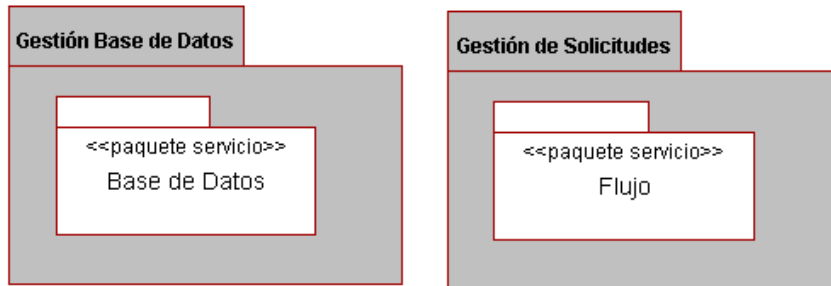
En la figura, el paquete de *Gestión de Solicitudes*, contiene todos los casos de uso concernientes a las solicitudes que maneja el sistema; dichas solicitudes generan un proceso al momento de iniciar su trámite. El paquete de *Gestión de Registro*, contiene todos los casos de uso que manejan acciones sobre las solicitudes como visar, aprobar, rechazar y funcionalidades de registro sobre la base de datos. El paquete de *Gestión de Consultas*, maneja todo lo concerniente a consultas desde la captura de criterios hasta la presentación de resultados. *Gestión de Ingreso*, contiene los casos de uso encargados de verificar el ingreso de los usuarios al sistema. Finalmente, el paquete de *Gestión de Informes*, contiene el caso de uso encargado del envío y registro de informes de ciertas solicitudes que el sistema maneja.

Figura 43. Paquetes de Análisis relacionados con casos de uso.



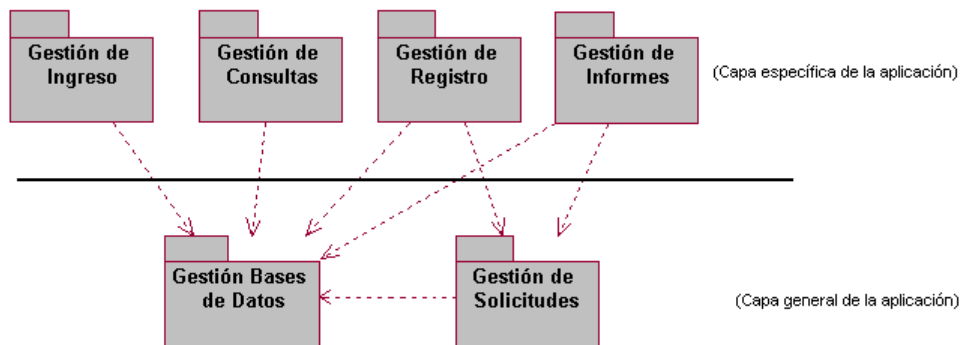
Como se muestra en la figura 44, además del paquete de *Gestión de base de datos* encontrado en la Fase de Inicio, se integró al paquete de *Gestión de Solicitudes* el paquete de *Flujo* presentado durante la misma fase. Este paquete contiene casos de uso abstractos, los cuales se encargan de encaminar las solicitudes por el proceso, actividades y el flujo de las mismas adecuado a su trámite.

Figura 44. Paquetes de servicio



La figura 45, presenta las dependencias de los paquetes presentados anteriormente. Allí se puede apreciar que todos los paquetes utilizan los casos de uso que contiene el paquete *Gestión de Base de Datos*. Así como, el paquete de servicio contenido en *Gestión de Solicitudes*, es usado por los paquetes *Gestión de Registro* y *Gestión de Informes*, con el fin de encaminar los casos de uso asociados en estos paquetes de acuerdo al tipo de solicitud que se trate.

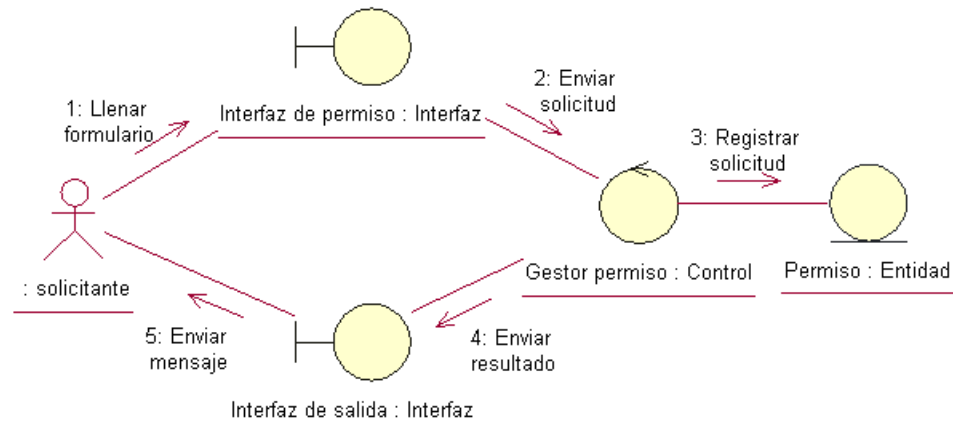
Figura 45. Dependencia entre paquetes.



4.2.2 Análisis de casos de uso. Los mensajes que presentan cada una de las figuras corresponden a la interacción entre instancia de objetos creados para cada una de las clases que se presentan. Los nombres de los actores corresponden a la jerarquía presentada en la figura 29.

- **Solicitar Permiso.** La figura 46¹⁵ presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 46. Diagrama de colaboración – Solicitar Permiso.



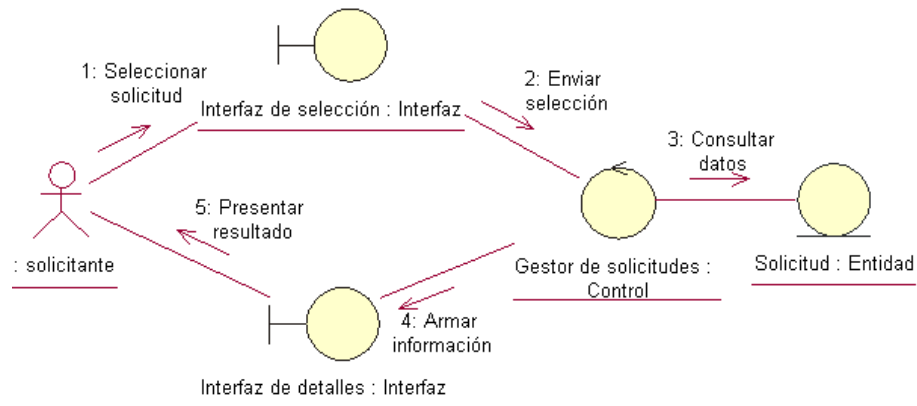
El objeto “Interfaz de Permiso” proporciona el formulario de Solicitud de Permiso correspondiente, el cual procede a llenar (1). El objeto “Interfaz de Permiso” se encarga de verificar que ciertos campos del formulario estén correctamente diligenciados, para luego enviarlo al objeto “Gestor Permiso” (2). Este se encarga de registrar la solicitud en las correspondientes tablas de la base de datos del sistema, identificar el proceso y actividad inicial asociada (3). A continuación el mismo objeto envía a “Interfaz de Salida” los parámetros que contendrá el mensaje de confirmación que se desplegará al actor (4). El objeto “Interfaz de Salida” presenta el mensaje (5).

Si el solicitante tuvo algún error cuando llenó el formulario, el objeto “Interfaz de Solicitud” devuelve el formulario indicando los campos que deben ser reingresados (6), de igual manera sucede con el objeto “Gestor permiso” el cual no permite el registro de la solicitud si ésta contiene errores.

¹⁵ En la figura se muestra al actor como “:solicitante”, esto indica que quién realiza el caso de uso es cualquier usuario, de acuerdo a la jerarquía mostrada.

- **Ver Estado.** La figura 47 presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 47. Diagrama de colaboración – Ver Estado.

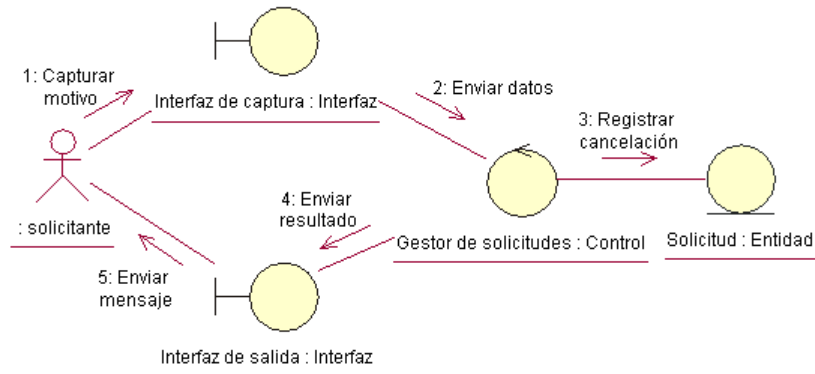


Una vez seleccionada la solicitud (1), se envían datos que identifican a dicha solicitud, como puede ser el número y el tipo (2), al objeto *Gestor de solicitudes*. Este objeto se encarga de consultar los datos correspondientes a la solicitud seleccionada (3). A continuación el mismo objeto (*Gestor de solicitudes*), se encarga de armar la información correspondiente a la solicitud, con los datos que fueron consultados (4). Finalmente el caso de uso termina cuando el objeto *Interfaz de detalles* presenta al actor la información solicitada.

- **Cancelar Solicitud.** La figura 48 presenta como se desarrolla el caso de uso a través de clases de análisis.

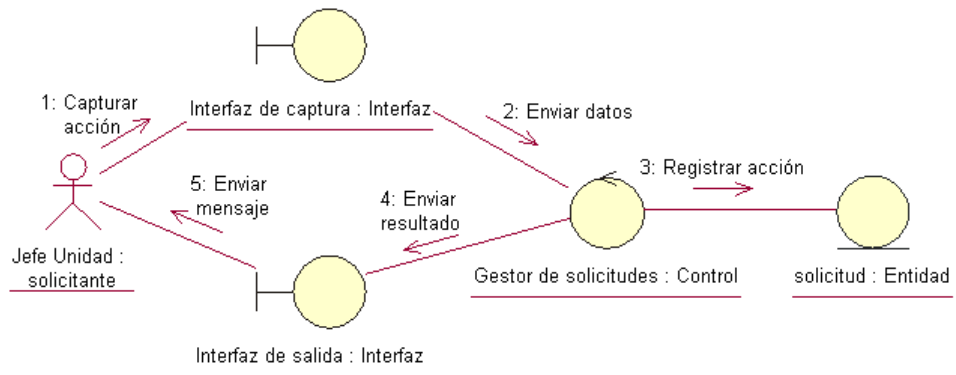
El actor ingresa el motivo por el cual quiere cancelar la solicitud, a través del objeto *Interfaz de captura* (1). Dicho objeto envía los datos necesarios a *Gestor de solicitudes*, con el fin de realizar la cancelación (2). Dicho objeto se encarga de registrar la cancelación para la solicitud en el sistema (3). Además se encarga de enviar el resultado del registro al objeto *Interfaz de salida* (4). La instancia del caso de uso finaliza una vez que el objeto presente el mensaje correspondiente (5).

Figura 48. Diagrama de colaboración – Cancelar Solicitud.



- **Aprobar/Visar Solicitud.** La figura 49 presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 49. Diagrama de colaboración – Aprobar/Visar Solicitud.



A través del objeto *Interfaz de captura*, el actor Jefe de Unidad registra la aprobación de la solicitud junto con las observaciones que se consideren (1). Este objeto se encarga de enviar los datos requeridos por *Gestor de solicitudes* para registrar dicha aprobación (2). El objeto *Gestor de solicitudes* se encarga de verificar si la acción que se registra es de aprobación o de visado, lo cual es transparente para el actor¹⁶, seguidamente este mismo objeto se encarga del

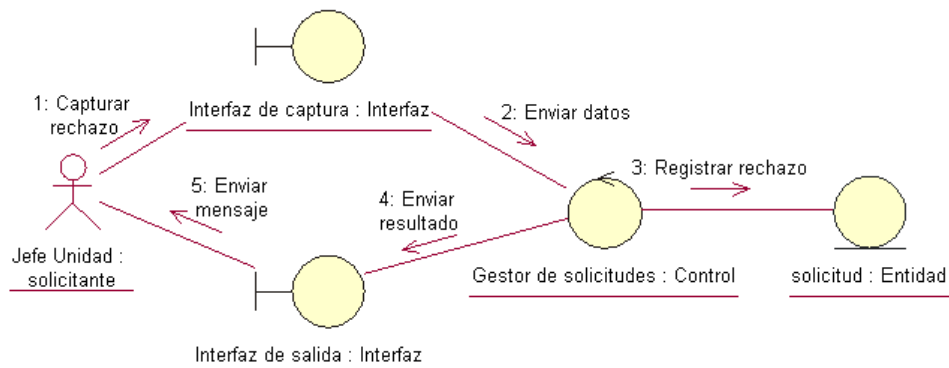
¹⁶ En la figura 49 aparece el actor Jefe Unidad, pero de acuerdo con la jerarquía establecida en la figura 29 este caso de uso es llevado a cabo también por los actores: Decano, Vicerrector y Rector.

registro sobre la solicitud (3). Luego, se encarga de enviar el resultado obtenido al objeto *Interfaz de salida* (4), el cual presenta el mensaje al actor (5).

- **Rechazar Solicitud.** La figura 50 presenta como se desarrolla el caso de uso a través de clases de análisis.

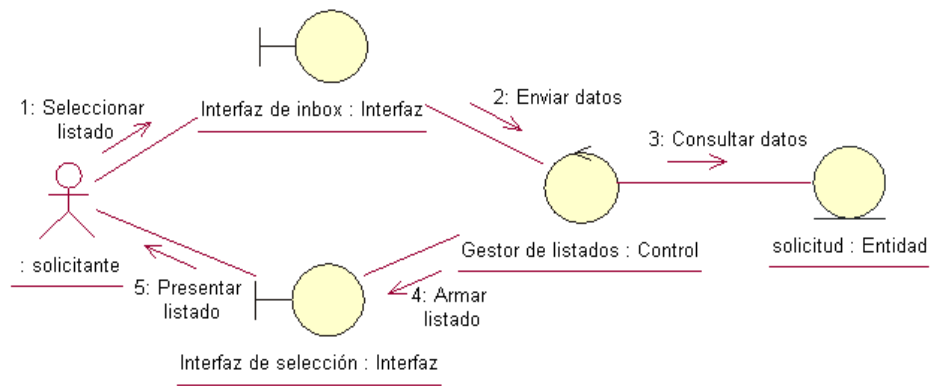
El actor Jefe Unidad a través del objeto *Interfaz de captura* registra el rechazo de la solicitud junto con los motivos por los cuales se rechaza (1). Dicho objeto envía éstos datos, junto con otros necesarios para realizar el registro sobre la solicitud (2). El objeto *Gestor de solicitudes* se encarga de registrar el rechazo de la solicitud junto con el motivo (3). Este objeto se encarga también de enviar el resultado del registro al objeto *Interfaz de salida* (4). Finalmente, éste último presenta un mensaje al actor (5).

Figura 50. Diagrama de colaboración – Rechazar solicitud.



- **Ver pendientes.** La figura 51 presenta como se desarrolla el caso de uso a través de clases de análisis.

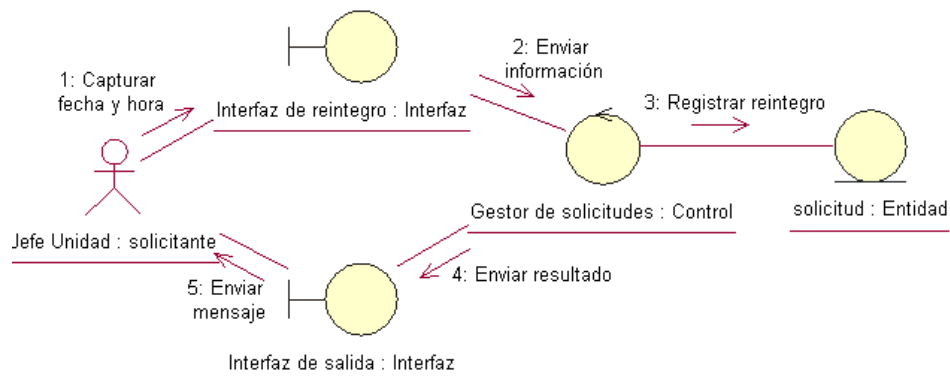
Figura 51. Diagrama de colaboración – Ver pendientes.



El actor mediante su *Interfaz de Inbox*, selecciona el listado que desea consultar (1). Estos datos son tomados por este objeto y enviados al *Gestor de Listados* (2). Dicho objeto se encarga de consultar los datos necesarios de acuerdo al listado seleccionado (3), además arma el listado con los datos consultados (4). Seguidamente el listado es recibido por el objeto *Interfaz de selección*, el cual se encarga de presentar el listado solicitado al actor (5).

- **Registrar Reintegro.** La figura 52 presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 52. Diagrama de colaboración – Registrar Reintegro

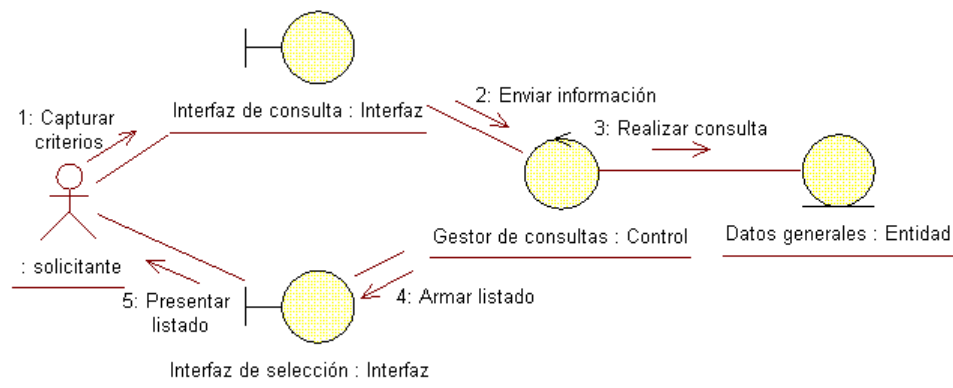


Una vez seleccionada la solicitud para la cual el actor registra el reintegro, el objeto *Interfaz de reintegro* presenta la solicitud y captura la fecha y hora del

reintegro (1). Seguidamente el mismo objeto envía a *Gestor de solicitudes*, los datos necesarios para realizar el reintegro (2). El objeto *Interfaz de reintegro*, junto con *Gestor de solicitudes*, se encargan de verificar los datos ingresados en (1). El objeto *Gestor* se encarga de hacer el registro sobre la solicitud cuando la verificación de datos sea correcta (3). Además se encarga de enviar el resultado del registro y armar el mensaje que luego será presentado al actor (4). Finalmente, el mensaje es presentado mediante el objeto *Interfaz de salida* (5).

- **Consultar Información.** La figura 53 presenta como se desarrolla el caso de uso a través de clases de análisis.

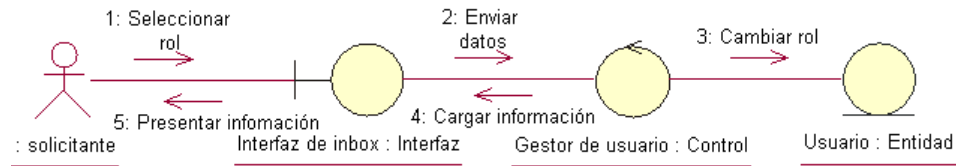
Figura 53. Diagrama de colaboración – Consultar Información.



El objeto *Interfaz de consulta*, presenta los campos mediante los cuales captura los datos requeridos para realizar la consulta (1). Este objeto se encarga de enviar dichos datos a *Gestor de consultas* (2). Seguidamente se realiza la consulta de acuerdo a los criterios ingresados (3). El mismo objeto se encarga de armar el listado, que será presentado como resultado al actor (4). Finalmente, el objeto *Interfaz de selección*, se encarga de presentar el listado (5). La consultas se hacen sobre datos generales tales como documento de identidad, nombres, apellidos y cargos, entre otros.

- **Cambiar Rol.** La figura 54 presenta como se desarrolla el caso de uso a través de clases de análisis.

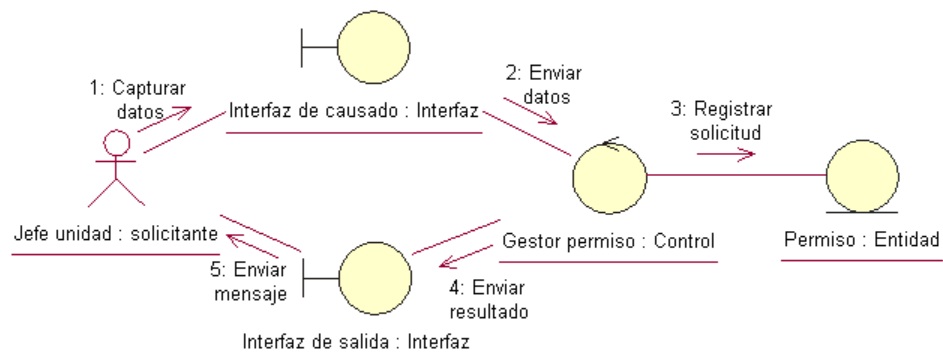
Figura 54. Diagrama de colaboración – Cambiar Rol.



El objeto *Interfaz de Inbox* presenta un listado con todos los roles válidos para usuario en el sistema. El actor selecciona el nuevo rol, a través del mismo objeto (1). Éste se encarga de enviar los datos necesarios para cambiar el rol actual al objeto *Gestor de usuario* (2). Seguidamente dicho objeto se encarga de consultar, cambiar el rol (3) y de obtener la nueva información que se va a presentar para el rol seleccionado (4). Finalmente, el objeto *Interfaz de Inbox*, presenta ésta nueva información al usuario. La nueva información, depende de las restricciones asociadas al rol que fue seleccionado.

- **Registrar Causados.** La figura 55 presenta como se desarrolla el caso de uso a través de clases de análisis.

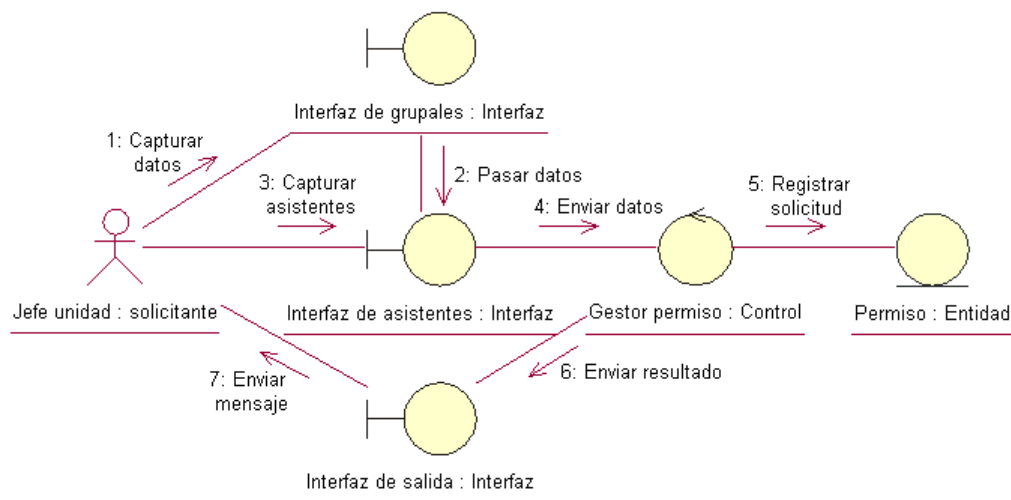
Figura 55. Diagrama de colaboración – Registrar Causados.



Los datos para realizar el registro de permisos causados son capturados mediante el objeto *Interfaz de causado* (1). Los datos solicitados son: el período disfrutado, el motivo y el tipo de situación de permiso que se dió. Dicho objeto se encarga de enviar los datos a un objeto de control, *Gestor permiso* (2). Este objeto es el encargado de registrar la solicitud en el sistema (3). Además se encarga de enviar el resultado del registro al objeto *Interfaz de salida* (4). Finalmente, éste objeto envía el mensaje generado por el objeto *Gestor permiso* (5). En la etapa de captura de datos ocurren algunas validaciones, en particular con el período que se está tratando de registrar, el horario del funcionario y los descuentos que tiene para un posible permiso grupal, entre otros.

- **Registrar Grupales.** La figura 56 presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 56. Diagrama de colaboración – Registrar Grupales.

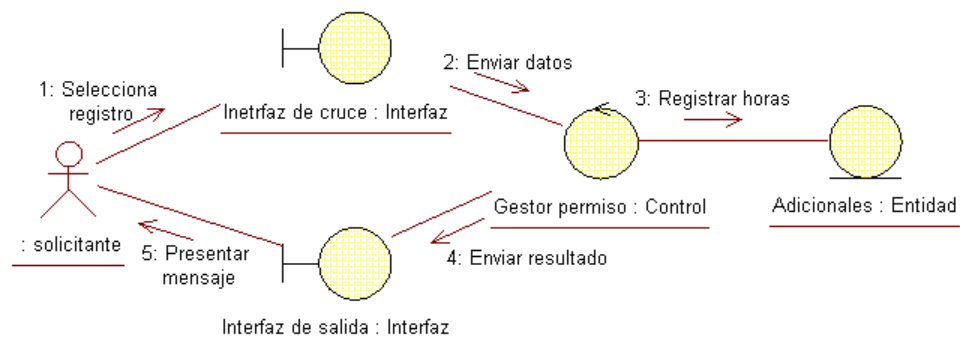


Para este caso de uso, la captura de los datos ha sido dividida en dos interfaces. En la primera de ellas, se ingresan datos como el período, el motivo y el tipo de asistentes a la situación de permiso (1). Éstos datos son pasados a la segunda interfaz de ingreso de datos (2), en la cual el actor correspondiente, ingresa los asistentes al permiso que está registrando (3), para ello utiliza algunos criterios que presenta esta interfaz. Seguidamente los datos en conjunto son enviado a

la clase de control (4), la cual se encarga de realizar el registro correspondiente de la solicitud (5), para cada uno de los asistentes ingresados en el paso anterior. Además el objeto *Gestor permiso*, se encarga de enviar el resultado del registro al objeto *Interfaz de salida* (6). El caso de uso termina, cuando este último objeto presenta el mensaje al actor (7) y el permiso queda a espera de trámite por parte de cada uno de los asistentes relacionados.

- **Cruzar Tiempo.** La figura 57 presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 57. Diagrama de colaboración – Cruzar Tiempo.



Para este caso de uso el actor solicitante debe contar con tiempo adicional trabajado, así como de alguna solicitud de permiso del tipo *Permiso para compensar*. El caso de uso comienza con la selección de los registros de tiempo adicional trabajado que quiere utilizar para compensar el permiso (1). Los datos son enviados a un objeto de control *Gestor permiso* (2). Dicho objeto se encarga de hacer el registro de las horas (3), descontando del tiempo adicional trabajado, así como de descargar las horas de la solicitud que se está tratando; además, se encarga de enviar el resultado a un objeto interfaz (4), el cual se encarga de presentarlo al actor (5).

4.2.3 Análisis de clases. La tabla No.4 contiene la descripción de las clases de análisis involucradas en la realización de los casos de uso detallados anteriormente.

Esta descripción incluye las responsabilidades de cada clase y sus posibles atributos (las clases de control por lo general no poseen atributos).

Tabla 4. Descripción de clases de análisis.

TIPO	NOMBRE	RESPONSABILIDADES	ATRIBUTOS
Interfaz	Interfaz de Solicitud	<ul style="list-style-type: none"> • Proporcionar el formulario de solicitud. • Verificar el formato correcto de los campos del formulario. • Informar en lo posible al usuario de los errores cometidos al diligenciar el formulario. • Presentar confirmación de los datos a enviar. • Enviar los datos recibidos del usuario al objeto "Gestor de Permiso". 	Contiene atributos para la captura de datos de una solicitud como fechas y horas del período a solicitar, así como el tipo y la justificación de la solicitud.
	Interfaz de Salida	<ul style="list-style-type: none"> • Recibir información del objeto "Gestor Permiso" y "Gestor de solicitudes" • Presentar al usuario los mensajes de confirmación, generados por los objetos de las clases de control. 	Presenta mensajes de acciones realizadas en el sistema. Ofrece la posibilidad de salir o continuar en el sistema.
	Interfaz de selección	<ul style="list-style-type: none"> • Presentar listado de solicitudes. • Permitir elegir una solicitud del listado. 	Carga los listados de solicitudes que se le presentarán al actor. Entre otros datos incluye el tipo de solicitud y el nombre de la persona que la tramitó.
	Interfaz de detalles	<ul style="list-style-type: none"> • Presentar los detalles de una solicitud. 	Presenta todas las actividades por las que ha pasado una solicitud.
	Interfaz de captura	<ul style="list-style-type: none"> • Enviar datos al objeto "Gestor de solicitudes". • Capturar la acción de visado o aprobación. • Capturar el rechazo de una solicitud. • Capturar las observaciones y motivos para las acciones sobre las solicitudes. • Capturar datos para cancelar una solicitud (tipo, motivo, entre otros.). 	Se encarga de capturar datos que después servirán como criterios para realizar inserciones o actualizaciones en el sistema.

Interfaz	Interfaz de Inbox	<ul style="list-style-type: none"> • Presentar el menú de opciones del sistema. • Soportar objetos de "Interfaz de selección". • Enviar datos al objeto "Gestor de listados". • Presentar todos los roles válidos para un usuario. • Enviar datos al objeto "Gestor de usuario". • Presentar la información para el actor que se encuentre en el sistema . 	Es la interfaz principal del sistema. Presenta todas las opciones y funcionalidades del sistema para el rol en que se encuentre el usuario en el momento.
	Interfaz de reintegro	<ul style="list-style-type: none"> • Presentar la solicitud que se le va a registrar reintegro. • Capturar fecha y hora de reintegro. • Validar los datos respecto del horario y el período disfrutado. • Enviar información al objeto "Gestor de solicitudes". 	Presenta datos relevantes sobre la solicitud a la que se le va a registrar el reintegro del funcionario. Además captura los datos necesarios para llevar a cabo la acción.
	Interfaz de consulta	<ul style="list-style-type: none"> • Capturar criterios para realizar consultas a la base de datos. • Enviar información al objeto "Gestor de solicitudes". 	Se encarga de presentar medios para capturar los datos que serán usados como criterios de consulta en el sistema.
	Interfaz de causado	<ul style="list-style-type: none"> • Capturar los datos del permiso causado que se quiere registrar. • Enviar estos datos a "Gestor permiso". 	Captura los datos necesarios para registrar un permiso causado en el sistema.
	Interfaz de grupales	<ul style="list-style-type: none"> • Capturar los datos para el registro de un permiso grupal. • Pasar estos datos al objeto "Interfaz de asistentes". 	Captura y envía los datos requeridos para el registro de un permiso grupal.
	Interfaz de asistentes	<ul style="list-style-type: none"> • Capturar los asistentes a un permiso grupal. • Recibir y conservar los datos provenientes del objeto "Interfaz de grupales". • Enviar todos estos datos al objeto "Gestor permiso". 	Se encarga de presentar criterios para la selección de asistentes a un permiso grupal.
	Interfaz de cruce	<ul style="list-style-type: none"> • Presentar el tiempo adicional trabajado, registrado para un actor del sistema. • Permitir seleccionar registros de dicho tiempo, con el fin de pagar por concepto de permisos para compensar. • Enviar los datos al objeto "Gestor permiso". 	Presenta los registros de tiempo adicional, con el fin que sean utilizados para descontar tiempo de una solicitud de permiso para compensar.

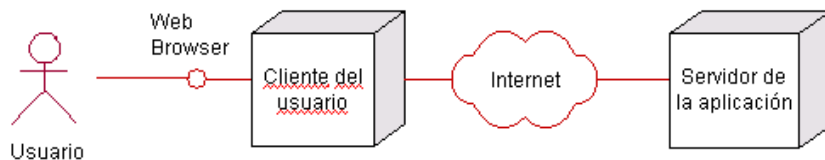
Control	Gestor Permiso	<ul style="list-style-type: none"> • Recibir datos proporcionados por el objeto "Interfaz de Solicitud". • Revisar los datos recibidos. • Consultar el consecutivo de la solicitud (Número de solicitud). • Identificar el flujo de la solicitud de acuerdo a su tipo. • Actualizar el sistema, registrando la solicitud en el mismo. • Enviar información de confirmación al objeto "Interfaz de Salida". • Registrar permisos causados. • Registrar un permiso grupal. • Registrar el descuento de horas de una solicitud de permiso para compensar. 	
	Gestor solicitudes	<ul style="list-style-type: none"> • Analizar el tipo de solicitud y encaminarla de acuerdo a un flujo establecido. • Indicar cuales datos deben ser cargados para luego presentarse al actor. • Registrar la cancelación de una solicitud. • Registrar acciones sobre la solicitud (aprobación o visado). • Registrar el rechazo de una solicitud. • Registrar las observaciones de las acciones para una solicitud. • Enviar resultados al objeto "Interfaz de salida". • Registrar el reintegro para el funcionario de una solicitud. • Realizar consultas de acuerdo a los criterios capturados. • Armar información para ser presentada por el objeto "Interfaz de detalles". 	
	Gestor de listados	<ul style="list-style-type: none"> • Generar listados de información. • Enviar resultados a "Interfaz de pendientes". 	
	Gestor de usuario	<ul style="list-style-type: none"> • Consultar y cambiar de rol a un usuario. • Cargar información para el nuevo rol. • Enviar información al objeto "Interfaz de Inbox". 	

	Gestor de consultas	<ul style="list-style-type: none"> • Consultar datos generales en el sistema, a partir de ciertos criterios. • Enviar resultados a "Interfaz de Selección" 	
Entidad	Solicitud	<ul style="list-style-type: none"> • Proporcionar a los objetos de las clases de control, la información que éstos requieran de la base de datos. • Mantener los datos que los objetos de las clases de control han registrado en la base de datos. 	<ul style="list-style-type: none"> • Apuntador a la Base de Datos (conexión abierta) • Realizador de consultas • Realizador de actualizaciones, inserciones y borrado.
	Permiso	<ul style="list-style-type: none"> • Mantener todos los datos asociados al procedimiento de permisos. 	
	Usuario	<ul style="list-style-type: none"> • Mantener todos los datos asociados a los usuarios del sistema. • Proporcionar al objeto "Gestor de usuario" la información que éste requiera de la base de datos. 	
	Datos generales	<ul style="list-style-type: none"> • Algunos de los datos generales son: unidades académico-administrativas, los mensajes generados por el sistema y cargos, entre otros. Esta clase de entidad se encarga de proporcionar los datos requeridos por las clases de control. 	

4.3 FLUJO DE DISEÑO

4.3.1 Diseño de la arquitectura

Figura 58. Modelo de despliegue.

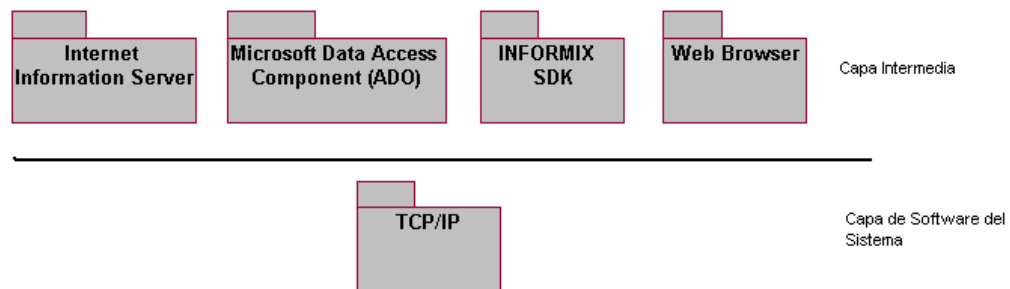


El sistema se ejecutará sobre dos nodos, un nodo servidor y una serie de nodos clientes, como lo muestra la figura 58. Se tiene un nodo servidor en el cual se ubica tanto la lógica del negocio como la base de datos (aunque esta última puede

ubicarse físicamente en otro equipo de cómputo). Un usuario accede al sistema mediante nodos clientes, a través de un navegador (*Web Browser*), los cuales se comunican con el nodo servidor utilizando el protocolo TCP/IP de Internet. Como se explicó anteriormente en el nodo cliente se ubican el actor Solicitante, y los actores asociados: Docente, Decano, Jefe de Unidad, Jefe División de Recursos Humanos, Vicerrector y Rector.

- **Subsistemas intermedios y de software del sistema.** Para la implementación del sistema se utilizará la tecnología de creación de páginas Web *ASP* y como motor de bases de datos se usará *INFORMIX*. La distribución de estos subsistemas en las capas aparece en la figura 59.

Figura 59. Subsistemas intermedios y de software del sistema.



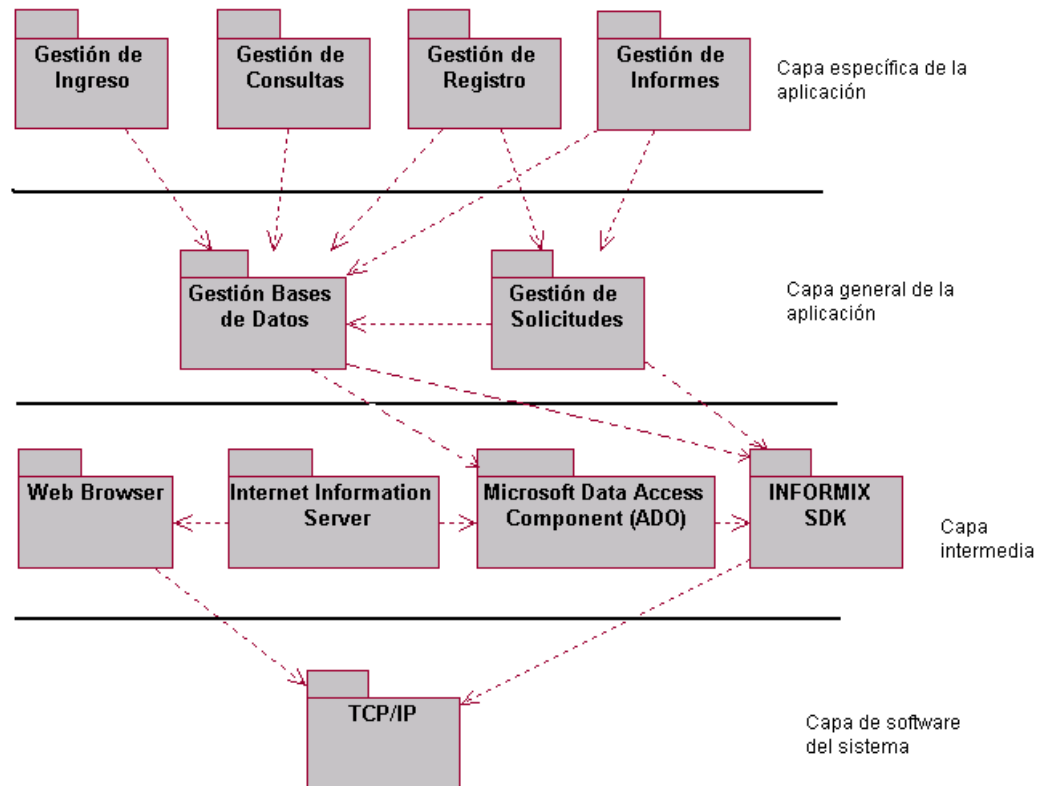
El paquete *Internet Information Server*, agrupa todas las funcionalidades que el sistema utiliza de este servidor de aplicaciones; *Microsoft Data Access Components (ADO)*, son todos los componentes que permiten el acceso a los datos; *INFORMIX SDK* es el paquete de desarrollo de Informix, el cual permite a las aplicaciones interactuar con esta base de datos a través de ODBC. El *Web Browser*, corresponde al navegador de Internet y *TCP/IP* hace referencia al protocolo TCP/IP de Internet.

La dependencia entre los subsistemas identificados hasta el momento se encuentran en la figura 60.

Los subsistemas de las dos primeras capas, corresponden a los paquetes de análisis identificados durante el flujo anterior (figura 45). Estos adquieren la

concepción de subsistemas ya que pasan de ser de la agrupación de casos de uso a un conjunto de métodos que realizan la funcionalidad mencionada.

Figura 60. Dependencias entre subsistemas.



4.3.2 Diseño de los casos de uso. Con el diseño de los casos de uso más significativos para la arquitectura (los calificados como críticos) se busca identificar las clases de diseño que serán necesarias para llevar a cabo el flujo de implementación.

Para realizar el diseño de cada uno de los casos de uso se identificarán primero las clases de diseño participantes en éstos a partir de las clases de análisis presentadas en el flujo anterior y posteriormente se describirán las interacciones entre instancias de dichas clases de diseño.

- **Solicitar Permiso.** La tabla 5 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

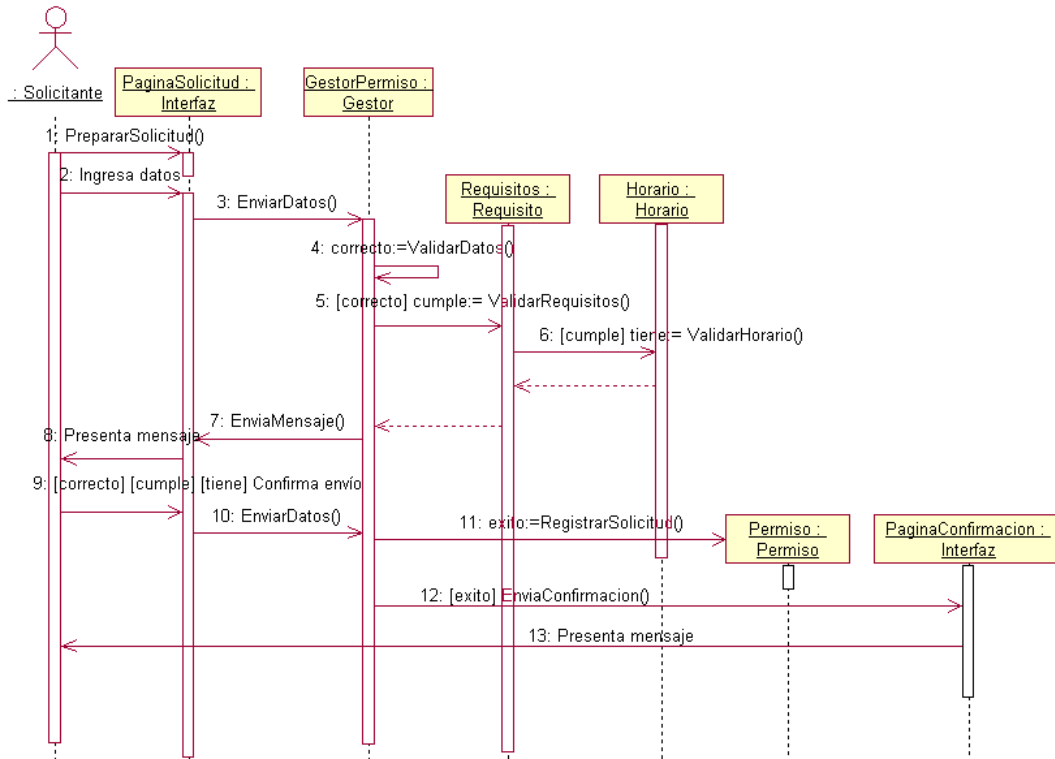
Tabla 5. Clases de diseño – Solicitar Permiso

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de permiso	PaginaSolicitud
Interfaz de salida	PaginaConfirmacion
Gestor permiso	GestorPermiso
	Requisitos
	Horario
Permiso	Permiso

A continuación, la figura 61 presenta el diagrama de secuencia de estas clases.

El caso de uso comienza cuando un objeto de la clase *Interfaz* (*PaginaSolicitud*) es solicitado por parte de un actor del sistema. En este momento, se prepara la solicitud de permiso, cargando la interfaz de captura de datos (1). En ella el actor ingresa los datos que hacen parte del permiso (2), entre los cuales se encuentran las fechas y horas de inicio y fin y la justificación. Cuando el actor decide enviar su solicitud, el sistema lleva los datos a una clase *Gestor* (3), esta clase se encarga de realizar una comprobaciones iniciales de los datos ingresados; algunas de ellas son: verificar la correspondencia de los formatos de las fechas ingresadas, que las mismas no sean a días festivos, entre otras (4).

Figura 61. Diagrama de secuencia – Solicitar Permiso.



Seguidamente se validan otros requisitos mediante la clase *Requisito*, como son el tiempo de anticipación y el límite para el permiso seleccionado (5). Además por medio de la clase *Horario*, se valida que la persona haya solicitado el permiso en un período laboral de acuerdo a su horario de trabajo (6). Una vez hechas todas las comprobaciones requeridas, la clase *Gestor* envía un mensaje a la clase *Interfaz* (7), la cual se encarga de presentarlo al actor (8). Si cumplió con todas las validaciones y el usuario confirma el envío de la solicitud (9), los datos son llevados de nuevo a la clase *Gestor* (10), para hacer el registro correspondiente de la solicitud en el sistema.

En este momento se crea un objeto de la clase *Permiso*, el cual contiene todos los datos ingresados por el actor, esta información se registra en el sistema (11). Si todo ocurre exitosamente, se envía un mensaje a una clase de *Interfaz* (12), con el fin de presentarlo al actor (13), con lo que finaliza la realización del caso de uso.

Durante la acción (7), si la solicitud no cumple con alguna validación en cualquiera de las clases que intervienen, la clase *Gestor*, se encarga de presentar los errores ocurridos. Además si el actor desea continuar, el proceso comienza nuevamente (vuelve a la acción 2, en este caso con los nuevos datos ingresados o corregidos).

Si en la acción (11), se llega a presentar algún fallo en el registro del permiso en el sistema, la clase *Gestor*, prepara un mensaje el cual informa que el proceso no se cumplió satisfactoriamente y la clase *Interfaz* (*PaginaConfirmacion*) se encarga de presentarlo al actor.

Las líneas punteadas que no tienen una etiqueta asociada corresponden a los mensajes que retornan las funciones, que por lo general son valores lógicos (verdadero o falso).

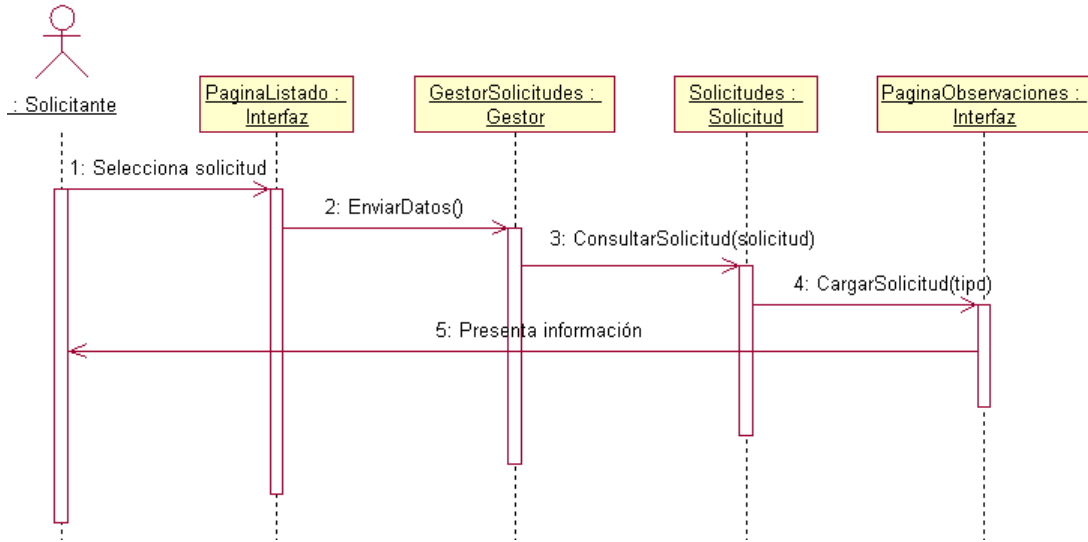
- **Ver Estado.** La tabla 6 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 6. Clases de diseño – Ver Estado.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de selección	PaginaListado
Interfaz de detalles	PaginaObservaciones
Gestor de Solicitudes	GestorSolicitudes
Solicitud	Solicitudes

A continuación, la figura 62 presenta el diagrama de secuencia de estas clases.

Figura 62. Diagrama de secuencia - Ver Estado.



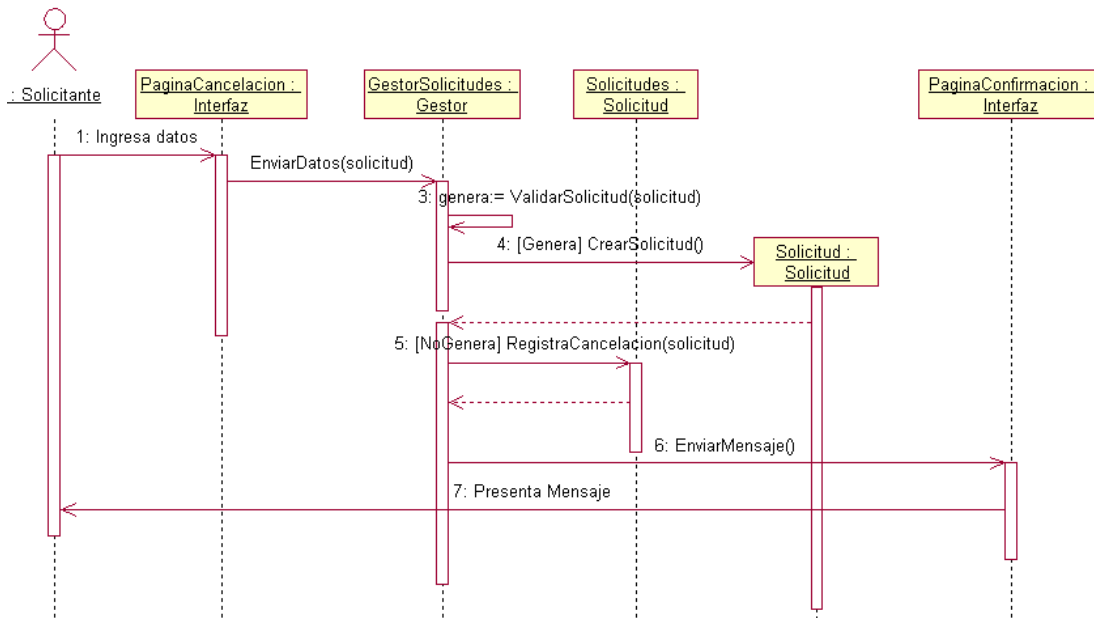
La secuencia de acciones para este caso de uso comienza cuando, un actor selecciona de una interfaz de listado de solicitudes, la solicitud que quiere consultar el estado (1). El dato seleccionado se envía a una clase *Gestor* (*GestorSolicitudes*) (2). Esta clase se encarga de consultar los datos para la solicitud (3). Dicha consulta se lleva a cabo en un objeto de la clase *Solicitud*, la cual además se encarga de seleccionar el tipo de interfaz en el cual se va a presentar la solicitud seleccionada en (1), de acuerdo al tipo de la solicitud que fue seleccionada. Finalmente, un objeto de la clase *Interfaz* (*Observaciones*) presenta la información que fue consultada para esa solicitud.

- **Cancelar Solicitud.** La tabla 7 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso. a figura 63 presenta el diagrama de secuencia de estas clases.

Tabla 7. Clases de diseño – Cancelar Solicitud.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de captura	PaginaCancelacion
Interfaz de salida	PaginaConfirmacion
Gestor de Solicitudes	GestorSolicitudes
Solicitud	Solicitudes
	Solicitud

Figura 63. Diagrama de secuencia – Cancelar Solicitud.



Este caso de uso comienza una vez que el actor haya seleccionado del listado presentado la solicitud que desea cancelar. Seguidamente a esto, el sistema le presenta una interfaz donde se muestra la solicitud elegida, además de ciertos campos que se deben llenar (1). Los campos requeridos son: el tipo de cancelación (parcial o total), el motivo y una fecha de inicio de cancelación cuando el tipo de cancelación escogida fue parcial.

Los datos capturados para la solicitud son enviados a la clase *Gestor* (2), la cual se encarga de validar los datos ingresados así como de determinar si se genera una solicitud de cancelación o simplemente se cancela directamente la solicitud elegida (3). Para el caso en que se genere una nueva solicitud, que en este caso será una solicitud de cancelación, ésta se crea con los datos ingresados en la interfaz (4). Una vez creada la solicitud, la clase se encarga de generar un mensaje (6) que será presentado al actor mediante una clase de *Interfaz* (7).

En el caso de no generar una nueva solicitud, se registra la cancelación para la solicitud que fue seleccionada, actualizando el estado de dicha solicitud (5).

En el diagrama, para evitar confusiones se decidió llamar de forma diferente las dos instancias de la clase *Solicitud*: Solicitudes, para manejar la solicitud que se está cancelando y Solicitud para indicar la creación de una nueva.

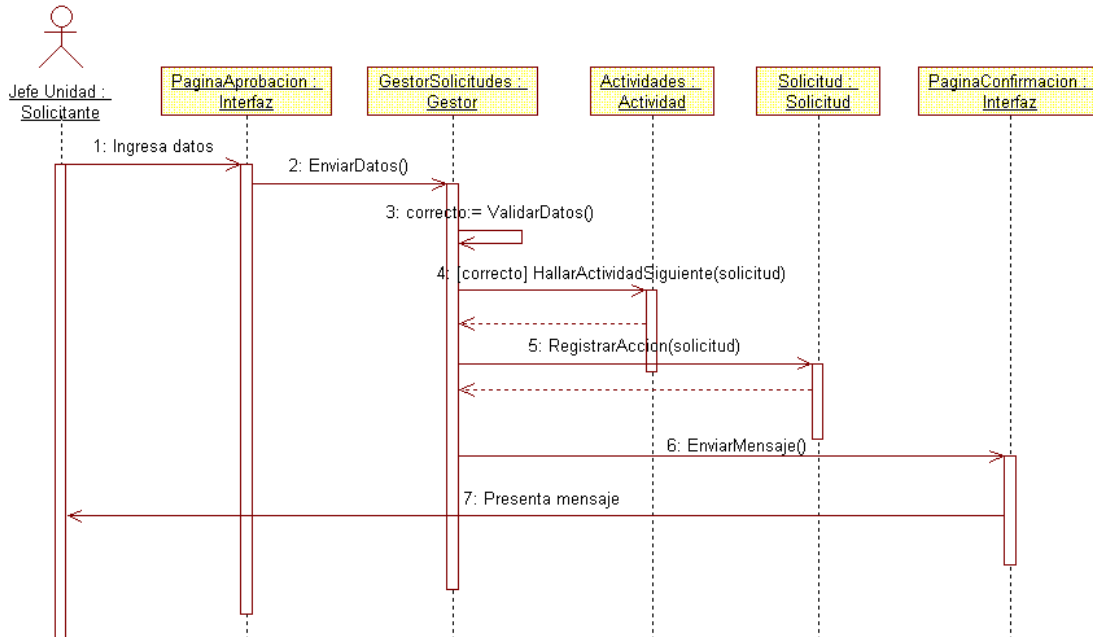
- **Aprobar/visar Solicitud.** La tabla 8 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 8. Clases de diseño – Aprobar/visar Solicitud.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de captura	PaginaAprobacion
Interfaz de salida	PaginaConfirmacion
Gestor de Solicitudes	GestorSolicitudes
Solicitud	Solicitud
	Actividades

A continuación, la figura 64 presenta el diagrama de secuencia de éstas clases.

Figura 64. Diagrama de secuencia – Aprobar/visar Solicitud.



El actor Jefe Unidad ingresa los datos correspondientes en la clase de Interfaz, *PaginaAprobacion* (1). Estos datos dependen tanto del tipo de solicitud, como de la actividad que se vaya a realizar sobre la misma; la clase se encarga de enviar los datos a una clase *Gestor* (2). En dicha clase se validan los datos que se ingresan (3). Una vez se compruebe que los datos fueron ingresados correctamente, se procede a buscar la siguiente actividad, dentro del flujo del proceso que sigue la solicitud (4). Con los datos ingresados y la actividad encontrada en el paso anterior, la clase *Gestor* procede a registrar la acción que se realizó sobre la solicitud (5), ya sea de aprobación o visado. Seguidamente la misma clase se encarga de enviar un mensaje de confirmación a la clase *Interfaz* (6), con el fin de presentarlo al actor (7).

La secuencia de acciones que se llevan a cabo durante el rechazo de una solicitud, es la misma que presenta la figura 64, con lo cual este mismo diagrama es el que sigue el caso de uso *Rechazar Solicitud*.

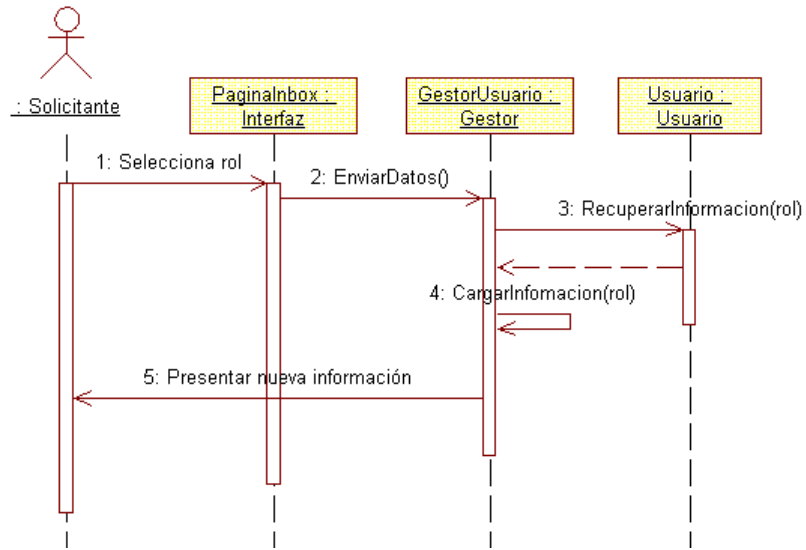
- **Cambiar Rol.** La tabla 9 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 9. Clases de diseño – Cambiar Rol.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de inbox	PaginaInbox
Gestor de Usuario	GestorUsuario
Usuario	Usuario

La figura 65 presenta el diagrama de secuencia de estas clases.

Figura 65. Diagrama de secuencia – Cambiar Rol.



Por medio de la clase de interfaz *PaginaInbox*, el actor puede seleccionar uno de los roles válidos con los que puede ingresar al sistema (1). Una vez seleccionado uno de ellos, los datos son enviados a una clase *Gestor* (2), la cual se encarga de recuperar la información correspondiente al rol seleccionado (3). Además, carga la información consultada (4), la cual será presentada en la misma interfaz donde se seleccionó el rol (5).

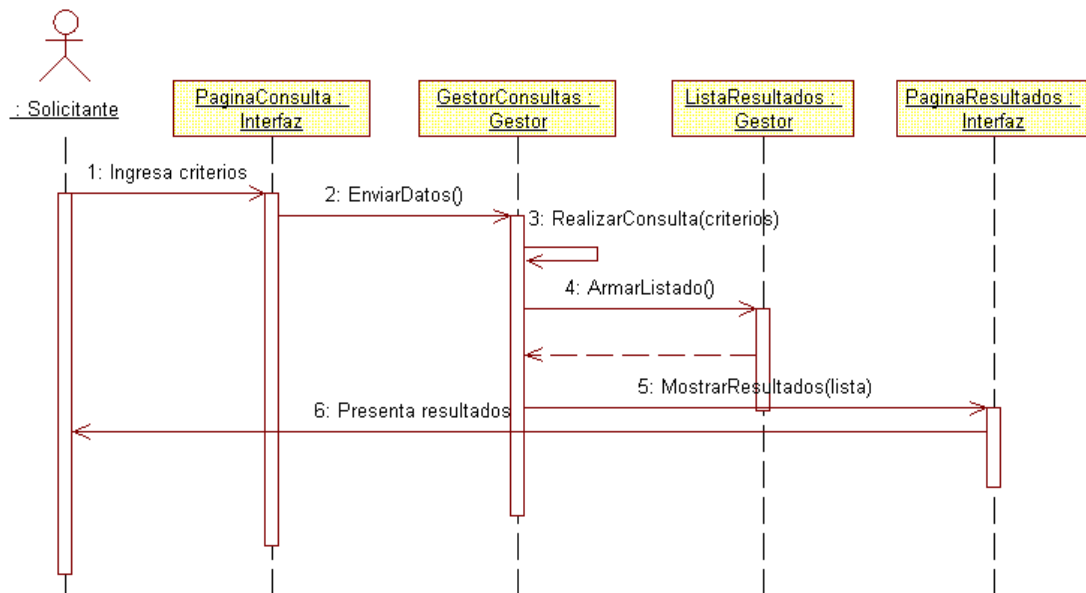
- **Consultar Información.** La tabla 10 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 10. Clases de diseño – Consultar Información.

Clases de análisis	Clases de diseño
Interfaz de consulta	PaginaConsulta
Interfaz de selección	PaginaResultados
Gestor de Consultas	GestorConsultas
	ListaResultados

A continuación, la figura 66 presenta el diagrama de secuencia de estas clases.

Figura 66. Diagrama de secuencia – Consultar Información.



Por medio de una clase de interfaz llamada *PaginaConsulta*, el actor ingresa unos criterios para la búsqueda de información (1). Estos criterios son enviados a una clase *Gestor* (2), la cual se encarga de realizar la consulta en la base de datos de acuerdo a los criterio ingresados (3). Una vez hecha la consulta se crea un listado (4), el cual será presentado como resultado. Seguidamente la instancia *GestorConsultas* de la clase *Gestor* se encarga de dar un formato al resultado de la consulta (5), el cual es presentado mediante una clase de interfaz, *PaginaResultados* (6).

Los criterios para la búsqueda dependen de dónde se utilice el caso de uso, el formato que se aplica a los datos antes de presentarlos, permiten darle mejor presentación y pueden llegar a incluir consultas adicionales.

- **Registrar Causados.** La tabla 11 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

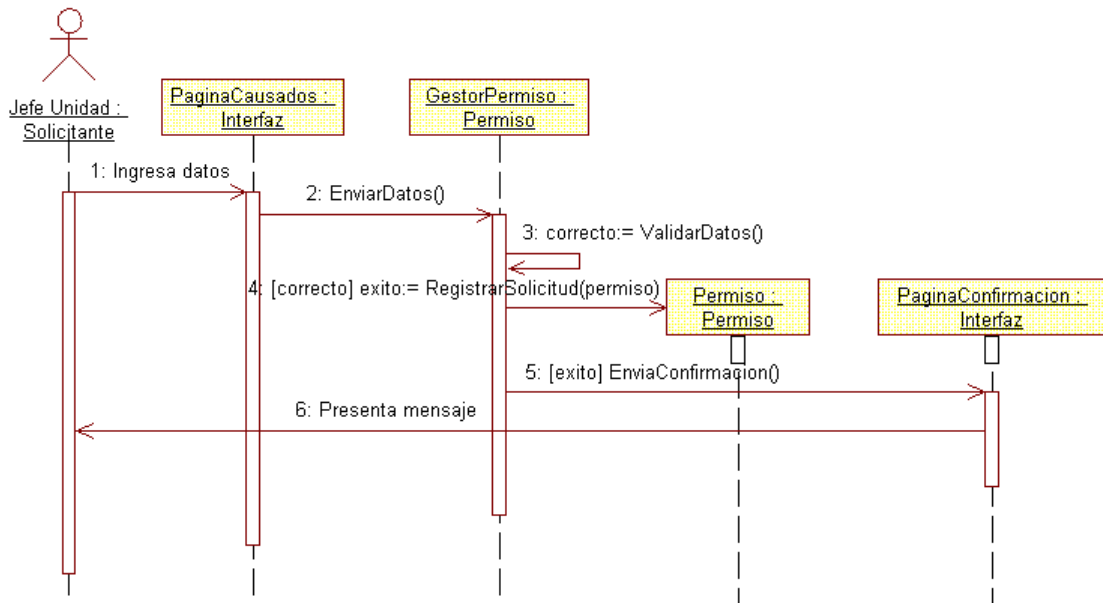
Tabla 11. Clases de diseño – Registrar Causados.

Clases de análisis	Clases de diseño
Interfaz de causado	PaginaCausados
Interfaz de salida	PaginaConfirmacion
Gestor permiso	GestorPermiso
Permiso	Permiso

A continuación, la figura 67 presenta el diagrama de secuencia de estas clases.

El caso de uso comienza cuando el actor Jefe Unidad decide registrar un permiso causado para alguno de los empleados a su cargo. A través de la clase de Interfaz *PaginaCausados*, ingresa los datos correspondientes a esta situación, como son el período que se disfrutó y la justificación (2). Estos datos son enviados a una clase *Gestor* (2), con el fin de validar los datos (3). Las validaciones corresponden a las que se realizan en el caso de uso *Solicitar Permiso*, incluyendo las que se realizan con el horario de la persona a la cual se le está registrando el permiso. Una vez se comprueben que los datos fueron ingresados correctamente, se procede a registrar la solicitud en el sistema (4). Además la clase *Gestor* se encarga de enviar un mensaje de confirmación (5), el cual es presentado por medio de la clase de interfaz *PaginaConfirmacion*.

Figura 67. Diagrama de secuencia – Registrar Causados.



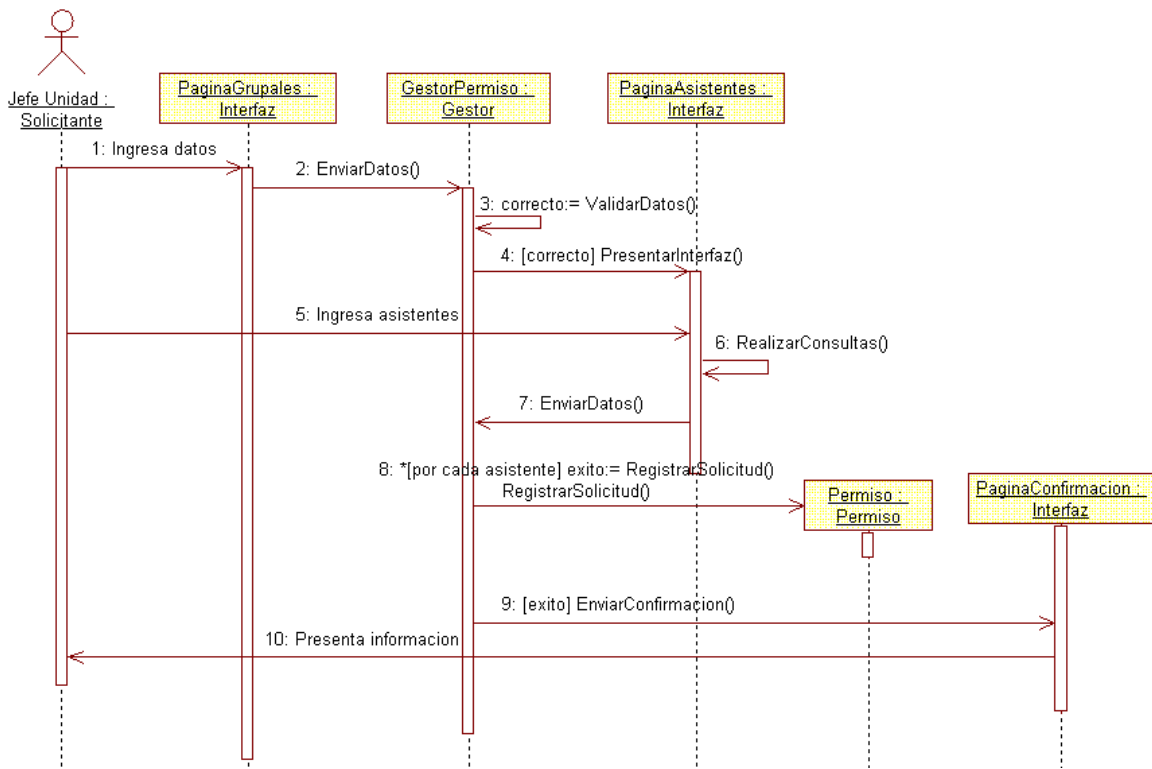
- **Registrar Grupales.** La tabla 12 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 12. Clases de diseño – Registrar Grupales.

Clases de análisis	Clases de diseño
Interfaz de grupales	PaginaGrupales
Interfaz de asistentes	PaginaAsistentes
Interfaz de salida	PaginaConfirmación
Gestor permiso	GestorPermiso
Permiso	Permiso

La figura 68 presenta el diagrama de secuencia de éstas clases.

Figura 68. Diagrama de secuencia – Registrar Grupales.



El actor *Jefe Unidad* ingresa los datos generales correspondientes a este tipo de permiso, a través de una clase de Interfaz, *PaginaGrupales* (1). Estos datos son enviados a una clase *Gestor* (2), la cual se encarga de validar los datos que fueron ingresados (3). Estas validaciones revisa que el actor haya escogido un tipo de asistentes para el permiso que está registrando. Seguidamente, si los datos fueron correctos, se presenta un objeto de la clase Interfaz, *PaginaAsistentes* (4), por medio del cual el actor ingresa los asistentes (5), a partir de ciertos criterios como son: el documento de identificación, el cargo o la unidad académico-administrativa a la cual pertenecen, entre otros. Para ello se hace uso del caso de uso *Consultar información*, en el cual se describe la manera en la que se hacen las consultas en el sistema (6). Una vez el actor confirma el envío de la solicitud, los datos son enviados a la clase gestor *GestorPermiso* (7), la cual se encarga de registrar la solicitud en el sistema (8). El registro incluye la creación de una solicitud, por cada uno de los asistentes

seleccionados en la interfaz anterior. Si el registro ocurrió satisfactoriamente, la clase gestor se encarga de enviar un mensaje de confirmación (9), el cual es presentado al actor (10).

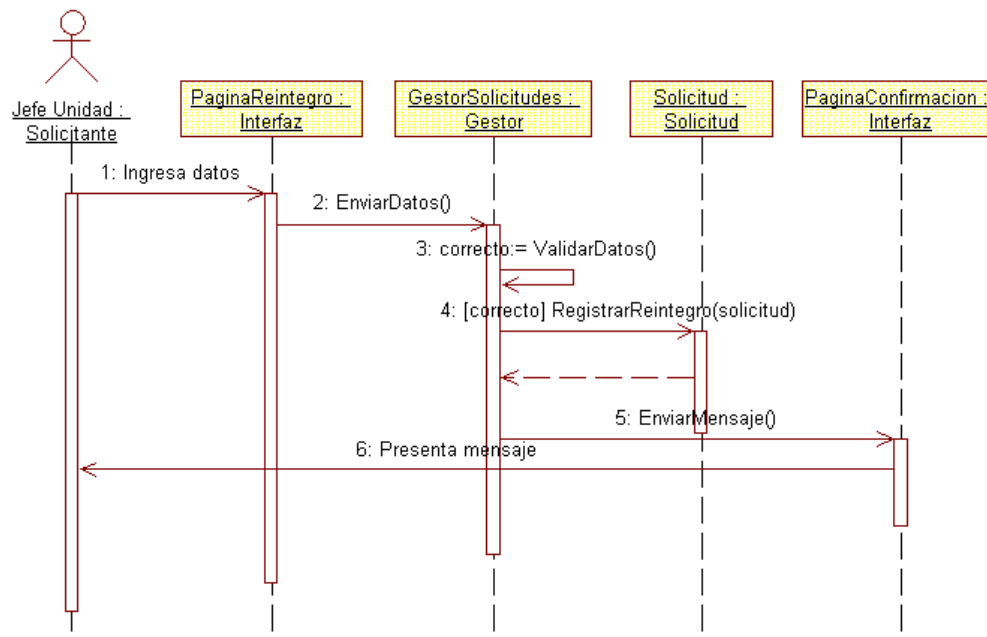
- **Registrar Reintegro.** La tabla 13 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 13. Clases de diseño – Registrar Reintegro.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de reintegro	PaginaReintegro
Interfaz de salida	PaginaConfirmación
Gestor de solicitudes	GestorSolicitudes
Solicitud	Solicitud

A continuación, la figura 69 presenta el diagrama de secuencia de estas clases.

Figura 69. Diagrama de secuencia – Registrar Reintegro.



El actor Jefe Unidad por medio de un objeto de la clase interfaz, *PaginaReintegro*, ingresa los datos correspondientes para el registro del reintegro (1). Los datos son enviados a una clase *Gestor* (2), la cual se encarga de validarlos (3). Las validaciones incluyen comprobaciones contra el período de la solicitud y con el horario del funcionario a reintegrarse, entre otras. Si los datos son correctos (4), se registra el reintegro de la solicitud. El reintegro puede incluir no sólo el ingreso de una fecha sino una hora también, dependiendo del tipo de solicitud. Además, una vez terminado el registro, la clase *Gestor* se encarga de enviar un mensaje de confirmación (5), el cual es presentado al actor mediante una clase de interfaz, *PaginaConfirmacion* (6).

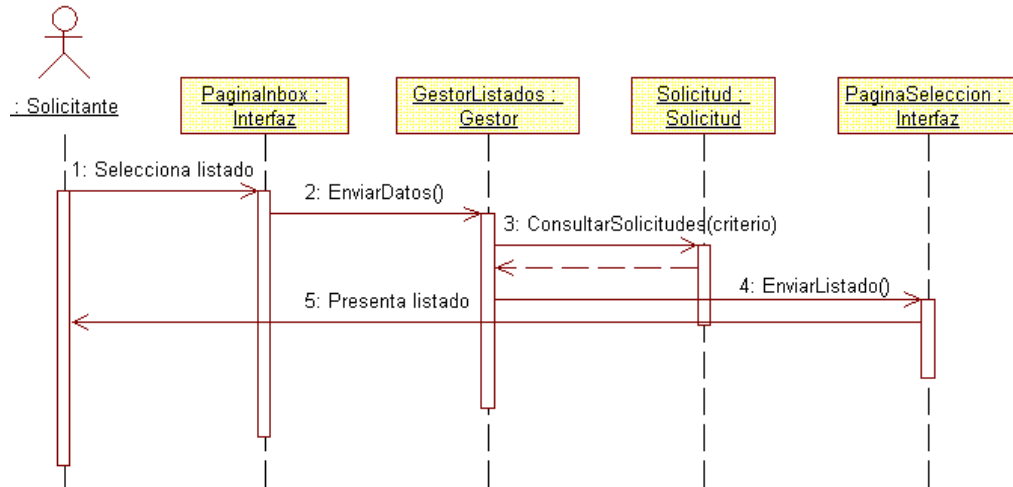
- **Ver Pendientes.** La tabla 14 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 14. Clases de diseño – Ver Pendientes.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de Inbox	PaginaInbox
Interfaz de selección	PaginaSeleccion
Gestor de listados	GestorListados
Solicitud	Solicitud

A continuación, la figura 70 presenta el diagrama de secuencia de éstas clases.

Figura 70. Diagrama de secuencia – Ver Pendientes.



Por medio de la clase de interfaz *PaginaInbox*, el actor selecciona el listado que desea ver (1). Los datos son enviados a una clase *Gestor* (2), los cuales incluyen un criterio que indica el listado que fue seleccionado. Seguidamente, la clase se encarga de realizar la consulta llamando a una clase *Solicitud*, además de devolver la consulta en forma de listado. Una vez la consulta ha terminado, la clase *Gestor* envía el listado a una clase interfaz, *PaginaSeleccion* (4), para ser presentado al actor (5).

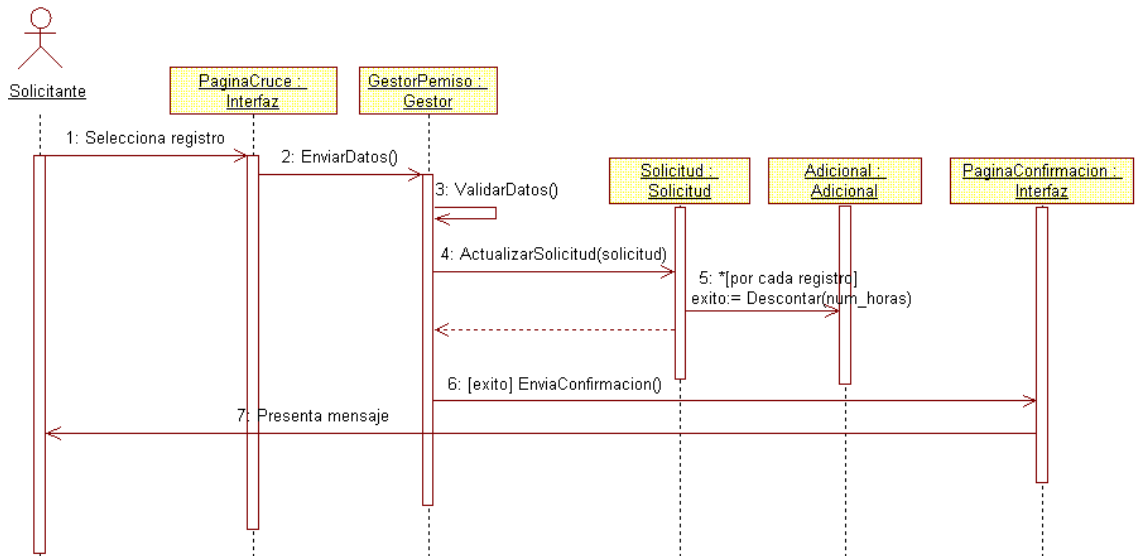
- **Cruzar tiempo.** La tabla 15 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 15. Clases de diseño – Cruzar Tiempo.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de cruce	PaginaCruce
Interfaz de salida	PaginaConfirmación
Gestor permiso	GestorPermiso
Adicionales	Solicitud
	Adicional

La figura 71 presenta el diagrama de secuencia de estas clases.

Figura 71. Diagrama de secuencia – Cruzar Tempo



La secuencia de acciones comienza cuando el actor haya seleccionado una solicitud de un listado de pendientes de cruce. El sistema presenta entonces una clase de interfaz, *PaginaCruce*, en la cual se muestra la solicitud elegida, así como los registros de tiempo adicional registrados para el actor en el sistema. De estos registros el actor hace uso para pagar en forma parcial o total el tiempo de la solicitud del permiso Por Compensar seleccionada (1). Los datos son enviados a una clase Gestor (2), la cual se encarga de validarlos (3). Además actualiza el número de horas que adeuda de la solicitud, ya sea total o parcialmente (4). Seguidamente, por cada registro que utilizó se descuentan las horas elegidas (5); si no ocurren errores durante este proceso, la clase Gestor envía un mensaje de confirmación (6), el cual es presentado al actor mediante la clase de interfaz, *PaginaConfirmación* (7).

4.3.3 Diseño de las clases. El diseño de las clases está basado en la secuencia de acciones de los casos de uso presentados anteriormente. De acuerdo a lo anterior, las clases se pueden agrupar en:

- Clases de interfaz
- Clases gestor
- Clases de persistencia de los datos

Clases de interfaz: Son las clases que durante la implementación corresponderán directamente a páginas *ASP*. La labor principal de estas clases es la captura y envío de datos a objetos de las clases de control, así como la presentación de datos a los actores del sistema.

Clases gestor: Corresponderán durante la implementación a las clases que contendrá el archivo *DLL*. En estas clases estará reunida no sólo la lógica del sistema (control de las actividades de las solicitudes), sino que contendrá métodos para: recepción y entrega de datos a objetos de la clase Interfaz, solicitudes de modificación, creación y borrado a los objetos encargados de manejar la persistencia de la información.

Clases que manejan la persistencia de los datos: Las cuales corresponderán a tablas del modelo de datos del Sistema Gestor de Bases de Datos utilizado, *INFORMIX*.

4.4 FLUJO DE IMPLEMENTACIÓN

Durante este flujo se implementaron los casos de uso que se documentaron en esta fase. Esto se debe a que ellos corresponden a la arquitectura básica del sistema que implementa el objetivo del manejo de solicitudes; que incluye la generación y administración de las mismas.

En primer lugar se muestran algunas interfaces de las mencionadas en el flujo anterior, junto con los elementos HTML que se utilizaron. Seguidamente se muestra el modelo de clases que se siguió y finalmente el modelo de datos correspondiente a la implementación de los casos de uso presentados.

4.4.1 Interfaces. Para las interfaces se utilizó el lenguaje de programación de páginas Web dinámicas ASP. Éste lenguaje permite que el código de programación que se escribe genere dinámicamente el lenguaje HTML que será presentado al actor a través de un navegador Web.

Para la captura de los datos, se utilizaron formularios (etiquetas <form>, en HTML). Estos formularios contienen otras etiquetas HTML, como son: cajas de texto, cajas de selección y cajas de chequeo, entre otras. A continuación se presenta la interfaz correspondiente al caso de uso *Solicitar Permiso*.

Figura 72. Interfaz del caso de uso Solicitar Permiso.

Botón  **Ayuda**

Permiso Individual - Llenado del formulario de permiso Ene 18 de 2004

ESTADO ACTUAL DE HORAS DE PERMISO INDIVIDUAL			
Horas Máximas de Permiso	Horas Disfrutadas	Horas en Trámite	Horas Máximas Disponibles
9999	0	0	9999

Dirigido a ENRIQUE TORRES LOPEZ **Tipo de Permiso** Seleccione

ESPECIFICACIONES DEL PERMISO

Fecha Desde (mm/dd/yyyy) Hora Desde* Fecha Hasta (mm/dd/yyyy) Hora Hasta*

Justificación

200 Caracteres restantes (*) Datos opcionales

Siguiente **Cancelar**

Botones

En la figura 72 se pueden apreciar algunos de los elementos nombrados anteriormente. Se utilizan los siguientes elementos:

- Cajas de texto: utilizadas para capturar textos cortos escritos por los actores del sistema. En este caso se usan para capturar el período de la solicitud.
- Cajas de selección: en las cuales el sistema, a través de una consulta, presenta cierta información que es conocida y cerrada (se elige alguna de las opciones); en este caso se usa para escoger entre otros el tipo de permiso.

- Cajas de áreas de texto: las cuales se utilizan para capturar textos largos. En este caso se utilizó para capturar la justificación de la solicitud.

Otros elementos que contiene los formularios, son los botones, cuya labor principal es la de enviar información al servidor. Además pueden tener métodos asociados, para permitir la realización de diferentes tareas, según la selección realizada, en caso que el formulario contenga más de uno. Para el caso mostrado en la figura 72, existen 3 botones. El botón *Siguiente*, envía los datos capturados al servidor para que sean evaluados; el botón *Cancelar* tiene un método asociado para cancelar el trámite de la solicitud; de la misma manera el botón *Ayuda* contiene un método encargado de cargar la página de ayuda.

Existen interfaces que capturan datos no sólo por medio del ingreso directo de los mismos, sino a través de otras interfaces, como por ejemplo las consultas. En este caso la interfaz presenta un vínculo hacia una consulta, el resultado de la cual será enviado al servidor para su evaluación. La figura 73 presenta un ejemplo de esto.

Figura 73. Interfaz PaginaAsistentes caso de uso Registrar Grupales.

 **Ayuda**

Permiso Grupal - Selección de asistentes Ene 18 de 2004

Clase del Permiso	Tipo del Permiso	Fecha desde	Hora desde	Fecha hasta	Hora hasta
GRUPAL FAVUIS	ASAMBLEA FAVUIS	Ene 20 de 2004	8:00 a.m.	Ene 30 de 2004	6:00 p.m.

[Inclusión por documento](#) - [Inclusión por criterio](#)

Inclusión de asistentes por criterio

Género Tiempo de vinculación de año(s) hasta año(s) Fecha nacimiento

Inclusión de Unidades y Personal por código de unidad y código de cargo

Código(s) Unidad(es) [Buscar](#)

Código(s) Cargo(s) Base [Buscar](#)

Código(s) Cargo(s) Actual [Buscar](#)

} Vínculos a consultas

Finalmente otro tipo de interfaz utilizada, es aquella que sólo presenta información al actor. La figura 74 presenta un ejemplo de esto.

Figura 74. Interfaz del caso de uso Ver Estado.

Número Solicitud	2827
Tipo Solicitud	PERMISO INDIVIDUAL
Fecha Solicitud	Oct 29 de 2003

ESPECIFICACIONES DE LA SOLICITUD			
Fecha Desde	Hora Desde	Fecha Hasta	Hora Hasta
Dic 01 de 2003	08:00 a.m.	Dic 01 de 2003	06:00 p.m.

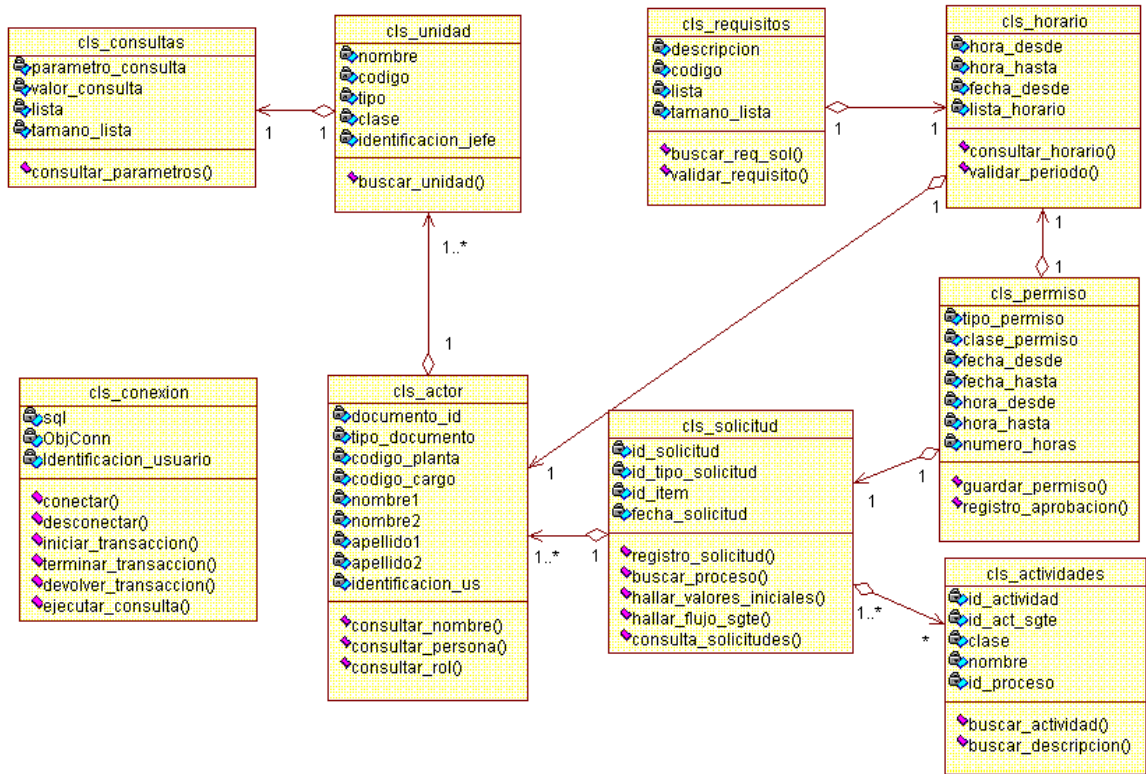
Justificación
PERMISO PERSONAL DE PRUEBA...

TRAMITE DE LA SOLICITUD	
Actividad	Envio de la solicitud
Fecha y hora Actividad	Oct 29 de 2003 a las 2:17 p.m.
Realizada por	OLGA GAMBOA PORRAS
Funcionario siguiente	Jefe de Division - DIVISION DE SERVICIOS DE INFORMACION
Actividad siguiente	Espera aprobacion

En estas interfaces los datos son presentados con un formato adecuado, el cual se encuentra dado por la clase que controla la visualización.

4.4.2 Clases. Las clases fueron realizadas con el lenguaje de programación *Visual Basic* en la versión 6.0. Las clases corresponden a objetos que se manejan en el dominio de la aplicación, para este caso: solicitudes, actores y horarios, entre otros. Dichas clases fueron agrupadas mediante un archivo *DLL*. Cada una de ellas contiene métodos asociados los cuales se encargan de utilizar los atributos con el fin de producir un resultado de valor para los actores del sistema o para otros métodos de la misma o inclusive de otras clases. La figura 47, presenta el diagrama de clases implementado durante esta fase.

Figura 75. Diagrama de clases – Fase de Elaboración.



Las clases se relacionan mediante agregación, es decir que están asociadas para representar conceptos más completos que las solas clases por sí mismas. Para ello, en el extremo donde se encuentra el rombo, en la figura 75, fue incluido un objeto de la clase con la cual se relaciona; éste objeto es agregado como atributo de la clase. Lo anterior permite utilizar los métodos y atributos específicos que se encuentran creados en ellas con el fin de no repetirlos.

Todos los métodos que requieran establecer una conexión con la base de datos para realizar su función, se relacionan de manera indirecta con objetos de la clase *cls_conexion*. Esta relación se establece en el momento de hacer el llamado a los métodos, los cuales requieren como parámetro un objeto de esta clase. Este objeto debe estar previamente creado y conectado, esto implica hacer un llamado al método *conectar()*, el cual se encarga de abrir una conexión con la base. Cuando ya no se requiera estar conectado, se hace un llamado al método *desconectar()*, el cual se encarga de cerrar la conexión con la base de datos. Los métodos harán uso de

ejecutar_consulta que se encuentra en *cls_conexion* para ejecutar sentencias SQL, así como de los métodos que requieran usar transacciones como: iniciar, devolver y terminar.

La clase *cls_permiso*, contiene métodos y atributos correspondientes a la solicitud de permiso. También tiene como atributo un objeto de la clase *cls_solicitud*, el cual le permite utilizar métodos y atributos que se encuentran en esta clase tales como: el número de la solicitud (*id_solicitud*), la fecha de solicitud y el ítem correspondiente a una persona en la solicitud de permiso (*id_item*) , entre otros. Además la clase *cls_permiso*, contiene un objeto de la clase *cls_horario*, mediante el cual se hacen validaciones con respecto al horario de la persona, haciendo uso de la clase *cls_requisitos*.

La clase *cls_solicitud*, se relaciona con la clase *cls_actor*, con el fin de representar el vínculo entre las personas y las solicitudes (permisos en este caso). Esta contiene métodos y atributos generales a las personas, siendo los más importantes los nombres, el documento de identificación y aspectos importantes para el sistema como la identificación de usuario de la persona, así como el código de planta y cargo. La clase *cls_solicitud* se relaciona además con la clase *cls_actividades*, la cual contiene métodos y atributos que representan las actividades que sigue una solicitud dentro del flujo establecido para la misma, de acuerdo a su tipo.

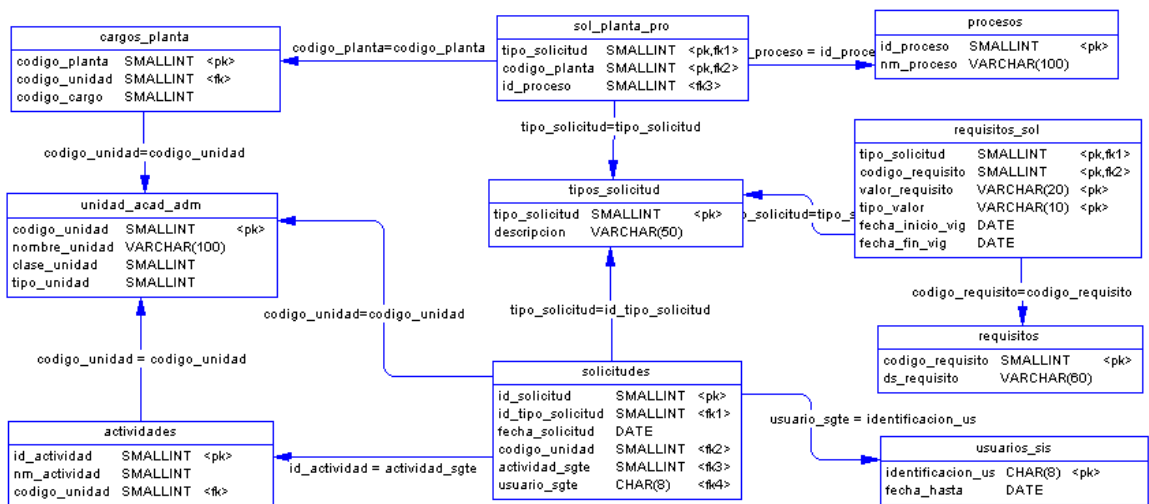
Por su parte, la clase *cls_actor*, se relaciona con la clase *cls_unidad*, la cual contiene aspectos generales a las unidades académico administrativas de la Universidad, como: el código, nombre, el tipo de unidad, la identificación del jefe correspondiente, entre otros. Esta clase a su vez se relaciona con la clase *cls_consultas*, la cual contiene, en su mayoría, métodos que generan listados de personas de acuerdo a ciertos criterios, siendo la unidad el más importante de ellos.

4.4.3 Modelo de datos. Teniendo en cuenta el diagrama de clases presentado anteriormente, así como las tablas existentes en el modelo de datos del Sistema de Información de Recursos Humanos de la Universidad, se planteó un modelo de datos para la primera solicitud: Permisos.

Debido a la cantidad de tablas y a los diferentes conceptos que éstas manejan, se hizo necesario presentar varios submodelos que representan las funcionalidades cubiertas durante la Fase de Elaboración. Existen ciertos modelos que serán utilizados por otras solicitudes diferentes a la que se implementó durante esta Fase.

La figura 76, presenta el modelo de datos que se sigue para el manejo de requisitos de las solicitudes.

Figura 76. Modelo de datos para requisitos de solicitud.



Las tablas marcadas con (*) fueron creadas durante el desarrollo del presente proyecto.

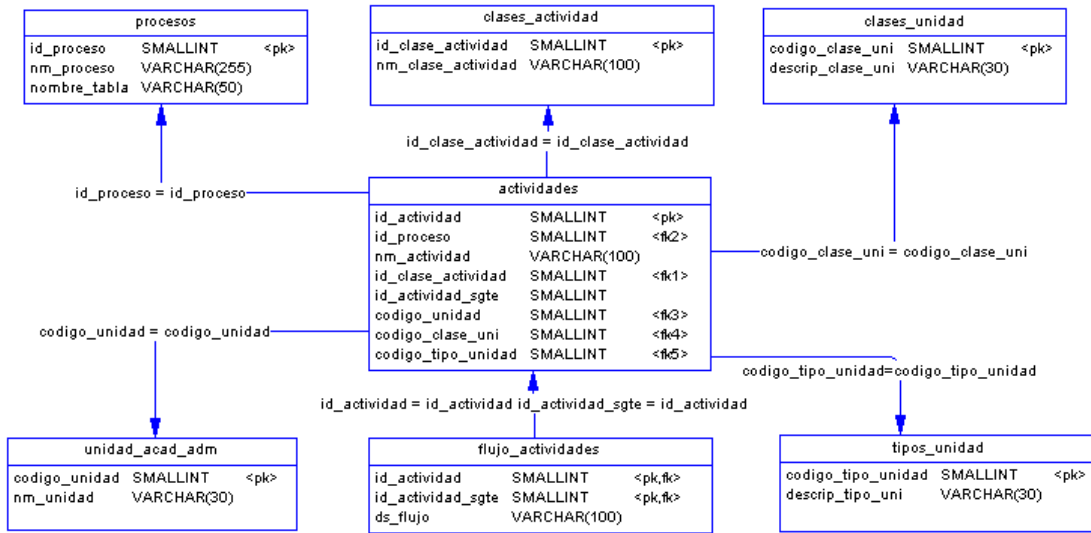
Los requisitos de las solicitudes se encuentran en la tabla *requisitos*. Los diferentes tipos de solicitudes que maneja el sistema se encuentran en la tabla *tipos_solicitud*.

La tabla *requisitos_sol*, es utilizada para hacer el cruce entre los tipos de solicitud y los requisitos correspondientes; dicha tabla contiene los valores para los requisitos de cada tipo de solicitud. Además la figura 76 presenta la tabla *sol_planta_pro*, la cual es utilizada para seleccionar el proceso correspondiente a un tipo de solicitud, de acuerdo al código de planta del solicitante. Los códigos de planta se encuentran

en la tabla *cargos_planta*, que además contiene atributos como el código de cargo y el código de unidad correspondiente a cada código de planta.

La figura 77, presenta el modelo de datos que siguen las actividades de los diferentes procesos de las solicitudes que maneja el sistema.

Figura 77. Modelo de datos para actividades.



La tabla *actividades*, contiene la información de las actividades existentes en los procesos, con la información requerida para ubicar el usuario a realizar dicha actividad, de acuerdo a la jerarquía del organigrama correspondiente y a partir de quien esté realizando la actividad actual.

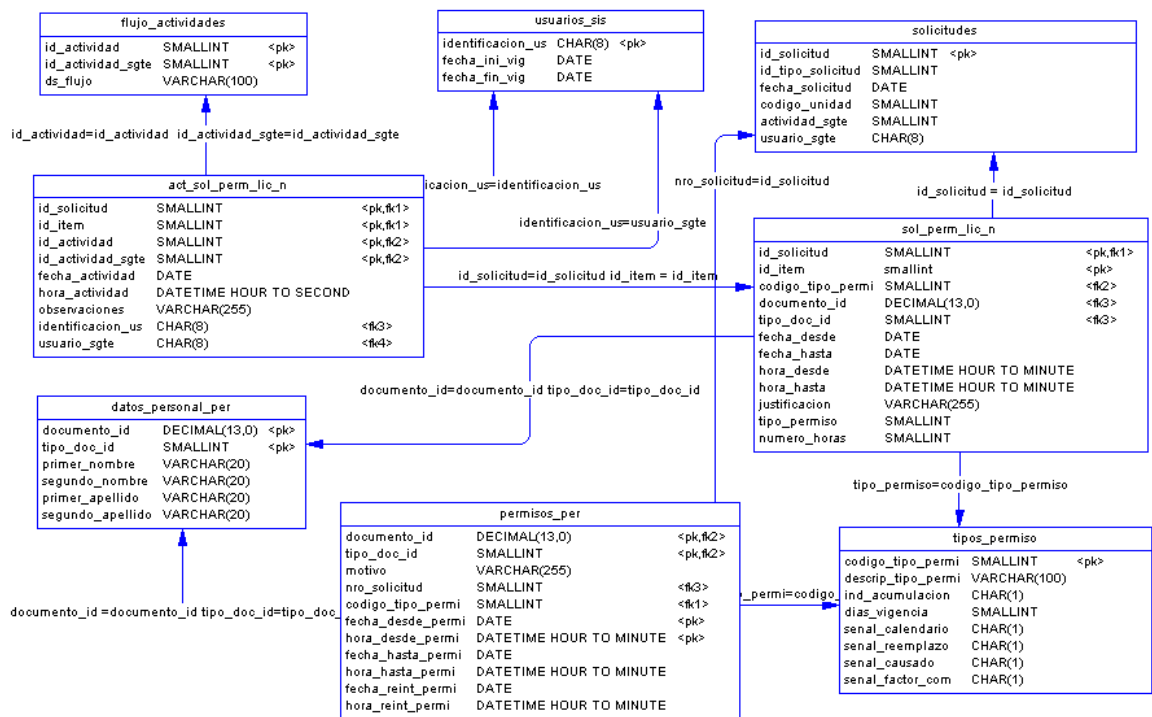
La figura 77 también presenta la tabla *flujo_actividades*, la cual permite almacenar los flujos válidos de actividades para atender el trámite de las solicitudes, así como una breve descripción del mismo.

La tabla *procesos* contiene todos los procesos asociados a las solicitudes que maneja el sistema, contiene un identificador de la primera actividad en el flujo del procesos, así como el nombre de la tabla en la cual se registran los mismos.

La tabla *clases_actividad* contiene el identificador de las clases de actividad que el sistema realiza, lo cual permite saber el estado actual de cada una de las solicitudes. Las clases son: Aprobada, En trámite, Cancelada, Revisada y Rechazada.

La figura 78, presenta el modelo de datos para el manejo de la solicitud de Permiso.

Figura 78. Modelo de datos para la solicitud de Permiso.



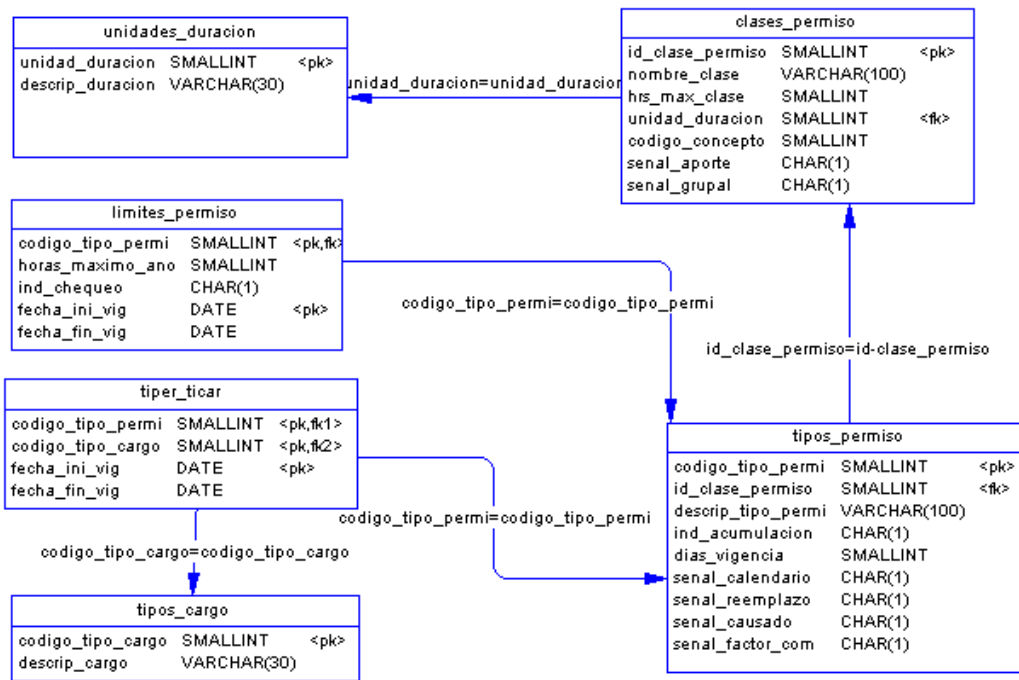
La solicitud de permiso en el momento del inicio del trámite es registrada en las tablas *solicitudes*, *sol_perm_lic_n* y *act_sol_perm_lic_n*, en las cuales se registran características generales a las solicitudes, específicas de la solicitud de permiso y el flujo que toma la misma, respectivamente. Una vez la solicitud sea aprobada, se toman los datos de la tabla *sol_perm_lic_n* y se ingresan en la tabla *permisos_per*.

Es decir, en la tabla *permisos_per* permanecen todas las solicitudes de permiso aprobadas y sobre esta tabla se hace el registro de reintegro, una vez éste se

produzca. La figura 78 presenta también la tabla *datos_personal_per*, la cual contiene todos los datos generales de las personas vinculadas con la Universidad.

La figura 79, presenta el modelo de datos para el manejo de los diferentes tipos de permisos.

Figura 79. Modelo de datos para tipos de permiso.



Los tipos de permisos se encuentran en la tabla *tipos_permiso*, cada uno de ellos pertenece a una clase de permiso y contiene una serie de señales propias para cada tipo. La tabla *clases_permiso* contiene, como su nombre lo indica, las diferentes clases de permiso que el sistema maneja, así como ciertos atributos de control como: las horas máximas que se pueden disfrutar por cada clase, la unidad para estos máximos (año, mes, entre otros.), una señal que indica si el permiso se puede tramitar como grupal y un código de descuento¹⁷ que permite identificar las personas afiliadas a ciertos organismos por los cuales se presentan permisos grupales.

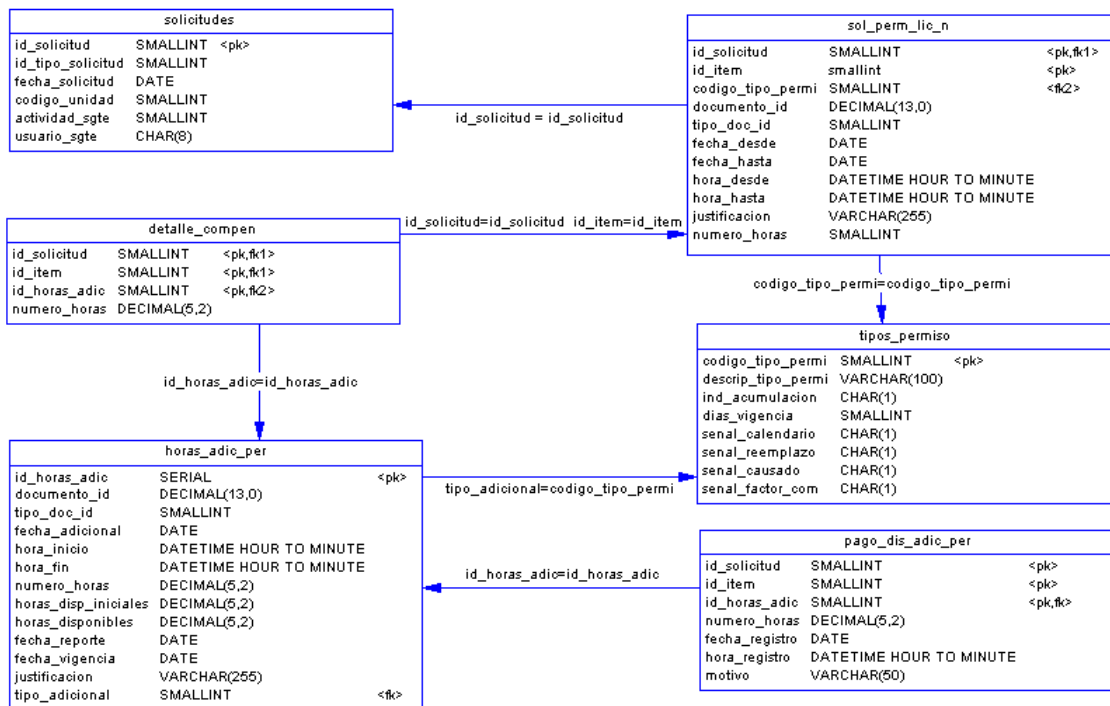
¹⁷ También se encuentra una señal que indica si el descuento corresponde a un aporte del funcionario.

La tabla *limites_permiso*, agrupa el límite en horas al año asociado a cada tipo de permiso para una vigencia establecida. Y la tabla *tiper_ticar*, es usada durante el trámite de una solicitud de permiso grupal y asocia ciertos tipos de permisos a los cargos de empleados de la Universidad (trabajadores oficiales y empleados públicos), para los cuales también pueden presentarse solicitudes de permisos grupales.

La figura 80, presenta el modelo de datos para el tipo permiso compensatorio.

A diferencia de los demás tipos de permisos, los permisos compensatorios presentan unas características muy particulares. Primero, es necesario que la persona que vaya a tramitar una solicitud de este tipo, cuente con tiempo adicional registrado en la base de datos. Este tiempo es registrado en la tabla *horas_dic_per*, la cual también incluye el tiempo otorgado por los conceptos de votación y jurado en los períodos de elecciones, junto con las vigencias establecidas para su posterior disfrute.

Figura 80. Modelo de datos para permisos compensatorios.



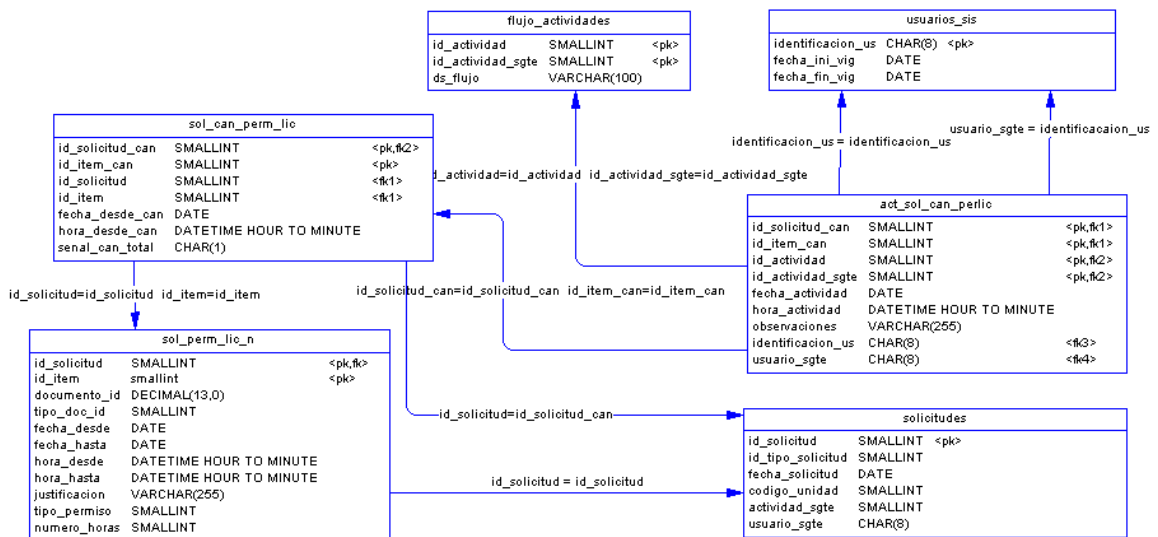
Una solicitud de permiso compensatorio se registra en la tabla *sol_perm_lic_n*, donde se registran todas las solicitudes de permiso. Además se registra en la tabla *detalle_compen*, en la cual se relacionan los registros de tiempo adicional vinculados a la solicitud que se está tramitando.

Finalmente cuando la solicitud es aprobada, además del registro de la solicitud en la tabla *permisos_per*, se hace el descuento con las horas seleccionadas en la tabla *horas_dic_per* y se insertan los valores de *detalle_compen* en *pago_dis_adic_per*.

En la tabla *pago_dis_adic_per*, se lleva el registro de las horas de tiempo adicional trabajado disfrutadas, así como el pago de horas de permisos del tipo Para Compensar que se puede realizar también con registros de tiempo adicional.

La figura 81, presenta el modelo de datos para las solicitudes de cancelación de permisos.

Figura 81. Modelo de datos para la solicitud de cancelación de permisos.



Una solicitud de cancelación es generada sólo cuando la solicitud que se quiere cancelar ya se encuentra aprobada. Cuando una solicitud esté en trámite, éste se

suspende y la solicitud pasa al estado cancelada. Una solicitud de cancelación puede ser parcial o total, respecto de la solicitud que está aprobada.

Una solicitud de cancelación se registra en las tablas *solicitudes*, *sol_can_perm_lic* y *act_sol_can_perlic* en las cuales, de manera similar a la solicitud tramitada inicialmente (la que está aprobada), se registran el flujo que sigue la solicitud y las características generales y específicas (en este caso, de la cancelación).

Si la cancelación es parcial, se actualizan los nuevos valores límite (fecha y hora de fin), en la tabla *permisos_per* para la solicitud inicial que ya se encontraba aprobada.

Sí la cancelación es total, la solicitud aprobada es borrada de la tabla *permisos_per*, quedando el registro correspondiente en la tabla *act_sol_perm_lic*, que la solicitud fue cancelada.

4.5 CONCLUSIONES DE LA FASE DE ELABORACIÓN.

Durante esta fase se implementó completamente el procedimiento de Permisos. Éste establece la arquitectura que tendrá el sistema, para el manejo de los demás procedimientos (solicitudes) contemplados.

El manejo de los procedimientos involucra desde la generación de la solicitud (registro por parte del solicitante), la secuencia de actividades de trámite hasta la aprobación y el registro del reintegro del empleado cuando la situación termine.

Además, las validaciones correspondientes a cada procedimiento, la posible cancelación en cualquier parte de su flujo de trámite, las consultas de apoyo tanto para la generación de la solicitud como para su evaluación durante el trámite y la posibilidad de consultar el estado de la solicitud en todo momento, entre otras funcionalidades.

Los casos de uso denominados con prioridad crítica, fueron los que se tuvieron en cuenta durante esta fase. Esto porque la mayoría, serán utilizados por los demás

procedimientos que el sistema manejará. Además del planteamiento e implementación de los modelos de datos de horarios y requisitos que también son comunes a todos los procedimientos.

Se mejoró en las interfaces de captura y presentación de los datos a medida que se implementaban los diferentes tipos de permisos, lo cual permitió establecer una estructura genérica de la conformación de las interfaces para los demás procedimientos. También a medida que se avanzó durante la Fase se fueron depurando las clases, con el fin de optimizar y reducir significativamente la redundancia de atributos, métodos y estableciendo un marco para el diseño de clases de los próximos procedimientos.

La implementación se llevó a cabo bajo el sistema operativo *Windows 2000*, con la herramienta de desarrollo *Visual Basic 6.0* y el manejador de bases de datos *INFORMIX* y para las pruebas se contó con el navegador *Internet Explorer*, presente con la versión de Windows utilizada.

El trabajo realizado durante la fase de elaboración se encuentra resumido en la tabla 16.

Tabla 16. Resumen de actividades de la Fase de Elaboración.

FLUJO	ACTIVIDAD
Requisitos	Jerarquización de los actores del sistema.
	Identificación y descripción de nuevos casos de uso.
	Categorización de los casos de uso encontrados.
Análisis	Identificación de paquetes de análisis y de servicio.
	Análisis de los casos de uso catalogados como críticos, mediante diagramas de colaboración.
	Descripción de todas las clases de análisis identificadas.
Diseño	Modelo de despliegue.
	Diseño de los casos de uso críticos para la arquitectura del sistema, mediante diagramas de secuencia.
	Diseño de las clases identificadas.

Implementación	Implementación de los casos de uso categorizados como críticos.
	Pruebas del funcionamiento de los casos de uso para el procedimiento de Permisos.

Para la siguiente fase se planea la implementación y prueba de los demás procedimientos, utilizando la arquitectura definida. Dicha implementación cubrirá los casos de uso categorizados como importantes y secundarios, en la tabla 1 del presente documento.

5. FASE DE CONSTRUCCIÓN

Durante la Fase de construcción se lleva el sistema hacia una primera versión operativa, dando cumplimiento a los objetivos planteados inicialmente.

De manera similar a la Fase anterior, el trabajo se desarrolla por iteraciones. Las iteraciones se realizan por los flujos de requisitos, análisis, diseño, implementación y pruebas. Durante la Fase de Construcción, hay una fuerte incidencia sobre los dos últimos flujos, esto con el fin de realizar la implementación de los objetivos planteados.

Esta Fase se lleva a cabo en tres iteraciones. En la primera de ellas se concluye lo concerniente al paquete de Solicitudes. Para la segunda se contempla la construcción del paquete de Informes. Y finalmente en la tercera los casos de usos restantes del paquete de Registro.

En el flujo de requisitos se estructuró la versión final del diagrama de casos de uso. Para el flujo de análisis, se complementó la descripción de los casos de uso, de acuerdo a la realizada en la Fase anterior. En el flujo de diseño, se construyeron las clases requeridas para el manejo de los procedimientos considerados y para los flujos de implementación y pruebas, se llevó a cabo el desarrollo de los casos de uso planteados, utilizando la arquitectura definida en la Fase anterior.

5.1 Flujo de Requisitos

Los procedimientos no sufrieron modificación, respecto a los presentados al inicio de la Fase anterior.

Para el listado de funcionalidades, se suman las siguientes a las presentadas en la Fase anterior:

- Permitir el registro de números de actas como respaldo a una acción realizada sobre una solicitud.
- Permitir el registro de información, como resoluciones y legalizaciones¹⁸, para los Jefes de Unidad que sea requerido.

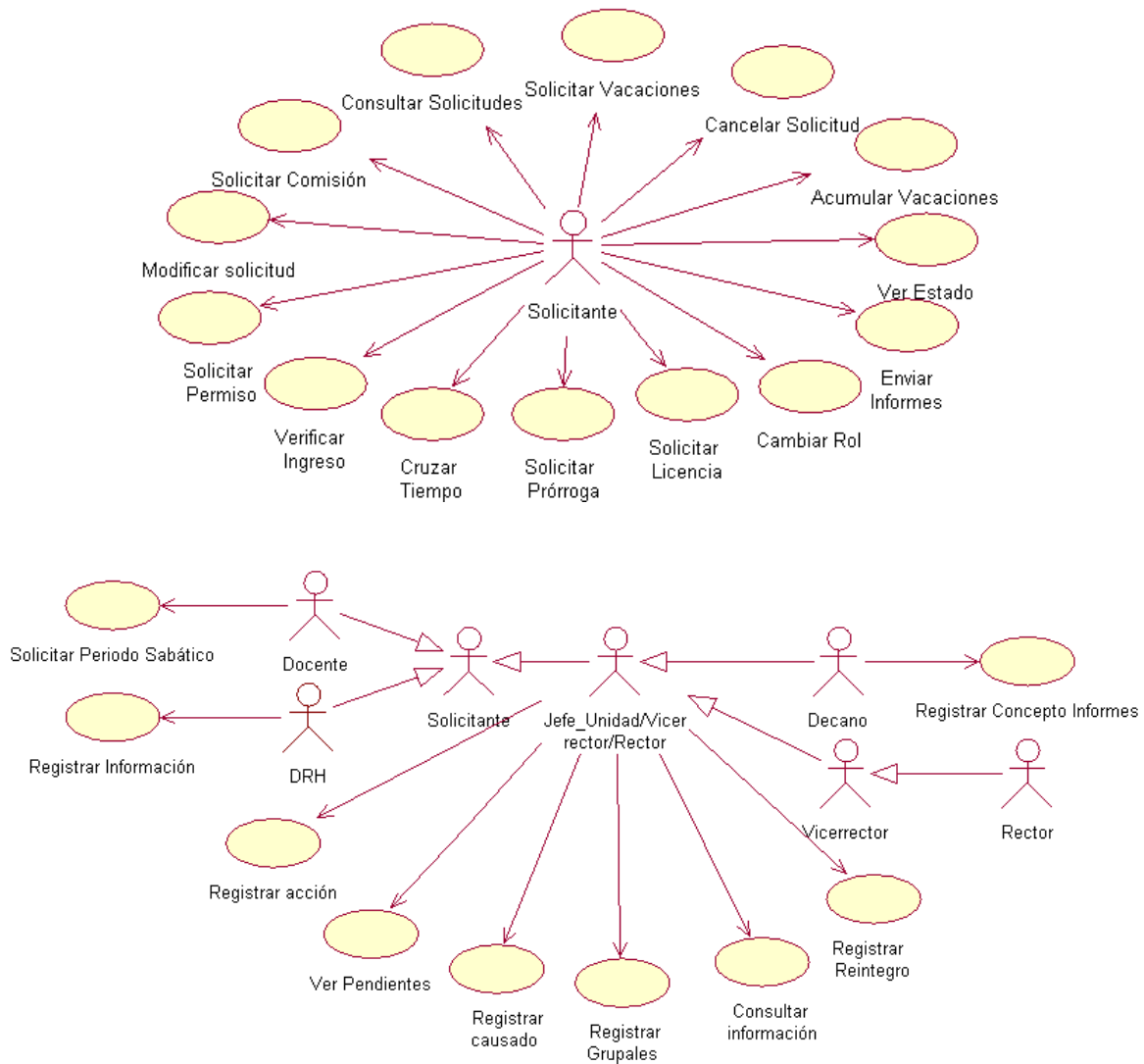
El listado de actores se mantiene, así mismo las relaciones de jerarquización que se presentaron en la Fase anterior.

Para el modelo de casos de uso del sistema, de acuerdo con la implementación realizada en la Fase anterior, se puede llegar a una simplificación del mismo. Esto con respecto a los casos de uso: *Aprobar solicitud*, *Visar solicitud* y *Rechazar solicitud*. Para la versión final del modelo, se decidió agrupar las tres funcionalidades en un solo caso de uso denominado: *Registrar acción*, el cual sigue la misma secuencia de acciones que se presentaron para los tres casos de uso mencionados, en la Fase anterior.

A continuación, en la figura 82 se presenta el modelo de casos de uso final del sistema. El diagrama se presenta, de manera similar que en la Fase anterior, dividido en dos partes: la primera mostrando los casos de uso referentes al actor *Solicitante* y la segunda al actor *Jefe Unidad*, con la respectiva jerarquización.

¹⁸ La legalización se presenta para ciertos procedimientos como en *Comisiones de Estudio*, una breve explicación acerca de este estado de las solicitudes se puede encontrar en el capítulo de Fase de Inicio.

Figura 82. Modelo de casos de uso – Fase de Construcción.



Se cambió de nombre al caso de uso *Solicitar Modificación* presentado en la Fase de Inicio, porque el sentido que adquirió este caso es de modificar los términos de la solicitud (como se encuentra descrito en la Fase de Inicio) y no genera una nueva solicitud. Por tanto, el nuevo nombre para esta funcionalidad es: *Modificar Solicitud*.

5.1.1 Casos de uso detallados. Durante esta Fase se complementarán los detalles de los casos de uso del sistema. Se comenzará con los casos de uso denominados *Importantes*, para finalizar con los casos de uso *Secundarios*¹⁹. Los

¹⁹ Esta clasificación se puede encontrar en la tabla 1, de la Fase de Inicio.

casos de uso que hacen referencia a solicitud (incluyendo el caso Acumular vacaciones), tienen todos como precondition, que hayan seleccionado el procedimiento desde el menú del sistema.

- **Solicitar Licencia.**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: Permite al solicitante iniciar el trámite de una licencia no remunerada.

Resumen: El caso de uso permite el registro en el sistema de solicitudes de licencia. Las licencias que se tuvieron en cuenta para la implementación fueron las *No Remuneradas*. El período solicitado es por días completos, a diferencia de un *Permiso*, que puede ser solicitado por horas.

Precondición: Escoger en la correspondiente página de Inbox, la opción que permite iniciar el trámite de una solicitud de *Licencia No Remunerada*.

Flujo de sucesos: El caso de uso *Solicitar Licencia*, presenta los mismos estados que *Solicitar Permiso* (del tipo Individual), caso de uso detallado en la Fase de Inicio. Exceptuando, que para el caso de las licencias el período son días completos.

- **Solicitar Comisión.**

Actor: Cualquier actor que pertenezca al escalafón profesional bajo el rol de *Solicitante*.

Propósito: Permite al solicitante el registro en el sistema de una comisión, ya sea de servicios o de estudios.

Resumen: Se consideraron dos tipos de Comisiones: De Estudios y de Servicios. Las *Comisiones de Estudio* tienen como propósito obtener un título académico de nivel superior (Especialización, Maestría y Doctorado) o realizar una pasantía o investigación.

Las *Comisiones de Servicio*, pueden ser otorgadas con el fin de: Asistir a un evento o realizar un trabajo para (o en nombre de) la Universidad, entre otros; sin embargo, el objetivo denominado Desempeñar cargo, no fue considerado para ser manejado

en el sistema, debido a que actualmente no se maneja dentro del marco de una solicitud, sino que la gestión de la comisión la realiza directamente la entidad externa interesada con la Rectoría. Otra diferencia entre los dos tipos de comisiones es en el período, así las *Comisiones de Estudio* se solicitan por días completos con una dedicación específica, a diferencia de las de Servicio, que pueden solicitarse en horas.

Precondición: Si el actor pertenece ya sea al escalafón docente o profesional de la Universidad y selecciona una de las opciones de Comisión (de servicios o de estudios), del menú de solicitudes desde la página de Inbox correspondiente.

Flujo de sucesos: El caso de uso *Solicitar Comisión*, sigue los mismos estados que se presentaron para el caso de uso *Solicitar Permiso* (del tipo Individual), descritos en la Fase de Inicio, con la diferencia en que el período solicitado puede ser por horas o días completos como se menciono anteriormente según el tipo de comisión.

- **Solicitar Período Sabático.**

Actor: Docentes de la Universidad que pertenezcan a las categorías Titular y Asociado.

Propósito: Permite al actor correspondiente el registro de una solicitud de período sabático.

Resumen: El período sabático se otorga por un año y sólo una vez, este período no es prorrogable. Tiene como objetivo, el desarrollo de una investigación, la elaboración de un libro o la realización de una pasantía y de manera similar a las comisiones de estudio, se deben presentar informes de seguimiento.

Precondición: Escoger en la correspondiente página de Inbox, la opción que permite iniciar el trámite de una solicitud de *Período Sabático*.

Flujo de sucesos: El caso de uso *Solicitar Período Sabático*, sigue el mismo flujo de sucesos presentado en la Fase de Inicio, para el caso *Solicitar Permiso* (del tipo Individual).

- **Solicitar Vacaciones.**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: Permite el registro de una solicitud de disfrute días de vacaciones acumulados.

Resumen: Por medio de este caso de uso, se puede iniciar el trámite de una solicitud de disfrute de días acumulados de vacaciones; la solicitud se realiza por días completos.

Flujo de sucesos: El caso de uso *Solicitar Vacaciones*, sigue el mismo flujo de sucesos presentado en la Fase de Inicio, para el caso *Solicitar Permiso* (del tipo Individual).

- **Acumular Vacaciones.**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: Por medio de este caso de uso, se puede iniciar el trámite de una solicitud que permita la acumulación de un período correspondiente de vacaciones.

Resumen: En la acumulación, la persona no hace uso del período de vacaciones otorgado por la Universidad, por razones de servicio.

Precondición: Escoger en la correspondiente página de Inbox, la opción que permite iniciar el trámite de una solicitud de acumulación de vacaciones.

Flujo de sucesos: El caso de uso *Acumular Vacaciones*, sigue el mismo flujo de sucesos presentado en la Fase de Inicio, para el caso *Solicitar Permiso* (del tipo Individual).

- **Solicitar Prórroga.**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: Por medio de este caso de uso, se puede solicitar una prórroga de una situación que es prorrogable y que ya se encuentre en vigencia.

Resumen: La prórroga es una solicitud, en la cual se busca una prolongación del período aprobado inicialmente para un procedimiento. En el sistema se consideraron, las prórrogas para los procedimientos de *Licencias No Remuneradas*, *Comisiones de Estudio* y *Comisiones de Servicio*.

Precondición: Escoger de un listado la situación que se quiere prorrogar.

Flujo de sucesos: El caso de uso *Solicitar Prórroga*, sigue el mismo flujo de sucesos presentado en la Fase de Inicio, para el caso *Solicitar Permiso* (del tipo Individual).

- **Enviar Informes.**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: Permitir el envío de informes de seguimiento para los procedimientos de *Comisiones de Estudio* y *Período Sabático*.

Resumen: De acuerdo al reglamento, los procedimientos anteriormente mencionados requieren de la presentación de informes de seguimiento, de las actividades propuestas.

Precondición: Tener pendiente el envío de algún informe para una solicitud, ya sea de *Comisión de estudios* o *Período Sabático*.

Flujo de sucesos: El flujo de sucesos para este caso de uso es presentado en la figura 83.

Figura 83. Diagrama de estados – Enviar Informes.



Camino básico:

Verificar requisitos: Se verifica que el funcionario tenga pendiente el envío del algún informe para cualquiera de las situaciones anteriormente mencionadas.

Capturar archivo: Durante este estado el sistema presenta una funcionalidad similar al envío de archivos adjuntos de correo electrónico, en la cual se permite buscar y vincular localmente el archivo que posteriormente se enviará.

Registrar envío: Una vez capturado el archivo y al realizar el envío del mismo se registra el informe en el sistema.

Fin: La instancia del caso de uso llega a su fin cuando el archivo seleccionado es colocado en un área específica del servidor donde se encuentra la aplicación.

Camino alternativo:

Verificar requisitos – Fin: Cuando no se cumpla con algún requisito que no se haya considerado manejar como advertencia, el sistema no permitirá el envío de informes.

- **Registrar Concepto Informes.**

Actor: Decano.

Propósito: Permite el registro del concepto emitido en la evaluación de informes presentados.

Resumen: Este caso de uso no sólo agrupa el registro del concepto de informes, sino opcionalmente ofrece la posibilidad de descargar el archivo enviado y permite el registro del concepto sobre el mismo.

Precondición: Tener pendiente el registro del concepto, sobre un informe enviado de *Comisión de Estudios o Período Sabático*.

Flujo de sucesos: El flujo de sucesos para este caso es presentado en la figura 84.

Figura 84. Diagrama de estados – Registrar concepto informes.



Camino básico:

Obtener archivo: En este estado (opcional), se permite descargar el archivo para su revisión.

Capturar concepto: Una vez hecha la revisión del documento enviado el sistema captura el concepto emitido.

Registrar concepto: En este estado el sistema se encarga de registrar el concepto capturado en el estado anterior.

Fin: La instancia del caso de uso llega a su fin cuando se confirma el registro del concepto en el sistema.

Caminos alternativos:

El estado *Obtener archivo*, es opcional porque puede ser que el informe ya haya sido revisado con anterioridad y sólo se ingrese al sistema a registrar el concepto.

De igual manera se puede ingresar al sistema sólo para obtener el archivo y en otro momento registrar el concepto sobre el mismo.

- **Modificar Solicitud.**

Actor: Cualquier actor bajo el rol de *Solicitante*.

Propósito: A través de esta funcionalidad se permite modificar los términos de una solicitud como son: el período y la justificación, entre otros.

Resumen: La modificación de una solicitud es válida, durante su trámite. Específicamente, mientras no haya recibido una acción, ya sea de visado o de aprobación, porque sí se ha registrado como mínimo un visado, debería cancelar dicha solicitud e iniciar nuevamente el trámite para la solicitud modificada.

Precondición: Seleccionar alguna solicitud en trámite, que no haya recibido ninguna acción. Las solicitudes con posibilidad de modificación son: Licencias no Remuneradas, Comisiones de servicio, Comisiones de estudio y Períodos sabáticos.

Flujo de sucesos: El flujo de sucesos para este caso es presentado en la figura 85.

Figura 85. Diagrama de estados – Modificar Solicitud.



Camino básico:

Capturar cambios: En este estado se muestra la solicitud, con el fin de ingresar los nuevos valores. Se realizan las mismas validaciones que se llevaron a cabo en el momento del trámite inicial, según el tipo de la solicitud.

Verificar requisitos: Una vez los datos son enviados, se realizan las validaciones correspondientes al tipo de solicitud. Estas son las mismas que se efectuaron en el momento en que se tramitó la solicitud que se esta modificando.

Registrar solicitud: Cuando se confirma el envío de la solicitud modificada, se registran los nuevos valores en la base de datos.

Fin: La instancia del caso de uso llega a su fin cuando se confirma que la solicitud ha sido modificada satisfactoriamente.

Caminos alternativos:

Verificar requisitos – Capturar cambios: Este camino dentro del caso de uso, ocurre cuando se presenta algún error en el momento de verificar los nuevos datos ingresados. El sistema presenta una serie de mensajes, los cuales indican acerca del error presentado.

Verificar requisitos – Fin: En caso que el actor decida no realizar la modificación de la solicitud, ésta se puede cancelar y la solicitud se mantiene con los datos ingresados inicialmente (cuando se tramitó por primera vez).

- **Registrar Información.**

Actor: El jefe de la unidad de Recursos Humanos y los miembros principales²⁰ de los consejos o comités, bajo el rol de Jefe Unidad.

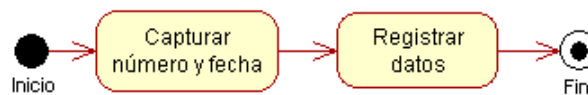
Propósito: Registrar información de respaldo a acciones realizadas sobre las solicitudes.

Resumen: Este caso de uso agrupa de manera similar, la funcionalidad de registro de fecha y número de actas, utilizadas también por Consejos y Comités de la Universidad, como respaldo a decisiones sobre alguna solicitud; en estos casos el actor específicamente puede que no sea el jefe de la unidad de Recursos Humanos, sino cualquier actor del tipo Jefe Unidad, que sea el miembro principal de dicha unidad.

Precondición: Para el caso de registro de Resoluciones, deben existir solicitudes de *Licencia No Remunerada*, *Período Sabático* o *Comisiones de Estudio* aprobadas; para el caso de registro de Actas, el actor debe ser el miembro principal de un Comité o un Consejo.

Flujo de Sucesos: El flujo de sucesos para este caso es presentado en la figura 5.

Figura 86. Diagrama de Estados – Registrar Información.



Camino básico:

Capturar número y fecha: En este estado se capturan los datos requeridos para realizar el registro. La información solicitada es el número y la fecha de expedición del documento (Acta o Resolución).

Registrar datos: Se lleva a cabo el registro de los datos de acuerdo al tipo de la solicitud y al momento en el cual se hace uso de éste caso, se registran Actas o Resoluciones, según corresponda.

²⁰ Los miembros principales hacen referencia a las jefes de unidades académico administrativas cuya clase de unidad es un consejo o comité. Ejemplo: CONSEJO DE ESCUELA DE ING.DE SISTEMAS, cuyo miembro principal es el jefe de la unidad ESCUELA DE ING.DE SISTEMAS.

Fin: la instancia del caso de uso llega a su fin cuando se confirma el registro de los datos ingresados.

5.2 FLUJO DE ANÁLISIS

5.2.1 Análisis de la Arquitectura. Durante esta Fase se continuó con la arquitectura planteada en la Fase de Elaboración. Es decir, el diagrama de paquetes y los paquetes de servicio, así como la dependencia entre éstos, no tiene variación en relación con la presentada en la Fase anterior del desarrollo.

5.2.1.1 Análisis de casos de uso. Los casos de uso que incluyen solicitudes, presentan el mismo desarrollo en clases de análisis que el caso de uso *Solicitar Permiso*, presentado en la Fase de Elaboración. Sólo cambian las clases involucradas en cada tipo de solicitud. La tabla 17 presenta dichos cambios.

Los mensajes que se envían entre las clases también son los mismos que en el caso de uso de referencia (*Solicitar Permiso*).

Tabla 17. Cuadro comparativo de los casos de uso de solicitud.

CASO DE USO	CLASES DE INTERFAZ	CLASE DE CONTROL	CLASE DE ENTIDAD
Solicitar Permiso	Interfaz de Permiso, Interfaz de Salida	Gestor Permiso	Permiso
Solicitar Licencia	Interfaz de Licencia, Interfaz de Salida	Gestor Licencia	Licencia
Solicitar Comisión ²¹	Interfaz de Comisión, Interfaz de Salida	Gestor Comisión	Comisión
Solicitar Período Sabático	Interfaz de Período, Interfaz de Salida	Gestor Período	Período
Solicitar Vacaciones	Interfaz de Vacaciones, Interfaz de Salida	Gestor Vacaciones	Vacaciones
Acumular	Interfaz de Acumulación, Interfaz de	Gestor Vacaciones	Vacaciones

²¹ Incluye los dos tipos de comisión, cada uno tiene diferente tipo de interfaz, clase de control y entidad, pero la funcionalidad es la misma en ambos casos.

Vacaciones ²²	Salida		
Solicitar Prórroga	Interfaz de Prórroga, Interfaz de Salida	Gestor Prórroga	Prórroga

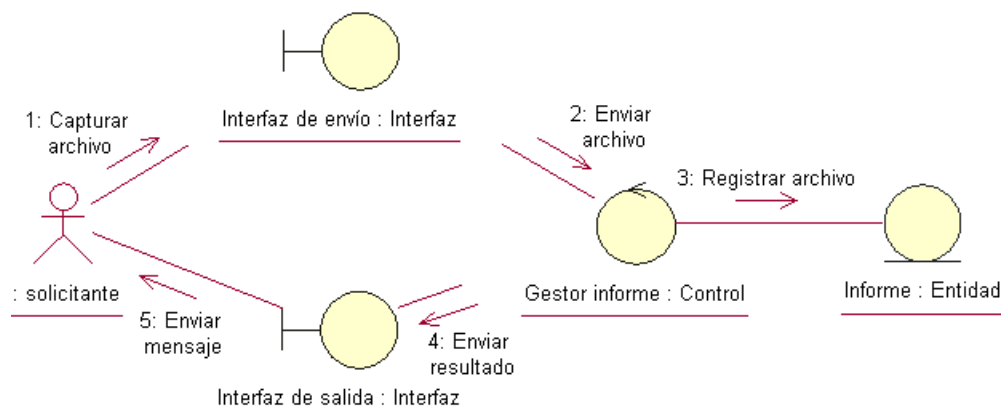
El análisis de los casos de uso considerados durante esta fase, se presenta a continuación.

- **Enviar Informes.** La figura 87 presenta el desarrollo del caso de uso a través de clases de análisis.

El sistema presenta una interfaz de envío (1), mediante la cual se captura el archivo; en esta interfaz el actor correspondiente selecciona el archivo que desea enviar.

El objeto “Interfaz de envío”, se encarga de enviar el archivo a un objeto de la clase de control “Gestor informe” (2), para que sea registrado en el sistema (3). Seguidamente, dicho objeto envía el resultado de la operación anterior (4), a un objeto de la clase “Interfaz de salida”; con el fin de presentarlo al actor en forma de un mensaje (5). El registro del informe incluye generar mensajes a los actores involucrados en el proceso de evaluación.

Figura 87. Diagrama de colaboración – Enviar Informes.

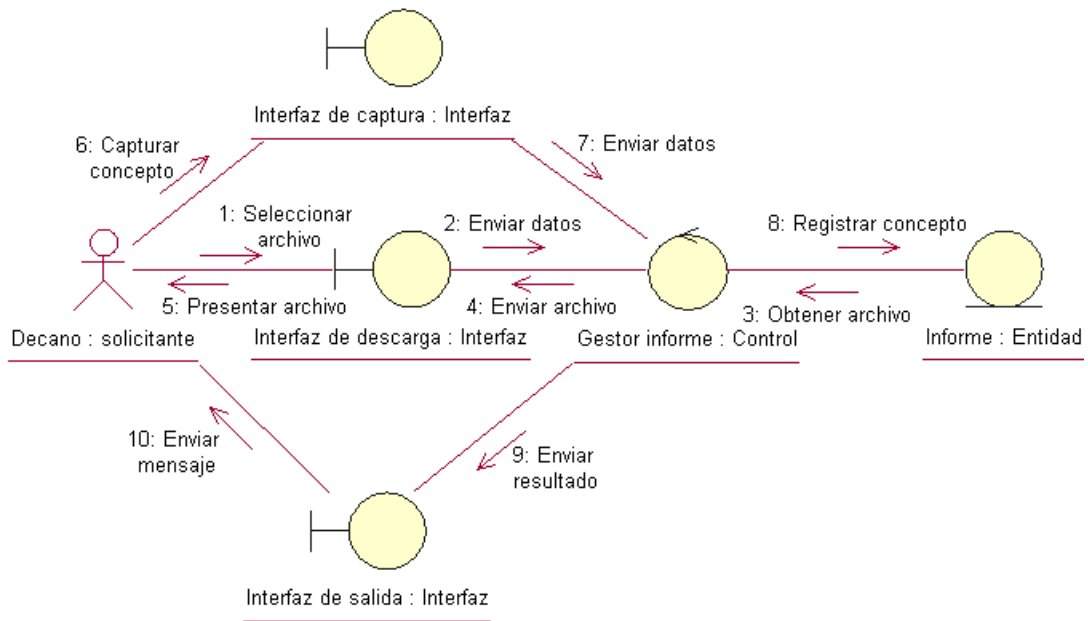


²² A pesar que tiene las mismas clases de control y entidad que el caso de uso Solicitar Vacaciones, se decidió tratarlo como una caso de uso independiente, debido a que el sentido es diferente.

- **Registrar Concepto Informes.** La figura 88 presenta el desarrollo del caso de uso a través de clases de análisis.

El desarrollo del caso de uso comienza cuando el actor Decano selecciona el archivo (informe) que desea descargar²³ (1), esto se hace mediante un objeto de la clase “Interfaz de descarga”, el cual envía los datos necesarios para identificar el archivo seleccionado (2). Un objeto de la clase “Gestor informe” se encarga de obtener el archivo (3), además de enviar el archivo al objeto “Interfaz de descarga” (4), para que sea presentado para descargar (5). En el momento que se desee registrar el concepto sobre el informe, éste se captura mediante un objeto de la clase “Interfaz de captura” (6); los datos son enviados a “Gestor informe” (7) con el fin de ser registrado en el sistema (8). Seguidamente, el objeto de esta clase se encarga también de enviar el resultado a un objeto de la clase “Interfaz de salida”, para ser presentados al actor (10). El caso de uso puede finalizar, en una primera parte, cuando se descarga el archivo debido a que posteriormente sólo se iniciará el caso de uso para registrar el concepto.

Figura 88. Diagrama de colaboración – Registrar concepto informes.

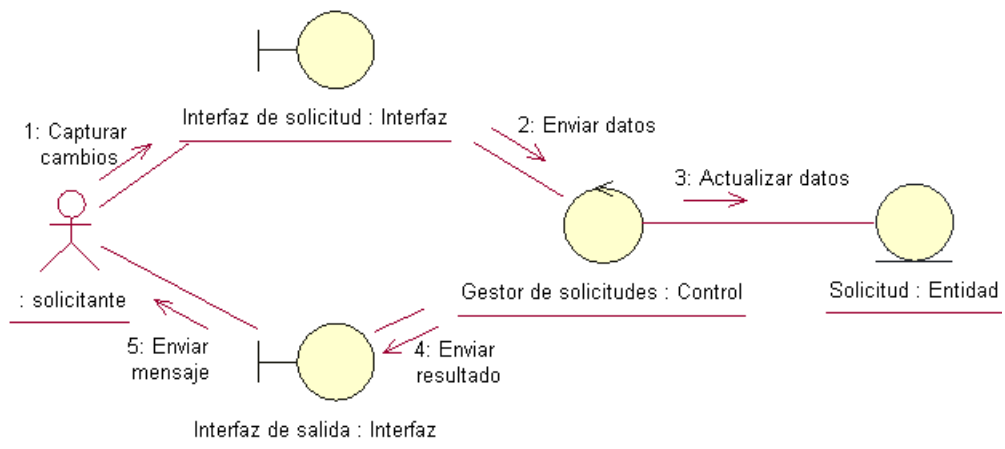


²³ La descarga se hace con el fin de revisar el documento para posteriormente registrar el concepto.

- **Modificar Solicitud.** La figura 89 presenta el desarrollo del caso de uso a través de clases de análisis.

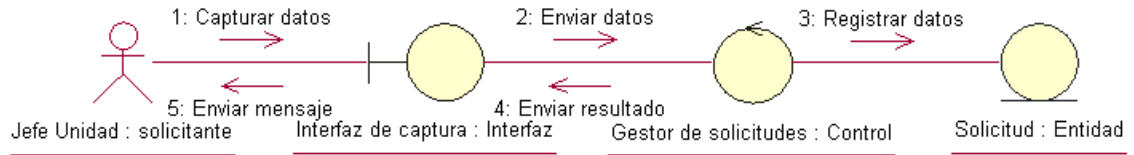
El actor mediante un objeto de la clase “Interfaz de solicitud”, hace los cambios necesarios sobre la solicitud que previamente seleccionó para modificar (1). Los nuevos datos son enviados a un objeto de la clase “Gestor de solicitudes” (2), para que la solicitud sea actualizada en el sistema (3). Este mismo objeto, se encarga de enviar el resultado de la modificación de los datos a un objeto de la clase “Interfaz de salida” (4), con el fin de presentarlos al actor correspondiente (5). En el momento de hacer modificaciones sobre la solicitud que ya se encuentra registrada en el sistema, se realizarán las mismas comprobaciones y verificaciones hechas a la solicitud inicial.

Figura 89. Diagrama de Colaboración – Modificar solicitud.



- **Registrar Información.** La figura 90 presenta como se desarrolla el caso de uso a través de clases de análisis.

Figura 90. Diagrama de Colaboración – Registrar Información.



Los datos que se desean registrar se capturan mediante un objeto de la clase “Interfaz de captura” (1).

Estos datos son enviados a un objeto de la clase “Gestor de solicitudes” (2), el cual se encarga de registrar la información en el sistema (3). Una vez los datos hayan sido registrados, el objeto de la clase de control, envía el resultado del registro a “Interfaz de captura” con el fin de ser presentados al actor mediante un mensaje (5).

5.2.2 Análisis de clases. La tabla 18 presenta una descripción de las clases de análisis, similar a la presentada en la Fase anterior. Se describen las clases nuevas encontradas para los casos de uso documentados durante esta Fase. Para las demás clases se sigue la misma descripción presentada en la Fase anterior.

Tabla 18. Descripción de clases de análisis – Fase de Construcción.

TIPO	NOMBRE	RESPONSABILIDADES	ATRIBUTOS
Interfaz	Interfaz de Envío	<ul style="list-style-type: none"> Mostrar un listado de archivos y permitir la selección de uno de ellos. Enviar datos al objeto “Gestor informe”. Permitir la captura de algún texto corto que explique el contenido del archivo enviado. 	Contiene atributos para la captura de archivos, así como para un texto corto informativo.

	Interfaz de descarga	<ul style="list-style-type: none"> • Permitir la descarga de algún documento enviado. • Seleccionar el documento que se desea descargar. • Enviar datos al objeto "Gestor informe". 	Contiene atributos para la descarga y presentación de archivos.
Control	Gestor Informe	<ul style="list-style-type: none"> • Recibir datos de "Interfaz de envío". • Administrar el registro de archivos enviados. • Enviar datos a "Interfaz de salida". 	
Entidad	Informe	<ul style="list-style-type: none"> • Mantener los documentos enviados, así como la información referente a los mismos. 	<ul style="list-style-type: none"> • Apuntador a la Base de Datos (conexión abierta). • Realizador de consultas. • Realizador de actualizaciones, inserciones y borrado.

5.3 FLUJO DE DISEÑO

Como se dijo anteriormente, se continuó con el diseño de la arquitectura planteada durante la Fase de Elaboración. Esto incluye el modelo de despliegue, así como los subsistemas intermedios y de software del sistema y la dependencia entre ellos.

Durante este flujo se continúa con el diseño de los casos de uso faltantes a los presentados en la Fase anterior. Los casos de uso del paquete de "Gestión de Solicitudes", presentan una secuencia en clases de diseño similar al del caso de uso *Solicitar permiso*. Es por esto que la tabla 19 presenta la relación entre las clases de análisis y las clases de diseño de los diferentes casos de uso asociados a este paquete.

Tabla 19. Relación entre clases de análisis y diseño.

CASO DE USO	CLASES DE ANÁLISIS	CLASES DE DISEÑO
Solicitar licencia	Interfaz de Licencia	PaginaSolicitud
	Interfaz de Salida	PaginaConfirmacion
	Gestor Licencia	GestorLicencia
		Requisito
		Horario
Licencia	Licencia	
Solicitar Comisión	Interfaz de Comisión	PaginaSolicitud
	Interfaz de Salida	PaginaConfirmacion
	Gestor Comisión	GestorComision
		Requisito
		Horario
Comisión	Comisión	
Solicitar Período Sabático	Interfaz de Período	PaginaSolicitud
	Interfaz de Salida	PaginaConfirmacion
	Gestor Período	GestorPeriodo
		Requisito
		Horario
Período	Periodo	
Solicitar Vacaciones – Acumular Vacaciones	Interfaz de Vacaciones	PaginaSolicitud
	Interfaz de Acumulación	PaginaSolicitud
	Interfaz de Salida	PaginaConfirmacion
	Gestor Vacaciones	GestorVacaciones
		Requisito
Horario		
Vacaciones	Vacaciones	
Solicitar Prórroga	Interfaz de Prórroga	PaginaSolicitud
	Interfaz de Salida	PaginaConfirmacion
	Gestor Prórroga	GestorProrroga
		Requisito
		Horario
Prórroga	Prorroga	

Las instancias de la clase de diseño *PaginaSolicitud*, son específicas para cada caso de uso. Lo anterior significa que físicamente cada caso de uso usa una instancia diferente de esta clase, pero indistintamente de la presentación, se lleva a cabo la misma funcionalidad.

Los casos de uso de Vacaciones (Solicitar Vacaciones, Acumular Vacaciones), utilizan las mismas clases, pero como se puede observar en la tabla 3, cada una tiene su propia instancia de la clase *PaginaSolicitud*, como se explicaba en el párrafo anterior.

Las instancias de las clases de diseño *Requisito* y *Horario* son usadas por las instancias de las clases de diseño *Gestor* (*GestorLicencia* y *GestorComision*, entre otras), para hacer validaciones con los conceptos que éstas clases manejan: requisitos necesarios para el trámite, períodos y horarios de disponibilidad de las personas, entre otros.

5.3.1 Diseño de los casos de uso

Los restantes casos de uso son presentados a continuación.

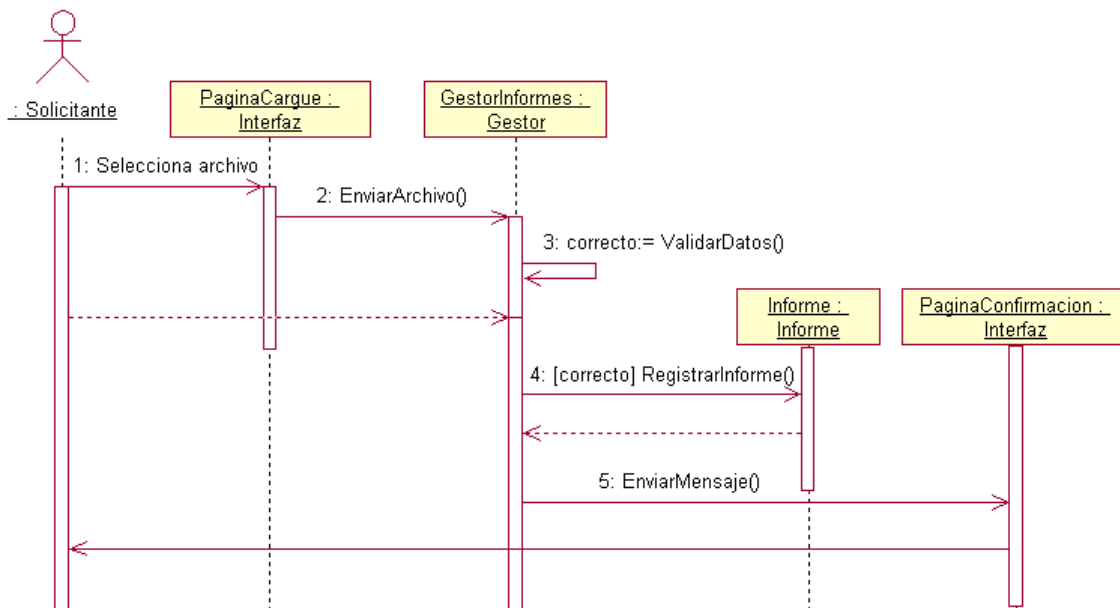
- **Enviar Informes.** La tabla 20 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 20. Clases de diseño – Enviar Informes.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de envío	PaginaCargue
Interfaz de salida	PaginaConfirmacion
Gestor Informe	GestorInformes
Informe	Informe

A continuación, la figura 91 presenta el diagrama de secuencia de estas clases.

Figura 91. Diagrama de secuencia – Enviar informes.



El actor selecciona por medio de la clase de interfaz, *PaginaCargue*, el archivo que desea enviar (1). La selección se realiza de manera similar al proceso de adjuntar archivos en un correo electrónico a través de Internet, vinculando archivos locales al equipo de cómputo donde se encuentre el actor. Una vez concluya la selección del archivo, la clase de interfaz se encarga de enviar el archivo junto con otros datos que identifican al informe (2), como la solicitud que generó el informe, entre otros.

Estos datos son validados por la clase *GestorInformes* (3). Si el usuario decide confirmar el envío del archivo, la clase gestor registra el envío del informe en el sistema (4), actualizando el listado de informes pendientes para el actor.

Seguidamente la misma clase se encarga de enviar un mensaje a la clase de Interfaz *PaginaConfirmacion*, con el fin de presentar un mensaje al actor (5).

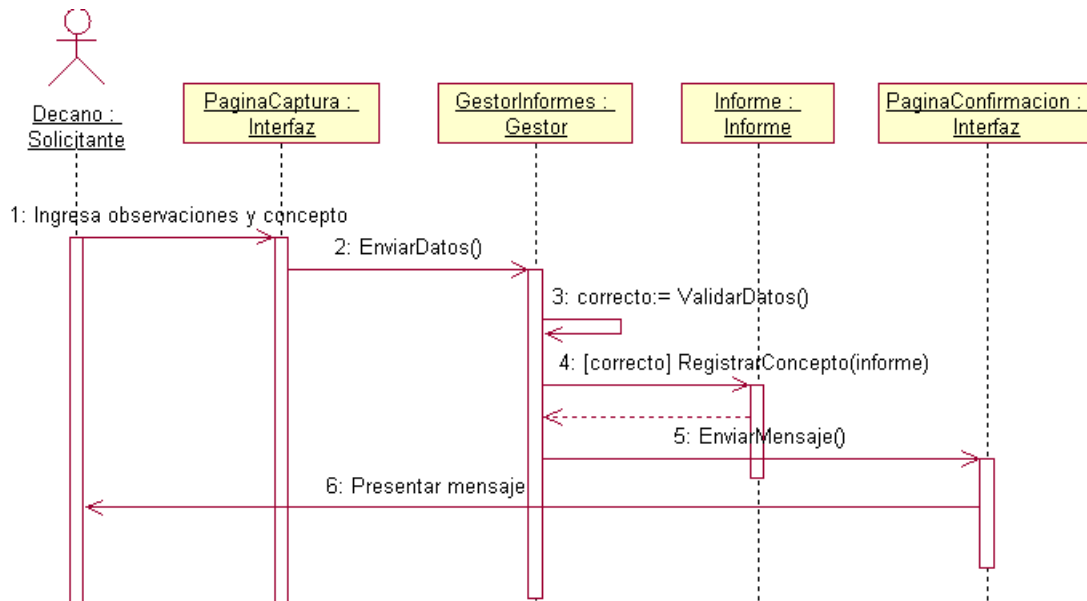
- **Registrar Concepto Informes.** La tabla 21 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 21. Clases de diseño – Registrar Concepto Informes.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
interfaz de captura	PaginaCaptura
interfaz de descarga	PaginaDescarga
interfaz de salida	PaginaConfirmacion
Gestor Informe	GestorInformes
Informe	Informe

A continuación, la figura 92 presenta el diagrama de secuencia de estas clases.

Figura 92. Diagrama de secuencia – Registrar Concepto Informes

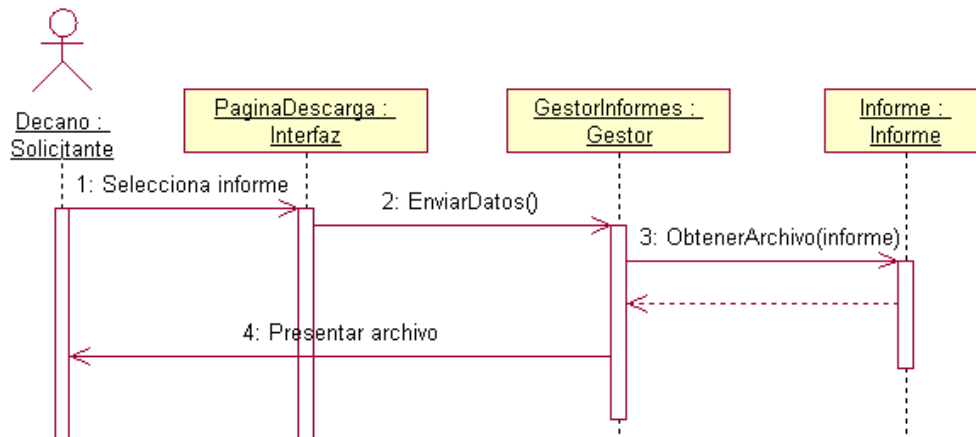


El actor Decano por medio de una clase de interfaz ingresa las observaciones correspondientes y el concepto de la evaluación realizada al informe (1). Estos datos son enviados a la clase de control (2), con el fin de ser validados (3) y posteriormente ser registrados en el sistema (4). Finalmente, se prepara un

mensaje de confirmación que se envía a la clase de interfaz (5), *PaginaConfirmacion*, con el fin de ser presentado al actor (6).

Una secuencia de acciones opcional a la presentada en el párrafo anterior, es cuando el actor decide primero descargar el informe y posteriormente ingresar el concepto. El descargue, se realiza mediante una clase de interfaz, *PaginaDescarga*, la cual proporciona toda la funcionalidad requerida para descargar el archivo. El diagrama de secuencias para esta opción es presentado en la figura 93.

Figura 93. Diagrama de secuencia – Registrar Concepto Informes (opcional).



Como lo muestra la figura anterior, una vez seleccionado el archivo que se quiera descargar, se procede a llamar a la clase gestor *GestorInformes*, con el fin de obtener el archivo (3) y ser presentado para descarga al actor correspondiente (4).

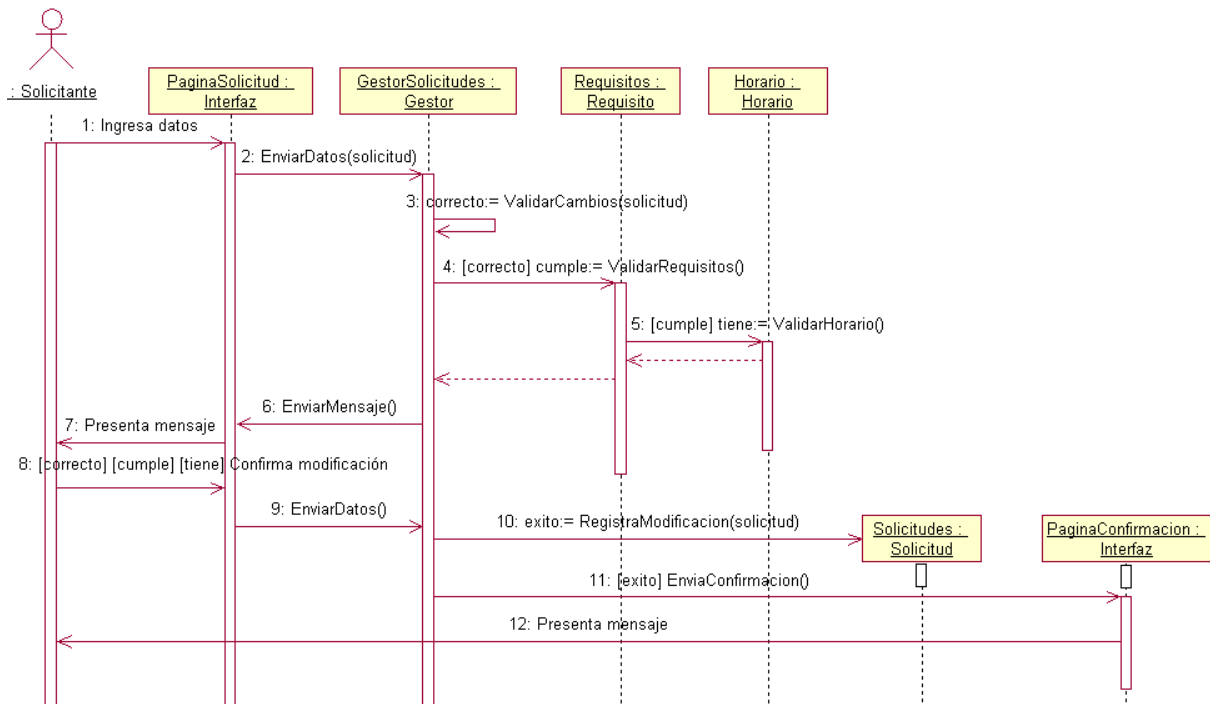
- **Modificar Solicitud.** La tabla 22 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

Tabla 22. Clases de diseño – Modificar Solicitud.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
interfaz de solicitud	PaginaSolicitud
interfaz de salida	PaginaConfirmacion
Gestor de solicitudes	GestorSolicitudes
	Requisitos
	Horario
Solicitud	Solicitudes

A continuación, la figura 94 presenta el diagrama de secuencia de estas clases.

Figura 94. Diagrama de secuencia – Modificar Solicitud.



Una vez seleccionada la solicitud que se desea modificar, ésta es presentada en una clase de interfaz, *PaginaSolicitud*, la cual corresponde al tipo de solicitud seleccionada (1). Por medio de esta interfaz se presentan los datos de la solicitud para que sean modificados. Si el usuario continúa con la modificación,

los datos de la solicitud son enviados a la clase *GestorSolicitudes* (2), la cual se encarga de realizar las validaciones correspondientes (3), así como las mismas comprobaciones que se llevaron a cabo cuando la solicitud fue registrada inicialmente (4) y (5).

Luego se muestra al actor un mensaje de confirmación de envío (6) y (7). Si el usuario confirma y cumple con todos los requisitos (8), los datos son enviados a la clase Gestor (9) la cual se encarga de actualizar los valores que están registrados en el sistema (10). La secuencia de acciones finaliza cuando la clase Gestor envía un mensaje a una clase de Interfaz (11), con el fin de presentarlo al actor (12).

- **Registrar Información.** La tabla 23 muestra la relación que existe entre las clases de análisis y las clases de diseño para este caso de uso.

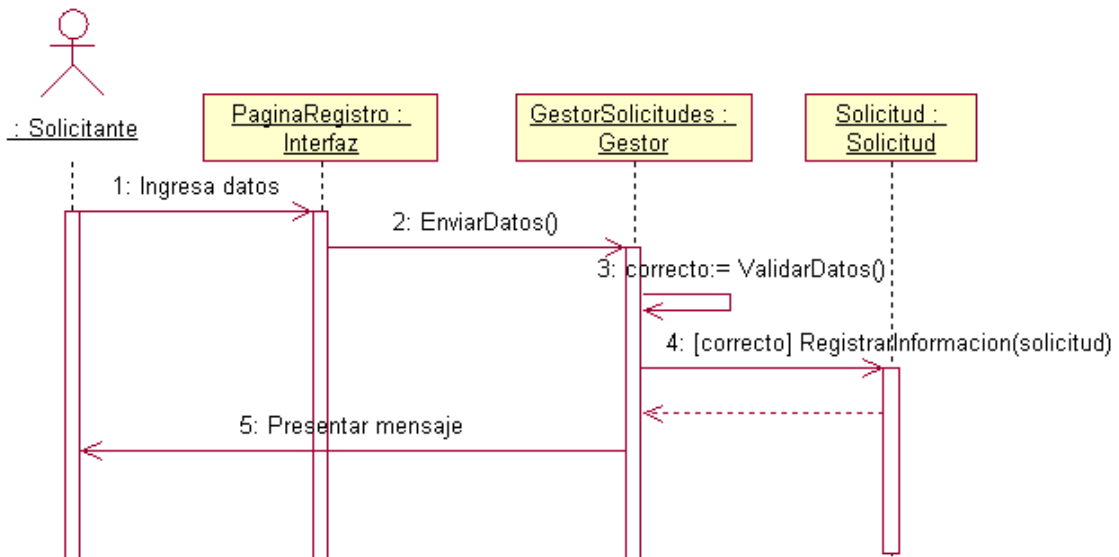
Tabla 23. Clases de diseño – Registrar Información.

CLASES DE ANÁLISIS	CLASES DE DISEÑO
Interfaz de captura	PaginaRegistro
Gestor de solicitudes	GestorSolicitudes
Solicitud	Solicitud

La figura 95 presenta el diagrama de secuencia de estas clases.

Por medio de la clase de interfaz, *PaginaRegistro*, se capturan los datos que se quieren registrar en el sistema (1). Estos datos son enviados a la clase de control *GestorSolicitudes* (2), con el fin de ser validados (3) y posteriormente registrados en el sistema, sí son correctos (4). Esta clase, prepara un mensaje que se presenta al actor en la misma clase de interfaz inicial *PaginaRegistro* (5), confirmando el éxito del registro.

Figura 95. Diagrama de secuencia – Registrar información.



5.3.2 Diseño de las clases. Las clases siguen las mismas directrices que se presentaron en la Fase anterior, es decir:

- Clases de interfaz: corresponden a las páginas ASP.
- Clases gestor: corresponden a las clases del archivo DLL.
- Clases de entidad: manejan la persistencia de los datos y corresponden a las tablas de la base de datos, que se encuentra en *INFORMIX*.

5.4 FLUJO DE IMPLEMENTACIÓN

Durante este flujo se lleva a cabo la implementación de los casos de uso documentados, con lo cual se lleva al sistema a una etapa totalmente funcional.

De manera similar a la Fase anterior, se presenta la implementación de las clases de interfaz, gestor y entidad.

5.4.1 Interfaces. Las nuevas clases de interfaz continúan utilizando los elementos que se presentaron en la Fase anterior, como son los formularios, las cajas de selección y las cajas de texto, entre otros. A través de éstos elementos se logran la captura, ingreso o selección de los datos presentados.

5.4.2 Clases. Se agregaron clases a las presentadas en el Fase anterior, para el manejo de las solicitudes de vacaciones y el envío de informes, así como métodos nuevos a las clases que existían para el manejo de las modificaciones y prórrogas.

Un complemento al modelo de clases presentado en la Fase anterior, en la figura 96.

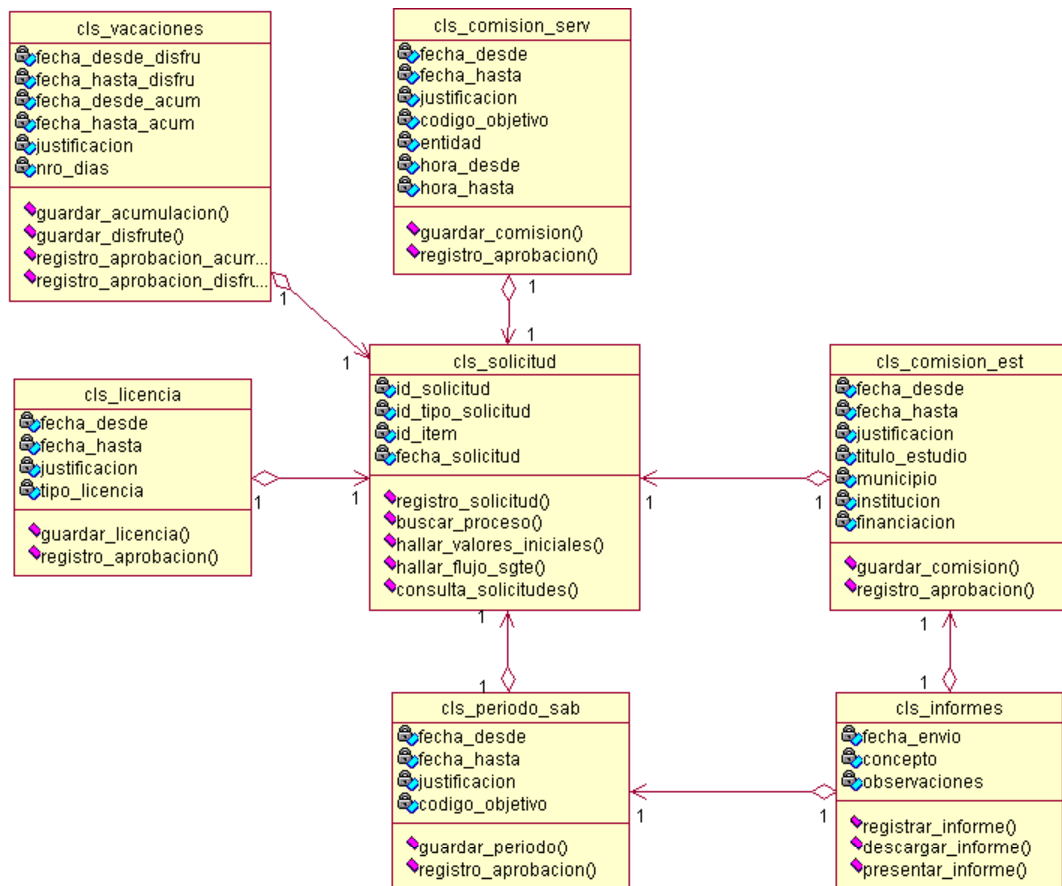
Existe una clase por cada solicitud que maneja el sistema, el nombre de la clase indica el tipo de solicitud, además se incluyó una solicitud para el manejo de informes.

Cada solicitud se realiza por un período de tiempo y con un motivo, representado en las clases por los atributos *fecha_desde*, *fecha_hasta* y *justificación*. Los atributos no fueron colocados en la clase *cls_solicitud*, con el fin de evitar errores sobre todo en ciertos métodos que manejan varias solicitudes y sus respectivos períodos.

La clase *cls_informes*, contiene un objeto de cada una de las clases que manejan solicitudes con informes (*cls_comision_est* y *cls_periodo_sab*), éstos con el fin de tener acceso a los métodos de estas clases y evitar confusiones.

La clase *cls_vacaciones*, agrupa los dos conceptos de vacaciones manejados por el sistema: Acumulación y Disfrute. Por esto los atributos que contiene son para cada uno de los conceptos.

Figura 96. Diagrama de clases – Fase de Construcción.



Cada una de las clases contiene una implementación adecuada de los métodos que se encargan de guardar la solicitud en el sistema, los que registran la aprobación de las mismas, así como los que se encargan de actualizar el flujo. Actualizar el flujo, indica cuál será el próximo usuario que revisará la solicitud y la actividad que va a realizar (ya sea de Aprobación o Visado).

5.4.3 Modelo de datos. Las funcionalidades implementadas durante esta Fase, utilizan algunos de los modelos presentados en la Fase anterior. Entre otros, los modelos de requisitos y actividades.

Cuando los procedimientos que el sistema maneja son aprobados, pasan a las tablas donde actualmente se registran las solicitudes que llegan a la División de

Recursos Humanos. A continuación, la tabla 24 presenta la relación entre los diferentes tipos de solicitud y las tablas correspondientes.

Tabla 24. Relación entre las solicitudes y tablas del modelo de datos.

TIPO DE SOLICITUD	TABLA DEL MODELO DE DATOS
Permisos	permisos_per
Licencias No Remuneradas	licencias_per
Comisiones de Servicio	comisiones_ser_per
Comisiones de Estudios	comisiones_est_per
Período Sabático	periodos_sabaticos
Acumulación de vacaciones	vacaciones_causada
Disfrute de Vacaciones	vacaciones_disfru
Informes	informes_comi_per; inform_periodo_per

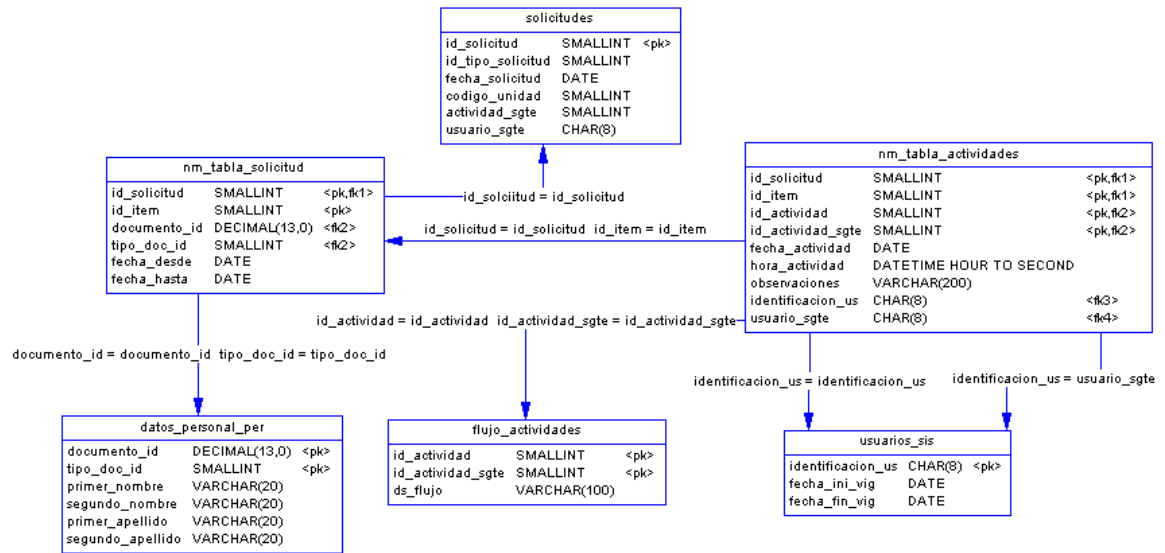
Para la solicitud de Licencias No Remuneradas, se utilizó el modelo presentado para la solicitud de Permisos, teniendo en cuenta que cuando la solicitud sea aprobada se registre en la tabla *licencias_per*, tal como lo muestra la tabla 25.

Para todas las solicitudes existen dos tipos de tablas. La primera de ellas donde se registran los datos de la solicitud una vez que un actor decida iniciar su trámite; y la segunda donde se registran las actividades por las cuales transita la solicitud de acuerdo a su flujo establecido. La figura 97, presenta un modelo general de cómo se integran éstas tablas con el modelo de datos del Sistema de Información de Recursos Humanos.

Las tablas fueron llamadas *nm_tabla_solicitud*, *nm_tabla_actividades* y corresponden a las tablas de las solicitudes y de las actividades de las mismas, respectivamente. A manera general, las tablas para el manejo de las solicitudes presentan las relaciones mostradas en la figura 97. Los atributos dependen del tipo de solicitud que se maneje y por lo general corresponden a los mismo atributos (con sus relaciones) para las tablas donde se registran las solicitudes aprobadas,

presentadas en la tabla 25. Además de datos generales como la identificación de la persona que tramita la solicitud (tipo y documento de identificación) y el período solicitado (expresado en fechas y horas dependiendo del tipo de solicitud).

Figura 97. Modelo de datos para las solicitudes.



La tabla 25 presenta los nombres de las tablas que manejan los diferentes tipos de solicitud.

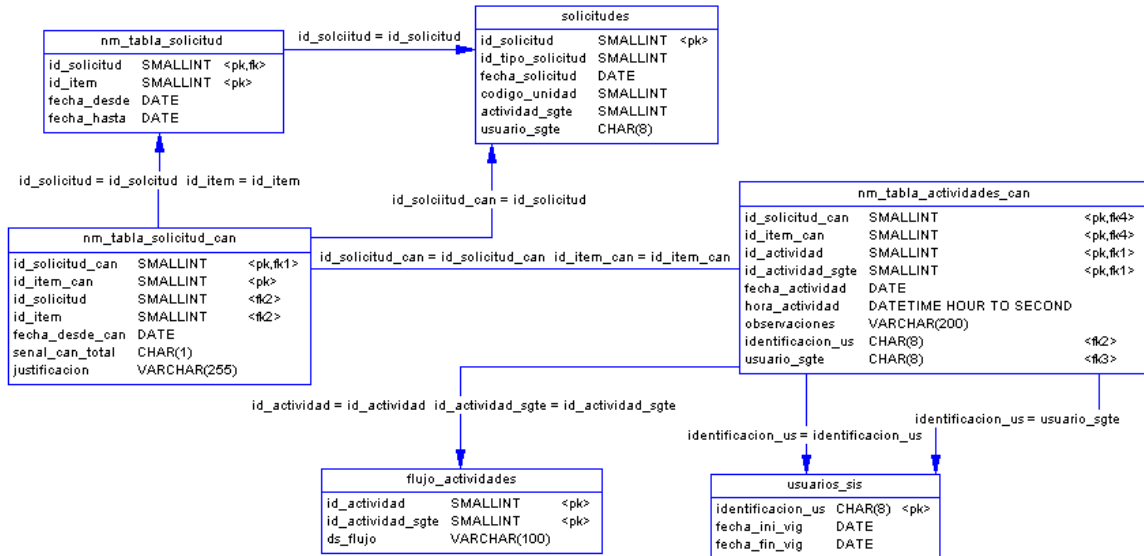
Tabla 25. Relación entre tipos de solicitud y tablas de manejo.

TIPO DE SOLICITUD	TABLA DE SOLICITUD	TABLA DE ACTIVIDADES
Permisos ²⁴ y Licencias No Remuneradas	sol_perm_lic_n	act_sol_perm_lic
Comisiones de Servicio	sol_comisiones_ser	act_sol_comi_ser
Comisiones de Estudios	sol_comisiones_est	act_sol_comi_est
Período Sabático	sol_periodo_sap	act_sol_per_sab
Acumulación de vacaciones	sol_acumula_vac	act_sol_acum._vac
Disfrute de Vacaciones	sol_disfru_vac	act_sol_disfru_vac

²⁴ El modelo completo para esta solicitud fue presentado en el capítulo Fase de Elaboración.

Para las solicitudes de cancelación el modelo de datos es muy similar al presentado anteriormente. La figura 98 muestra este modelo de datos.

Figura 98. Modelo de datos para las solicitudes de cancelación.



Las tablas del modelo de datos de solicitudes de cancelación se representan mediante la tabla *nm_solicitud_can*, donde se registra el identificador de la solicitud que se va a cancelar, la fecha a partir de la cual se cancela²⁵, la justificación de la cancelación y una señal que indica el tipo de cancelación (parcial o total). Las tablas de cancelación se relacionan con: las tablas de actividades que sigue la solicitud (*nm_tabla_actividades_can*); con las tablas de la solicitud dependiendo de su tipo (*nm_tabla_solicitud*) y con la tabla general de solicitudes (*solicitudes*) porque se registra la nueva solicitud que genera la cancelación.

Las tablas de actividades de cancelación se relacionan de la misma manera que las tablas de actividades de las solicitudes.

La tabla 26, describe lo anteriormente expuesto. En la columna TABLA DE SOLICITUD DE CANCELACION, se encuentran los nombres de las tablas donde se

²⁵ Sí la solicitud maneja horas, se registra también la hora desde de cancelación.

registran las solicitudes de cancelación y en la columna TABLA DE ACTIVIDADES DE CANCELACION, los nombres de las tablas donde se registra el flujo de actividades de las solicitudes de cancelación.

Tabla 26. Relación entre tipos de solicitud y tablas de manejo de cancelaciones.

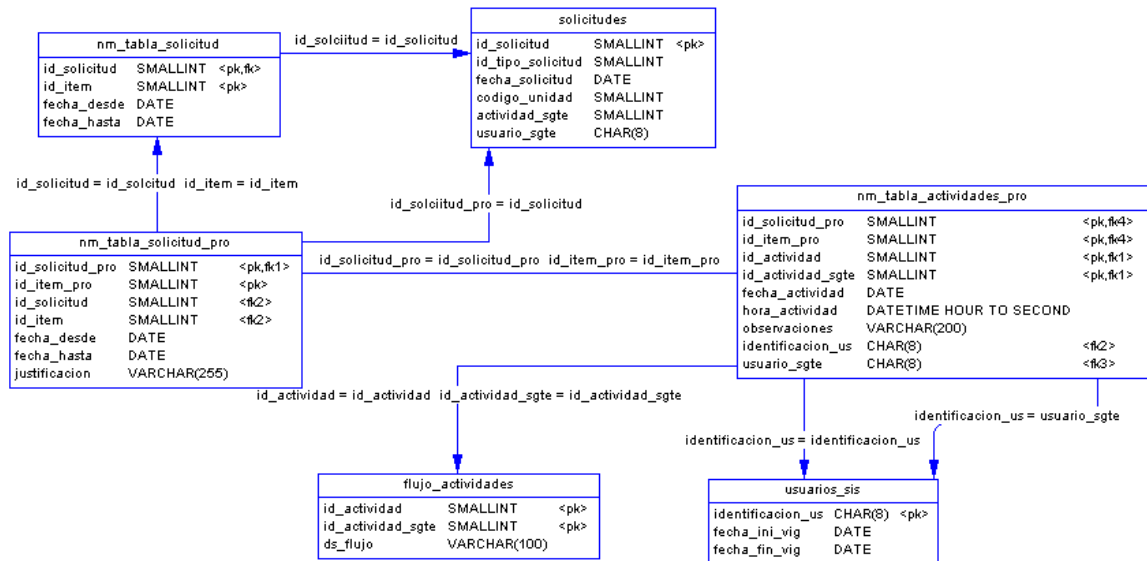
TIPO DE SOLICITUD	TABLA DE SOLICITUD DE CANCELACION	TABLA DE ACTIVIDADES DE CANCELACION
Permisos y Licencias No Remuneradas	Sol_can_perm_lic	act_sol_can_per_lic
Comisiones de Servicio	Sol_can_comi_ser	act_sol_can_comiser
Comisiones de Estudios	Sol_can_comi_est	act_sol_can_comiest
Período Sabático	Sol_can_periodo_sab	act_sol_can_persab

Cuando una solicitud de cancelación es aprobada, se afecta la solicitud que está registrada en la tabla del modelo de datos del Sistema de Información de Recursos Humanos, que de acuerdo a su tipo se muestran en la tabla 25.

Para las solicitudes de prórroga, se consideraron las solicitudes de *Comisión de Estudios*, *Comisión de Servicios* y *Licencias No Remuneradas*. El modelo de datos general para estas solicitudes se presenta en la figura 99.

Las tablas de solicitudes de prórroga se representan mediante la tabla *nm_tabla_solicitud_pro*. En estas tablas se registran el identificador de la solicitud que quiere prorrogar, el período y la justificación de la prórroga solicitada. Se relacionan con las siguientes tablas: La que contiene las actividades que sigue la solicitud (*nm_tabla_actividades_pro*), la tabla donde se encuentra la solicitud que se quiere prorrogar (*nm_tabla_solicitud*) y con la tabla general de solicitudes (*solicitudes*), al generar la nueva solicitud.

Figura 99. Modelo de datos para las solicitudes de prórroga.



Las tablas de actividades de las solicitudes de prórrogas se relacionan de manera similar a las presentadas en las figuras 97 y 98.

5.5 FLUJO DE PRUEBAS

Durante este flujo se buscan las posibles fallas y faltas en la funcionalidad de cada uno de los casos de uso desarrollados.

En cada una de las Fases del desarrollo se llevaron a cabo distintas pruebas, tal como lo sugiere la metodología de desarrollo.

En la Fase de Inicio, se realizaron pruebas con el fin de obtener una arquitectura candidata para el sistema. Para ello se implementaron unas interfaces preliminares en el lenguaje HTML, con el propósito de verificar el ingreso de los usuarios e identificar la manera más adecuada de realizar la captura de los datos para las solicitudes consideradas.

Además se crearon los primeros métodos en la DLL con el fin de probar el funcionamiento del Driver SDK Informix para realizar la conexión y las consultas sencillas con esta base de datos.

Para la Fase de Elaboración, en la cual se implementaron los casos de uso denominados críticos, las pruebas se concentraron en el funcionamiento de la arquitectura seleccionada. Para ello las interfaces pasaron de HTML a ASP; se incluyó código de Script (JavaScript) para validar los datos capturados en estas interfaces; se crearon las clases para el control de las solicitudes²⁶, y las funcionalidades de los casos de uso considerados y se incluyó al Sistema de Información de Recursos Humanos los modelos de datos necesarios para ello. Con esto, se realizaron solicitudes de prueba de cada uno de los tipos desarrollados, con el fin de optimizar y establecer un marco general para el manejo de las demás solicitudes.

De esta manera durante esta Fase, las pruebas se fijaron en: registrar solicitudes, establecer los flujos o caminos que debían seguir, darles trámite (aprobar o rechazar) y finalmente registrar el reintegro, si la solicitud fue aprobada.

Para la Fase de Construcción, las pruebas estuvieron orientadas a integrar las demás solicitudes a la estructura general planteada. Se trató de adaptar en lo posible las interfaces a los restantes casos de uso; se optimizaron los métodos en la DLL tanto para las solicitudes ya implementadas de la Fase anterior como para los nuevos casos de uso y se incluyeron otros modelos de datos al Sistema de Información. Las pruebas, además de ser similares a las realizadas en la Fase anterior, estuvieron centradas en optimizar el funcionamiento en conjunto, el desempeño de métodos compartidos en el manejo de los casos de uso desarrollados y la correcta realización de los diferentes caminos presentados durante los flujos de análisis y diseño.

²⁶ A pesar que el procedimiento tratado en detalle fue el de Permiso, este contiene diferentes tipos de solicitud: Permisos Individuales, Grupales y Causados, entre otros.

De acuerdo con PRESSMAN²⁷, un producto de ingeniería puede probarse de dos formas:

- Pruebas de Caja Blanca: Se basa en el minucioso examen de los detalles procedimentales (los procedimientos y métodos creados).
- Pruebas de Caja Negra: Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software.

Las pruebas de Caja Blanca fueron realizadas en el lenguaje de programación *Visual Basic*, el mismo en que fue desarrollada la DLL. Utilizando este lenguaje de programación es posible realizar las pruebas de los métodos antes de utilizarlos desde las páginas ASP. Para ello se creó un formulario de pruebas, el cual tenía como propósito probar los procedimientos creados, evaluando los caminos que siguen los datos de entrada y las salidas respectivas.

El formulario de pruebas, básicamente contiene las entradas a los métodos de cada una de las clases creadas y realiza los llamados correspondientes a los mismos, tal como se realiza desde una página ASP.

A continuación se muestra un ejemplo de la utilización de este formulario.

El método `Consultar_Dia` (que se encuentra en la clase `cls_horario`), recibe como argumento la conexión a la base de datos y como parámetro de consulta una fecha (en el formato `mm/dd/yyyy`); tiene como propósito obtener una señal, la cual se obtiene de la tabla `dias_nohabil_uis`, que indicará si la fecha ingresada corresponde a un día festivo o no. El código para este método es:

```
Public Sub Consultar_Dia(mobjconn As cls_conexion)
Dim sql As String
Dim valor As String
```

²⁷ PRESSMAN, Roger S. Ingeniería del Software. Quinta Edición. Mc Graw Hill. Madrid 2002. Pág. 285

```

sql = "SELECT senal_festivo " & _
      "FROM dias_nohabil_uis " & _
      "WHERE fecha = '" & sParametro_consulta & "'"
mobjconn.setSql = sql
Set rc = mobjconn.ejecutarConsulta
valor = ""
While Not (rc.EOF)
    valor = rc.Fields("senal_festivo").Value
    rc.MoveNext
Wend
rc.Close
Set rc = Nothing
sValor_consulta = valor
Exit sub
End sub

```

Para la prueba del método anterior desde el formulario, se instancian los valores requeridos y se hace el llamado al método, el código de esta prueba es el siguiente.

```

Dim con as new cls_conexión
Dim horario as new cls_horario
horario.setParametro_consulta = "12/25/2003"
call con.conectar
call horario.Consultar_Dia(con)
senal = horario.getValor_consulta
call con.desconectar()

```

Previamente al llamado del método existen una serie de validaciones que aseguran que el parámetro de entrada corresponde a una fecha y se encuentra en el formato requerido. Si la fecha no fue encontrada, la variable instanciada tendrá el valor de vacío, el cual será utilizado para indicar que la fecha no fue encontrada.

Utilizando las herramientas de depuración con las que cuenta este lenguaje, es posible realizar un seguimiento del comportamiento de los datos ingresados línea por línea dentro del método.

Las pruebas de Caja Negra, pretende entre otros objetivos:

- Hallar funciones incorrectas o ausentes.
- Establecer posibles errores de interfaz.
- Encontrar errores de rendimiento.
- Errores de iniciación y terminación (por ejemplo, de conexiones a la base de datos).

Según los datos que se capturan en cada una de las interfaces, se estableció cuales eran las funciones JavaScript requeridas para validar los datos antes de ser enviados. Lo anterior permitió la creación de una serie de funciones comunes y utilizadas a lo largo del desarrollo, además de funciones específicas para ciertas interfaces que permiten una mejor verificación de los datos que van a ser enviados.

Durante el avance en el desarrollo, se encontraron las posibles fallas en relación con los datos que las interfaces presentaban. Resultado de las pruebas se establecieron tanto la forma en la cual se debían capturar los datos, como el orden de validación y la manera en la cual se presentan los datos de solicitudes para su evaluación.

La figura 100, presenta la interfaz de evaluación para una solicitud de Acumulación de vacaciones. En la parte superior derecha se muestra el vínculo referente al trámite de la solicitud, esto con el propósito que el actor actual tenga la posibilidad de mirar el flujo de actividades que ha seguido la solicitud y las respectivas observaciones que se han hecho. La sección 1, presenta datos generales a todas las solicitudes como son: la fecha en la que se realizó la solicitud, el nombre el solicitante y la unidad a la que pertenece. La sección 2, presenta datos específicos de la solicitud como son: el período en fechas (sí la solicitud maneja horas también

se presentan) y el número de días que se solicitan, entre otros. Adicionalmente mostrado, se presenta el período por el cual se causaron las vacaciones.

Figura 100. Modelo de interfaz de evaluación.

Vínculos: [Ver el trámite de la solicitud](#)

ESPECIFICACIONES DE LA SOLICITUD DE ACUMULACION DE VACACIONES NUMERO 3069				
Sección 1	Fecha Solicitud	Solicitante		Unidad
	Dic 22 de 2003	OLGA GAMBOA PORRAS		DIVISION DE SERVICIOS DE INFORMACION

ESPECIFICACIONES DE LA ACUMULACION				
Sección 2	Inicio Periodo	Fin Periodo	Fecha Desde	Fecha Hasta
	Jul 26 de 2003	Jul 25 de 2004	Dic 24 de 2003	Ene 06 de 2004
Días Hábilies				
8				
Justificación				
SOLICITUD DE ACUMULACION DE VACACIONES DE PRUEBA				

ACCION SOBRE ESTA SOLICITUD			
Aprobar	Rechazar	Observaciones	Caracteres restantes
<input type="radio"/>	<input type="radio"/>	<input type="text"/>	100

Finalmente la sección 3, presenta las opciones que se pueden realizar sobre la solicitud (se escoje una de las dos) y las observaciones que el actor puede realizar sobre la solicitud.

Respecto a los errores de rendimiento, estos fueron mitigados a lo largo del desarrollo mediante la colocación del código encargado de realizar la lógica del negocio en la DLL y no en las páginas. Esto con el propósito que el código ASP se encargue de dar el formato necesario a los datos que se van a presentar y no de realizar comprobaciones que exigen estar conectado a la base de datos.

Respecto de los errores de iniciación y terminación, en lo referente a conexiones con la base de datos y los objetos de la DLL creados en las páginas, resultado de las pruebas de las interfaces se estableció que la forma en la cual se garantizaba un cierre correcto de la conexión y la liberación de los objetos en el servidor, era la desconexión a la base de datos y la destrucción de todos los objetos creados, al final de cada página; contrastando con lo realizado al inicio del desarrollo en el cual se realizaban conexiones y desconexiones a la base de datos, en cada uno de los

métodos, lo cual generaba continuas solicitudes de conexión para una misma página.

Para las últimas fases se construyeron y se llevaron a cabo casos de prueba, que mostrarán una aproximación de lo que se podría presentar una vez el sistema entre en funcionamiento. Entre otros casos se pueden mencionar:

- Registro de reintegros antes y después del tiempo solicitado de acuerdo al horario del funcionario.
- Posibles cruces entre solicitudes del mismo tipo.
- Incumplimiento de requisitos en el inicio del trámite de una solicitud.
- Validaciones de los períodos solicitados contra días hábiles y horarios establecidos.
- Cancelaciones y prórrogas de solicitudes aprobadas.
- Envíos y evaluaciones de informes.
- Verificación de requisitos para cada solicitud por ejemplo: tiempo de servicio, tiempo máximo solicitado y categoría del funcionario, entre otros.

5.6 CONCLUSIONES DE LA FASE DE CONSTRUCCIÓN

Durante esta Fase se llevaron a cabo las siguientes actividades:

Se completó la especificación de los casos de uso planteados. Para ello se realizó la descripción de los casos de uso denominados Importantes y Secundarios durante la Fase anterior.

Se finalizó la inclusión de los modelos de datos al Sistema de Información de Recursos Humanos necesarios para el funcionamiento de la aplicación.

Se optimizó el funcionamiento de métodos compartidos por diferentes solicitudes, como los encargados de actualizar el flujo que siguen las solicitudes.

En las tres iteraciones planeadas se finalizó la implementación y se comprobó la funcionalidad de los subsistemas específicos de la capa de aplicación.

Se obtuvo la primera versión de la aplicación, la cual implementa los casos de uso planteados al inicio del proyecto, utilizando la arquitectura definida en las Fases anteriores.

Las pruebas descritas, tanto de Caja Blanca con el formulario de pruebas como las de Caja Negra realizadas con las interfaces y su funcionamiento en conjunto con los objetos creados de la DLL, permitieron llevar el sistema hasta un punto de desarrollo donde se considera cubierta la funcionalidad planificada. Se recomienda para el Flujo de Transición realizar pruebas de tensión o estrés y pruebas negativas²⁸, con un número considerable de usuarios. Lo anterior con el propósito de observar el comportamiento del sistema en ambientes de concurrencia y optimizar el funcionamiento del mismo bajo estas condiciones.

²⁸ JACOBSON, Ivar. BOOCH, Grady. RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Addison Wesley. Madrid 1999. Página 285.

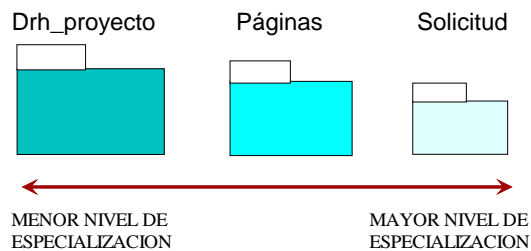
6. ESTRUCTURA DEL SISTEMA

6.1 DISTRIBUCIÓN DE PÁGINAS EN EL SERVIDOR

La distribución a la que se hace referencia, corresponde a la estructura manejada para el almacenamiento de páginas y archivos relacionados del Sistema de Información Web de la División de Recursos Humanos de la Universidad Industrial de Santander.

Con el fin de lograr un sistema estructurado y por consiguiente ordenado y de fácil acceso y seguimiento, se clasificaron y empaquetaron las páginas ASP y los archivos requeridos por el sistema en carpetas y subcarpetas; creando así un árbol jerárquico de almacenamiento que va desde los más general a lo más específico.

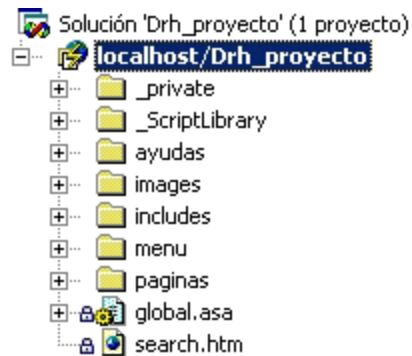
Figura 101. Niveles jerárquicos en la distribución de archivos en el sistema.



La distribución de páginas en el servidor Internet Information Server (IIS) del sistema inicia en la raíz del sistema o punto de partida para cada uno de los archivos que conforman el sitio, la cual se denomina “localhost/Drh_proyecto”.

Seguidamente se encuentran dos carpetas propias del IIS, `_private` y `_ScriptLibrary`; destinadas a almacenar los archivos que no serán visibles para el visitante (`_private`) y el código ASP script que será solicitado por los clientes utilizando JavaScript (`_ScriptLibrary`).

Figura 102. Distribución de Páginas y archivos en el servidor IIS.



Seguidamente se encuentran las carpetas de: ayudas, images, incluye, menu y paginas.

- Ayudas: Contiene las páginas de ayuda correspondientes a cada una de las páginas con las cuales interactúa el usuario. Por medio de estas páginas el usuario cuenta con un soporte inmediato que lo guía mientras se encuentra en el sistema.
- Images: Contiene todas las imágenes presentes en las páginas ASP del sistema.
- Includes: Contiene el archivo de estilos en el cual se definen estilos específicos para la presentación de títulos, subtítulos, tablas y viñetas, entre otros. Es de gran importancia contar con este archivo de estilos en donde se especifican los formatos gráficos de los componentes de las páginas Web, en lugar de formatear los mismos y de manera independiente en cada una de las páginas.

En el momento de realizar un cambio significativo en el diseño gráfico del sistema y no contar con el archivo de estilos, el código de cada una de las páginas que conforman el sitio debe ser alterado. Con el archivo de estilos este cambio es reducido significativamente debido a que los estilos se afectan en un único archivo que es accedido por las páginas.

En cada una de las páginas del sistema encontramos la siguiente directiva HTML

```
<link REL="stylesheet" TYPE="text/css" HREF="<ruta><nombre_archivo_css>">
```

Ejemplo:

```
<link REL="stylesheet" TYPE="text/css" HREF="../../includes/estilos.css">
```

Esta directiva dice al servidor que debe asociar el archivo de estilos llamado “estilos.css” a la página que se encuentra procesando esta información.

El archivo de estilos contiene información estructurada de la siguiente manera:

```
.link
{
  COLOR: #407931;
  TEXT-DECORATION: underline
  FONT-FAMILY: Verdana, Arial, Helvetica, sans-serif;
  FONT-SIZE: 10px;
  FONT-WEIGHT: normal
}
```

Este código describe la clase “link”, especificando el color (en hexadecimal), la decoración del texto (subrayado=underline), la familia de la fuente, el tamaño de la fuente (10 pixeles) y el ancho de la misma (normal).

El llamado a esta clase es realizado de la siguiente forma:

```
<a href="javascript:ver_horas();" class="Link">Ver el estado de horas</a>
```

Este código dice al navegador que para la etiqueta HTML que describe un link “<A>” relacione el estilo descrito por la clase “link” mediante la directiva class=“Link” para obtener el resultado deseado: Ver estado de horas.

- Menu: Contiene todos los archivos necesarios para que el navegador interprete y despliegue en pantalla el menú dinámico presente en el inbox del sistema.

- Páginas: Corresponde a la carpeta principal del sistema de información, en la cual se almacenan todas las páginas por medio de las que el usuario hará uso de las funcionalidades del Sistema de Información de Recursos Humanos.

Subcarpetas presentes en la carpeta principal “paginas”:

- Cancelación: Almacena las páginas necesarias para iniciar procesos de cancelación de solicitudes.
- Comisiones: Almacena las páginas necesarias para iniciar los procedimientos de comisión de estudios y de servicios.
- Consultas: Almacena las páginas necesarias para la realización de consultas de apoyo para el trámite de cualquier solicitud o su correspondiente visado, aprobación, rechazo o cancelación.
- Funcionalidades: Almacena las páginas necesarias para realizar las acciones de visado, aprobación, rechazo, reintegro, cruce de tiempos, asignar resolución, confirmación y cancelación de permisos grupales.
- Informes: Almacena las páginas necesarias para el manejo de informes de las situaciones administrativas que lo requieran.
- Licencias: Almacena las páginas necesarias para iniciar el procedimiento de licencia no remunerada.
- Modificación: Almacena las páginas necesarias para realizar la modificación de situaciones administrativas permitidas.
- Permisos: Almacena las páginas necesarias para iniciar los procedimientos de permiso causado, compensatorio, grupal, individual y para compensar.

- Prorroga: Almacena las páginas necesarias para realizar la solicitud de prórroga de las situaciones administrativas prorrogables.
- Sabático: Almacena las páginas necesarias para iniciar el procedimiento de periodo sabático.
- Vacaciones: Almacena las páginas necesarias para tramitar la acumulación y el disfrute de vacaciones.

El archivo Global.asa al final del árbol de directorios del proyecto es un archivo opcional que puede contener declaraciones de objetos, variables y métodos, los cuales se utilizan por las páginas de una aplicación ASP. Todos los navegadores que soporten scripts (JavaScript, VBScript, etc) pueden trabajar con el archivo Global.asa.

El archivo Global.asa puede contener únicamente los siguientes elementos:

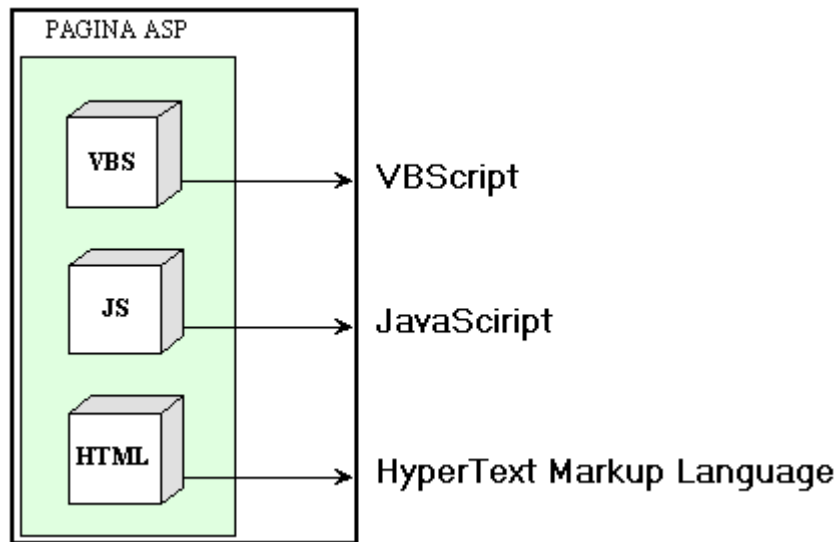
- Eventos de la aplicación: Usados para definir lo que hace la aplicación cuando inicie o finalice.
- Eventos de la sesión: Usados para definir lo que hace la aplicación cuando una sesión inicie o finalice.
- Declaración de objetos: Es posible declarar objetos de sesión o de extensión de la aplicación de acuerdo a los requerimientos de los mismos.
- Declaraciones TypeLibrary: Estas declaraciones empaquetan el contenido de una Dll correspondiente a un objeto COM.

Nota: El archivo Global.asa debe ser almacenado en el directorio raíz de la aplicación ASP, y cada aplicación puede tener únicamente un archivo Global.asa.

6.2 ESTRUCTURA DE PÁGINAS ASP DEL SISTEMA

Las páginas ASP (Active Server Pages) del sistema contienen código Visual Basic Script, Java Script y HTML, cada uno de los cuales cuenta con una función específica descrita a continuación.

Figura 103. Esquema de pagina ASP.



El código Visual Basic Script es el encargado de realizar todas las operaciones del lado del servidor en las cuales sea necesario el acceso a las clases y métodos de la librería dinámica (DII).

El código Java Script es el encargado de realizar operaciones del lado del cliente (validación de formularios).

El código HTML el cual es ejecutado del lado del cliente es el encargado de presentar la interfaz al usuario.

A continuación un ejemplo del uso del código Visual Basic Script:

<pre><%@ Language=VBScript %> <% CODIGO VISUAL BASIC SCRIPT %></pre>	<p>Definición del lenguaje Abre para iniciar secuencia de código VBS Líneas de código VBS Cierra para finalizar código VBS</p>
--	--

Ejemplo de empleo del código Java Script

<pre><script> CODIGO JAVA SCRIPT </script></pre>	<p>Abre para iniciar secuencia de código JS. Líneas de código JS. Cierra para finalizar código JS.</p>
--	--

Ejemplo de empleo del código HTML

```
<html>
<head>
<title>Título de la página</title>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</head>
<body>
<form name="Nombre del formulario" method="post">
    CODIGO HTML
</form>
</body>
</html>
```

6.3 SEGURIDAD DEL SISTEMA

La seguridad en el sistema permite a través de Visual Basic Script validar la seguridad mediante el uso de variables de sesión.

La seguridad del sistema fue desarrollada con el fin de lograr:

- *Obligatoriedad de Identificación.* Corresponde a la obligatoriedad de identificarse ante el sistema por medio de un nombre de usuario y contraseña en la página de ingreso "pg_ingreso.asp".
- *Control de navegación.* Validar que la navegación por el sitio sea de acuerdo a la permitida por los flujos de los procedimientos, denegando el acceso a páginas

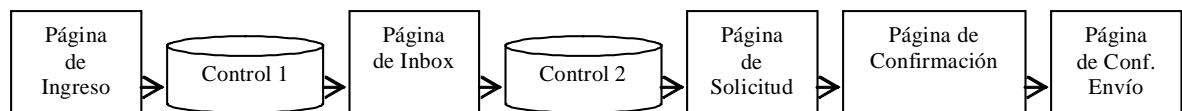
mediante el uso de la barra de direcciones del Internet Explorer o del browser que se encuentre utilizando, evitando así el salto entre páginas no subyacentes.

- *Control de recarga de páginas.* Controlar el uso continuo del botón de refresco o tecla F5 en Internet Explorer.

Las variables de sesión son objetos instanciados en un punto determinado de la aplicación, y cuyo valor perdura durante un tiempo asignado al objeto mediante el método "session.timeout = valor_en_minutos", o hasta que el usuario finalice su sesión de trabajo utilizando indirectamente por medio del sistema el método "session.abandon".

Para el correspondiente manejo de la seguridad en las páginas del sitio, el servidor continuamente valida la existencia de las variables de sesión durante la navegación a través del mismo.

Figura 104. Flujo de navegación en el sistema.



Descripción del flujo de navegación:

- *Página de Ingreso:* Corresponde a la página en la cual el usuario se autentifica ante el sistema. Esta es una página provisional para la realización del caso de uso "Ingreso".

El objeto sesión, se instancia con las siguientes variables en el momento en que un usuario ingresa al sistema:

session("Nombre_Completo") = Nombre completo del usuario.
session("Identificacion") = Identificación en el sistema.

session("Documento")	= Documento de Identificación.
session("tipo_doc")	= Tipo del documento.
session("Codigo_uni")	= Código de la Unidad a la que pertenece.

- *Contro1,2*: Estas páginas tienen la funcionalidad de asignar variables de sesión para el control del flujo de navegación, así como para mantener datos significativos del usuario almacenados en una variable global, para que se puedan utilizar en el momento en que se necesiten en las páginas del sitio; cuando finaliza la sesión estas variables son descargadas de memoria. Estas dos páginas son transparentes al usuario.

Una funcionalidad importante de control 1 y control 2 es que no permiten que el usuario trate de enviar una solicitud bajo el mismo número utilizando los botones atrás y delante de su Explorador Web.

- *Inbox*: Inbox del sitio de acuerdo al actor.
- *Página de solicitud*: Corresponde a la interfaz del formulario de captura de datos (formulario de solicitud).
- *Página de confirmación*: Presenta al solicitante los datos que se van a enviar de la solicitud en sí.
- *Página de conf. de envío*: (Página de confirmación de envío) Corresponde a la página que confirma al usuario el resultado del envío de su solicitud, si fue o no satisfactorio; en caso de ser satisfactorio presenta el número de solicitud asignado.

En la página de solicitud se valida que la sesión del usuario no haya caducado o que el usuario no haya accedido a dicha página mediante la utilización de la barra de direcciones de su navegador (realizar salto en el flujo de páginas); para tal caso,

el sistema cuenta con una variable de sesión que controla que la página sea encaminada por el sistema y no manualmente por el usuario.

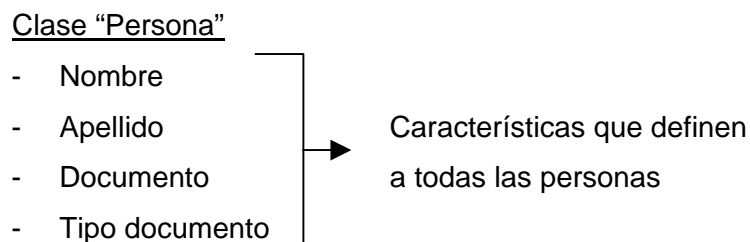
Si el sistema verifica que se ha realizado un salto entre páginas, desvía el flujo a la página de inbox, presentando un mensaje de error al usuario, el cual informa que ha intentado acceder a una página por una ruta no especificada o que su sesión ya ha caducado.

6.4 LIBRERÍAS DINÁMICAS (DLL) – SOLDRH.DLL

Como su nombre sugiere, los archivos DLL son colecciones de procedimientos en código máquina encapsulados. Estos procedimientos se vinculan a una aplicación solo en tiempo de ejecución, esto implica que las bibliotecas se pueden actualizar independientemente de la aplicación.

6.4.1 Clases y definición de clases. Una clase es el conjunto de todas las entidades que existen (o que pudieran existir) en la realidad y que tienen características definidas. La definición de una clase (un componente .cls en Visual Basic) es una representación de esa clase que ofrece alguna utilidad desde el punto de vista del sistema. Una variable, declarada como un miembro de esa clase con sus propiedades debidamente configuradas, representa una instancia u objeto de esa clase.

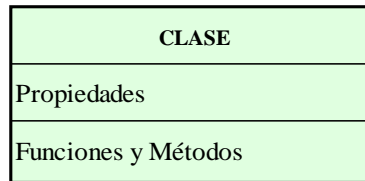
Ejemplo:



Teniendo en cuenta que la definición describe a cualquier miembro de la clase, cada "persona" corresponde a un objeto particular (o instancia) de la clase persona.

En Visual Basic, una clase no es más que un objeto; este objeto, tiene propiedades, funciones y métodos. Las clases son utilizadas para manipular la información interna de un programa o sistema, manteniendo un código ordenado y reutilizable en cualquier aplicación.

Figura 105. Componentes de una clase.



6.4.2 Definiciones relacionadas con el concepto de clase

- *Objeto*: Un objeto es una entidad que tiene asociado un conjunto de métodos, eventos y propiedades. Hay muchas clases de objetos, y por tanto, puede llegar a haber tantos métodos, eventos y propiedades distintas como objetos diferentes.
- *Propiedades*: Puede definirse a una propiedad como un elemento de un objeto que puede ser alterado, ya sea directamente (mediante el código) o indirectamente (mediante las acciones de un usuario). Las propiedades de los controles tienden a ser persistentes durante la vida de una aplicación y se establecen mediante tres procedimientos: Property Set, Property Let y Property Get.

Property Set: se ejecuta cuando el usuario asigna un objeto a una propiedad.

Property Let: se ejecuta cuando el usuario asigna un valor a una propiedad.

Property Get: se ejecuta cuando el usuario lee una propiedad.

- *Métodos*: Son procedimientos definidos en Visual Basic para realizar operaciones específicas sobre los objetos (Controles o Formularios). Estos son

la forma en que el programador hace que el control ejecute alguna acción. Los métodos se instauran mediante procedimientos públicos Function o Sub, lo cual depende si el método devolverá o no información.

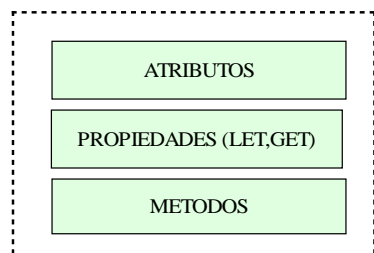
6.4.3 Estructura de las clases de la librería dinámica. La librería dinámica “Dll” denominada “soldrh.dll”, desarrollada bajo el lenguaje de programación Visual Basic 6.0 contiene las clases las cuales contienen los métodos, atributos y propiedades necesarias para cumplir con cada procedimiento y funcionalidad con que cuenta el Sistema de Información.

Para cada una de las solicitudes contempladas por el sistema de información se desarrolló una clase encargada de realizar su correspondiente gestión (cls_permiso, cls_licencia, cls_vacaciones, entre otras), con el fin de disminuir al máximo la dependencia entre los mismos.

La estructura de cada una de estas clases se define en tres secciones:

- Definición de atributos
- Definición de propiedades
- Definición de métodos (funciones o procedimientos)

Figura 106. Estructura de una clase.



6.4.4 Clase de ejemplo “persona.cls”. A continuación se presenta la clase de ejemplo denominada persona.cls en donde se identificarán el código de cada una de las secciones de la misma.

cls_persona

'DEFINICIÓN DE ATRIBUTOS

```
Private sNombre_persona as string
Private sApellido_persona as string
Private iDocumento as integer
Private bytTipo_documento as byte
```

'DEFINICIÓN DE PROPIEDADES

```
Public Property Let setNombre_persona(ByVal vdata As string)
    sNombre_persona = vdata
End Property
Public Property Get getNombre_persona() As string
    getNombre_persona = sNombre_persona
End Property
```

...

'SE DEFINEN LAS PROPIEDADES SET Y GET PARA CADA ATRIBUTO
DE LA CLASE

...

'DEFINICIÓN DE MÉTODOS

```
Public Sub Consultar_persona(mobjconn as cls_conexion)
```

'UTILIZANDO EL DOCUMENTO Y EL TIPO DE DOCUMENTO EL
'METODO

'CONSULTA EL NOMBRE Y APELLIDO DE LA PERSONA

```
End sub
```

Como se puede observar, en la sección de Definición de Atributos se anteponen al nombre de la variable un identificador correspondiente al tipo de datos de la misma. Para tal fin la convención fue la siguiente:

Tabla 27. Identificadores de tipos de datos.

IDENTIFICADOR	TIPO DE DATO
byt	BYTE
i	INTEGER
f	DOUBLE
bln	BOOLEAN
s	STRING
l	COLLECTION
cl	OBJETO DE CLASE

En la definición de propiedades se definen las propiedades SET (instanciar) y GET (recuperar) a cada uno de los atributos de la clase. La propiedad SET va acompañada de un parámetro que se transfiere a la función al momento de instanciar la variable, el cual debe corresponder al tipo de datos de la variable.

Ejemplo (página ASP):

Una vez creado un objeto de la clase persona y conexión con las instrucciones

```
Set persona = Server.CreateObject("nombre_dll.nombre_clase")  
Set mobjconn = Server.CreateObject("nombre_dll.nombre_clase")
```

Se instancian los parámetros que el método CONSULTAR_PERSONA necesita, en este caso el documento y el tipo de documento

```
persona.setDocumento = "documento_de_la_persona"  
persona.setTipo_documento = "tipo_del_documento_de_la_persona"
```

se llama el método CONSULTAR_PERSONA pasando como parámetro la conexión a la base de datos

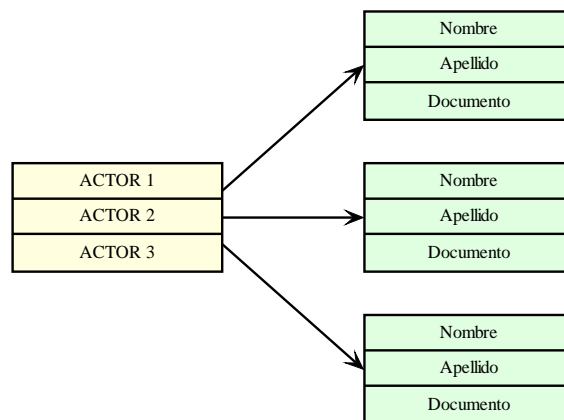
```
persona.Consultar_persona(mobjconn)
```

una vez ejecutado el método esta instancia las variables nombre y apellido para que luego se recuperen por medio de la propiedad GET.

```
Nombre = persona.getNombre  
Apellido = persona.getApellido
```

Cuando un método instancie múltiples niveles de resultados, como por ejemplo una lista de usuarios y sus respectivos datos, como se muestra en la figura 107, el procedimiento a seguir para el almacenamiento y recuperación de dicha información se realiza mediante la utilización de listas de listas. Una lista de listas es un vector de elementos en donde cada posición del vector contiene a otro vector.

Figura 107. Lista de listas.



Ejemplo (dll):

En la clase que devolverá la lista de listas se crea la colección, mediante la instrucción

```
Private Nombre_lista_ptal As New Collection
```

En el método se instancia dicha lista con sublistas como se muestra a continuación, en donde la lista principal es instanciada por una sublista creada en uno de los métodos de la clase:

```

Public sub Instanciar_lista()
Dim nombre_sublista()
Redim nombre_sublista(3)
    For i = 0 to 3
        Redim preserve Nombre_lista_ptal(i)
        Nombre_sublista(i) = "Nombre"
        Nombre_sublista(i+1) = "Apellido"
        Nombre_sublista(i+2) = "Documento"
        Nombre_lista_ptal (i) = nombre_sublista()
    Next
end sub

```

Para instanciar y recuperar valores de la lista es necesaria la creación de las propiedades SET y GET para la lista principal o global.

```

Public Property Let setNombre_lista_ptal (s As Variant)
    Nombre_lista_pptal = s
End Property
Public Property Get getNombre_lista_ptal () As Variant
    get Nombre_lista_pptal = Nombre_lista_pptal
End Property

```

6.4.5 Definición de las clases principales

Cls_conexion: Clase encargada de gestionar la interacción entre el sistema web y la base de datos. En ella encontramos los métodos necesarios para conectar a la base de datos y así consultar, actualizar, agregar o eliminar registros.

Cls_actor: Clase encargada de gestionar los usuarios del sistema.

Cls_consultas: Clase encargada de realizar las consultas globales a la base de datos.

Cls_requisitos: Clase encargada de la verificación de requisitos para cada una de las solicitudes.

Cls_mensaje: Clase encargada de gestionar los mensajes presentados al usuario por el sistema en casos específicos.

6.5 ESTRUCTURA DEL INBOX

El INBOX corresponde a la interfaz principal presentada al usuario al momento de ingresar al sistema de Recursos Humanos.

Figura 108. INBOX del sistema.

Item	Solicitud Número	Fecha Solicitud	Tipo Solicitud	Funcionario
1	3073	Dic 24 de 2003	ACUMULACION DE VACACIONES	OLGA GAMBOA PORRAS
2	3091	Ene 08 de 2004	DISFRUTE DE VACACIONES	ELBERTO CARRILLO RINCON
3	3102	Ene 15 de 2004	PRORROGA DE COMISION DE ESTUDIOS	JOSE DE JESUS LEON PEREIRA

Esta interfaz se encuentra dividida en 4 secciones:

- Sección de encabezado: En donde se presenta el baner del Sistema de Información de Recursos Humanos de la Universidad Industrial de Santander.

Figura 109. Baner del Sistema de Recursos Humanos.



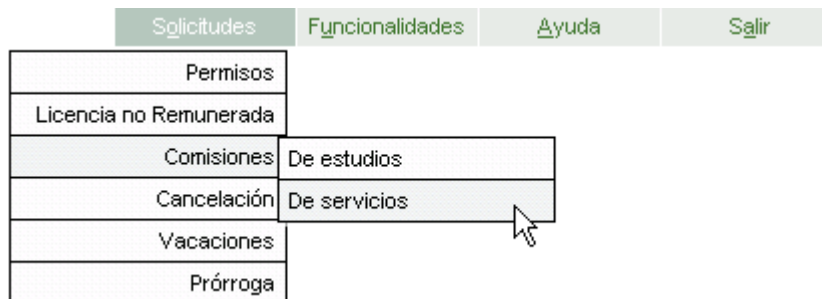
- Sección de cambio de rol: Esta sección brinda al usuario la posibilidad de cambiar de rol desde el INBOX; para ello presenta una lista de los roles vigentes, entre los cuales se puede escoger el rol requerido. El rol por defecto cuando se ingresa al sistema es el rol de empleado.

Figura 110. Combo para cambio de rol.



- Sección de menú dinámico: Presenta la vía de acceso a todas las páginas de solicitudes y funcionalidades a las que el usuario tiene derecho. Depende directamente de la definición del rol bajo el cual el usuario está trabajando.

Figura 111. Menú dinámico del sistema.



A continuación se describen las secciones del menú dinámico

- *Solicitudes:* En este menú se encuentra el vínculo a: Permisos, Licencia no Remunerada, Comisiones, Cancelación, Vacaciones y Prórrogas.
- *Funcionalidades:* En este menú se encuentra el vínculo a: Cruce de Tiempos, Registro Permiso Grupal, Cancelar Permiso Grupal, Registro Permiso Causado y Enviar Informes.
- *Ayuda:* Es el vínculo a la ayuda correspondiente al INBOX del sistema.

- *Salir*: Vinculo por medio del cual el usuario finaliza su sesión de trabajo en el sistema.
- Sección de cuerpo principal: Presenta las carpetas en las cuales se agrupan las diferentes solicitudes correspondientes a un usuario. Como son:
 - Pendientes: Presenta el listado de solicitudes que el usuario tiene actualmente pendientes para realizar visado o aprobación sobre las mismas.
 - Personales: Presenta el listado de solicitudes que han sido tramitadas por el usuario.
 - Mensajes: Presenta un listado de mensajes informativos sobre las acciones que se han realizado sobre una solicitud tramitada por el usuario. Los mensajes pueden ser de confirmación de aprobación, de confirmación de cancelación total o parcial, de rechazo o de modificación.
 - Reintegrar: Esta carpeta solo es visible para los actores jefes de unidad, los cuales son los encargados de registrar el reintegro de sus funcionarios a cargo.

Cada listado presenta el número de solicitud a manera de enlace (link) por medio del cual se accede al detalle de la solicitud.

Figura 112. Cuerpo principal de INBOX.

Pendientes Personales Reintegrar Mensajes				
Item	Solicitud Número	Fecha Solicitud	Tipo Solicitud	Funcionario
1	3073	Dic 24 de 2003	ACUMULACION DE VACACIONES	OLGA GAMBOA PORRAS
2	3091	Ene 08 de 2004	DISFRUTE DE VACACIONES	ELBERTO CARRILLO RINCON
3	3102	Ene 15 de 2004	PRORROGA DE COMISION DE ESTUDIOS	JOSE DE JESUS LEON PEREIRA

7. CONCLUSIONES

- El Sistema de Información de Recursos Humanos representa para la Universidad Industrial de Santander un importante aporte en lo referente a la adopción y buena utilización de nuevas tecnologías en el manejo de la información con el fin de brindar servicios que apoyen la gestión de situaciones administrativas del personal de planta.
- El Desarrollo de un módulo con tecnologías Web para el trámite de las situaciones administrativas propias de la gestión del Recurso Humano de la Universidad, presenta una alternativa ágil que brinda una mayor cobertura y autonomía a sus usuarios en lo referente al trámite de solicitudes.
- El obtener y manipular la información de forma sencilla, hace del sistema una herramienta robusta para sus usuarios, ofreciendo un conjunto de funcionalidades que apoyan labores administrativas de la Universidad.
- El Proceso Unificado como metodología de desarrollo proporciona una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.
- La utilización de la arquitectura de tres capas permite el desarrollo de aplicaciones modulares, en las cuales se distinguen adecuadamente los elementos que las constituyen, como son: La presentación de información, la lógica del negocio y los datos. Así mismo brinda la posibilidad de realizar operaciones posteriores sobre la aplicación como la adición de nuevas funcionalidades o mantenimiento de la existente.
- Debido a que la construcción de los subsistemas se realizó con base a estándares y componentes software reutilizables, el sistema es de fácil

mantenimiento, depuración y análisis, lo cual le permite ser actualizable según los requerimientos y necesidades futuras, obteniendo de esta manera una de las características más importantes y poderosas en los sistemas actuales que es la escalabilidad.

- Como modelo de Sistema de Información Web para el trámite de solicitudes del Área de Recursos Humanos de la Universidad Industrial de Santander, este proyecto no pretende ser el último esfuerzo en esta área de desarrollo institucional, ya que el sistema tendrá que ser sometido a modificaciones y actualizaciones a medida que los procesos correspondientes vayan cambiando o se vayan agregando nuevas funcionalidades, lo que conllevará a posteriores desarrollos.
- El Sistema de Información se realizó bajo la modalidad de práctica empresarial, aprovechando la infraestructura tecnológica de la División de Servicios de Información, la cual proporcionó a los desarrolladores las garantías y condiciones para un desarrollo confiable, y brindó la oportunidad de enriquecer sus conocimientos en el desarrollo de aplicaciones Web y la utilización de las tecnologías ASP.
- La experiencia de compartir el ambiente laboral en la División de Servicios de Información es un valioso aporte en el desarrollo profesional de los autores de este proyecto para futuros desempeños.

8. RECOMENDACIONES

Se recomienda la creación de un módulo de interfaz Web que permita en el mismo ambiente en que se tramita la solicitud, realizar el seguimiento a situaciones como:

- El vencimiento de reintegros.
- El vencimiento de reporte de informes.
- Solicitudes pendientes de aprobación que ya hayan entrado en vigencia.

Se recomienda mantener actualizada la información del personal de planta de la Universidad, prestando especial atención a situaciones como: cambios de cargo, ingresos y retiros del personal, horarios y categorías, entre otras. Lo cual permitirá mantener actualizado en cada puesto de trabajo los usuarios a cargo, de tal manera que sean atendidas oportunamente las solicitudes del personal, por los funcionarios que corresponda. Además de crear una funcionalidad en el sistema que apoye el monitoreo de estas situaciones.

Se recomienda en la página Inbox, incluir una sección dónde se observe de manera resumida todos los pendientes que se tienen en cada una de las carpetas que esta página maneja, dependiendo del actor que haya ingresado; esto con el fin de gestionar de manera oportuna y prioritaria la información requerida.

Se recomienda la creación de una funcionalidad que permita el registro del tiempo adicional trabajado, para ser usado por los Jefes de Unidad con el fin de llevar un control adecuado de los tiempos adicionales trabajados por los funcionarios de la entidad.

Se recomienda la creación de un módulo de sugerencias, con el fin de que los usuarios tengan un canal de comunicación directo con el administrador del sistema. Esto permitirá identificar más eficientemente las necesidades y nuevos requerimientos para el sistema.

Se recomienda la inclusión de nuevas solicitudes, para fortalecer el módulo de solicitudes de cancelación, las cuales no fueron consideradas en este desarrollo de acuerdo a las prioridades establecidas en el mismo. Entre otras se recomienda la inclusión de:

- Solicitudes de cancelación de prórrogas de Licencias no Remuneradas, Comisión de Servicio y Comisión de Estudio.
- Solicitudes de cancelación para los procedimientos de vacaciones: Acumulación y Disfrute.

Se recomienda la ampliación del módulo de consultas del sistema. La inclusión de una sección donde se permita realizar consultas a las solicitudes utilizando diferentes criterios como los tipos y los períodos solicitados, entre otros. Esta sección será de gran ayuda para los Jefes de las unidades académico administrativas, porque permitirá realizar un seguimiento de las solicitudes generadas desde y hacia su unidad.

Se recomienda el establecimiento de un plan de trabajo para la implantación del sistema desarrollado, diseñando estrategias que permitan paulatinamente ir eliminando el registro de solicitudes aprobadas, que actualmente se realiza en la División de Recursos Humanos. Lo anterior con el fin de no entorpecer la labor de control que realiza esta División y para lograr la utilización del sistema desarrollado adecuadamente.

Se recomienda dentro del plan de trabajo para la implantación del sistema, realizar una serie de capacitaciones dirigida a la comunidad universitaria potencial usuaria del sistema, como son: el Rector y Vicerrectores, los Jefes de unidades académico administrativas y los funcionarios de planta.

Se recomienda para la implantación, que el sistema quede soportado en un equipo con características de servidor con el fin de ofrecer todas las garantías de seguridad y eficiencia requeridas.

BIBLIOGRAFÍA

CARCAMO SEPULVEDA, José. Bases de Datos Relacionales: Un enfoque práctico de diseño. Universidad Industrial de Santander. 1994.

CEBALLOS, Francisco Javier. Visual Basic 6 Curso de programación. ALFAOMEGA Grupo Editor, S.A 1999.

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

DECRETO NÚMERO 55, 10 de Enero de 1994. DECRETO NUMERO 1444, 3 de Septiembre 1992. Universidad Industrial de Santander, Publicaciones UIS.

DE MIGUEL, Adoración. PIATTINI, Mario. MARCOS, Esperanza. Diseño de Bases de Datos Relacionales. Editorial AlfaOmega Rama. Año 2000.

FOWLER, Martin, KENDALL, Scott. UML GOTA A GOTA. Addison Wesley Longman de México S.A 1999.

GÓMEZ, Flórez Luis Carlos. Planeación de Proyectos. Universidad Industrial de Santander. Bucaramanga, 2001.

GROFF, James R. - WEINBERG, Paul N. APLIQUE SQL. OSBORNE- McGRAW-HILL, 1991.

JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El Lenguaje Unificado de Modelado. Primera edición. Addison Wesley. España, 2000.

_____. El Proceso Unificado de Desarrollo de Software. Addison Wesley. España 2000.

McCONNEL, Steve. Desarrollo y gestión de proyectos informáticos. Editorial McGraw Hill.

PRESSMAN, Roger. Ingeniería del software. Un enfoque práctico. Cuarta edición. McGraw Hill. España, 1998.

REGLAMENTO DEL PROFESOR, Acuerdo No.063 del 5 de Octubre de 1994. Aprobado por el Consejo Superior, Bucaramanga 1994. Universidad Industrial de Santander, Publicaciones UIS.

ULLMAN, Chris, BUSER, David, DUCKETT, Jon, FRANCIS, Brian, KAUFFMAN, John, LLIBRE, Juan T, SUSSMAN, David. Beginning ASP 3.0. Development. Wrox Press Ltd. 1999.

BIBLIOGRAFIA EN INTERNET

<http://programacion.com> Sitio dedicado a dar soporte para aplicaciones Web, contiene documentación acerca de ASP, JSP y Javascript

<http://webexperto.com> Sitio dedicado a dar soporte para aplicaciones Web, contiene documentación acerca de ASP, JSP y Javascript

<http://javascript.com> Contiene documentos y ejemplos de interés sobre este lenguaje.

<http://msdn.microsoft.com/library> Sitio dedicado a dar soporte para aplicaciones desarrolladas sobre plataforma Windows™.

<http://www.asptutor.com> Sitio dedicado a programadores de páginas ASP. Contiene manuales, tutoriales y códigos de ejemplo.