

**SISTEMA DE INFORMACIÓN WEB PARA LA PERSONERÍA
DISTRITAL DE CARTAGENA DE INDIAS - SIPEDCAR**

**CARLOS MANUEL CARABALLO GARCIA
MARIO RAFAEL POMARES CANTILLO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2004

**SISTEMA DE INFORMACIÓN WEB PARA LA PERSONERÍA
DISTRITAL DE CARTAGENA DE INDIAS - SIPEDCAR**

**CARLOS MANUEL CARABALLO GARCIA
MARIO RAFAEL POMARES CANTILLO**

**Proyecto de grado para optar al título de
Ingeniero de Sistemas**

Director

Ing. LUIS CARLOS GÓMEZ FLÓREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2004

RESUMEN

**Título : SISTEMA DE INFORMACIÓN WEB PARA LA PERSONERÍA DISTRITAL DE
CARTAGENA – SIPEDCAR-***

AUTORES:

CARLOS MANUEL CARABALLO GARCIA**

MARIO RAFAEL POMARES CANTILLO**

Palabras claves: *Proceso Unificado, Tecnología Web, Intranet, Software libre, Sistema de Información, Desarrollo Web, Proceso Unificado, Casos de Uso.*

Este proyecto es el resultado de la convicción y necesidad de adaptar e implementar las Tecnologías de la Información (TIC) en entidades gubernamentales como apoyo a sus procesos de gestión y acercamiento a la comunidad. Particularmente en la Personería Distrital de Cartagena para facilitar la interactividad y la extensión de servicios de esta hacia la comunidad; así como para crear un canal de comunicación permanente que informa oportunamente de las acciones y progresos realizados por los miembros de la entidad en la solución de denuncias planteadas por la comunidad (Web Site).

De manera similar al interior de la entidad (Intranet), lograr de manera ordenada y controlada la programación de dichas actividades en procura de dicho fin. Todo esto de la mano de las políticas estatales de masificación y uso de las TIC (Agenda de Conectividad – Estrategia de Gobierno en línea).

En el proyecto se profundiza el uso de estas tecnologías, al aportar soluciones innovadoras a problemas que surgen en sistemas desarrollados para ambientes tan complejos como internet. Como lo es el uso de sesiones de usuario y Plantillas (“templates”), para evitar accesos indebidos o no autorizados a sectores restringidos del sistema y para lograr la separación de capas (la capa de aplicación y la de presentación), respectivamente.

Este sistema se implementó bajo la plataforma Linux y fue desarrollado con software libre. Se logra de esta manera una solución económica sin perder robustez y la seguridad brindada por los sistemas UNIX, razón importante para incentivar el uso de estas herramientas que permiten desarrollar excelentes soluciones informáticas ajustadas a las necesidades presupuestales y particulares de nuestro país.

* Proyecto de Grado.

** Facultad de Ciencias Fisicomecánicas – Escuela de Ingeniería de Sistemas. Director de Proyecto:
M.S.C. Luis Carlos Gómez Flores.

El proceso unificado, metodología usada en este proyecto, proporciona normas para el desarrollo eficiente de software de calidad dentro de plazos y presupuestos planeados, al permitir mitigar riesgos críticos presentes en el desarrollo del sistema.

SUMMARY

**TITLE : WEB INFORMATION SYSTEM FOR PERSONERIA DISTRITAL DE
CARTAGENA – SIPEDCAR-***

AUTHORS:

CARLOS MANUEL CARABALLO GARCIA**

MARIO RAFAEL POMARES CANTILLO**

Key words: Unified process, Technology Web, Intranet, free Software, Information system, Development Web, Unified Process, Cases of Use.

This project is the result of the conviction and necessity to adapt and to implement the Technologies of the Information (TIC) in governmental organizations like support to its processes of management and approach to the community. Particularly in the Personería Distrital de Cartagena to facilitate the interactivity and the extension of services of this towards the community; as well as to create a permanent communication channel that opportunely informs into the actions and progresses made by the members of the organization in the solution of denunciations raised by the community (Web site).

Of way similar to inside of the organization (Intranet), to obtain of way ordered and controlled the programming of these activities in it tries of this aim. All this of the hand of the state policies of masification and use of the TIC (Connectivity Agenda - Government on line Strategy).

In the project the use of these technologies is deepened, when contributing innovating solutions to problems that arise in systems developed for as complex atmospheres as Internet. As it is it the use of user sessions and Groups ("templates"), to avoid illegal accesses or nonauthorized to restricted sectors of the system and to obtain the separation of layers (the application layer and the one of presentation), respectively.

This system was implemented under the Linux platform and was developed with free software. An economic solution without losing robustness is obtained this way and the security offered by the UNIX systems, reason important to stimulate the use of these tools that allow developing excellent computer science solutions fit to the budgetary and particular necessities of our country.

* Degree Project.

** Fisicomecánicas Sciences Faculty - School of Engineering of Systems. Project Director: M.S.C. Luís Carlos Gomez Flores.

The unified process, methodology used in this project, provides planned norms for the efficient development of software of quality within terms and budgets, when allowing mitigating present critical irrigations in the development of the system.

TABLA DE CONTENIDO

INTRODUCCIÓN	16
PARTE I	20
CAPITULO 1	21
ASPECTOS GENERALES.....	21
1.1. JUSTIFICACIÓN	21
1.1.1 Antecedentes y descripción del problema	21
1.1.2. Personería Distrital De Cartagena De Indias (P. D. C)	26
1.2. OBJETIVOS	28
1.2.1. Objetivo General.....	28
1.2.2. Objetivos Específicos.....	28
1.3. DESCRIPCIÓN DE OBJETIVOS.....	30
1.4. ALCANCES DEL PROYECTO	34
1.4.1. Social	34
1.4.2. Tecnológico.....	34
1.4.3. Económico	35
CAPITULO 2	37
MARCO TEORICO, METODOLOGICO Y REFERENCIAL	37
2.1. INFORMÁTICA JURÍDICA.....	37
2.2. INTERNET	39
Soluciones basadas en tecnología Internet.....	39
2.2.1. Intranet	39
2.2.2. Extranet.....	40
2.3. SISTEMAS DE INFORMACIÓN	40
2.4. INGENIERÍA DEL SOFTWARE (IS).....	41
2.5. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE	61
2.6. LENGUAJE UNIFICADO DE MODELADO (UNIFIED MODELING LANGUAGE, UML).	64
2.7. INGENIERÍA WEB (IWEB).....	64
2.8. ARQUITECTURA DE SOFTWARE	67
2.8.1. Cliente – servidor	69
2.8.2. Múltiples capas	70
2.8.2.1. Capa de Datos	72
2.8.2.2. Capa de Aplicación o de Negocios.....	72
2.8.2.3. Capa de Presentación	72
2.9. APLICACIONES WEB	73
2.9.1. Calidad de una aplicación web.....	74
2.9.2. Tecnologías disponibles para el desarrollo	74
2.9.2.1 Programación en el cliente:	75
2.9.2.2. Programación en el servidor:.....	75
2.9.2.3. Esquema mixto: (programación en el cliente y en el servidor).....	76
2.9.3. Programación del lado del cliente.....	77
2.9.4. Programación del lado del Servidor.....	78
2.9.4.1. El protocolo CGI:.....	78
2.9.4.2 Uso de un Módulo del Servidor Web	79
2.10. PHP (Hypertext Preprocessor)	81
2.11 Linux.....	83
2.12. Servidor Web Apache	87
2.13. Postgresql	88
PARTE II	90
CAPITULO 3	92

PLANEACION Y ESTABLECIMIENTOS DE CRITERIOS DE EVALUACIÓN EN LA FASE DE INICIO	92
3.1. CRITERIOS DE EVALUACIÓN	92
3.2. PRODUCTOS A ENTREGAR.....	93
CAPITULO 4	94
VIABILIDAD DEL PROYECTO	94
4.2. REQUISITOS	94
4.2.1. Lista de características	95
4.2.2. Contexto del sistema	99
4.2.3. Misión y visión de la Personería Distrital de Cartagena	99
4.2.4. Modelo De Negocio	101
4.2.4.1. Modelo De Casos De Uso De Negocio	102
4.2.4.2. Modelo de Objetos del negocio.	103
4.2.5. Riesgos Críticos	104
4.2.6. Modelo De Casos De Uso	107
4.2.6.1. Actores	107
4.2.6.2. Casos de uso	108
4.2.6.3. Casos De Uso En Detalle	109
CONSULTAR BOLETÍN.....	113
4.3. ANÁLISIS	116
4.3.1. Análisis De La Arquitectura.....	116
4.3.2. Análisis De Caso De Uso.....	119
4.4. DISEÑO	120
4.4.1. Diseño De La Arquitectura.....	120
4.5. RESUMEN	122
4.5.1. Vista de la Arquitectura	123
4.5.2. Cumplimiento De Criterios	123
PARTE III	125
CAPITULO 5	126
PLANEACION Y ESTABLECIMIENTOS DE CRITERIOS DE EVALUACIÓN EN LA FASE DE ELABORACION	126
5.1. CRITERIOS DE EVALUACIÓN	126
5.2. PRODUCTOS A ENTREGAR.....	127
CAPITULO 6	128
LINEA BASE DE LA ARQUITECTURA	128
6.1. Introducción.....	128
6.2. REQUISITOS	128
6.2.1. Casos de uso	128
6.2.1.1. Paquete de casos de uso – Gestión de quejas	129
6.2.1.2. Paquete de casos de uso - Sitio Web.....	130
6.2.1.3. Paquete de casos de uso – Agenda de actividades.....	131
6.2.1.4. Paquete de casos de uso – Mecanismos de Intercambio de información	132
6.2.1.5. Paquete de casos de uso – Administración del sistema	133
6.2.2. Casos de uso en detalle.	134
Consultar queja por Internet	134
Participar en espacios de opinión	135
Consultar queja	135
Diligenciar queja.....	135
Agregar nueva actividad a la agenda	136
Participar en espacios de discusión	136
6.3. ANÁLISIS	137
6.3.1. Análisis De La Arquitectura.....	137
6.3.2. Análisis De Casos De Uso.....	140

6.4. DISEÑO	143
6.4.1. Diseño de la Arquitectura.....	143
6.4.1.1. Identificación de nodos y configuraciones de red.....	143
6.4.1.2. Modelo de arquitectura de tres capas	144
6.4.1.3. Diseño a partir de plantillas (Templates).....	145
6.4.2. Diseño De La Interfaz	146
6.4.3. Diseño De La Base De Datos.....	151
6.4.4. Diagrama Entidad Relación De La Base De Datos Del Sitio Web	153
6.4.5. Diagrama Entidad Relación De La Base De Datos Del Sistema Interno	154
6.5. IMPLEMENTACION.....	155
6.5.1. Software Del Sistema Y La Capa Intermedia	156
6.5.2. Manejo De Sesiones.....	157
6.6. RIESGOS CRITICOS.....	159
6.7. RESUMEN	160
6.7.1. Cumplimiento de criterios	160
PARTE IV.....	161
CAPITULO 7	162
PLANEACION Y ESTABLECIMIENTOS DE CRITERIOS DE EVALUACIÓN EN LA FASE DE CONSTRUCCION.....	162
ACTIVIDADES Y PRODUCTOS A ENTREGAR	162
CAPITULO 8	163
ESTANDARES DE IMPLEMENTACION	163
8.1 Introducción.....	163
8.2 CONVENCIONES DE CODIFICACIÓN	163
8.3 CONVENCIONES DE INTERFAZ	166
8.4 MENSAJES DEL SISTEMA.....	166
8.5 IMPLEMENTACION.....	167
8.5.1 Subsistemas De Gestión De Sesiones E Interfaz	169
8.5.1.1 Subsistema De Gestión De Sesiones.....	169
8.5.1.1 PRUEBAS	172
8.5.2 SUBSISTEMA DE GESTION DE INTERFAZ	173
8.5.2.1 PRUEBAS	176
CAPITULO 9	177
SUBSISTEMAS DE APLICACION.....	177
9.1. SITIO WEB.....	177
9.2. SUBSISTEMA DE EVENTOS.....	180
9.2.1 Pruebas.....	180
9.3. SUBSISTEMA DE QUEJA.....	181
9.3.1 Pruebas	182
9.4. SUBSISTEMA DE COMUNICACIÓN	183
9.4.1 Pruebas	183
9.5. SUBSISTEMA ADMINISTRACIÓN.....	184
9.7. RESUMEN	187
9.7.1 Elaboración de Manuales de Usuario.....	187
CONCLUSIONES	189
RECOMENDACIONES	191
BIBLIOGRAFÍA	193

LISTA DE TABLAS

Tabla 1. Guía para la selección del ciclo de vida a utilizar en un proyecto.....	43
Tabla 2. Resumen de los principales Modelos de Ciclo de Vida de Desarrollo del Software.....	49
Tabla 3. Requisitos de calidad.....	74
Tabla 4. Lenguajes del lado del Cliente.....	77
Tabla 5. Lenguajes del lado del servidor.....	80
Tabla 6. Comparación de lenguajes del lado del servidor.....	80
Tabla 7. Lista de características.....	98
Tabla 8. Personerías Delegadas y función.....	100
Tabla 9. Casos de uso del negocio.....	103
Tabla 10. Contenido de la lista de riesgos críticos.....	105
Tabla 11. Riesgos críticos.....	106
Tabla 12. Actores del sistema.....	107
Tabla 13. Descripción de casos de usos del modelo general del sistema.....	109
Tabla 14. Contenido de detalle de casos de uso.....	109
Tabla 15. Detalle de caso uso enviar queja por Internet.....	110
Tabla 16. Detalle de caso uso Revisar queja.....	110
Tabla 16. Detalle de caso uso Revisar queja.....	111
Tabla 17. Detalle de caso de uso Consultar agenda.....	112
Tabla 18. Detalle de caso de uso Consular Boletín.....	113
Tabla 19. Detalle de caso de uso Crear Usuario.....	114
Tabla 20. Detalle de caso de uso Incluir evento en cartelera.....	115
Tabla 21. Clases de análisis para el caso de uso Revisar queja.....	119
Tabla 22. Descripción de los casos de uso del paquete sitio web.....	129
Tabla 23. Descripción de los casos de uso del paquete Gestión de quejas.....	130
Tabla 24. Descripción de los casos de uso del paquete Agenda de actividades.....	131
Tabla 25. Descripción de los casos de uso del paquete Intercambio de información.....	132
Tabla 26. Descripción de los casos de uso del paquete Administración del sistema.....	133
Tabla 27. Detalle del caso uso consultar queja por Internet.....	134
Tabla 28. Detalle del caso uso Participar en espacios de opinión.....	135
Tabla 29. Detalle del caso uso Consultar Queja.....	135
Tabla 30. Detalle de caso uso Diligenciar Queja.....	135
Tabla 31. Detalle de caso uso Agregar nueva actividad a la agenda.....	136
Tabla 32. Detalle de caso uso Participar en espacios de discusión.....	136
Tabla 33. Contenido del menú de navegación de la página inicial del sitio web.....	147
Tabla 34. Servicios del sistema de información web según el perfil de usuario.....	150
Tabla 35. Descripción de las tablas en el Diseño de la Base de Datos.....	151
Tabla 36. Descripción de las tablas de la base de datos del sistema interno.....	155
Tabla 37. Software del sistema.....	157
Tabla 38. Principales métodos de la clase FastTemplate.....	157
Tabla 39. Comentario de cabecera de función.....	165
Tabla 40. Código para el inicio de sesión y registro de las variables.....	171
Tabla 41. Prueba uno subsistema de sesión.....	172
Tabla 42. Prueba con datos inválidos/ subsistema de sesión.....	172
Tabla 43. Prueba con acceso ilegal / subsistema de sesión.....	172
Tabla 44. Prueba con acceso y datos válidos / subsistema de sesión.....	172
Tabla 45. Código de la página noticias.php del sitio web.....	174
Tabla 46. Código de tabla_noticias.tpl.....	175
Tabla 47. Código de filas_tabla_noticias.tpl.....	175
Tabla 48. Zonas de la página inicial del sitio web.....	178
Tabla 49. Opciones del menú de navegación del sitio web.....	178

Tabla 50. Acceso de los usuarios a las opciones de eventos, según su perfil.	180
Tabla 51. Acceso de los usuarios a las opciones de quejas según su perfil.	182
Tabla 52. Acceso de los usuarios a las opciones de comunicación según su perfil.....	183
Tabla 53. Opciones de administración del sitio web	185
Tabla 54. Opciones de administración	186

LISTA DE FIGURAS

Figura 1. Estructura del estado Colombiano	27
Figura 2. Alcance de Intranet, Extranet e Internet	39
Figura 3. Flujos de trabajo en las cuatro fases del proceso unificado	63
Figura 4. El modelo de proceso web.	66
Figura 5 Arquitectura Cliente - Servidor.	69
Figura 6 Arquitectura de Tres capas. Caso mas general de arquitectura multicapas.	71
Figura 7. Programación en el lado del cliente.	75
Figura 8. Programación en el lado del Servidor.	76
Figura 9. Esquema mixto. Programación en el cliente y en el servidor.	76
Figura 10. Modelo de Casos de Uso de Negocios	102
Figura 11. Modelo de Objetos.....	103
Figura 12. Modelo de casos de usos general del sistema	108
Figura 13. Diagrama de estados del caso de uso - Enviar queja por Internet	110
Figura 14. Diagrama de estados del caso de uso revisar queja	111
Figura 15. Diagrama de estados del caso de uso consultar agenda.	112
Figura 16. Diagrama de estados del caso de uso consultar Boletín	113
Figura 17. Diagrama de estados del caso de uso Crear usuario	114
Figura 18. Identificación de paquetes de análisis a partir de los casos de uso.	117
Figura 19. Paquetes de análisis a partir del modelo de objeto del negocio	117
Figura 20. Dependencia de paquetes de análisis	118
Figura 21. Diagrama de colaboración de la realización del caso de uso Revisar Queja	120
Figura 22. Diagrama de despliegue del sistema.	122
Figura 23. Diagrama de despliegue del sistema interno.	122
Figura 24. Casos de uso del paquete Gestión de quejas.....	129
Figura 25. Casos de uso del paquete Sitio Web	130
Figura 26 - Casos de uso del paquete Agenda de actividades	131
Figura 27. Casos de uso del paquete Mecanismos de intercambio de información	132
Figura 28. Casos de uso del paquete Administración del sistema.....	133
Figura 29. Paquete de análisis Sitio Web	137
Figura 30. Paquete de análisis Agenda de actividades.....	137
Figura 31. Paquete de análisis Gestión de quejas	138
Figura 32. Paquete de análisis Mecanismos de intercambio de información	139
Figura 33. Paquete de análisis Administración del sistema	139
Figura 34. Diagrama de colaboración del caso de uso Consultar queja por Internet.	140
Figura 35. Diagrama de colaboración del caso de uso Participar en espacio de opinión (foro)	141
Figura 36. Diagrama de colaboración de la realización del caso de uso Consultar Boletín	142
Figura 37. Diagrama de colaboración de la realización del caso de uso Crear cuenta de Usuario.	142
Figura 38. Estructura de malla completa o Web pura.	146
Figura 39. Interfaz de la página inicial del sitio web	147
Figura 40. Interfaz de las páginas interiores del sitio web.....	149
Figura 41. Diseño de la página de inicio del sistema web de información interno.....	149
Figura 42. Interfaz de las páginas interiores del sistema interno	150
Figura 43. Diagrama E/R de la base de datos del sitio web.....	153
Figura 45 Diagrama de restricción de acceso usando sesiones.....	159
Figura 46. Mensaje En una ventana propia del navegador.....	167
Figura 47. Mensaje En la misma pagina que generó el error.....	167
Figura48. Mensaje En una pagina nueva	167

Figura 49. Paquetes de las Fases de inicio y Elaboración y subsistemas de implementación	168
Figura 50. Interfaz de noticias.php creada a partir de plantillas	176
Figura 51. Interfaz de la página inicial del sitio web	177
Figura 52. Interfaz de las páginas interiores del sitio web	179
Figura 53. Diagrama de estados interposición de queja por Internet.....	179
Figura 54. Interfaz de la página del subsistema eventos	180
Figura 55. Interfaz de la página del subsistema queja	181
Figura 56. Interfaz de la página del subsistema queja	183
Figura 57. Interfaz de la página de ingreso a administración.....	184
Figura 58. Interfaz de la página de ingreso a administración.....	184
Figura 59. Interfaz de la página del perfil administrador	187

INTRODUCCIÓN

Las entidades del estado son organizaciones que manejan un gran volumen de información la cual en su mayoría se manipula físicamente , debido en parte a la complejidad para el almacenamiento, gestión y control a través de medios electrónicos.

Gran parte de ésta información es de tipo jurídica y por lo tanto se caracteriza por su extensibilidad y diversa interpretación, lo cual hace difícil su posible cuantificación. Por lo tanto desde hace algún tiempo se ha tratado de crear una nueva forma de estudiar esta información, combinando lo jurídico y lo tecnológico, utilizando para ello los conceptos y técnicas propios de la informática, dando lugar a la llamada *informática jurídica* ¹, un campo de aplicación de la informática que se ha visto limitado muchas veces por las barreras tecnológicas impuestas en ese momento como lo demuestran muchos de los casos estudiados en el contexto colombiano, los cuales no se pudieron implantar o no dieron los resultados que se esperaban.

Actualmente gracias al avance vertiginoso de la tecnología y en especial a la masificación de Internet y todo lo que tiene que ver con las redes de comunicación, este campo de acción cobra una gran importancia y se convierte en un espacio propicio para el desarrollo de sistemas que faciliten la gestión de las entidades que se relacionan con la creación, flujo, clasificación, organización, sistematización y utilización de datos requeridos en la producción y/o en la administración de lo jurídico.

En nuestro país, este hecho ha sido impulsado gracias a una política establecida por el gobierno, llamada *Agenda de Conectividad* ² que busca masificar y mejorar el uso de las Tecnologías de la Información y Comunicación, como Internet, para acelerar el desarrollo económico, social y político de nuestro país y para lo cual se

¹ Para mayor información ver Anexo A.

² Agenda de Conectividad: El Salto a Internet - Documento CONPES No. 3072 / 9 de febrero de 2000.

ha definido la *Estrategia de gobierno en línea*³ la cual es de obligatorio cumplimiento para las entidades del estado.

Ante la necesidad de adoptar las instrucciones de la Estrategia de Gobierno en Línea, La Personería Distrital de Cartagena de Indias, organismo de control Distrital perteneciente al Ministerio Público, cuya función es la guarda y promoción de los derechos humanos, la protección del interés general y vigilancia de la conducta de quienes desempeñan funciones públicas(excepto el alcalde Distrital), ofrece la oportunidad mediante el desarrollo de este sistema de información web para facilitar su gestión y apoyarla en su función como punto de contacto entre la comunidad y el municipio, de hacer un valioso aporte en el campo de la informática jurídica de gestión y control, ya que ésta entidad al manejar en un alto porcentaje información relacionada con el derecho, se convierte en un espacio propicio para el desarrollo de este tipo de sistemas, convirtiendo a la comunidad en la principal beneficiaria al recibir una atención mucho más eficiente y de acuerdo a la importancia que se merece. Con este sistema la comunidad a través del sitio web tendrá la oportunidad de acceder a múltiples servicios y consultar diversa información relacionada con la labor de la entidad. El sistema interno agilizará el trámite de los procesos y hará más fácil llevar un seguimiento de éstos, evitando vencimientos y proporcionando a la comunidad información actualizada en el momento que lo requiera, además que permitirá compartir toda clase de conocimiento relativo a la entidad y facilitará el movimiento y control de la información que se manipula constantemente.

Este sistema demanda la aplicación de muchos conceptos propios de la Ingeniería de sistemas y ha sido guiado por una metodología de desarrollo denominada *Proceso Unificado del Desarrollo de Software*, el cual nos brinda una visión moderna de cómo se desarrolla un producto software; proporcionando un método común que nos da una guía para ordenar las actividades de un equipo de desarrollo, especificar los componentes que deben desarrollarse, los criterios para el control y la medición de los productos, y las actividades de un proyecto software, todo esto nos ofrece la

³ Directiva presidencial 02 de Agosto 28 del 2000 – Ver Apéndice A

seguridad de culminar un proyecto con calidad, dentro de los plazos y presupuesto planeados.

La información que se ofrece en el presente documento ha sido clasificada principalmente en cuatro partes esenciales:

En la **Parte I**, se establecen los fundamentos generales del proyecto. Esta compuesta por dos capítulos:

El Capítulo 1, presenta los objetivos y la definición de la situación problema, es decir las necesidades que dieron origen al proyecto, describiendo brevemente los antecedentes encontrados y los alcances(impacto y viabilidad) que éste proyecto ofrece.

El capítulo 2 , muestra la información teórica considerada como necesaria para una buena comprensión del proyecto, dentro de esta información se encuentra los conceptos generales sobre tecnologías web, PHP, Linux, Postgresql, Apache y la arquitectura de software que puede servir de base para un desarrollo de este tipo. Además se encuentra información sobre informática jurídica , proceso unificado de desarrollo de software y UML (Lenguaje de Modelado Unificado).

Las siguientes tres partes se centran en mostrar el desarrollo del proyecto a través de cada una de las fases expuestas por la metodología escogida. Estas son inicio, elaboración y construcción.

En la **parte II**, se presenta la fase de inicio, la cual busca desarrollar el análisis de negocio hasta el punto necesario para justificar la puesta en marcha del proyecto. La componen dos capítulos.

El *capítulo 3*, presenta la planeación de la fase, el establecimiento de los criterios de evaluación, las actividades y los productos a obtener.

El *capítulo 4*, describe los flujos fundamentales que se llevan a cabo en esta fase y al final del capítulo se presenta un resumen que le permite al lector tener una idea general de los productos obtenidos, comprobar si se cumplieron los criterios de evaluación y lo que seguirá en el capítulo próximo .

La fase de elaboración, presentada en la **parte III**, orientará al lector en la obtención de una arquitectura estable que guiará la construcción del sistema. En ésta parte se encuentran los *capítulos 5 y 6* que tendrán una estructura similar a la expuesta en los capítulos 3 y 4 de la parte II, pero particularmente para la fase de elaboración.

Por último la fase de construcción, la cual se encuentra en la **parte IV**, presenta las iteraciones necesarias para obtener el sistema propuesto. Esta conformada por los capítulos 7,8 y 9. El capítulo 7 nos muestra la planeación de esta fase, el capítulo 8 nos presenta los estándares que se utilizaron en la implementación del sistema(convenciones de codificación, de interfaz y mensajes) y el capítulo 9 muestra la implementación de los diferentes subsistemas que fueron diseñados.

Para finalizar, se presentan las conclusiones y recomendaciones del trabajo realizado. Así como los anexos y apéndices respectivos.

PARTE I

FUNDAMENTACION GENERAL

A lo largo de esta parte encontraremos información de carácter general agrupada en dos capítulos que contendrán las especificaciones del proyecto y fundamentos teóricos, referencial y metodológicos requeridos para la comprensión del mismo.

Esta información es muy importante para ubicar al lector en el ¿qué?, el ¿cómo? y ¿para que? se hizo este proyecto, para lograr esto se formulan y describen detalladamente los objetivos del proyecto, se presenta la justificación, se analiza el impacto del proyecto y las proyecciones esperadas. Más adelante se muestra un marco teórico necesario para el entendimiento de este trabajo. Igualmente se describe el marco metodológico que guió su desarrollo y un concepto general del marco referencial como lo es la informática jurídica.

Es importante revisar el Anexo B sobre Proceso Unificado y UML ya que durante el desarrollo de éste documento se utilizaran definiciones, conceptos, diagramas propios de la metodología

CAPITULO 1

ASPECTOS GENERALES

Los aspectos generales del proyecto que se tratan en este apartado comprenden: el análisis de la problemática que lo originó, los objetivos que guiaron la planeación, el diseño y el desarrollo de la solución propuesta; como también sus alcances en los aspectos social, tecnológico y económico.

El objetivo general describirá el proyecto en un pequeño párrafo, la cual incluye el producto que se va a obtener, como se va a obtener y para que va a servir. Luego se presentan las características del producto en cuanto a funcionalidad, utilidad, estructura, tecnología y metodología de desarrollo, al igual que una descripción detallada de éstas.

Se presentan los antecedentes y estado del arte de algunos sistemas similares que han sido desarrollados en Colombia para luego mostrar algunos aspectos sobre La Personería Distrital de Cartagena, planteando su necesidad y solución. Se indican además los aspectos más relevantes de la Agenda de conectividad, en especial de la Estrategia de gobierno en línea.

1.1. JUSTIFICACIÓN

1.1.1 Antecedentes y descripción del problema

El gobierno colombiano desde tiempo atrás ha elaborado políticas de desarrollo informático y sistematización de entidades del sector público, así por ejemplo:

1968: Comisión de normas.

1974: Secretaría de Organización e Información de la Administración Pública.

1976: Comisión Nacional de Sistemas.

1983: Centro Latinoamericano de Informática y Recursos Humanos.

1988: Modernización de la Comisión Nacional de Sistemas.

1989: Programa de Informática en el Sector Público (INSEP)⁴ dentro del cual se desarrolla JURISCOL⁵.

⁴ Representa una estrategia para hacer avanzar el país en aspectos técnicos y tecnológicos, en el campo de la informática.

Es innegable que la implementación e implantación de herramientas informáticas y últimamente el uso de las llamadas **Tecnologías de la Información (T.I)**, ofrecen una oportunidad única para que los países en vía de desarrollo den un salto en su evolución económica, política, social y cultural, disminuyendo la brecha que los separa de los países desarrollados. Estas tecnologías y en especial Internet han trascendido los campos tecnológico y científico, constituyéndose hoy en día en herramientas que se encuentran al alcance y servicio de toda la comunidad en los ámbitos económico, educativo y de salud entre otros, permitiendo el desarrollo de una nueva economía, la universalización del acceso a la información, y la adquisición y eficaz utilización del conocimiento, todos estos elementos fundamentales para el desarrollo de la sociedad moderna.

El gobierno colombiano consciente de esta oportunidad busca masificar el uso de las T.I. y con ello aumentar la competitividad del sector productivo, así como también modernizar las instituciones públicas y socializar el acceso a la información, para lograr esto se ha diseñado una política de estado presentada y aprobada por el Conpes ⁶ mediante el documento 3072 del 9 de febrero de 2000, denominada *La Agenda de Conectividad* ⁷.

La Agenda de Conectividad se fundamenta en un conjunto de estrategias desarrolladas a través de programas y proyectos articulados entre sí, con el propósito de lograr que Colombia aproveche el uso de las tecnologías de la información para su desarrollo económico, social y político. En la Agenda se contemplan acciones concretas para asegurarle al país que con la voluntad del sector público y privado, es viable expandir las tecnologías de la informática y las comunicaciones a todo el territorio nacional, de tal forma que la mayoría de los colombianos tengan acceso a la información y al conocimiento. Dentro de este contexto, el Estado, a través de la coordinación de la Agenda, actuará como el

⁵ Banco de datos documental que contiene toda la legislación del país, la jurisprudencia de la Corte Constitucional, de la Corte Suprema de Justicia y del Consejo de Estado.

⁶ CONPES: Consejo Nacional de Política Económica y Social. Dirección en Internet: <http://www.dnp.gov.co>.

⁷ <http://www.agenda.gov.co>

principal facilitador y promotor de la inversión nacional e internacional de empresas de telecomunicaciones en territorio colombiano.

Dentro de esta política, el gobierno debe ser un usuario modelo motivo por el cual se ha definido la estrategia de *Gobierno en Línea* que establece como uno de sus objetivos el proveer al Estado la conectividad que facilite la gestión en línea de los organismos gubernamentales y apoye su función de servicio al ciudadano, como un complemento al esquema actual, en el que muchos de estos procesos se realizan en forma presencial en las oficinas del gobierno y se sustentan con documentos. Esta es de obligatorio cumplimiento y con ella deben estar comprometidas todas las entidades públicas, según se instrumenta por medio de la directiva presidencial No. 02 de Agosto 28 del 2000 (Apéndice A).

Esta consta de tres fases, la primera es proveer información en línea a los ciudadanos, la segunda es ofrecer servicios y tramites en línea a los ciudadanos y la tercera es la contratación en línea.

Además contiene una serie de instrucciones generales, que son responsabilidad de las entidades darle cumplimiento. Como son:

- ✓ Identificar las necesidades de tipo normativo para atenderla satisfactoriamente y en caso de ser necesario proponer o tramitar las modificaciones normativas requeridas, ante las autoridades competentes.
- ✓ Incluir el cumplimiento de estas instrucciones, correspondientes a la estrategia de Gobierno en Línea de la Agenda de Conectividad, dentro del Plan Estratégico de cada entidad, con la mayor prioridad para su desarrollo eficiente y oportuno.
- ✓ Adecuar la infraestructura de redes y telecomunicaciones para garantizar el nivel de servicio exigido.

A la luz de esta nueva política de modernización del estado cobra gran importancia la aplicación de las tecnologías de la información en el ámbito jurídico, puesto que en muchas entidades públicas se maneja información de este tipo o que tenga que ver con el derecho; cabe anotar que desde los años 70 Colombia entró la aplicación del computador en el campo de lo jurídico, los juristas de una manera casi desprevénida comenzaron a utilizar y a mezclar las actividades de trabajo con

las nuevas tecnologías, familiarizándose con sus métodos y términos, conceptualizando las bases de la llamada *Informática Jurídica*⁸. En el país se han realizado varios proyectos considerados como esfuerzos aislados en lo que se conoce como informática jurídica registral, operacional y algo de documental, dejando de lado la decisional la cual tiene más acogida y desarrollo en el resto del mundo.

Varias experiencias pueden ser reseñadas en cuanto a la sistematización de la producción Jurídica nacional, algunos de ellos son⁹:

- **PROYECTO CIDOC (Centro de Información de la Cámara de Representantes)**

Se inició en 1972 y fue suspendido en 1973, luego de una evidente falta de voluntad política a pesar de la también evidente necesidad de los parlamentarios de tecnificarse a la par del Ejecutivo. El proyecto pretendía cubrir varias áreas: estadística, economía, ciencia política, ciencias sociales en general.

La falta de *hardware* y de *software* en esa época, hizo difícil la realización del trabajo. Además, los sistemas manuales del Congreso eran muy deficientes. Para tomar modelos, se tomaron experiencias de otros países: de Bélgica el CREDOC, de Estados Unidos y del Brasil, el PRODASEN. (Cf. Fernando Jordán F. Informática jurídica. Opus cit. p.p. 165 – 171). En 1974, el proyecto CIDOC murió, no obstante, en 1975 el Banco Popular retomó el proyecto bajo el nombre de Sistema de Información de la Legislación Colombiana (SILCO).

- **PROYECTO SILCO**

Duró cuatro años y se especializó en el proceso de análisis e incorporación de La información legal, en lo relativo a leyes y decretos con fuerza de ley. Por el sistema imperante en esa época de tarjetas perforadas, el proyecto fracasó dado que la información era recuperada a través de las palabras naturales en que está redactado el documento y esto es muy lento y costoso. Además,

⁸ Anexo A Informática Jurídica.

⁹ Tomados de Manual de Informática Jurídica, Orlando Solano B, Ediciones Jurídicas Gustavo Ibáñez,

producía silencio informático muy pronunciado por imposibilidad de hallar desinencias gramaticales distintas, sinónimos o contenidos conceptuales implícitos.

SILCO contaba con un computador en cuyos archivos se encontraban almacenados los textos de las normas legales, junto con cierta información complementaria como: tipo de norma, número, fecha, diario donde se publicó e historia.

Las siguientes entidades: Presidencia de la República, Corte Suprema, Consejo de Estado, Ministerio de Justicia, Banco de la República, entre otras, debían disponer de un terminal conectado con el computador central por línea telefónica. Pese a estas bondades, el proyecto se vino abajo.

- **PROYECTO DE SISTEMATIZACION DE OFICINAS JURÍDICAS (SISJURIS)**

Busca facilitar el trabajo de oficinas públicas y privadas con el fin de asumir el grado de complejidad creciente que las agobia por falta de tecnificación del trabajo. El sistema está compuesto por varios módulos de servicio que son: contabilidad, conceptos, reparto y control de procesos, control de contratos, elaboración de textos, legislación, doctrina y jurisprudencia.

- **PROYECTO DE SISTEMATIZACIÓN DE PROCESOS EN LOS CONSULTORIOS JURÍDICOS**

Es una aplicación del SISJURIS por parte del consultorio jurídico de la Universidad de los Andes, pero no se pudo implantar.

Se puede concluir que en nuestro medio aún se trabaja sobre la informática jurídica de gestión y documental, en busca de un mejoramiento significativo de las estructuras jurídico-administrativas que permitan a las entidades públicas cumplir con sus funciones determinadas por ley. Dentro de estas entidades públicas en donde la información que se maneja a diario es de tipo jurídico y la aplicación de herramientas informáticas contribuyen al cumplimiento de sus funciones, encontramos a La Personería Distrital de Cartagena de Indias.

1.1.2. Personería Distrital De Cartagena De Indias (P. D. C)

La Personería Distrital de Cartagena de Indias es una entidad pública de orden Distrital creada por la Constitución Nacional Art. 118,¹⁰ Ley 136 de 1994¹¹, como un organismo de control perteneciente al Ministerio Público¹² (Ver estructura del Estado Colombiano Figura 1). Cuya misión es “Velar por los Derechos de todos, mediante la guardia de los bienes del Distrito, la Vigilancia del cumplimiento de los Derechos de la Comunidad para que estos no sean vulnerados, la debida ejecución de la ley y la vigilancia de la conducta oficial de quienes ejercen funciones publicas”.

La personería como instrumento municipal, o en este caso Distrital, ha sido tradicionalmente el punto de contacto entre las autoridades y la comunidad. Hoy en día están encargadas de ejercer el control administrativo en el municipio. Tiene autonomía presupuestal y administrativa. Está conformada por un personero Distrital, una planta de personal y por Personerías Delegadas, que conforman las llamadas personerías delegadas de acuerdo con las necesidades del municipio; estos son los encargados de resolver los diferentes casos que son de competencia de la entidad.

El personero, bajo la dirección suprema del Procurador General¹³, ejerce en el municipio las funciones del Ministerio Público como lo es la guarda y promoción de los derechos humanos, la protección del interés general y la vigilancia de la conducta de quienes desempeñan funciones públicas, además de las funciones que reciba por delegación de la Procuraduría General ¹⁴ de la Nación. Los personeros son elegidos a partir de 1995 por el consejo municipal o Distrital para períodos de tres años.

El personero tiene un amplio campo de acción en el municipio en cuanto a la

¹⁰ El ministerio público será ejercido por el procurador General de la Nación, por el Defensor del pueblo, por los procuradores delegados y los agentes del ministerio público, ante las autoridades jurisdiccionales, por los **personeros municipales** y por los demás funcionarios que determine la ley.

¹¹ Ley 136 de 1994. Nuevo Régimen Municipal Colombiano, XI Personeros Municipales

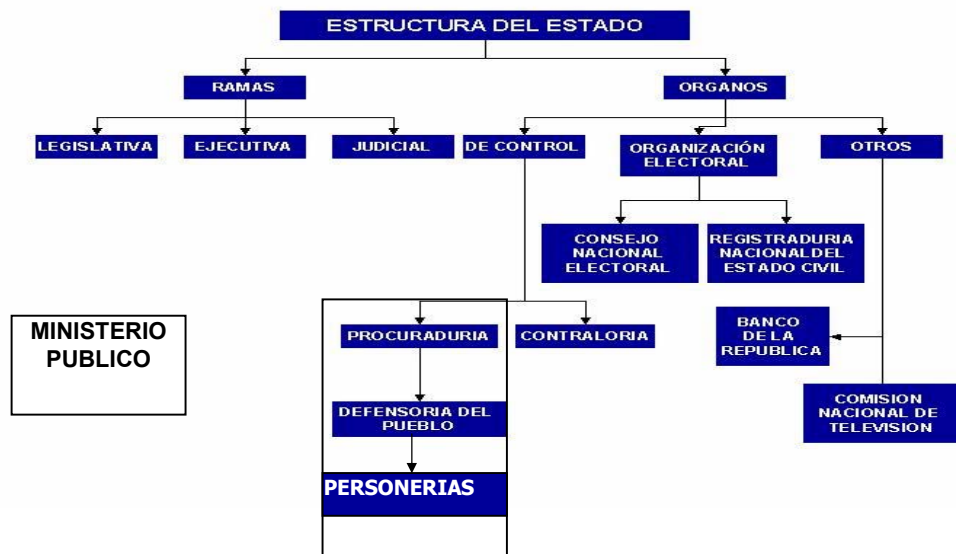
¹² Constitución Política de Colombia , Título X Capítulo II.

¹³ Supremo director del Ministerio Público.

¹⁴ Constitución Política de Colombia , Título X Capítulo II.

protección de los derechos humanos de los sectores vulnerables de la población es decir a favor de ancianos, menores, presos, indigentes, mujeres, desplazados, etc.

Figura 1. Estructura del estado Colombiano



Otras actividades desempeñadas por el Personero son: Las de vigilar que en el sector salud en su municipio se lleve a cabo las campañas de vacunación, que los hospitales funcionen de manera adecuada. En cuanto a las actividades encaminadas a la prevención de crímenes de lesa humanidad (torturas, masacres, desapariciones forzadas), el personero se convierte en la mano derecha del Procurador Delegado para los Derechos Humanos, y en el caso de ser cometido es él quien dirige las más urgentes diligencias necesarias para adelantar los procesos penales y disciplinarios requeridos.

Enmarcados dentro de la política de modernización del estado, buscando incrementar cuantitativa y cualitativamente los niveles de eficacia en la gestión interna y servicio a la comunidad, conscientes del papel social que en el

cumplimiento de sus funciones de ministerio público, tiene la Personería, surge la idea de diseñar e implementar un sistema de información web para la Personería Distrital de Cartagena.

1.2. OBJETIVOS

1.2.1. Objetivo General

Diseñar e implementar un sistema de información web, soportado en tecnologías Internet y con base en la Directiva Presidencial 02 de Agosto 28 del 2000 “Gobierno en Línea”, que facilite la gestión de la Personería Distrital de Cartagena de Indas y sirva de apoyo en el cumplimiento de su función de Ministerio Público

1.2.2. Objetivos Específicos

1.2.2.1. Apoyar la gestión administrativa de la P. D. C mediante el desarrollo de aplicaciones con tecnología Internet para:

- Ejercer seguimiento y control al proceso Quejas, conociendo su evolución y evitar el vencimiento de términos (tiempo) fijados por la ley, para ello se utilizará información como :
 - Personero encargado del proceso.
 - Datos generales del quejosos.
 - Tipo de proceso.
 - Estado del proceso.
 - Fecha de la última actuación.
 - Forma de terminación del proceso.

- Facilitar y agilizar la elaboración y trámite de informes periódicos que sobre los procesos a su cargo deben rendir los Personeros.

- Facilitar la elaboración de una agenda de actividades que permita, a cada Personero, programar las tareas relacionadas con los

procesos a su cargo, logrando una mejor distribución del tiempo para la solución de los mismos.

- Crear mecanismos de intercambio de información que propicien una comunicación efectiva entre los miembros de la entidad, como son:
 - Correo electrónico.
 - Cartelera de eventos y noticias.
 - Mecanismos de discusión en grupo.
 - Transferencia de archivos.

- Publicar documentos y suministrar información de interés general para la comunidad mediante un sitio web que permita prestar servicios y realizar trámites a los ciudadanos a través de Internet posibilitando la interacción de la comunidad con la P. D. C, cumpliendo con lo estipulado en las Fases 1 y 2 de la directiva presidencial 02, servicios como:
 - Recepción y atención de quejas, solicitudes e inquietudes.
 - Consulta del estado de los procesos .
 - Mecanismos de opinión.

1.2.2.2 Utilizar el **Proceso Unificado de Desarrollo de Software** como metodología que proporciona una familia de técnicas que soportan el ciclo completo de desarrollo de software, empleando el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas del sistema (modelo de casos de uso, modelo de diseño y modelo de implementación).

1.2.2.3 Desarrollar el sistema de información web bajo la arquitectura multicapas, capa de presentación, capa de reglas de aplicación y capa de datos, la clasificación de cada capa esta dada por su funcionalidad, así el mantenimiento y administración del sistema se convierte en una labor sencilla.

1.2.2.4 Desarrollar el sistema usando herramientas con licencias GNU ¹⁵, entre las cuales tenemos el sistema operativo Linux, el servidor web Apache y el manejador de base de datos Postgresql y MySQL.

1.3. DESCRIPCIÓN DE OBJETIVOS

El sistema propuesto permitirá publicar y divulgar a través de un sitio web, información de interés general para la comunidad exigida en la Fase 1 y 2 de la directiva y que se aplique a la Personería, tal como:

- Funciones, organigrama, localización física y electrónica, teléfonos y fax, horarios de trabajo y demás indicaciones que sean necesarias para que las personas puedan cumplir sus obligaciones o ejercer sus derechos.
- Funcionarios principales.
- Normas básicas que regulan la actividad de la entidad.
- Regulaciones y procedimientos que le son aplicables a la entidad (incluidos los internos).
- Regulaciones, procedimientos y tramites a que están sujetas las actuaciones de los particulares, precisando de manera detallada los documentos que deben ser suministrados, dependencias responsables y plazos que se deben cumplir.
- Planes y programas en ejecución y futuros.
- Entidades de control que vigilan a la entidad y mecanismos de control.
- Contratos realizados, indicando identificación del contratista, objeto, monto, plazo y demás condiciones.
- Mecanismos de rendición de cuentas a la ciudadanía tales como metas e informes de gestión.

Además el sitio web permitirá a la P.D.C prestar servicios a través de Internet , como es el caso de la recepción y atención de quejas, solicitudes e inquietudes,

¹⁵ GNU: es la licencia que se da al software de libre distribución. Toda herramienta software que esté registrada en este tipo de licencia debe ser entregada a los usuarios con el código fuente y puede ser

con el fin de complementar y ampliar la forma en que este servicio es prestado por la entidad. A su vez la comunidad podrá conocer el resultado de algunos procesos a través de los **Boletines de Prensa**, donde periódicamente se tendrá información de procesos que la P.D.C tiene a su cargo prestando especial atención a las normas sobre privacidad y derecho a la información. La ciudadanía podrá interactuar con la Personería y podrá opinar o discutir sobre temas que sean propuestos por esta o por la misma comunidad, a través de votaciones, encuestas y foros.

Para apoyar la gestión interna de la P.D.C se propone desarrollar un conjunto de aplicaciones basadas en el web que le brindará a los funcionarios la oportunidad de aprovechar al máximo los recursos informáticos y telemáticos disponibles. Uno de los procesos más importantes y determinantes en el desempeño de la Personería son las quejas de los ciudadanos, a través de las cuales la entidad puede cumplir su función como órgano de control en el municipio. Por lo tanto es necesario una aplicación que facilite su recepción y seguimiento, permitiendo a los Personeros conocer en cualquier momento el progreso de los procesos mediante información descrita en el objetivos específico 1.2.2.1 y evitar el vencimiento de términos (tiempo) en algunos de ellos.

Se facilitará la elaboración y trámite automatizado de los informes eventuales o periódicos que los Personeros Delegados deben rendir ante solicitud del Personero Distrital sobre los procesos a su cargo, logrando una mayor eficiencia en la gestión de estos documentos.

Los Personeros en el desarrollo de sus procesos necesitan llevar a cabo varias actividades como visitar hospitales y centros de salud, establecimientos carcelarios, de policía y de internación psiquiátricas, hogares para menores y ancianos, es decir, practicar visitas ordinarias y extraordinarias en las diferentes oficinas del orden Distrital sobre las cuales se ejerce la vigilancia, para lo cual se desarrollará una aplicación que posibilite la programación automática diaria, semanal o mensual de

estás actividades para cada uno de los Personeros teniendo en cuenta disponibilidades horarias y la intensidad horaria de las actividades, logrando una distribución adecuada del tiempo para cumplir con las tareas asignadas y evitar interferencias y conflictos de programación de actividades. El sistema prestará servicios básicos que mejorarán el trabajo en equipo, como **Correo electrónico** (Enviar, recibir y archivar mensajes y adjuntar archivos con un mensaje), **Mecanismos de discusión en grupo** (Para la discusión de ideas, críticas y coordinación en la elaboración de conceptos, con el objeto de mantener uniformidad de criterios), **Cartelera de eventos y noticias**, **Transferencia de archivos** (Propiciar el intercambio de información y que por su tamaño supera el admitido por un servidor de correo, impidiendo pueda ser enviada como e-mail).

En la actualidad la tendencia en el desarrollo de software lleva a la construcción de sistemas más grandes y complejos, así como también a software que esté mejor adaptado a nuestras necesidades ; lo que requiere de métodos más confiables de los que se han venido usando hace 25 años. Para este proyecto emplearemos El Proceso Unificado del Desarrollo de Software, el cual nos brinda una visión moderna de cómo se desarrolla un producto software; proporcionando un método común que nos da una guía para ordenar las actividades de un equipo de desarrollo, especificar los componentes que deben desarrollarse, los criterios para el control y la medición de los productos, y las actividades de un proyecto software, todo esto nos ofrece la seguridad de culminar un proyecto con calidad, dentro de los plazos y presupuesto planeados.

Por ultimo destacamos que la implementación de este sistema bajo la plataforma Linux y el uso de herramientas de libre distribución para su desarrollo, presenta una solución bastante económica en contraste con sistemas similares bajo otra plataforma. Una de las características principales de Linux consiste en su fácil acceso gracias a que todo el código fuente del sistema Linux incluyendo el kernel¹⁶, los manejadores de dispositivos, librerías, programas de usuario y herramientas de desarrollo son de distribución gratuita, lo que permite una reducción de costos hasta

¹⁶ Núcleo del sistema operativo.

10 veces menor que al utilizar otras plataformas. Está implementado en módulos que pueden ser agregados, actualizados o eliminados según las necesidades, lo que permitirá liberar al servidor de procesos que nunca serán utilizados para aprovechar más eficientemente los recursos del sistema.

El servidor web a utilizar será Apache, la principal función de un servidor Web es responder rápidamente cuando un usuario solicite una página. Para cada transacción el servidor debe realizar dos acciones básicas: integrar todos los componentes de la pagina (texto, imágenes, vídeo, scripts, etc.) y enviarla rápidamente al usuario. Apache es el servidor web por defecto para casi todas las distribuciones Linux y esta basado en HTTPd17 de NCSA(Centro Nacional de Aplicaciones de Supercomputación de la Universidad de Illinois), es el servidor web adecuado para este sistema.

El manejador de base de datos será Postgresql. Entre sus principales características tenemos:

- Relacional, incluye características como optimización de sentencias, control concurrente de transacciones y soporte multiusuario.
- Expandible, se pueden definir diferentes tipos de datos, operadores, funciones y métodos de acceso.
- Utiliza SQL, permite implementar un subconjunto de instrucciones del estándar ANSI SQL.
- Portable, se pueden programar scripts para ser utilizados por la mayoría de lenguajes.
- Independiente de la plataforma, se puede utilizar con un sistema operativo como Linux, Solaris, SunOS, Digital Unix, FreeBSD, entre otros.

Actualmente, el grupo de investigación en Sistemas y Tecnologías de la información-STI; viene adelantando una serie de investigaciones y proyectos, dentro de los cuales se incluye Sistema de Información web para la Personería distrital de Cartagena de indias - SIPEDCAR, en el área de la informática jurídica, esto con el fin de generar soluciones informáticas tanto para la comunidad Universitaria como para organizaciones estatales y académicas, favoreciendo la modernización y el

reconocimiento de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander como líder en el área de tecnologías de la información en la región y fuera de ella.

1.4. ALCANCES DEL PROYECTO

1.4.1. Social

El desarrollo del Sistema de Información Web para La Personería Distrital de Cartagena de Indias, le proporcionará a esta entidad una herramienta con la cual estará a la vanguardia, dentro de las instituciones de su tipo en la región, en la cual se llevan a cabo actividades de índole jurídico. Contando así no solo con un sistema que además de permitirle agilizar su gestión interna, cumple con las especificaciones gubernamentales en cuanto a la sistematización de los entes del estado, garantizando disponibilidad de la información independiente de la hora y el lugar , es decir, un nivel mínimo de servicio en un 95% del tiempo. Siendo la ciudadanía la principal beneficiaria de "... una institución moderna.. que responde de manera eficaz y eficiente a las comunidades actuando con celeridad, responsabilidad y objetividad.", como se contempla en la visión de la entidad.

1.4.2. Tecnológico

La creación de este sistema demanda la aplicación de muchos conocimientos propios de la ingeniería de sistemas, en especial lo relacionado con el desarrollo y documentación de sistemas de información, redes, tecnologías Internet , sistemas operacionales y de bases de datos entre otros. Además, la utilización y adaptación de diversas tecnologías de información a un ambiente poco explorado en Colombia convierte este proyecto en un valioso aporte no solo en su parte técnica sino en la concepción del uso y aprovechamiento al máximo de los recursos informáticos y telemáticos disponibles en el ámbito y quehacer de las

¹⁷ HTTPd, Hypertext Transfer Protocol daemon

entidades de orden jurídico con el fin de mejorar su gestión y capacidad de servicio.

1.4.3. Económico

Por ultimo destacamos que la implantación de este sistema permitirá a la institución, el uso de tecnologías de la información en red con un costo relativamente módico ya que el software utilizado por el servidor posee licencias GNU (Linux y herramientas de libre distribución), sin que ello implique perdida de robustez o seguridad que otros sistemas (UNIX, Windows) pueden brindar. El uso de este tipo de herramientas con licencias GNU, proporcionan una buena perspectiva para el desarrollo de nuevas tecnologías en diversos campos de aplicación y que en un entorno económico como el de nuestro país, surgen como la mejor alternativa.

CAPITULO 2

MARCO TEORICO, METODOLOGICO Y REFERENCIAL

Este capítulo presenta información relevante para una buena comprensión de la teoría y metodología empleada en el desarrollo del sistema. Aunque no ha sido organizada estructuralmente, se pueden distinguir tres categorías para facilitar su consulta, se encuentra la información de referencia que se utiliza para conceptualizar el problema y su solución, como lo es la informática jurídica, esta información se incluye en un apéndice resultado del estudio por parte de la línea de acción en ingeniería del software del grupo STI perteneciente a la escuela de Ingeniería de Sistemas e Informática de la Universidad industrial de Santander. Los desarrollos hechos bajo el enfoque de informática jurídica se tomaron como referencia en la solución de la situación problema. Existe otro tipo de información como es el marco metodológico, donde se describe el proceso unificado de desarrollo de software y algunas características puntuales del Lenguaje de Modelamiento Unificado (UML) además de los conceptos de la ingeniería del software así como algunas nociones de la denominada Ingeniería web.

Finalmente se presenta temas sobre las herramientas y tecnologías que ayudaron en la construcción de la solución , entre ellas todo lo relacionado con las herramientas web, el sistema operativo Linux, el servidor web Apache, el motor de bases de datos Postgresql y la arquitectura de múltiples capas.

2.1. INFORMÁTICA JURÍDICA

Básicamente la informática jurídica se puede definir como: “ La utilización de los diferentes conceptos, categorías, métodos y técnicas propios de la *informática* en el ámbito de lo *jurídico*. Se relaciona con la creación, flujo, clasificación, organización, sistematización y utilización de datos requeridos en la producción y/o en la administración de lo jurídico” ¹⁸.

¹⁸ Fernando Jordán (Opus cit. P.49). Tomado del libro Manual de Informática Jurídica de Orlando Solano .

La informática jurídica se puede clasificar en tres grandes ramas dependiendo en que centran su atención cada una de ellas y la manera como fueron apareciendo.

Informática Jurídica Documentaria: Centra su atención en el *Documento*, es decir en el almacenamiento y recuperación de información que contienen datos principalmente jurídicos, documentos como: Constitución, Leyes, Resoluciones, Jurisprudencia, textos jurídicos , etc.

Informática Jurídica de Gestión y Control: Sirve para organizar y hacer más funcionales los despachos que puedan de alguna forma tratar temas jurídicos, está dirigida a actuar dentro de un proceso fundamentalmente administrativo.

Informática Metadocumentaria o decisional : Emplea el ordenador en la ayuda de toma de decisiones a través de la solución de casos (Inteligencia Artificial).

Este proyecto lo hemos situado dentro de la Informática Jurídica de Gestión y Control, este campo de acción se convierte en nuestro marco de referencia , ya que su estudio nos ayudó a la identificación del contexto dentro del cual se enmarca el presente proyecto, gracias a ello pudimos llegar a estudiar desarrollos similares llevados a cabo por el gobierno en este campo a través de la historia permitiéndonos conocer sus características, ventajas y limitaciones dándonos una idea del sistema que queríamos construir.

Si está interesado en conocer más acerca de este tema, le sugerimos leer el Apéndice B donde encontrará un resumen sobre la informática jurídica (definición, clasificación y aplicaciones) resultado del estudio y la investigación realizada por la línea de informática jurídica, perteneciente al Grupo de Investigación en Sistemas y Tecnologías de la Información S T I ¹⁹ de la Universidad Industrial de Santander, sobre algunos textos de autores Colombianos y extranjeros existentes en este campo.

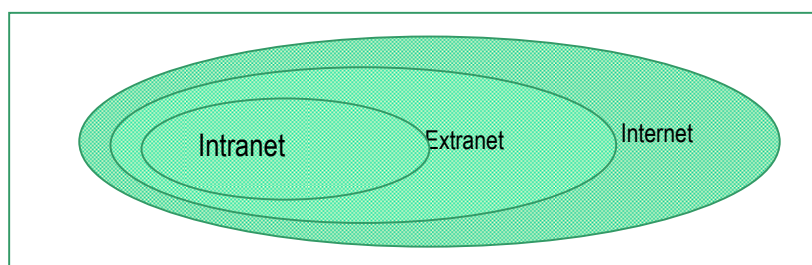
¹⁹ <http://www.uis.edu.co/investigacion/grupos/sti/STI/documentosframe.htm> En esta dirección encontrará información completa sobre informática jurídica desarrollada por el grupo STI.(Sesiones del curso, marco teórico, estado del arte y resúmenes de los proyectos).

2.2. INTERNET

Es el nombre de un sistema de alcance mundial que consta de información, personas y computadoras; es la red por excelencia que permite a computadoras de todo el mundo conectarse entre si. Las comunicaciones en Internet son posibles entre redes de diferentes ambientes y plataformas. Este intercambio dinámico de datos se ha logrado debido al desarrollo de los protocolos ²⁰ de comunicación.

Soluciones basadas en tecnología Internet

Figura 2. Alcance de Intranet, Extranet e Internet



2.2.1. Intranet

El uso más común de las tecnologías de Internet, por los negocios y organizaciones, es interno a sus redes de área local o de área amplia. Una LAN o WAN ²¹ que utilice las tecnologías de Internet es llamada una Intranet. Las Intranets brindan a los usuarios la capacidad de compartir dinámicamente recursos internos de la misma forma que los usuarios de Internet lo hacen.

Internet y las Intranets comparten los mismos estándares de comunicación, por lo que su diferencia no es tecnológica, su diferencia consiste en:

- Nivel de acceso. El alcance u audiencia de una Intranet está estrictamente limitado a un grupo de trabajo, un departamento o una corporación, pero sirve a una comunidad de usuarios bien definida y limitada.

²⁰ Los protocolos son un conjunto de reglas para el intercambio de datos que permiten a los usuarios comunicarse entre diferentes redes.

²¹ LAN: Red de área local, WAN: Red de área extensa

- La forma como se utilizan las tecnologías para el establecimiento de comunicaciones. Los paquetes de información transportados por una Intranet utilizan por lo general redes privadas, locales o de área amplia definidas por la estructura de red de la organización, o incluso por Internet cuando se cuenta con los requerimientos de velocidad y seguridad necesarios para asegurar la integridad de la información.
- Los objetivos de las partes en comunicación. Debido a su carácter corporativo, las Intranet poseen un estudio y diseño basados en las necesidades informáticas internas, al igual que unos propósitos bien definidos.

2.2.2. Extranet

Una extranet es una solución Internet que permite el intercambio de información de manera permanente, confidencial y segura con personas fuera de la organización como socios de negocios, clientes, proveedores, etc y que posee las siguientes características generales:

- Es un sistema de acceso restringido basado en tecnologías Internet.
- La comunicación se da solo entre la empresa (dueña del sistema) y su entorno. Las otras empresas usuarias de la Extranet no necesariamente se comunican entre sí.

La única diferencia significativa entre una Extranet y una Intranet es que en esta última generalmente sólo acceden empleados de la empresa, mientras que a una Extranet se permite el acceso tanto a empleados como a clientes, proveedores, colaboradores, etc. Por tanto se puede pensar en una Extranet como una extensión de la Intranet a usuarios externos autorizados.

2.3. SISTEMAS DE INFORMACIÓN

Un sistema de información puede definirse como “Un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución “²².

²² PRESSMAN, Roger S. Ingeniería del software un enfoque práctico. McGraw-Hill, 1993

Para el desarrollo de un sistema de información, el personal de desarrollo de una organización debe adoptar una estrategia de desarrollo llamada generalmente modelo de proceso o paradigma de ingeniería del software.

2.4. INGENIERÍA DEL SOFTWARE (IS).

La ingeniería del software se ha constituido en una de las ramas más importantes de la ingeniería de sistemas, debido a que suministra el fundamento necesario para construir y mantener sistemas de información. Para lo anterior, la ingeniería del software involucra tres elementos importantes: Métodos, herramientas, y Procedimientos.

- **Métodos.** Suministran la estructura básica y las instrucciones de diseño para desarrollar el software. Indican “como” construir técnicamente el software. Incluye tareas tales como: planeación de proyectos, análisis de los requerimientos del sistema, diseño de las estructuras de datos, condición, pruebas y mantenimiento.
- **Herramientas.** Son utilizadas para el desarrollo del software y suministran un soporte automático o semiautomático para los métodos. En la actualidad, se integran todas las herramientas de tal forma que la información generada por una de ellas puede ser usada por otra, creando lo que se conoce como Ingeniería del Software Asistida por Computador.
- **Procedimientos.** Son los elementos integrantes de los métodos y las herramientas, facilitando un desarrollo racional y oportuno del software. Definen la secuencia en que se aplican los métodos, las entregas de resultados que se requieren, los controles que ayudan a asegurar la calidad y las directrices para la evaluación del progreso del proyecto informático. En pocas palabras, los procedimientos permiten controlar el proceso de desarrollo de software.

Con estos elementos la Ingeniería del Software es una forma efectiva de generar software económico, eficiente y confiable.

Existen diversos métodos que han planteado los profesionales del software para definir las actividades que se deben llevar a cabo en un proyecto de desarrollo de sistemas, y al mismo tiempo lograr congruencia con todos los demás proyectos de este tipo, que se llevan a cabo dentro de una misma organización. Estas deben seleccionarse de acuerdo a las características del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren.

Entre los métodos de procesos para la ingeniería del software se tienen:

Cascadas, DRA(Desarrollo Rápido de aplicaciones), Prototipado, Entrega por etapas, Espiral y Proceso Unificado.

En la Tabla 1 se muestra un resumen de los diferentes modelos de procesos para la IS. y que constituyen el ciclo de vida en el desarrollo del software.

La elección del modelo de vida mas efectivo depende del contexto en el que se vaya a utilizar. Determinados modelos de ciclos de vida son considerados más rápidos que otros, pero cada uno de ellos será más rápido en determinadas situaciones, y mas lento en otras. Según McConnell en su libro *Desarrollo y Gestión de proyectos informáticos*, en la selección del modelo de desarrollo más conveniente se debe examinar el proyecto y responder a las siguientes preguntas:

- *¿Me compenetro con el cliente para la especificación de los requerimientos al comienzo del proyecto? ¿Es probable que el entendimiento de las dos partes cambie significativamente a medida que se avance en el proyecto?*
- *¿Comprendo bien la arquitectura del sistema?¿Es probable que necesite llevar a cabo modificaciones importantes en la arquitectura a mitad del proyecto?*
- *¿Cuánta fiabilidad ²³ necesito?*
- *¿Cuánto tiempo extra necesito para planificar y diseñar durante el proyecto para las versiones futuras?*
- *¿Cuántos riesgos conlleva el proyecto?*
- *¿Estoy sometido a una planificación predefinida?*

²³ Se refiere a los defectos que pueden aparecer en un sistema desarrollado (cuando empieza a funcionar) con un modelo en particular.

- *¿Necesito poder realizar modificaciones a medio camino?*
- *¿Necesito proporcionar a mis clientes signos visibles de progreso durante el proyecto?*
- *¿Necesito ofrecer a la directiva signos visibles de progreso durante el proyecto?*
- *¿Cuánta sofisticación necesito para utilizar el modelo de ciclo de vida con éxito?*

Después de dar respuesta a estas preguntas, en la Tabla 1 encontramos una guía para decidir que ciclo de vida utilizar. La efectividad de cualquier modelo dependerá de cómo se implemente. De otra parte, si se sabe que un modelo es débil en un área particular, podemos controlar este inconveniente al principio de la planificación del proyecto y compensarlo creando una mezcla entre uno o mas modelos.

Tabla 1. Guía para la selección del ciclo de vida a utilizar en un proyecto.

Capacidades del modelo	Cascada Modificada	DRA	Prototipo Evolutivo	Espiral	Entrega por etapas	Proceso Unificado
Trabaja con poca identificación de los requerimientos	Medio a Excelente	Malo	Excelente	Excelente	Malo	Excelente
Trabaja con poca comprensión sobre la arquitectura	Medio a Excelente	Malo	Malo a Medio	Excelente	Malo	Malo o Medio
Genera un sistema altamente fiable	Excelente	Medio	Medio	Excelente	Excelente	Excelente
Genera un sistema con amplio desarrollo	Excelente	Medio a Excelente	Excelente	Excelente	Excelente	Excelente
Gestionar riesgos	Medio	Excelente	Medio	Excelente	Medio	Excelente
Estar sometido a una planificación predefinida	Medio	Excelente	Malo	Medio	Medio	Excelente
Requiere poco tiempo de gestión	Excelente	Medio	Medio	Medio	Medio	Malo a Medio
Permite modificaciones a medio camino	Medio	Medio	Excelente	Medio	Malo	Excelente
Ofrece a los clientes signos visibles de progreso	Medio	Excelente	Excelente	Excelente	Medio	Excelente
Ofrece a la directiva signos visibles de progreso	Medio de Excelente	Excelente	Medio	Excelente	Excelente	Excelente

Requiere poca sofisticación para los directivos y desarrolladores.	Malo a Medio	Medio	Malo	Malo	Medio	Excelente
--	--------------	-------	------	------	-------	-----------

Tomada del proyecto: SISTEMA INTRANET DE INFORMACIÓN PARA EL APOYO DE LA ACTIVIDAD ACADÉMICA EN EL PROGRAMA DE ESPECIALIZACION EN DOCENCIA UNIVERSITARIA DEL CEDEDUIS - SIEDUIS 1.0 - Realizado por Andrea Marcela Mora Villamizar, Marzo del 2002.

Luego de responder a las preguntas de McConnell y teniendo como guía la información de la tabla 1 ; en la realización de este proyecto se ha escogido el *Proceso Unificado de Desarrollo de Software* como metodología (Modelo) de desarrollo. Para la construcción del proyecto se cuenta con bastante conocimiento sobre los requisitos y arquitectura del sistema, representado en la información suministrada en la directiva presidencial 02 de agosto 28 del 2000 así como entrevistas realizadas con anterioridad a los miembros de la Personería. La información sobre la arquitectura del sistema cobra gran importancia en el uso de esta metodología de desarrollo, ya que a lo largo del desarrollo del proyecto esta debe permanecer estable para poder aprovechar todos los beneficios de este método.

Otro aspecto importante en el desarrollo del sistema, y en el que el Proceso Unificado juega un papel importante, es en la detección y reducción riesgos de forma temprana; también por requerir de un amplia gestión permite a los clientes y en especial al director del proyecto, ejercer un constante seguimiento del mismo, logrando así cumplir con la entrega del sistema con la calidad esperada, dentro de los plazos y presupuesto planeados.

Por último cabe anotar que el desarrollo de este proyecto estará enmarcado dentro de las tres primeras fases de la metodología (Proceso Unificado), quedando por fuera la fase de Transición, dado que las implicaciones directas que esta fase tiene en tamaño y duración, están fuera del alcance de este proyecto, así como también a la no disponibilidad inmediata de una infraestructura operativa apropiada para su implantación.

	SE UTILIZA CUANDO....	APLICACIÓN DE OTROS MODELOS	VENTAJAS	CONTACTO CON EL CLIENTE	DETECCIÓN DE ERRORES	PROBLEMAS Y RIESGOS
CASCADA PURA	<ul style="list-style-type: none"> • los requerimientos se pueden definir fácil y rápidamente a partir del conocimiento del área en estudio. • Se tiene una definición estable del producto, con variaciones mínimas de los requisitos. • Se trabaja con tecnologías conocidas • Se Construye una versión de mantenimiento de un producto. • Se migra un producto a una nueva plataforma. • Los requerimientos de calidad dominan sobre los de costes de planificación • Se dispone de personal inexperto en el manejo de una metodología en particular. 	Es una plantilla para otros métodos	<ul style="list-style-type: none"> • Presenta el proyecto con una estructura que ayuda a minimizar el esfuerzo inútil. • Ayuda a localizar errores en las primeras etapas del proyecto a un bajo costo (Si se tiene una definición estable del producto) • Permite enfrentarse a la complejidad de forma ordenada (En Proyectos complejos comprendidos totalmente) 	El cliente tiene contacto con el producto solo hasta el final del desarrollo.	Las pruebas para detección de errores se hacen al final del ciclo.	<ul style="list-style-type: none"> • No proporciona resultados de software hasta el final del ciclo • El cliente tiene contacto con el producto solo hasta el final del desarrollo. • Dificultad para especificar exactamente los requerimientos al comienzo del proyecto. • No permite flexibilidad en los cambios. Olvidar algo puede suponer un error costoso. • Discontinuidad de sus etapas, hasta no terminar una no se da inicio a la siguiente.. • Gran cantidad de documentación.

<p>DRA</p>	<ul style="list-style-type: none"> • Se comprenden bien los requisitos del cliente. • Se limita el ámbito del proyecto. • Se puede separar el desarrollo en módulos. • Proyectos donde no se requiera un alto rendimiento. • Se posee un buen conocimiento de las herramientas de desarrollo. 	<p>Adaptación de Cascada pura basado en componentes</p>	<ul style="list-style-type: none"> • Desarrollo de proyectos en periodos cortos de tiempo (60-90 días) • Enfatiza en el uso de componentes reutilizables. 	<p>Al concluir un modulo se entrega un producto de prueba al cliente.</p>	<p>Al final de cada modulo o subproyecto.</p>	<ul style="list-style-type: none"> • En proyectos grandes se requieren suficientes personas para crear los equipos. • Requiere clientes y desarrolladores comprometidos en completar un sistema en corto tiempo. • No es recomendable cuando los riesgos técnicos son altos: por el uso de tecnologías nuevas o si se requiere interoperatividad con programas ya existentes.
-------------------	--	---	---	---	---	--

PROTOTIPAD O EVOLUTIVO	<ul style="list-style-type: none"> • Los requerimientos cambian con rapidez • El cliente no especifica los requerimientos. • No se identifica fácilmente el área de aplicación • No se esta seguro de la arquitectura o los algoritmos adecuados a utilizar 	<p>Es una plantilla para otros métodos</p>	<p>El producto final cumplirá con todos los requerimientos del cliente.</p>	<p>Se presentan prototipos al cliente, continuando el desarrollo con los cambios sugeridos hasta ser aceptado como producto final.</p>	<p>Cada vez que el cliente prueba el prototipo</p>	<ul style="list-style-type: none"> • Imposibilidad de conocer al comienzo del proyecto lo que se tardara en crear un producto aceptable. • Se puede degenerar fácilmente en un desarrollo sin ninguna planeación (con el esquema de codificar y corregir)
ESPIRAL	<p>Es aplicable a todo tipo de desarrollo de software, especialmente en aquellos donde no se conocen bien los requerimientos del sistema.</p>	<ul style="list-style-type: none"> • Puede finalizar con un modelo en cascada, prototipado u otro modelo • Puede incorporar otros modelos en sus iteraciones. 	<ul style="list-style-type: none"> • Las primeras iteraciones son menos costosas • Mientras los costos suben los riesgos bajan. • Proporciona bastante control de gestión. • Reducción continua del nivel de riesgo en el proyecto. • Mejor visibilidad de progreso. 	<p>Hasta que se complete todo el numero especificado de iteraciones requeridas para completar el producto.</p>	<p>Las pruebas se hacen al final de cada ciclo.</p>	<ul style="list-style-type: none"> • Requiere mayor planificación • Requiere mayor seguimiento del proyecto. • Modelo complicado. • Puede ser difícil definir hitos y objetivos de comprobación que indiquen si esta preparado para pasar al siguiente nivel. • El cliente no siempre esta

						predispuesto a ser parte integral del desarrollo de software
ENTREGA POR ETAPAS O INCREMENTAL	<ul style="list-style-type: none"> • Se conoce exactamente lo que se va a construir. • El personal no esta disponible para una implementación completa 	Combina elementos del modelo en cascada con la filosofía iterativa de construcción por prototipos	<ul style="list-style-type: none"> • Presenta signos tangibles de progreso en el proyecto. • Proporciona una funcionalidad útil al cliente antes de entregar el 100% del proyecto. • Se pueden entregar las prestaciones mas importantes al inicio. 	El producto se entrega por partes como resultado de cada una de las etapas de desarrollo.	Las pruebas se hacen al final de cada etapa	No funciona sin una planificación adecuada tanto para niveles técnicos como para niveles de gestión.

<p>PROCESO UNIFICADO</p>	<ul style="list-style-type: none"> • Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. 	<p>En cada una de sus fases e iteraciones utiliza el método en cascada. Al inicio se puede utilizar prototipos de prueba.</p>	<ul style="list-style-type: none"> • El proceso unificado es un marco de trabajo genérico. • Busca eliminar los riesgos críticos primero. • Reduce el costo de riesgos. • Presenta signos visibles de progreso rápidamente. • Proporciona funcionalidad al cliente antes de entregar todo el proyecto. • Se entregan prestaciones importantes al inicio • Proporciona suficiente control de gestión 	<p>Se entrega un incremento al final de cada iteración, en especial en la etapa de construcción</p>	<p>Las pruebas se hacen al final de cada iteración. Además al final de cada fase se realiza una evaluación del plan de trabajo.</p>	<ul style="list-style-type: none"> • Requiere mayor planificación • Requiere mayor control de gestión. • Se necesita definir su arquitectura antes de dar inicio a las iteraciones de construcción.
---------------------------------	--	---	--	---	---	--

Tabla 2. Resumen de los principales Modelos de Ciclo de Vida de Desarrollo del Software.

Tomada del proyecto: SISTEMA INTRANET DE INFORMACIÓN PARA EL APOYO DE LA ACTIVIDAD ACADÉMICA EN EL PROGRAMA DE ESPECIALIZACION EN DOCENCIA UNIVERSITARIA DEL CEDEDUIS - SIEDUIS 1.0 -. Realizado por Andrea Marcela Mora Villamizar, Marzo del 2002.

2.5. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

La metodología a emplear en este proyecto es el *Proceso unificado de Desarrollo de Software*, el cual nos proporciona una guía para ordenar las actividades durante el desarrollo del proyecto, dirigiendo las tareas de cada desarrollador por separado y como un equipo, especificando los artefactos²⁴ que deben construirse, ofreciendo criterios para el control y medición de los productos. Entendiendo *Proceso de Desarrollo de Software* como el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software.

El *Proceso unificado* es iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso y los riesgos. Está basado en componentes, lo cual quiere decir que el sistema en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

- **Dirigido por casos de uso:** Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema lleva a cabo y que proporcionan un resultado de interés para un usuario determinado; dicho usuario no es necesariamente una persona, puede ser algún tipo de sistema externo que interactúe con el sistema en cuestión, por ejemplo sensores, temporizadores, etc.

El desarrollo del software se centra en la importancia del desarrollo para el usuario y no en términos de funciones que debe cumplir el sistema. Los Casos de Uso dirigen el proceso durante todos los flujos de trabajo (requisitos, análisis, diseño, implementación y prueba), de las distintas fases.

- **Centrado en la arquitectura:** La arquitectura es el conjunto de decisiones significativas acerca de la organización de un sistema, la selección de los elementos estructurales a partir de los cuales se compone el sistema y las interfaces entre ellos, junto con su comportamiento. Además se interesa por las

²⁴ **Artefacto:** Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades. Un artefacto puede ser un modelo, un elemento del modelo o un documento.

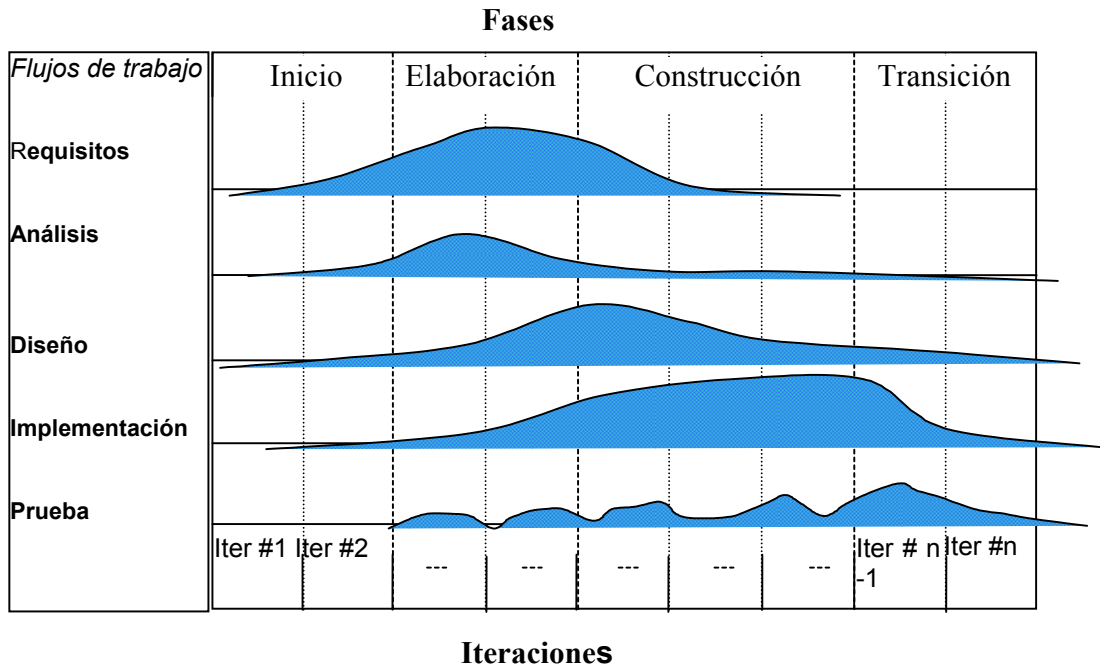
restricciones y compromisos de uso, funcionalidad, funcionamiento, flexibilidad al cambio, reutilización, comprensión, economía y tecnología, así como por aspectos estéticos. La arquitectura interactúa dinámicamente con los casos de uso, esto es, los casos de uso pueden llegar a alterar la arquitectura y esta a su vez puede hacer necesaria la inclusión de nuevos casos de uso. Centrado en la arquitectura significa que la arquitectura de un sistema se usa como un artefacto primordial para la conceptualización, construcción, gestión y evolución del sistema en desarrollo.

- **Iterativo e Incremental:** El Proceso unificado de Desarrollo de Software es iterativo e incremental porque permite controlar los riesgos críticos y significativos desde el principio del proceso. Las iteraciones hacen referencia a diversos pasos en los flujos de trabajo, y los pasos, al incremento del producto. Las iteraciones deben estar controladas, es decir, deben ejecutarse en forma planificada. En cada iteración, se identifican y especifican los casos de uso relevantes, se crea un diseño utilizando la arquitectura seleccionada como guía, se implementa el diseño mediante componentes, y se verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus objetivos el desarrollo continúa con la siguiente iteración, en caso contrario, se revisa las decisiones previas y se prueba un nuevo enfoque. Esto hace que el desarrollo a través de las iteraciones sea incremental.

La arquitectura proporciona la estructura sobre la cual guiar las iteraciones mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. De esta manera el proceso reduce el riesgo de grandes retrasos en la entrega de un producto, se fijan metas más inmediatas por lo cual se pueden controlar mejor el avance del proyecto

De esta manera el proceso reduce el riesgo de grandes retrasos en la entrega de un producto, se fijan metas más inmediatas, por lo cual se puede controlar mejor el avance del proyecto.

Figura 3. Flujos de trabajo en las cuatro fases del proceso unificado



El Proceso Unificado divide el proceso de desarrollo en ciclos, donde se obtiene una nueva versión del producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción, y 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. Cada una de estas fases se divide a su vez en iteraciones.

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. En la iteración también se incluye la planificación que precede a los flujos de trabajo y la evaluación que va detrás de ellos.

Por las características mencionadas anteriormente, el Proceso unificado de Desarrollo de Software se presenta como un marco ideal para el desarrollo de

sistemas y aplicaciones basadas en el web, como es el caso de este proyecto. Estas aplicaciones presentan atributos específicos los cuales no sufren alteraciones al seguir esta metodología de desarrollo, por el contrario ayudan a conservarlas y resaltarlas; como son: Inmediatez, evolución continuada, diseño basado en componentes, seguridad, intensivas de red²⁵ y estética.

Por otra parte el Proceso unificado de Desarrollo de Software, permite cumplir con los requisitos de calidad de una aplicación Web como la usabilidad, funcionalidad, fiabilidad, eficiencia y capacidad de mantenimiento.

2.6. LENGUAJE UNIFICADO DE MODELADO (UNIFIED MODELING LANGUAGE, UML).

UML es un lenguaje estándar para escribir planos de software. Se utiliza para visualizar, especificar, construir y documentar los artefactos o productos de un sistema que consta de una gran cantidad de software. Este lenguaje es adecuado para modelar desde sistemas de información en empresas hasta aplicaciones distribuidas basadas en la web (Como lo es el sistema que se construirá en este proyecto). Es un lenguaje que cubre todas las vistas necesarias para desarrollar y luego desplegar tales sistemas.

UML, como todo lenguaje, es tan solo una parte de un proceso o modelo de desarrollo de software; siendo independiente de este proceso. Para usarlo óptimamente se debería usar en un proceso que fuese dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental (Proceso Unificado de Desarrollo de Software). Para más detalles revisar el Anexo B.

2.7. INGENIERÍA WEB (IWEB).

¿Pueden aplicarse principios, conceptos y métodos de ingeniería en el desarrollo de las aplicaciones basados en el Web?

“Cualquier producto o sistema importante es merecedor de recibir una ingeniería. Antes de comenzar a construirlas, lo mejor es entender el problema, diseñar una solución viable,

²⁵ Por su propia naturaleza, una aplicación web es intensiva de red. Reside en una red y debe dar servicios a las necesidades de una comunidad diversa de clientes. Una aplicación Web puede residir en Internet, en una Intranet o en una Extranet.

implementarla de una manera sólida y probarla en profundidad. Probablemente también se deberían controlar los cambios a medida que el trabajo vaya avanzando, y disponer de mecanismos para asegurar la calidad del resultado final. Muchos de los que desarrollan Webs no dicen lo mismo; ellos piensan que su mundo es realmente diferente, y que simplemente no se van a aplicar los enfoques de ingeniería del software convencionales “²⁶

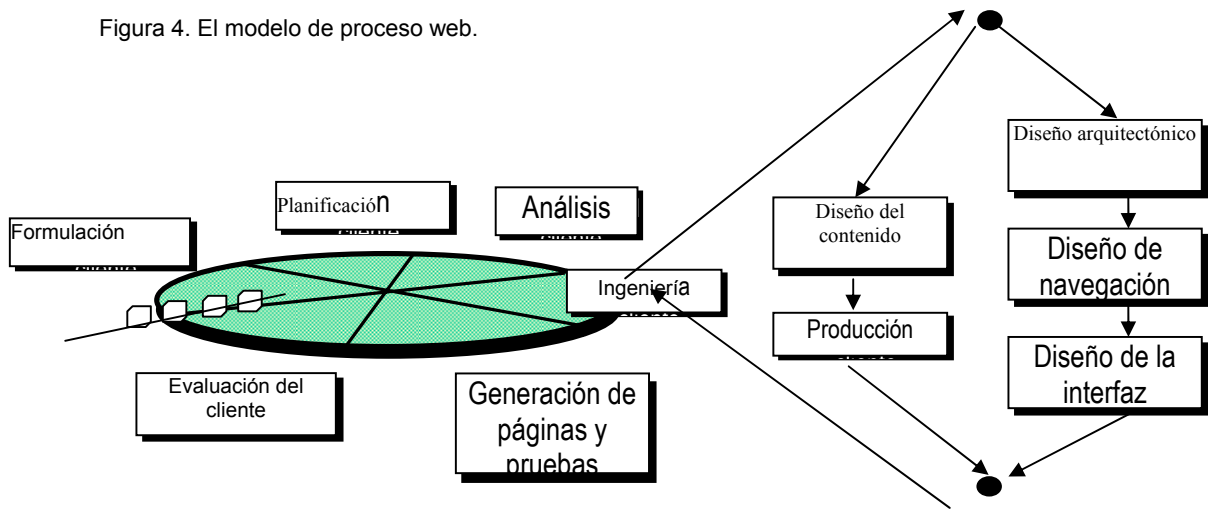
La ingeniería web no es una aplicación perfecta de la ingeniería del software, pero toma prestado muchos de los conceptos y principios básicos de esta. La lweb está relacionada con el establecimiento y utilización de principios científicos, de ingeniería y de gestión, con enfoques sistemáticos y disciplinados del éxito del desarrollo, empleo y mantenimiento de sistemas y aplicaciones basados en Web de alta calidad.

El proceso de lweb comienza con una formulación del problema que pasa a resolverse con una aplicación web. Se planifica el proyecto y se analizan los requisitos de la aplicación, entonces se lleva a cabo el diseño de interfaces arquitectónico y del navegador. El sistema se implementa utilizando lenguajes y herramientas especializados asociados con la Web, y entonces comienzan las pruebas. Dado que estas aplicaciones están en constante evolución, deben de establecerse los mecanismos para el control de configuraciones, garantía de calidad y soporte continuado.

A mediada que la evolución de las aplicaciones web pasan de utilizar recursos estáticos de información controlada por el contenido a utilizar entornos de aplicaciones dinámicos controlados por el usuario, se hace cada vez más importante

²⁶ PRESSMAN, Roger S. Ingeniería del software un enfoque práctico. 5 Edición McGraw-Hill. Pág. 521

Figura 4. El modelo de proceso web.



la necesidad de aplicar una gestión sólida y unos principios de ingeniería. Para conseguir esto es necesario desarrollar un marco de trabajo IWeb que acompañe a un modelo de proceso eficaz, popularizado por las actividades ²⁷ del marco de trabajo y por las tareas de ingeniería. En la figura 2.2 se muestra un modelo de proceso para IWeb.

El proceso inicia con **la Formulación**, es la actividad que identifica las metas y los objetivos de la aplicación y establece el ámbito del primer incremento.

La planificación estima el costo global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo y define una planificación del desarrollo detallada para el incremento final de la aplicación. **El análisis** establece los requisitos técnicos para la aplicación e identifica los elementos del contenido que se van a incorporar. También se definen los requisitos del diseño gráfico. La actividad de **ingeniería** incorpora dos tareas paralelas.

El diseño del contenido y la producción son tareas llevadas a cabo por personas no técnicas del equipo Iweb. El objetivo de estas tareas es diseñar, producir, y/o adquirir todo el contenido de texto, gráfico y vídeo que se vayan a integrar en la aplicación. Al mismo tiempo se llevan a cabo un conjunto de tareas de diseño. La **generación de páginas** es una actividad de construcción que hace mucho uso de

²⁷ Las actividades del marco de trabajo se realizan para todas las aplicaciones web; mientras que las tareas se adaptan al tamaño y a la complejidad de la aplicación web que se va a desarrollar.

las herramientas automatizadas para la creación de la aplicación web. El contenido definido en la actividad de ingeniería se fusiona con los diseños arquitectónicos, de navegación y de la interfaz.

Las **pruebas** ejercitan la navegación, intentan descubrir los errores de las applets²⁸, guiones y formularios, y ayuda a asegurar que la aplicación funcionará correctamente en diferentes entornos (Por ejemplo, con distintos navegadores).

Cada incremento producido como parte del proceso Iweb se revisa durante la actividad de **evaluación del cliente**.

2.8. ARQUITECTURA DE SOFTWARE

La arquitectura de un sistema es la vista conceptual de la estructura de este. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. Por tanto su arquitectura se define según como este distribuido este código.

En los últimos años la arquitectura de los sistemas de software desarrollados ha evolucionado junto con el resto de la tecnología informática. Hace no muchos años, cuando los mainframes eran una novedad, se desarrollaban sistemas monolíticos en donde un sistema no conocía nada mas allá de su entorno de operación.

Los continuos avances en las áreas de la computación y las comunicaciones, hardware más potente y barato (los PC actuales tiene más potencia que los mainframes de hace algunos años), sistemas de comunicación vía satélite, etc; propiciaron la conexión económica y eficiente de sistemas informáticos. Pasando de sistemas monolíticos al manejo de un nuevo concepto, los sistemas de computación distribuidos. Cuando la tecnología de redes aparece y se difunde, la industria no tardó en darse cuenta de la ventaja de desarrollar sistemas que contaran con la capacidad de interactuar con otros sistemas residentes en otras máquinas dentro de la red. De aquí surge el modelo Cliente-Servidor donde un "Cliente"²⁹ solicita

²⁸ Pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta como parte de un documento web

²⁹ Computadora que realiza un servicio que normalmente requiere de mucha potencia de procesamiento.

servicios de un "Servidor"³⁰ el cual gestiona las solicitudes de varios clientes a la vez.

Con el auge de Internet y el desarrollo de nuevas tecnologías para software por componentes, se pueden construir aplicaciones distribuidas que residan en uno o mas servidores de una red, disminuyendo así la necesidad de software cliente ya que generalmente se utiliza uno ya estandarizado: los navegadores de Internet, tales como Netscape, Internet Explorer, entre otros.

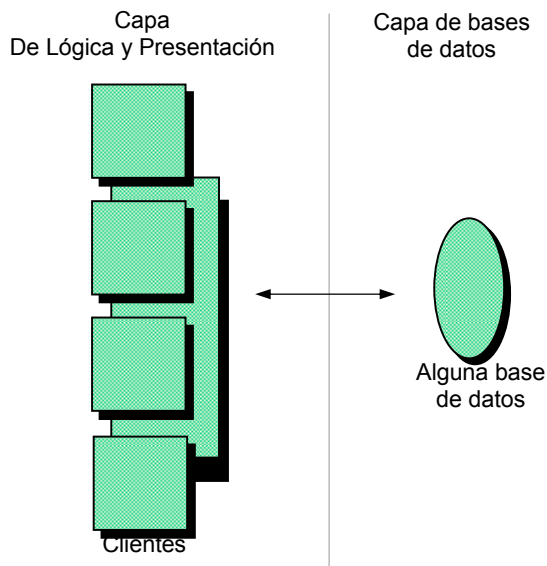
Las organizaciones modernas están aprovechando estas tecnologías que permiten a sus usuarios tener acceso sencillo y casi universal a las aplicaciones corporativas sin la necesidad de incurrir en todos los gastos de mantenimiento que conlleva el modelo cliente-servidor como fue planteado en sus inicios.

Para que todo esto funcione, se requiere incrementar la lógica de programación del lado del servidor ya que la funcionalidad del cliente es mínima. Para lograr esto en la actualidad se maneja el concepto de sistema multicapas o de "n" capas como una metodología para el diseño de sistemas distribuidos. Esta arquitectura se ha popularizado desde principios de los años 90 y en la actualidad se ha afianzado como la arquitectura de software de aplicación empresarial.

³⁰ Computadora que solicita servicios a uno o más servidores, realiza también algún tipo de procesamiento

2.8.1. Cliente – servidor

Figura 5 Arquitectura Cliente - Servidor.



Esta es una versión simplificada de la arquitectura en capas, donde la capa de aplicación se encuentra repartida entre el cliente y el servidor. Los sistemas basados en la arquitectura cliente/servidor están formados por dos partes lógicas o capas: un servidor que proporciona servicios (Back-End), y un cliente que solicita servicios del servidor o servidores (Front-End). Juntos, forman un sistema de computación completo con una clara división de responsabilidades.

Un cliente es un proceso que envía un mensaje a un proceso servidor, solicitando que realice determinada tarea. Los procesos cliente normalmente gestionan la porción de interfaz de usuario de la aplicación, validan los datos introducidos por el usuario, realizan las solicitudes a los servidores y, a veces, ejecutan cierta lógica de negocio. Una petición que realiza el cliente tiene la siguiente estructura:

- Dirección de la ubicación del servidor al cual se le hace la petición.
- Información o recurso que se solicita.
- Dirección de ubicación del cliente que realiza la petición para poder enviar la respuesta o información solicitada.

Un servidor es un proceso que contesta a la solicitud del cliente realizando la tarea propuesta por éste. Los clientes gestionan recursos compartidos como ficheros, impresoras, enlaces de comunicación, bases de datos, etc. Proporciona tanto

almacenamiento de datos como capacidad de procesamiento. Los programas servidores deben tener la capacidad de trabajar con múltiples conexiones simultáneas de clientes diferentes.

Normalmente la arquitectura de dos capas se utiliza cuando se requiere mucho procesamiento de datos, pero cuando una aplicación implica un considerable procesamiento lo mejor es optar por una arquitectura de tres capas.

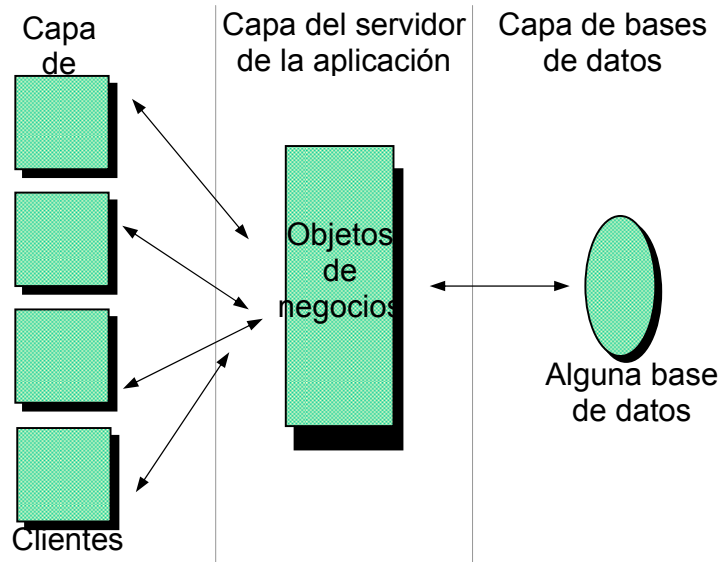
2.8.2. Múltiples capas

Este modelo es una adaptación del modelo DNA³¹ desarrollado por Microsoft Corporation y se basa en el concepto de que una aplicación distribuida debe estar separada lógicamente en tres partes o capas. La arquitectura de múltiples capas consiste en dividir la funcionalidad del sistema total en capas lógicas que pueden ser encapsuladas como componentes que interactúan entre ellos a alto nivel y supone una mayor escalabilidad de la aplicación, un mantenimiento menor y un incremento de la reutilización de componentes.

Las arquitecturas de múltiples capas, también son llamadas arquitecturas centradas en el servidor, debido a que solo permiten que la implementación de las reglas del negocio de una aplicación funcione en la capa intermedia del servidor, independiente tanto de la presentación como de la implementación de la base de datos

³¹ DNA, Arquitectura para Aplicaciones distribuidas en Internet, de Microsoft Corporation.

Figura 6 Arquitectura de Tres capas. Caso mas general de arquitectura multicapas..



El numero de capas típico que se utiliza en esta arquitectura es de tres; estas son:

- Capa de datos
- Capa de negocio
- Capa de presentación.

Así en la capa de "datos" se encapsulan todas las funciones de base de datos, las funciones de interfaz de usuario en la capa de "Presentación" y finalmente toda la lógica de operación en la capa de "Negocios".

Este encapsulamiento tiene como objeto ofrecer una colección mas simple de funciones que permiten desarrollar aplicaciones a alta velocidad y bajo costo. El costo de mantenimiento del sistema disminuye notablemente ya que una modificación en una capa no debe afectar a las demás.

La arquitectura por capas³² se especifica como:

- Desarrollar aplicaciones distribuidas de múltiples capas, escalables³³ y robustas.

³² Capas: Niveles de agregación que conforman un sistema construido bajo el enfoque DNA (Arquitectura para Aplicaciones distribuidas en Internet, de Microsoft Corporation).

- Extender datos existentes y aplicaciones externas para soportar el Internet.
- Soportar una amplia variedad de clientes, maximizando el alcance de las aplicaciones.

2.8.2.1. Capa de Datos

El nivel de servicios de datos es responsable de: Almacenar, recuperar y mantener los datos así como verificar la integridad de los mismos.

Los servicios de datos tienen una variedad de formas y tamaños, incluyendo los sistemas de administración de bases de datos relacionales, servidores de correo electrónico y sistemas de archivos.

2.8.2.2. Capa de Aplicación o de Negocios

La capa de aplicación es el “puente” entre un usuario, representado en la capa de presentación, y los servicios de datos, que se encuentran en la capa de datos. Los servicios de esta capa responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Esto aísla al usuario de la interacción directa con la base de datos. Esta capa es la que tiene capacidad de mantenimiento y de reutilización; puede contener objetos reutilizables definidos por clases que se pueden utilizar una y otra vez en otras aplicaciones. El lugar en donde puede residir esta capa depende de la decisión de diseño que se tome, puede residir en el mismo servidor que contenga la capa de datos o si así se requiere, dependiendo de factores como la carga que tiene un servidor en particular y la distancia a la que se encuentra de los clientes, esta puede residir en un servidor separado.

2.8.2.3. Capa de Presentación

Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios.

³³ La escalabilidad hace referencia a la capacidad de una aplicación de incrementar el número de usuarios sin que el desempeño de la misma se vea disminuido.

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. El cliente proporciona el contexto de presentación, generalmente un navegador como Microsoft Internet Explorer o Netscape, que permite ver los datos remotos a través de una capa de presentación HTML, u otro tipo de aplicación.

Mediante el uso de capas, se separa la programación que da acceso a los datos en las bases de datos de otros contenidos del sistema. Esto ayuda a asegurar que los desarrolladores estén libres para enfocarse en escribir su lógica de negocios en componentes sin preocuparse acerca de cómo se muestra la salida. Recíprocamente, esto da libertad a los diseñadores de usar herramientas familiares para modificar la interfaz.

2.9. APLICACIONES WEB

Los sistemas y aplicaciones basados en web hacen posible que una gran cantidad de usuarios finales puedan utilizar una amplia variedad de información y servicios. Estos sistemas presentan en su gran mayoría una serie de atributos que ayudan a caracterizarlos como son:

- *Intensivas de red.* Reside en una red y debe dar servicio a las necesidades de una comunidad diversa de clientes. Puede residir en Internet, se puede ubicar en una Intranet o una Extranet.
- *Controlada por el contenido.* En muchos casos la principal función de una aplicación web es utilizar hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y vídeo.
- *Evolución continua.* Las aplicaciones web están en constante evolución, diferente a otras aplicaciones de software convencional cuya evolución se lleva a cabo a través de versiones planificadas y un tiempo determinado entre cada una de ellas. Es posible que una aplicación web evolucione en cuestión de horas; pero esto no quiere decir que sea arbitraria y sin

fundamento para hacerla. Está es guiada por una serie de características que indican el momento en que deba actualizarse, tales como: **Inmediatez, Seguridad y Estética.**

2.9.1. Calidad de una aplicación web.

¿Como se percibe la calidad de una aplicación web?

Aunque todas las características generales de la calidad del software se aplican también a este tipo de aplicaciones, existen algunas que son más relevantes y que proporcionan una base útil para determinar la calidad del aplicación web, como se muestra a continuación en la tabla 2.1 - Requisitos de calidad.³⁴

Tabla 3. Requisitos de calidad.

Calidad de una aplicación web	Usabilidad	Capacidad de comprensión del sitio global. Servicios de ayuda y realimentación en línea. Capacidades estéticas y de interfaz. Servicios especiales.
	Funcionalidad	Capacidad de recuperación y búsqueda. Servicios de búsqueda y navegación. Servicios relacionados con el domino de la aplicación.
	Fiabilidad	Proceso correcto de enlace. Recuperación de errores. Validación y recuperación de la entrada del usuario.
	Eficiencia	Rendimiento del tiempo de respuesta. Velocidad de generación de páginas. Velocidad de generación de gráficos.
	Capacidad de mantenimiento	Facilidad de corrección. Adaptabilidad. Extensibilidad. Escalabilidad.

2.9.2. Tecnologías disponibles para el desarrollo³⁵

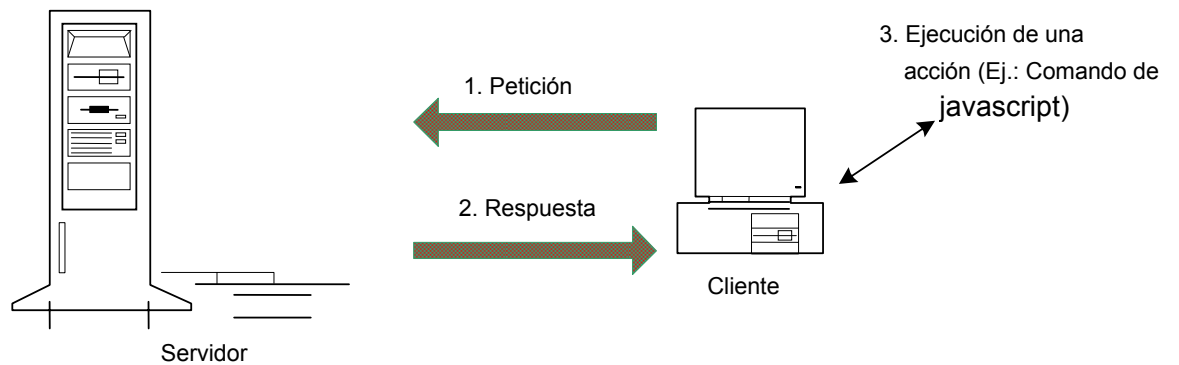
Para desarrollar aplicaciones y dotar a las páginas web de funcionalidad se puede trabajar tanto en el lado del cliente como en el lado del servidor, las variantes son:

³⁴ Olsina, L. Specifying Quality Characteristics and Attributes for Web Site. Los Angeles , Mayo de 1995.

³⁵ Tomado de : Mora Villamizar, Andrea Marcela, Sistema Intranet de Información para El Apoyo De La Actividad Académica en el Programa de Especialización En Docencia Universitaria Del Centro De Desarrollo Para La Docencia Universitaria-CEDEDUIS –SIEDUIS 1.0

2.9.2.1 Programación en el cliente:

Figura 7. Programación en el lado del cliente.

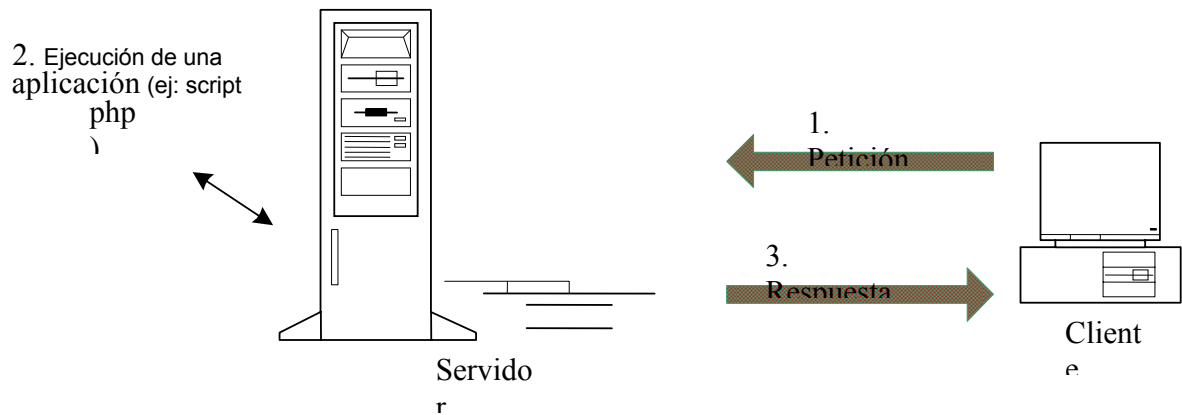


1. El navegador envía un request.
2. El servidor envía un response que contiene código que el navegador entiende.
3. El navegador interpreta el código enviado por el servidor y realiza una determinada acción.

2.9.2.2. Programación en el servidor:

1. El navegador envía un request.
2. El servidor ejecuta una aplicación que realiza una determinada acción.
3. El servidor envía el resultado de dicha aplicación al cliente.
4. El navegador muestra el resultado recibido del servidor.

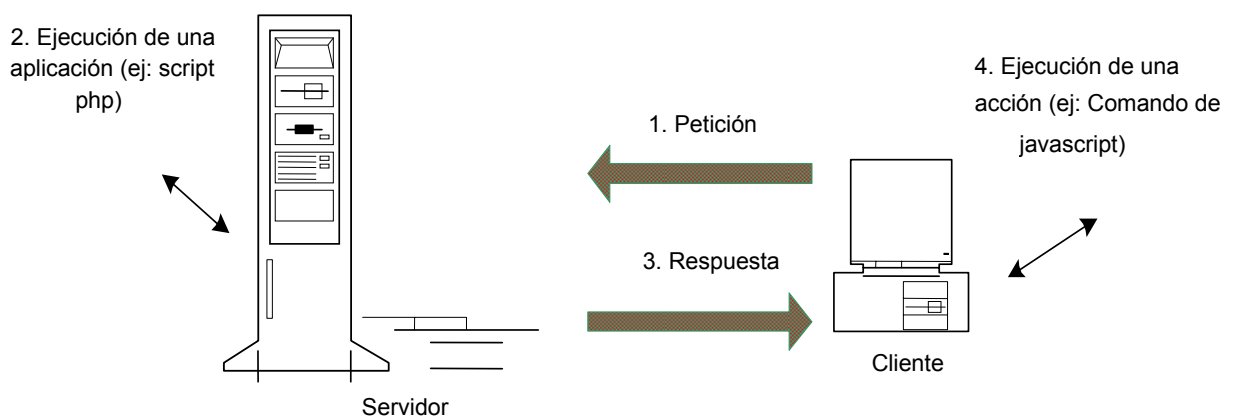
Figura 8. Programación en el lado del Servidor.



2.9.2.3. Esquema mixto: (programación en el cliente y en el servidor)

1. El navegador envía un request.
2. El servidor ejecuta una aplicación que realiza una determinada acción.
3. El servidor envía el resultado de dicha aplicación al cliente conteniendo código a interpretar por el navegador.
4. El navegador interpreta el código enviado por el servidor y realiza una determinada acción.

Figura 9. Esquema mixto. Programación en el cliente y en el servidor.



La programación del lado del cliente tiene como principal ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita recargar al servidor de

trabajo. El servidor sólo envía el código, y es tarea del navegador interpretarlo. La gran desventaja de esta metodología es que el código que el servidor envía es “sensible” a qué cosas puede o no hacer el navegador. El usuario puede, por ejemplo, decidir deshabilitar una funcionalidad del navegador que es necesaria para que se ejecute un determinado servicio o peor aún, navegadores distintos pueden interpretar el mismo código de distintas formas. Típicamente Netscape y Microsoft, que producen los dos navegadores más usados del mercado, no están de acuerdo sobre como se implementan diversas tecnologías en el cliente.

Programar del lado del servidor tiene como gran ventaja que cualquier cosa puede hacerse sin tener en cuenta el tipo de cliente, ya que la aplicación se ejecuta en el servidor que es un ambiente controlado. Una vez ejecutada la aplicación, el resultado que se envía al cliente puede estar en un formato “normalizado” que cualquier cliente puede mostrar. La desventaja reside en que el servidor se sobrecarga de trabajo ya que además de servir páginas, es responsable de ejecutar aplicaciones. A menudo esto redundante en requisitos de hardware mayores a medida que el servidor ejecuta más y más servicios.

Sin embargo, debido a las incompatibilidades existentes y a la posibilidad de que el usuario controle que cosas se ejecutan y cuales no, la programación del lado del cliente no es muy recomendable y debe limitarse a código altamente estándar que pueda interpretarse de cualquier forma en cualquier navegador, lo cual obliga a ejecutar la gran mayoría de las aplicaciones y servicios del lado del servidor.

2.9.3. Programación del lado del cliente

A continuación se presenta la tabla 4 que resume los principales lenguajes para la programación de los sitios web en el lado del cliente.

Tabla 4. Lenguajes del lado del Cliente.

Lenguaje	Descripción
HTML	Hypertext Markup Language. Lenguaje que se utiliza para la creación de páginas web. Consta de un conjunto de elementos, denominados etiquetas o marcas, que se utilizan para incluir texto, dibujos, tablas, cabeceras, etc.

DHTML	Extensiones de HTML que cubre el uso de hojas de estilo en cascada (CSS), uso de layers o capas para el posicionamiento y visibilidad de elementos en el navegador, programación con lenguajes de scripts y modelado de objetos de documentos (DOM).
JavaScript	Lenguaje interpretado incrustado en las paginas web para permitir ejecutar código en el cliente a través del manejo de objetos y la captura de eventos producidos en el navegador.
VBScript	Lenguaje interpretado similar a JavaScript proporcionado por Microsoft.
Applets de Java	Pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta como parte de un documento web.
ActiveX	Tecnología de Microsoft que permite ejecutar un programa que siga este estándar en los equipos de los usuarios al ser vinculado como objeto en una página web. Los programas ActiveX dependen del sistema operativo Windows, por lo tanto no funcionan fuera de este ambiente.

2.9.4. Programación del lado del Servidor

Para el desarrollo de aplicaciones del lado del servidor existen 3 grandes metodologías, utilizar el protocolo CGI, utilizar una API provista por el web-servidor o bien utilizar un “módulo” del web servidor.

2.9.4.1. El protocolo CGI:

El protocolo CGI (Common Gateway Interface) fue creado para establecer un protocolo estándar de comunicación entre el servidor web y cualquier lenguaje de programación de forma tal que desde el lenguaje “x” puedan recibirse datos que el usuario envía usando el método “POST” o “GET” y además el resultado de la aplicación sea enviado por el servidor al navegador. Típicamente para recibir datos se usa alguna biblioteca o módulo del lenguaje elegido que implementa el protocolo CGI y para enviar datos simplemente se envían a la salida estándar desde el lenguaje elegido y el servidor web se encarga de redireccionar esto al navegador.

De esta forma pueden realizarse aplicaciones para un sitio web en casi cualquier lenguaje. Los lenguajes interpretados rápidamente han ganado terreno ya que tienen un ciclo de desarrollo en tiempo inferior a los lenguajes compilados y son más fáciles de depurar dentro del ambiente CGI.

Los lenguajes no interpretados (C, C++) tienen como ventaja que requieren menos recursos del servidor al generarse el proceso CGI (no hace falta un intérprete) y además suelen ser mucho más veloces en su ejecución (no se necesita interpretar nada), sin embargo el desarrollar y depurar suelen ser tareas muy complejas y no siempre se justifica el esfuerzo si la aplicación es pequeña.

La desventaja de las aplicaciones CGI consiste en que el servidor debe realizar un proceso, y ejecutar la aplicación o bien el intérprete de la aplicación, y este ciclo, que se cumple cada vez que se ejecuta la aplicación, CGI consume muchos recursos y en general es costoso en tiempo para el servidor.

Uso de una API del servidor:

Otra técnica factible consiste en utilizar una API (Application Programming Interface) provista por el servidor web para desarrollar aplicaciones, es decir que el servidor provee un lenguaje en el cual se pueden desarrollar aplicaciones. Este esquema, como podemos apreciar, es mucho más eficiente que el anterior ya que el servidor web es el encargado de ejecutar las aplicaciones en forma directa sin necesidad de crear un proceso. Las desventajas son sin embargo importantes: En primer lugar las aplicaciones creadas en este marco no son portables ya que sólo pueden ejecutarse en un servidor web determinado, esto es una gran desventaja frente a las aplicaciones CGI que podían una vez desarrolladas ejecutarse en cualquier servidor. La segunda gran desventaja es que frecuentemente un error de programación de una aplicación podría ocasionar que el servidor deje de funcionar, genere un error, pierda memoria u otros problemas. Esto ocasiona que este tipo de aplicación no sea confiable.

2.9.4.2 Uso de un Módulo del Servidor Web

La tecnología más reciente para la ejecución de aplicaciones consiste en anexas a un servidor web “módulos” que le permitan interpretar un determinado lenguaje. De esta forma se logra eficiencia ya que el servidor no necesita crear un nuevo proceso por cada aplicación que ejecuta. Las aplicaciones son portables ya que son

desarrolladas en un lenguaje estándar que no depende del servidor web. Las aplicaciones son confiables ya que si bien pueden producir un error en el lenguaje en que están diseñadas, si el módulo es sólido, dichos errores no pueden comprometer al servidor.

En las siguientes tablas 4 y 5, se presenta un resumen de los principales lenguajes con una descripción de su uso en programación del lado del servidor así como una breve comparación de los mismos.

Tabla 5. Lenguajes del lado del servidor.

Lenguaje	Descripción
Perl	Practical Extraction and Report Language. Lenguaje interpretado creado con el objetivo principal de simplificar las tareas de administración de un sistema UNIX. Hoy en día se ha convertido en un lenguaje de propósito general.
Python	Es un lenguaje interpretado que permite escribir programas pequeños, utilizado en desarrollo web para la creación de CGI
C, C++	Utilizado para la creación de CGI
PHP	<i>PHP es un lenguaje interpretado diseñado para el desarrollo de sitios dinámicos. La distribución más popular de PHP es como módulo para el servidor Apache, aunque puede funcionar como un interprete para ejecutar aplicaciones CGI.</i>
ASP	Active Server Pages. Tecnología creada por Microsoft destinada a la creación de sitios web.
JSP	Java Server Pages (JSP). Es un lenguaje interpretado insertado en paginas web y basada en Java para el desarrollo de sitios dinámicos.
Mod_perl	Módulo de Perl para el servidor web
Mod_python	Módulo Python para el servidor web

Tabla 6. Comparación de lenguajes del lado del servidor.

	CGI (interpretado)	CGI (compilado)	API del servidor	Módulo del servidor
Ejemplos	Perl, Python	C, C++	Netscape Enterprise	PHP, ASP, JSP, Mod_perl, Mod_python, FastCGI
Tiempo de Desarrollo	Corto	Largo	Medio	Corto
Depuración	Sencilla	Compleja	Compleja	Sencilla

Confiabilidad	Alta	Alta	Baja	Alta
Eficiencia	Baja	Media	Alta	Alta

2.10. PHP (Hypertext Preprocessor)

El **PHP** inicio como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994. Su primer uso fue el de mantener un control sobre quien visitaba su curriculum en su web.

En los siguientes tres años, se fue convirtiendo en lo que se conoce como PHP/FI 2.0. Esta forma de programar llego a muchos usuarios, pero el lenguaje no tomo el peso actual hasta que Zeev Surasky y Andi Gutmans le incluyeron nuevas características en 1997, que dio por resultado el PHP 3.0. La versión 4 es la más reciente. Según estudios más de un millón de servidores tienen esta capacidad implementada y los números continúan creciendo.

Traduciendo la definición del FAQ de PHP.net: *"El PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de si mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de paginas"*.

Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes. Productos similares y propietarios son Active Server Pages (ASP) de Microsoft, ColdFusion de Allaire y Java Server Pages (JSP) de Sun.

PHP permite acceder unos veinte tipos de Bases de Datos distintas en diversos Sistemas Operativos y utilizando distintos servidores HTTP. Sus principales ventajas frente al resto de competidores en el sector, además del hecho de ser multiplataforma, son su ALTA velocidad de respuesta y que es "free software" bajo licencia GPL, esto es, gratuito y con permiso a modificar el código fuente original

(que está disponible). Además PHP ofrece de serie gran cantidad de características y añadidos (módulos) que en el resto de programas son extras que hay que obtener (comprar) a parte del paquete básico. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras.

Debido a su amplia distribución **PHP** esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, **PHP** goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de **PHP**. Es utilizado en aplicaciones Web-relacionadas por algunas de las organizaciones más prominentes tales como Mitsubishi, Redhat, Der Spiegel, MP3-Lycos, Ericsson y NASA.

PHP es fácil de aprender comparado con otros mecanismos para obtener la misma funcionalidad. A diferencia de JSP o CGI basados en C, PHP no requiere un conocimiento exhaustivo del lenguaje de programación. A diferencia de Perl, PHP tiene una sintaxis muy fácil de comprender y a diferencia de ASP, no requiere conocer más de un lenguaje de programación o de la instalación de módulos externos o comerciales para realizar tareas más complicadas no previstas en el lenguaje más usado (Visual Basic Script).

La mayoría de las funciones más útiles están predefinidas en PHP:

- Acceso a bases de datos: ODBC, Oracle, Postgres, SQL Server, MySQL, Informix, Interbase, SyBase, mSQL, dBase entre otros.
- Conectividad: HTTP, FTP, COM, YP/NIS, SNMP, Sockets, CORBA, LDAP.
- Servicios Correo y Noticias: POP, IMAP, SMTP, NNTP.
- Manejo de Textos y Gráficos: XML, HTML, PDF, GD, Flash.
- Funciones Matemáticas.
- POSIX: semáforos, memoria compartida, acceso a ficheros, expresiones regulares, cronómetros, etc.

- Comercio Electrónico: Cybercash, Verisign.
- Formularios.
- Encriptación y Compresión: MD5, Gzip, Bzip2, OpenSSL, etc.

2.11 Linux

Linux es un sistema operativo para equipos personales y estaciones de trabajo, que incorpora una interfaz gráfica de usuario (*GUI, Graphic User Interface*) como Windows y Mac, aunque más estable. Linux fue desarrollado a principio de la década de los noventa por el estudiante finlandés Linus Torvalds, junto con otros programadores de todo el mundo. Linux desempeña las mismas funciones que un sistema UNIX, Mac, Windows y Windows NT; sin embargo Linux se distingue entre estos por su potencia y flexibilidad.

Linux es una versión para PC de UNIX, el sistema operativo que ha sido utilizado por décadas en mainframes y minicomputadores y que en la actualidad es el sistema preferido para las estaciones de trabajo. Linux proporciona la velocidad, eficiencia y flexibilidad de UNIX a un PC, sacando partido de todas las funciones que los equipos personales en la actualidad pueden dar. A sus funciones de UNIX se le añaden potentes características de red, como por ejemplo, el soporte para Internet, intranets y redes Windows. Las diversas distribuciones de Linux integran servidores de Internet rápidos, eficientes, y estables, como servidores Web, FTP y Gopher, así como servidores de nombres de dominio, proxy, de noticias, de correo y servidores de indexación. Es decir, Linux ofrece todo lo necesario para instalar, brindar soporte y mantener una red completamente funcional.

Este sistema operativo corre en una variedad de plataformas, especialmente en computadores personales con procesadores Intel 80386 o superiores. Soporta un gran rango de software, desde el TeX³⁶ hasta el Sistema X Windows³⁷, desde el compilador GNU C/C++ hasta el protocolo TCP/IP. LINUX puede convertir cualquier PC 80386 o superior en una estación de trabajo que ofrece todo el poder de UNIX. LINUX es diferente porque es una implementación de UNIX gratuita, se distribuye gratuitamente bajo la licencia GPL (General Public License, Licencia Pública

³⁶ TeX : Sistema procesador de texto para plataformas UNIX / Linux, desarrollado por Donald Knuth.

General), de GNU, como especifica la Free Software Fundatition (Fundación de software Libre), de manera que queda a alcance de quien lo desea. Las siglas GNU procede de "Gnu`s not UNIX (GNU no es UNIX) y se trata de un proyecto iniciado y gestionadao por la Free Software Fundatition, para proporcionar software gratuito a los usuarios, desarrolladores y programadores.

Existen varias distribuciones de Linux, conservando todas el núcleo desarrollado por Torvalds, entre otras tenemos Open Linux, SUSE, Debian, Slackware, Mandrake y Red Hat Linux. Esta última publicada en el verano de 1994, por Red Hat Software³⁸, y aportó al Linux básico cosas como: soporte para más hardware, enormes avances en la seguridad, y un creciente uso de Linux por empresas de todo el mundo.

Red Hat Linux es fácil de administrar. Incluye un amplio conjunto de herramientas que reduce las molestias de la administración diaria del sistema.

Características de Linux

Entre las principales características de Linux encontramos:

Similitud con UNIX estándar. Linux posee muchas características presentes en varias implementaciones de UNIX, como por ejemplo un sistema operativo completamente multi-tareas y multi-usuario, esto significa que varios usuarios se pueden conectar a la misma máquina y correr programas simultáneamente. El ambiente de trabajo más común en UNIX también ha sido portado a Linux y todos los comandos estándar están incluidos. El sistema Linux es casi completamente compatible con varios estándares de UNIX a escala de fuente, incluyendo POSIX 1 de IEEE, Sistema UNIX V y la distribución del sistema UNIX de Berkeley. La mayoría de las utilidades básicas de Linux son software de GNU.

³⁷ Sistema manejador de una interface gráfica de ventanas para sistemas operativos derivados de UNIX

³⁸ Red Hat Software, <http://www.redhat.com>

Las utilidades GNU soportan características avanzadas que no se encuentran en versiones estándar de BSD y Sistemas UNIX V. Por ejemplo el clon de GNU de `vi`, `elvis`, incluye un macro lenguaje estructurado que difiere de su implementación original. Sin embargo las utilidades GNU tienden a ser compatibles con las diferentes versiones de UNIX. Inclusive, hay quienes consideran las versiones de GNU superiores a las originales.

Linux ofrece el completo ambiente de programación UNIX, que incluye las librerías estándar, herramientas de programación, compiladores y debuggers encontradas en UNIX. Estándares como POSIX 1 son soportados, lo que permite que el software escrito en Linux puede ser fácilmente portable a otros sistemas. Con Linux, se tiene acceso al conjunto completo de librerías y utilidades de programación y al código fuente del kernel y de las librerías. El compilador estándar de C y C++ para Linux es el gcc, el cual es un moderno y avanzado compilador que soporta C++, incluyendo las características AT&T 3.0, C-Objetivo y otros dialectos de C. Además de C y C++, otros programas compilados e interpretados han sido llevados a Linux, como el SmallTalk, FORTRAN, Java, Pascal, LISP, Scheme y Ada.

Otras características internas específicas de Linux, incluyen el control de trabajo POSIX (utilizado por shells³⁹ como csh y bash), pseudo - terminales (dispositivos pty), y soporte para manejadores de teclados cargables dinámicamente tanto nacionales como personalizados. También soporta las consolas virtuales que permiten moverse entre sesiones en la misma consola. De igual forma administra varios sistemas de archivos para almacenamiento de datos, como el sistema de archivos ext2, el cual fue desarrollado específicamente para Linux.

Compatibilidad del Sistema de Archivos

Linux soporta los sistemas de archivos del Sistema UNIX V y XENIX como el VFAT de Microsoft Windows95 y de MS-DOS en disco duro o disco flexible, así como el sistema de archivos ISO 9660 para CD-ROM.

³⁹ Programa que lee y ejecuta los comandos del usuario en el sistema operativo

Tipos de Shells

Existen varios tipos de shells disponibles para Linux. La diferencia más importante entre shells es el lenguaje de comandos. Por ejemplo, el Shell C (csh) utiliza un lenguaje de comandos similar al lenguaje de programación C. El clásico Shell Bourne (sh) utiliza otro lenguaje de comandos. La selección del shell, por lo general se basa en el lenguaje de comandos que ofrece, y determina las cualidades del ambiente de trabajo bajo Linux.

El Bourne Again Shell de GNU (bash) es una variante del Shell Bourne que incluye varias características avanzadas como control de trabajo, comandos históricos, finalización de comandos y nombres de archivo, y una interfaz estilo emacs para la edición de líneas de comandos, junto con otras poderosas extensiones del lenguaje del Shell Bourne estándar. Otro shell popular es el tcsh, una versión del shell C con funcionalidades avanzadas similares a las del bash. Entre otros shells, están el zsh, un pequeño shell estilo Bourne; el Shell Korn (ksh); BSD's ash; y rc.

El Kernel

El kernel de Linux está desarrollado para utilizar las características de modo de protección de los procesadores Intel 80386 y mejores. En particular, Linux utiliza el modo de protección, descriptor de base, paradigma de manejo de memoria y otras cualidades avanzadas. El kernel soporta la demanda por página de los ejecutables cargados. Solo aquellos programas que están en uso se reflejan en la memoria desde el disco duro. Si varias instancias de un programa están corriendo simultáneamente, comparten memoria física lo cual reduce la utilización total.

Para incrementar la cantidad de memoria disponible, Linux implementa la paginación del disco. Hasta 1 GB de espacio de intercambio (swap space) puede ser ubicado en el disco. Cuando el sistema necesita más memoria física, intercambia páginas inactivas con el disco, permitiendo el funcionamiento de aplicaciones grandes y soportando más usuarios. El kernel de Linux también implementa la pila de memoria unificada para programas de usuario y caché del disco. Toda la memoria disponible es usada por el caché, la cual es reducida cuando se corren programas grandes. Los ejecutables utilizan enlaces dinámicos, librerías compartidas. Los archivos ejecutables ocupan menos espacio en disco,

especialmente aquellos que utilizan varias funciones de librería. También existen librerías enlazadas estáticamente para objetos de pruebas y se mantiene un archivo binario "completo" cuando las librerías dinámicas no se encuentran instaladas. Las librerías son enlazadas dinámicamente al momento de ejecución, y el programador puede usar sus propias rutinas en lugar de las rutinas de librerías estándar.

Trabajo en red

Linux proporciona una implementación completa del software de red TCP/IP. Esto incluye los manejadores de dispositivos para muchas tarjetas Ethernet, SLIP (Serial Line Internet Protocol) y PPP (Point-to-Point Protocol) lo cual proporciona acceso a redes TCP/IP por medio de conexión serial, PLIP (Parallel Line Internet Protocol) y NFS (Network File System). De la misma manera, soporta los clientes y servicios TCP/IP como FTP, telnet, NNTP y SMTP.

Los servicios que ofrece la para la comunicación a través de una red podemos enumerar:

- ✓ NFS (Network File System) permite compartir sistemas de archivos con otras máquinas en la red.
- ✓ FTP (File Transfer Protocol) permite transferir archivos con otras máquinas.
- ✓ SENDMAIL envía y recibe correo electrónico vía protocolo SMTP.
- ✓ C-NEWS E INN con sistemas de noticias basados en NNTP; y
- ✓ Telnet, Rlogin, Y Rsh permiten acceso y ejecución de comandos desde otras máquinas en red. finger permite recuperar información de otros usuarios de Internet.

El sistema proporciona una interface de programación de socket estándar para UNIX. En general, cualquier programa que utilice TCP/IP puede ser llevado a Linux. El servidor X de Linux también soporta TCP/IP, y las aplicaciones que corren en otros sistemas pueden usar la presentación del sistema local.

2.12. Servidor Web Apache

La principal función de un servidor Web es responder rápidamente cuando un usuario solicite una página. Para cada transacción el servidor debe realizar dos acciones básicas: integrar todos los componentes de la página (texto, imágenes, vídeo, scripts CGI, etc.) y enviarla rápidamente al usuario. Como núcleo principal de la Intranet el servidor Web debe ser fácil de administrar.

Apache es el servidor web por defecto para casi todas las distribuciones Linux y está basado en HTTPd⁴⁰ de NCSA (*Centro Nacional de Aplicaciones de Supercomputación, de la Universidad de Illinois*). Apache está formado por una serie de parches incorporados al servidor NCSA, es decir, es un servidor parcheado. En general se ejecuta como el usuario 'nobody', lo cual le da muy poco acceso al sistema, adicionalmente proporciona un sistema de autenticación de usuario basado en ID⁴¹ de usuario y contraseña, y puede prohibir el acceso por nombres de dominio y direcciones IP⁴² específicas. En cuestión de seguridad, la mayoría de los servidores WWW simplemente toman datos del sistema y los transmiten, los mayores peligros no vienen de estos sino de programas que se ejecutan vía servidor.

2.13. Postgresql

Es un manejador de bases de datos. Se originó como un proyecto de investigación del profesor Michael Stonebraker en la Universidad de Berkeley. En esta primera versión se utilizó el lenguaje Postquel, que fue reemplazado posteriormente por el lenguaje estándar para el manejo de bases de datos SQL. La versión actual se conoce como Postgresql y es continuamente actualizada por un equipo de programadores coordinados a través de Internet por Marc G. Fournier.

Las principales características de Postgresql son:

- ✓ Relacional, incluye características como optimización de sentencias, control concurrente de transacciones y soporte multiusuario.
- ✓ Expandible, se pueden definir diferentes tipos de datos, operadores, funciones y métodos de acceso.

⁴⁰ HTTPd, Hypertext Transfer Protocol Daemon

⁴¹ ID, Nombre o número único que se utiliza para identificar al usuario.

- ✓ Utiliza SQL, permite implementar un subconjunto de instrucciones del estándar ANSI SQL.
- ✓ Portable, se pueden programar scripts para ser utilizados por la mayoría de lenguajes.
- ✓ Independiente de la plataforma, se puede utilizar con un sistema operativo como Linux, Solaris, SunOS, Digital Unix, FreeBSD, entre otros.

Postgres permite la extensión del lenguaje SQL ya que presenta: Funciones, Tipos, Operadores y agregados. Una diferencia clave entre Postgres y los sistemas relacionales tradicionales es que Postgres almacena mucha más información en sus catálogos - no solo información referente a tablas y columnas, sino también sus tipos, funciones, métodos de acceso y más. Estas clases pueden ser modificados por el usuario ya que Postgres basa su operación interna en estas clases, lo que significa que Postgres puede ser extendido por los usuarios.

Postgres es diferente a la mayoría de manejadores de datos ya que el servidor puede incorporar código escrito por el usuario en sí mismo utilizando carga dinámica. Es decir, el usuario puede especificar un código objeto (por ejemplo un archivo compilado o una librería compartida) que implemente un nuevo tipo o una función y Postgres la cargará a medida que sea necesaria. EL código escrito en SQL es más trivial de adicionar al servidor. Esta habilidad para modificar su operación "en el momento" hace de Postgres extraordinariamente adecuado para prototipados rápidos de nuevas aplicaciones y estructuras de almacenamiento.

⁴² Direcciones IP, Son direcciones de 32 bits utilizada por las redes TCP/IP para identificar las computadoras y las redes a las que están conectadas.

PARTE II

FASE DE INICIO

Durante la fase de inicio, se establece la planificación del proyecto y se delimita su alcance. La planificación del proyecto incluye los criterios de éxito, la evaluación de los riesgos, estimaciones de recursos que se necesitarán y un plan de flujos que muestre la planificación de los hitos principales.

El objetivo de la fase de inicio es establecer el análisis de negocio, el análisis para decidir si se sigue adelante con el proyecto. La fase de inicio no es un estudio completo del sistema propuesto, sino que en ella buscamos el porcentaje de casos de uso necesarios para fundamentar el análisis del negocio inicial, a través de los siguientes pasos:

- Delimitar el ámbito del sistema
- Describir una propuesta de la arquitectura del sistema.
- Identificar los riesgos críticos

Esta parte está conformada por el capítulo 3 y 4, el primero presenta la planeación de la fase y está relacionado con el conocimiento de la organización, el establecimiento de los criterios de evaluación y la descripción de las actividades y productos a obtener.

El segundo capítulo describe los flujos de trabajo fundamentales de acuerdo a la metodología, presentando los productos obtenidos a lo largo de cada uno de ellos y las actividades realizadas para la obtención de dichos productos. La mayor parte del trabajo de la fase de inicio se lleva a cabo en el primer flujo de trabajo, el de requisitos, seguido de algo de trabajo en los flujos de análisis y diseño. Hay poco trabajo que realizar en los flujos de implementación y pruebas.

Finalmente se hará un resumen que contendrá una revisión de los criterios de evaluación y los productos obtenidos, los cuales sirven de punto de partida para la próxima fase o por el contrario abandonar el proyecto.

CAPITULO 3

PLANEACION Y ESTABLECIMIENTOS DE CRITERIOS DE EVALUACIÓN EN LA FASE DE INICIO

El Esfuerzo involucrado en esta fase esta directamente relacionado con el conocimiento de la organización y de los requisitos candidatos para el nuevo sistema.

En este caso se posee un buen conocimiento de las actividades realizadas en La Personería Distrital de Cartagena y en particular en lo que respecta a la gestión de las quejas. Sin embargo al igual que en todo proyecto, se debe realizar un análisis de viabilidad ya que si existen riesgos que deben ser mitigados o eliminados oportunamente. Este capitulo muestra la planeación necesaria para el desarrollo de esta fase.

3.1. CRITERIOS DE EVALUACIÓN

El objetivo principal de la fase de inicio es dar respuesta a la pregunta: *¿Es el caso de negocios inicial lo suficientemente bueno para justificar que sigamos hacia delante con el proyecto?*

Para determinar cuando se ha cumplido cada uno de los cuatro objetivos de esta fase se definieron los siguientes criterios que permitan su evaluación:

Objetivo 1: Decidir el ámbito del sistema

- ¿Está Claro lo que va a formar parte del sistema?
- ¿Se han identificado todos los actores externos?
- ¿Se ha expuesto la naturaleza general de las interfaces con los actores? (interfaces con usuarios y protocolos de comunicación)
- ¿Lo que esta incluido en el ámbito puede constituir por si mismo un sistema que funcione?

Objetivo 2: Resolver ambigüedades en los requisitos necesarios en esta fase

- ¿Se han identificado y detallado los requisitos funcionales, no funcionales y adicionales del limitado número de casos de uso necesarios para el objetivo de esta fase?

Objetivo 3: Determinar una arquitectura candidata

- ¿Satisface esta arquitectura las necesidades de los usuarios?
- ¿Es verosímil que funcione? (Promesa de funcionar)
- ¿Puede utilizar en forma apropiada la tecnología sobre la que será construida?
- ¿Puede ser eficiente?
- ¿Puede explotar los recursos existentes?
- ¿Es flexible y tolerante a fallos?
- ¿Es robusta y flexible al cambio?
- ¿Evolucionará fácilmente si se añaden requisitos?

Objetivo 4: Mitigar los riesgos críticos

- ¿Se han identificado todos los riesgos críticos?
- ¿Se han mitigado los identificados o hay un plan para mitigarlos?

3.2. PRODUCTOS A ENTREGAR

La fase de inicio tiene como resultado los siguientes productos:

- Una lista de características.
- Una primera versión del modelo del negocio que describe el contexto del sistema.
- Un esbozo de los modelos que representan una primera versión del modelo de casos de uso, el modelo de análisis y el modelo de diseño. Respecto al modelo de implementación y el modelo de pruebas, puede existir algo rudimentario.
- Una descripción de la arquitectura candidata, a través de las vistas de los modelos de casos de uso, análisis, diseño e implementación.
- Una lista inicial de riesgos.
- Un borrador del plan para el proyecto en su totalidad.

CAPITULO 4

VIABILIDAD DEL PROYECTO

4.1. Introducción

La mayor parte del trabajo de la fase de inicio se lleva a cabo en el primer flujo de trabajo, el de requisitos, seguido de algo de trabajo en los flujos de análisis y diseño. Hay poco trabajo que realizar en los flujos de implementación y pruebas. En forma general se identificarán los casos de uso y actores que definen el ámbito del sistema, se establecen las prioridades entre los casos de uso, seleccionando aquellos importantes para establecer una arquitectura candidata, luego se describen y detallan algunos de ellos, dependiendo de su relevancia para identificar los riesgos críticos, en este punto tendremos una primera versión del modelo de casos de uso. Luego en el análisis se analizará la arquitectura y los casos de uso, obteniendo un modelo de análisis que sirve de base para la primera aproximación del modelo de diseño.

4.2. REQUISITOS

El propósito fundamental de este flujo es guiar el desarrollo hacia el sistema correcto, realizando una descripción de los requisitos del sistema que permita llegar a un acuerdo entre los clientes y desarrolladores sobre qué debe y qué no debe hacer el sistema.

Para capturar los requisitos de manera eficaz se emplean un conjunto de técnicas y artefactos que nos ayudan a obtener una visión suficientemente buena del sistema para avanzar en los flujos de trabajo subsiguientes. Técnicas que van desde enumerar requisitos candidatos, comprender el contexto del sistema, capturar los requisitos funcionales, hasta capturar los requisitos no funcionales y con las cuales obtenemos un conjunto de artefactos denominado conjunto de requisitos, formado por:

- Lista de características

- Modelo del negocio
- Modelo de casos de uso
- Lista de requisitos adicionales.

A continuación se presenta el conjunto de requisitos comenzando por una lista de características candidatas, el modelo del negocio que nos permite definir el contexto del sistema, una lista de riesgos críticos, el modelo de casos de uso y una lista de requisitos adicionales.

4.2.1. Lista de características

En esta sección presentamos una lista de ideas que consideramos como un conjunto de requisitos candidatos que podemos decidir implementar en esta versión del sistema.

Cada características tiene un nombre corto y una breve definición suficiente para poder hablar de ella durante la planificación de este proyecto, también un conjunto de valores de planificación que son:

- *Estado*, puede ser Propuesto, Aprobado, Incluido o Validado.
- *Prioridad*, puede ser Crítico, Importante o Secundario.
- *Nivel de riesgo*, puede ser Crítico, Significativo u Ordinario.

Estos requisitos están agrupados en características generales del sistema con relación a la función que desempeñan. Así por ejemplo Diligenciar Queja por Internet y Radicación de Correspondencia están agrupadas en Recepción de Quejas.

En la siguiente Tabla se muestran la lista de características candidatas del sistema.

**	NOMBRE	DEFINICION	ESTADO	PRIORIDAD	NIVEL DE RIESGO
	Recepción de Quejas	Recepción de quejas, solicitudes e inquietudes que llegan a la Personería para su posterior asignación a los funcionarios correspondientes.	Aprobado	Importante	Significativo
	Diligenciar una queja interpuesta personalmente	Se diligencia un formato con información sobre el quejoso, contra quien se interpone, asunto y personero delegado encargado de dar solución. Además se le asigna un número para radicar la misma (número de radicación)	Aprobado	Importante	Significativo
	Diligenciar una queja por Internet	Se diligencia un formato a través del sitio web de la entidad, con información sobre el quejoso, contra quien se interpone el asunto y datos sobre la denuncia; luego en la dependencia de quejas y reclamos se asigna al personero delgado encargado de dar solución.	Aprobado	Crítico	Ordinario
	Seguimiento y Control de Quejas	Conocer la evolución de un proceso y evitar el vencimiento de términos fijados por la ley.	Aprobado	Importante	Significativo
	Consultar Datos Generales del Proceso	Se podrá obtener información sobre el personero encargado del proceso, datos del quejoso, estado del proceso, que permiten conocer la evolución del caso, así como también la consulta de documentos relacionados con el mismo.	Aprobado	Importante	Significativo
	Agenda de Actividades	Los Personeros en el desarrollo de sus procesos necesitan practicar visitas ordinarias y extraordinarias en las diferentes oficinas del orden Distrital sobre las cuales se ejerce la vigilancia, para lo cual se hace necesario contar con una agenda que permita una distribución adecuada del tiempo para cumplir con las tareas asignadas, evitando interferencias y conflictos de programación.	Aprobado	Importante	Significativo
	Ingreso de Actividades, prioridades y disponibilidad horaria	Al ingresar las actividades se pueden establecer unas prioridades (prioridad entre actividades), las cuales permitan junto con la disponibilidad horaria del personero, una optima programación de la agenda y así evitar el conflicto entre las actividades.	Aprobado	Importante	Ordinario
	Publicación de Información	Publicación de información para los funcionarios de la Personería Distrital y para la comunidad en general.	Aprobado	Crítico	Significativo
	Seguimiento de la Fase I de la directiva.	Publicar y divulgar a través de un sitio web, información de interés general para la comunidad cumpliendo con lo estipulado en las Fases I y II de la directiva presidencial 02 de Agosto 28 del 2000 y que aplique a la Personería.	Aprobado	Crítico	Significativo
	Publicación de eventos y noticias	Facilitar la publicación de los eventos y actividades que se realicen al interior de la personería, así como también, la publicación en el sitio web de información generada por la oficina de divulgación y prensa para la ciudadanía en general.	Aprobado	Crítico	Significativo
	Publicación de Boletín informativo	En este Boletín, la comunidad a través de Internet podrá conocer el resultado de los procesos, teniendo en cuenta para esto, el tipo de información que se puede suministrar bajo criterios establecidos por la Personería Distrital.	Propuesto	Crítico	Significativo
	Elaboración y Trámites de Documentos	Elaboración de los documentos involucrados en la gestión de las	Aprobado	Importante	Significativo

		personerías delegadas, así como el trámite automático de los mismos.			
	Generación de Informes	Elaborar informes de los procesos a su cargo o de labores realizadas, que en forma mensual o previa solicitud; deben rendir los diferentes personeros delgados y el personero Distrital.	Aprobado	Importante	Ordinario
	Documentación interna	Elaborar documentos como Autos Comisorios, Oficios y Citaciones.	Aprobado	Importante	Ordinario
	Documentos legales	Elaborar documentos como Tutelas, Demandas, Denuncias, Recursos de apelación, Derecho de Petición, es decir, algunos documentos específicos que en el desarrollo de los procesos legales utilizan las personerías delegadas.	Aprobado	Importante	Ordinario
	Mecanismos de Intercambio de Información	Estos mecanismos propiciarán una comunicación efectiva entre los miembros de la entidad, mejorando el trabajo y actividades en grupo.	Aprobado	Importante	Significativo
	Correo electrónico	Mantener una comunicación permanente con los miembros de la entidad de manera asincrónica (sin coincidencia en lugar y tiempo). Se podrá enviar, recibir y archivar mensajes y adjuntar archivos con un mensaje.	Aprobado	Crítico	Significativo
	Transferencia de archivos	Propiciar el intercambio de información y que por su tamaño supera el admitido por un servidor de correo, impidiendo pueda ser enviada como e-mail (correo electrónico).	Aprobado	Secundario	Ordinario
	Discusión en grupo	Son espacios de encuentro a través del cual se pueden compartir e intercambiar ideas entorno a un tema planteado y finalmente construir en conjunto las conclusiones.	Aprobado	Importante	Crítico
	Mecanismos de contacto con la comunidad	A través del sitio web la ciudadanía podrá mediante votaciones, o sondeos de opinión, tener un mayor acercamiento con la Personería; confrontando sobre temas de mutuo interés .	Aprobado	Crítico	Significativo
	ADMINISTRACIÓN	El sistema permitirá ejercer control sobre el acceso de los usuarios, la seguridad de los datos y el movimiento de la información en la red.			
	Administración de cuentas de usuario	El sistema está diseñado para permitir cumplir con las labores de crear usuario, modificar cuentas de usuario y borrar un usuario.	Aprobado	Crítico	Ordinario
	Mantenimiento del contenido de las páginas y archivos	Búsqueda e inclusión de información al sistema en colaboración con la persona responsable de la información que se va a publicar, según los parámetros y criterios de contenidos que se manejan, actualizando las páginas y revisando o editando los hipervínculos a éstas, además se debe organizar los directorios, crear nuevos subdirectorios o eliminar los archivos no utilizados, de manera que los usuarios puedan acceder a la información sin ningún problema.	Aprobado	Crítico	Significativo
	Administración de eventos y noticias	El administrador podrá publicar, modificar o borrar un evento.	Aprobado	Crítico	Significativo
	Planeación y coordinación de las actividades grupales de discusión	Permitir la creación y eliminación de espacios de discusión, jugando un papel importante si se quiere en la planeación,	Propuesto	Crítico	Significativo

		moderación y gestión de las respuestas.			
	REQUISITOS ADICIONALES	Son los requisitos no funcionales, los cuales especifican propiedades del sistema en cuanto a:			
	Acceso al sistema	Se lleva a cabo mediante la autenticación de nombre y contraseña que le fueron asignados al usuario para poder ingresar al sistema.	Aprobado	Crítico	Significativo
	Restricciones de desarrollo e implementación	El sistema se desarrollará con mínimos costos, usando además para su Implementación herramientas que cuenten con licencias GNU, presentando una solución bastante económica , sin que ello implique pérdida de robustez o seguridad del sistema.	Aprobado	Crítico	Significativo
	Restricciones de diseño	El diseño del sistema se basa en la arquitectura multicapas (capas separadas y clasificadas por su funcionalidad), lo cual representa mantenimiento y administración sencilla del mismo.	Aprobado	Crítico	Significativo
	Restricciones de plataforma hardware	Servidor: PC Pentium III mínimo 128 RAM Clientes: PC mínimo Intel 486	Aprobado	Crítico	Crítico
	Restricciones de plataforma software	Software del Servidor: Sistema Operativo: Linux RedHat. Servidor Web: Apache. Software del Cliente: Internet Explorer 4.0 o mayor.	Aprobado	Crítico	Significativo
	Requisitos Normativos	Directiva presidencial 02 de Agosto 28 del 2000 – Estrategia de gobierno en línea.	Aprobado	Importante	Significativo

Tabla 7. Lista de características

4.2.2. Contexto del sistema

Definimos el contexto del sistema con el fin de limitar el alcance (ámbito) del sistema propuesto, necesitamos hacer esto para tener una mayor claridad sobre lo que debe cubrir el sistema, para comprender que debe cubrir la arquitectura, como también para definir los límites dentro de los cuales debemos buscar riesgos críticos.

Hay por lo menos dos aproximaciones para expresar el contexto de un sistema en una forma utilizable para desarrolladores de software: modelado del dominio y modelado del negocio. Un modelado del dominio describe los conceptos importantes del contexto como objetos del dominio, el identificar y asignar un nombre a estos objetos nos ayuda a desarrollar un glosario de términos que permitirá comunicarse mejor a todos los que están trabajando en el sistema.

El objetivo del modelado del negocio es describir los procesos, con el fin de comprenderlos. En este caso solo se modelan los procesos, de forma general, que soportará el sistema. En este modelo se establecen los trabajadores, responsabilidades y operaciones que llevan a cabo.

A continuación se muestra la misión, visión de la Personería Distrital de Cartagena, identificando luego el proceso fundamental que se lleva a cabo (atención de quejas) que es importante para la comprensión del contexto del sistema. Por último se desarrollará el modelo del negocio a través de los modelos de casos de uso y de objetos.

4.2.3. Misión y visión de la Personería Distrital de Cartagena

Visión: “La Personería Distrital de Cartagena, es una institución moderna, con personal altamente calificado que opera en la defensa de los Derechos Humanos y los intereses públicos que responde de manera eficaz y eficiente a las comunidades actuando con celeridad, responsabilidad y objetividad.”

Misión: “Velar por los Derechos de todos, mediante la guardia de los bienes del Distrito, la Vigilancia del cumplimiento de los Derechos de la Comunidad para que estos no sean vulnerados, la debida ejecución de la ley y la vigilancia de la

conducta oficial de quienes ejercen funciones publicas.”

La Personería Distrital de Cartagena de Indias está conformada por Personerías Delegadas. En la Tabla 8 se listan las diferentes Personerías Delegadas y la función que cumple cada una de ellas.

Tabla 8. Personerías Delegadas y función.

Personería delegada	Función
Personería delegada en lo civil	Interviene en todos los procesos civiles, laborales y administrativos en que sea parte el municipio o coadyuvando sus pretensiones de defensa y guarda de los bienes municipales.
Personería delegada para asuntos de la comunidad	Sirve de enlace entre el Personero Distrital y la comunidad y para ello se preocupará de promover la participación ciudadana en todos los asuntos de la administración; impulsar la organización popular y gremial para el fomento del desarrollo municipal.
Personería delegada en lo penal	Encargado de todos los aspectos legales que en materia penal tengan que ver con la Personería. Como la emisión de conceptos jurídicos relacionados con la entidad y la intervención en los procesos penales que cursen en los juzgados y la Fiscalía de la ciudad.
Personería delegada para la vigilancia y contratación administrativa	Vigila la conducta de todos los empleados de las dependencias distritales y promueve que se les exija la responsabilidad por faltas administrativas que cometan, con excepción del Alcalde Distrital. Adelanta investigaciones disciplinarias cuando se tenga certeza de la anuencia de hechos irregulares y formular los cargos contra el posible o los posibles responsables.
Personería delegada para la vigilancia de las tarifas de los servicios públicos	Interviene ante las entidades que prestan los servicios públicos para buscar soluciones de los problemas y quejas instauradas por los usuarios inconformes con la prestación de dichos servicios con el fin de coadyuvar en las reclamaciones ante las entidades que suministran dicho servicio.
Personería delegada para el control del desarrollo urbano, medio ambiente, vigilancia policiva y bienes distritales	Solicita a las autoridades competentes la vigilancia y cumplimiento de las disposiciones legales que regulan las intervenciones a los inmuebles, especialmente en la conservación de fachadas, construcciones, remodelaciones, demoliciones y embellecimiento del entorno urbanístico del distrito de Cartagena. Realiza actividades que permitan el desarrollo y la aplicación de conocimientos profesionales reconocidos por la ley en la defensa y protección del medio ambiente y los recursos naturales renovables, además ejerce las funciones de Ministerio Público en todos los procesos que se ventilen ante la inspección de policía; previene contravenciones; además de coadyuvar al mantenimiento del orden público y colaborar en la defensa de quienes carecen de recursos económicos para ello.
Personería delegada para la promoción, protección y defensa de los derechos humanos, tutela y acciones jurisdiccionales	Adelanta investigaciones delegadas por el Personero Distrital por las violaciones de los derechos fundamentales que establece la Constitución Nacional, recepción de quejas y reclamos que cualquier individuo o institución le hagan llegar, referente a la violación por parte de funcionarios del estado o por agentes ajenos al gobierno de los derechos civiles, políticos y de las garantías sociales. Promover la acción jurisdiccional en los casos que exista fundamento para ello, interviniendo en la práctica de todas las diligencias en forma activa, interrogando testigos, formulando cuestionarios a las partes, participando en las inspecciones judiciales, reconocimiento en fila de personas y demás actos que a juicio de funcionario deben ejecutarse.

El servicio fundamental que contribuye al cumplimiento de la misión de la

Personería es la atención de quejas. Las quejas pueden ser interpuestas por la comunidad a través de una oficina de quejas y reclamos de la Personería Distrital donde se reciben directamente de manera verbal. Otra manera es enviándola al Personero Distrital a través de correo ordinario. El proceso de atención de quejas lo podemos describir como un conjunto de actividades que se realizan cuando se interpone una queja, así:

- Se asigna la Personería Delegada a la cual le compete solucionar la queja interpuesta.
- Se elaboran una serie de documentos que permitan la ejecución de acciones legales en pro de dar solución a dicha queja.
- Se interponen, acciones populares, acciones de cumplimiento, tutelas o derechos de petición, si es necesario, para compensar los daños o perjuicios causados a la comunidad o al Distrito.
- Se publica y divulga información de interés para la comunidad sobre hechos, acciones y resultados de investigaciones adelantadas por la Personería Distrital.

Estas actividades descritas, nos permiten comprender el contexto en el que se desarrollará el sistema.

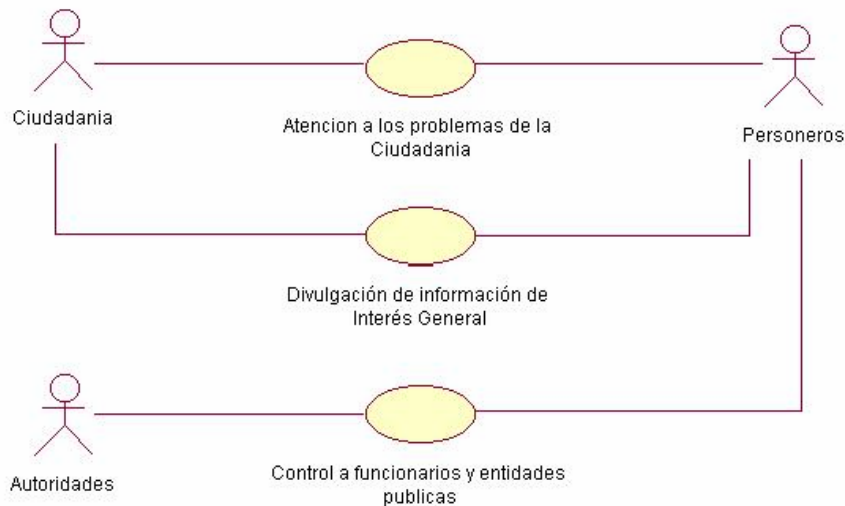
4.2.4. Modelo De Negocio

De la Misión y Visión de la P.D.C y de la descripción hecha del proceso fundamental que se lleva a cabo en ella (Atención de Quejas), se ha determinado el siguiente Modelo de Negocio.

El modelado del negocio está soportado por dos tipos de modelos de UML: Modelos de casos de Uso y Modelo de Objetos.

4.2.4.1. Modelo De Casos De Uso De Negocio

Figura 10. Modelo de Casos de Uso de Negocios



Cuando hablamos de un modelo de negocio, cada caso de uso corresponde a un proceso de negocio por lo tanto en este punto utilizaremos indistintamente ambos nombres.

Para la descripción de los casos de usos de negocio de la P. D. C. Utilizamos una tabla que contiene los siguientes campos:

- **Nombre** del caso de Uso de Negocio.
- **Objetivo** que se intenta alcanzar.
- **Acciones** que se realizan en el proceso de negocio.
- **Actor del negocio**, corresponden a los trabajadores o individuos identificados que participan en el proceso de negocio. Estos son:
 - **Autoridades:** Son todas las autoridades judiciales, civiles y militares que ayudan a la Personería en el cumplimiento de su función de vigilancia administrativa.
 - **Personeros:** Representa al Personero Distrital y a las Personerías Delegadas encargadas del funcionamiento básico de la institución y que hacen parte de ella.

- **Comunidad:** Representa cualquier individuo o institución que interpone una queja ante la entidad.

Tabla 9. Casos de uso del negocio

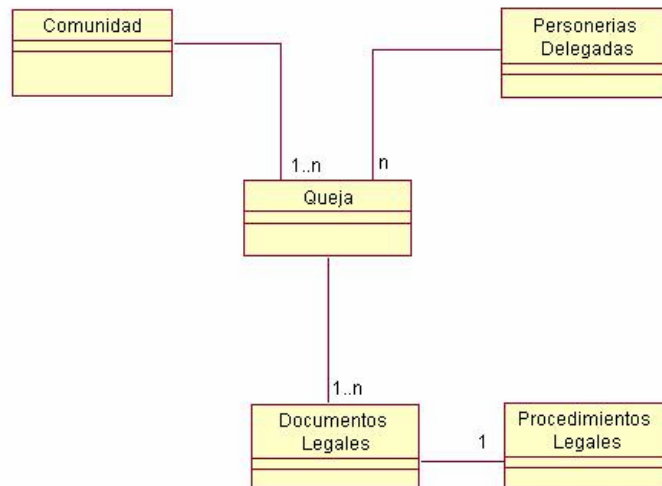
PROCESO DE NEGOCIO	OBJETIVO	ACCIONES	ACTOR
Atender los problemas de la Ciudadanía	Proteger y velar los derechos e intereses de la sociedad, así como defender el patrimonio público y los bienes del distrito.	<ol style="list-style-type: none"> 1. Recepcionar la queja. 1. Asignar la Personería Delegada competente. 1. Adelantar las investigaciones e interponer las acciones legales pertinentes para la solución de la queja. 	<ul style="list-style-type: none"> • Personeros • Comunidad
Divulgar información de Interés General	Publicar y divulgar información de interés para la ciudadanía a través de los medios de comunicación masivos.	<ol style="list-style-type: none"> 2. Recopilar información adecuada. 3. Elaborar un Boletín Informativo. 4. Distribuir la información a los medios para su promoción y divulgación. 	<ul style="list-style-type: none"> • Personeros • Comunidad
Controlar a funcionarios y entidades publicas	Vigilar la conducta oficial de quienes desempeñan funciones públicas municipales.	<ul style="list-style-type: none"> • Adelantar investigación disciplinaria cuando se tenga certeza de la aprobación de hechos irregulares, de acuerdo a procedimientos establecidos por la Procuraduría General de la Nación. • Formular cargos contra el posible o los posibles responsables. • Citar y hacer comparecer a las personas ajenas a la administración que deban colaborar dentro de la investigación. • Interponer ante la autoridad correspondiente las acciones legales pertinentes. 	<ul style="list-style-type: none"> • Personeros • Autoridades

4.2.4.2. Modelo de Objetos del negocio.

Describe como cada caso de uso del negocio es realizado por un conjunto de usuarios que utilizan un conjunto de entidades del negocio y de unidades de trabajo. Una entidad del negocio representa algo como un documento que puede ser manipulado, producido o utilizado en un caso de uso y una unidad de trabajo es un conjunto de estas entidades.

En la Figura 11 se ilustra el modelo de objetos del negocio para la Personería Distrital de Cartagena.

Figura 11. Modelo de Objetos



4.2.5. Riesgos Críticos

En esta fase se deben identificar los principales riesgos que podrían poner en peligro el éxito del proyecto, es decir, aquellos que determinarán si seremos capaces de construir el sistema o no. Estos riesgos críticos deben ser mitigados con el fin de reducir la probabilidad del riesgo, o para minimizar su gravedad si ocurriese.

Todo lo que es preciso saber sobre un riesgo para poder trabajar con él se encuentra en la siguiente lista de riesgos; está ordenada por nivel de prioridad y por influencia en el desarrollo. Además se presenta la forma en que serán mitigados sus efectos durante el desarrollo del sistema. Esta lista de los riesgos críticos incluye, Ver Tabla 10.

Tabla 10. Contenido de la lista de riesgos críticos

CONTENIDO LISTA DE RIESGOS CRÍTICOS	
<i>Descripción</i>	Comienza con una breve descripción a la cual se le van añadiendo detalles.
<i>Prioridad</i>	Se le asigna una prioridad al riesgo las cuales pueden ser; Crítico, significativo o rutinario.
<i>Impacto</i>	Indica qué partes del proyecto o del sistema se verán afectadas por el riesgo.
<i>Monitor</i>	Indica quien es responsable del seguimiento de un riesgo persistente.
<i>Responsabilidad</i>	Indica que individuo o unidad de la organización es responsable de eliminar el riesgo.
<i>Contingencia</i>	Indica lo que ha de hacerse en caso de que el riesgo se materialice.

Tabla

11.

Riesgos

críticos

DESCRIPCION	NIVEL DE RIESGO	IMPACTO	MONITOR	RESPONSABILIDAD	CONTINGENCIA
Sipedcar es un sistema basado en tecnología Internet que requiere una infraestructura de red mínima que corresponda a las necesidades de implantación de este tipo de tecnología; en la actualidad la Personería Distrital de Cartagena no cumple con este requisito.	Significativo	Muchos servicios del sistema no serán habilitados, ya que no se cuenta con la infraestructura necesaria para la implantación del sistema.	Personería Distrital de Cartagena e Ingenieros de desarrollo	Personería Distrital de Cartagena.	Este proyecto abarca hasta la fase de construcción, la fase de transición será contemplada para una segunda etapa mientras se estudia la compra de los equipos adecuados. Sin embargo, en este proyecto se dejará descrito las consideraciones sobre el Hardware requerido para la implantación del sistema.
No se cuenta con un registro de dominio, que permita acceso al sitio Web de la entidad a través de Internet.	Critico	Los usuarios (comunidad en general), no podrán interponer quejas o informarse sobre la labor que realiza la personería a través de Internet; además no se podrá dar cumplimiento a algunos de los objetivos planteados en las Fases I y II de la Directiva Presidencial 02.	Personería Distrital de Cartagena e Ingenieros de desarrollo	Personería Distrital de Cartagena	Se adelantará la gestión correspondiente para la consecución de un servicio de Web Hosting mientras se adecua la infraestructura interna.

4.2.6. Modelo De Casos De Uso

Este modelo nos da una primera idea del funcionamiento del sistema, ya que se capturan los requisitos funcionales, los cuales nos sirven para llegar a un acuerdo con los clientes. En esta fase se plantea una primera descripción de este modelo, el cual está compuesto por actores, casos de uso y sus relaciones.

4.2.6.1. Actores

Son todos los agentes que interactúan con el sistema, tanto para operarlo como de mantenimiento, incluyendo otros sistemas externos.

Para identificar estos actores partimos del modelo de negocio descrito anteriormente, ya que muchos actores del sistema se corresponden con trabajadores del negocio.

En la Tabla 12 se encuentran los actores identificados, describiendo el papel de cada uno en el proceso del negocio (responsabilidad) y para qué utilizará el sistema (necesidad)

Tabla 12. Actores del sistema

ACTOR	RESPONSABILIDAD	NECESIDAD
Personero Distrital	Es la persona encargada de ejercer en el municipio, bajo la dirección suprema del Procurador General de la nación, las funciones de Ministerio Público. Defender los intereses de la sociedad, el patrimonio público y el ambiente. Divulgar los derechos humanos y vigilar la conducta de los funcionarios que desempeñen cargos públicos.	<ul style="list-style-type: none"> • Hacer seguimiento y control a los procesos. • Elaborar documentos legales. • Programar las actividades. • Divulgar información de interés, eventos y noticias. • Participar en espacios de discusión. • Enviar y recibir correo electrónico. • Revisar informes de los Personeros Delegados.
Personero Delegado	Es la persona encargada hacer cumplir las funciones de la Personería en un área de trabajo específica. Puede ser en lo penal, civil, policial, familia, etc. Son los colaboradores directos del Personero Distrital en la solución de las quejas que llegan a la entidad.	<ul style="list-style-type: none"> • Hacer seguimiento y control a los procesos. • Elaborar documentos legales. • Programar las actividades. • Elaborar informes y enviarlos al Personero Distrital. • Consultar los eventos y noticias que se producen en la entidad. • Participar en espacios de discusión.
Visitante	Representa un individuo o institución que interpone una queja a través del sitio web de la entidad. Tiene la responsabilidad al poner la queja de suministrar todos los datos y elementos necesarios para que la personería pueda	<ul style="list-style-type: none"> • Acceder a información de interés. • Interponer una queja. • Conocer el estado de la queja • Opinar sobre temas de interés mediante votaciones o sondeo de opinión.

	realizar su trabajo satisfactoriamente.	<ul style="list-style-type: none"> • Consultar el Boletín de la Personería.
Abogado de reparto	Es la persona encargada de recibir la declaración verbal de la queja, transcribirla, y luego enviarla a la Personería Delegada que en su concepto corresponde darle solución.	<ul style="list-style-type: none"> • Diligenciar la queja (incluyendo las que llegan a través del sitio web) de una manera más ágil y eficaz. • Enviar la queja a través de la red a la Personera Delegada correspondiente. • Mantenerse al tanto de los eventos y noticias que se producen en la entidad.
Secretaria	Recibir y radicar la documentación que ingrese y salga de la dependencia. Proporcionar la información requerida siempre que no tenga carácter reservado.	<ul style="list-style-type: none"> • Elaborar documentos internos. • Consultar los eventos y noticias que se producen en la entidad. • Enviar correo electrónico.
Funcionarios	Son todas las personas encargadas de colaborar con el funcionamiento global de la institución y que podrían en algún momento ser usuarios del sistema.	<ul style="list-style-type: none"> • Consultar eventos y noticias. • Enviar correo electrónico.
Administrador del sistema	Es la persona responsable de la operación y mantenimiento de del sistema. Es el encargado de mantener el correcto funcionamiento del sistema y brindarles el soporte técnico a los usuarios.	<ul style="list-style-type: none"> • Administrar las cuentas de usuario, analizar la utilización del sistema, gestionar el correo electrónico. • Mantener y actualizar las páginas, los contenidos y los archivos. • Administrar la creación de los espacios de discusión, eventos y noticias.

4.2.6.2. Casos de uso

Representa la forma en que cada actor del negocio usa el sistema, se propone encontrar un caso de uso para cada rol de cada trabajador que participa en la realización de un proceso de negocio. En la Figura 12 se muestra el modelo de casos de uso del sistema. En la tabla 13 se hace una breve descripción de cada uno de ellos.

Figura 12. Modelo de casos de usos general del sistema

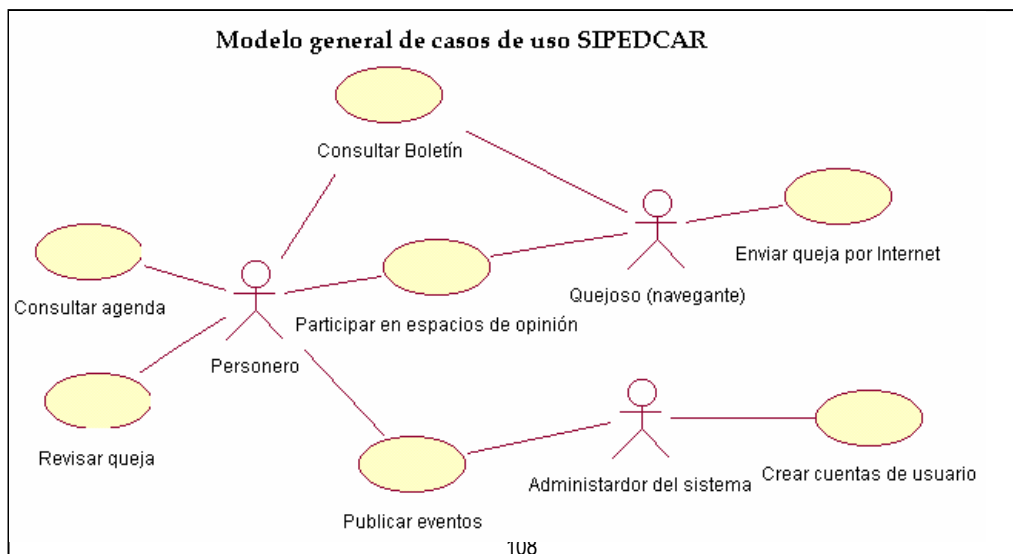


Tabla 13. Descripción de casos de usos del modelo general del sistema.

Caso de Uso	Descripción
Enviar queja por Internet	Permite al quejoso diligenciar y enviar una queja a través del sitio web de la Personería. Necesita conexión a Internet.
Revisar queja	Permite al Personero ver los datos generales de la queja, como el nombre de la persona que la interpuso, contra quien, la fecha y la declaración de la queja, para luego tomar la decisión del procedimiento a seguir.
Consultar agenda	Permite al personero verificar las actividades que a ingresado a la agenda del sistema, para conocer cuales debe realizar en el día, la semana o en el mes especificado.
Publicar evento	Este caso de uso permite al personero, actor especial(secretaría y funcionarios) o al administrador a través de un formato sencillo, publicar en la cartelera de eventos las actividades a realizarse
Consultar Boletín	Permite consultar información generada por la oficina de divulgación y prensa de la Personería Distrital de Cartagena. La información aquí contenida esta sujeta a criterio de la personería, quien es la que regula lo que se puede o no publicar.
Crear cuenta de usuario	Mediante este caso de uso el administrador tiene la facultad de crear, activar o desactivar cuentas de un usuario (Personero o funcionario de la Personería Distrital de Cartagena de Indias).

4.2.6.3. Casos De Uso En Detalle

El objetivo de detallar los casos de uso es describir el flujo de sucesos que ocurren en su desarrollo, es decir como comienza, termina e interactúa con los actores.

De la Tabla 15 hasta la Tabla 20 se detallan los casos de uso que fueron descritos con anterioridad y en las cuales se incluye la siguiente información contenida en la Tabla 14

Tabla 14. Contenido de detalle de casos de uso.

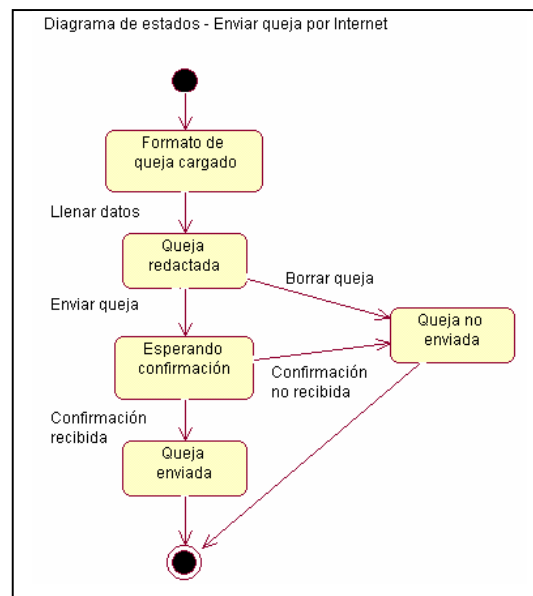
CONTENIDO DETALLE DE CASOS DE USO	
Nombre	El nombre del caso de uso.
Precondición	Son las condiciones previas que deben darse para la realización del caso de uso.
Flujo de sucesos	El conjunto de acciones básicas en un orden requerido que deben ejecutarse para el desarrollo del caso de uso.
Caminos alternativos	Son las acciones que complementan al flujo principal de sucesos o que describen una acción no permitida.
Postcondición	Define la forma como termina un caso de uso o su estado final.
Requisitos especiales	Son los requisitos no funcionales asociados a los casos de uso.

Para complementar el detalle se utilizarán los diagramas de estado de UML⁴³ para describir los estados de los casos de usos y las transiciones entre esos estados.

Tabla 15. Detalle de caso uso enviar queja por Internet

Caso de Uso	Enviar queja por Internet
Precondición	El usuario debe encontrarse registrado ya sea como anónimo o no; en caso de que no esté el sistema le presenta la opción para hacerlo.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El sistema le presenta un formato, el cual debe llenar con los datos respectivos, algunos serán obligatorios. 2. Se llenan los datos personales de quien interpone la queja. 3. Se llena la información de contra quien va dirigida la queja. 4. Se describe la queja en el espacio reservado para tal fin. 5. Cuando se ha terminado de escribir la queja se selecciona la opción enviar. 6. El sistema confirma el envío y el caso de uso termina.
Caminos Alternativos	En el paso 5, si no se decide a enviar la queja puede seleccionar la opción borrar
Poscondiciones	El caso de uso termina cuando la queja es enviada, o cuando el usuario sale de la sección.
Requisitos especiales	Se necesita conexión a Internet para poder acceder al sitio de la entidad.

Figura 13. Diagrama de estados del caso de uso - Enviar queja por Internet



⁴³ Ver Anexo B

Tabla 16. Detalle de caso uso Revisar queja.

Caso de Uso	Revisar queja
Precondición	Para poder revisar una queja ésta debió ser aceptada por el Personero a través del caso de uso Asignadas en el cual acepta la queja que el Abogado de reparto le asignó.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El personero selecciona un criterio de búsqueda (Identidad o nombre del quejoso, número radicado, número de la queja, fecha, etc.) para recuperar la queja 2. Selecciona la queja y el sistema le mostrará los datos detallados de ésta. 3. El sistema presenta la opción de elaborar una serie de documentos conforme a los procedimientos que deben realizarse al igual que puede ver los que ya ha elaborado. 4. El personero establece el estado de la queja y escribe una observación. 5. El caso de uso finaliza.
Caminos Alternativos	En la precondición, Si la queja es rechazada el personero se encargará de devolverla con sus respectivas observaciones para que sea reasignada.
Poscondiciones	El caso de uso termina cuando la queja es revisada y se le asigna un estado.

Figura 14. Diagrama de estados del caso de uso revisar queja

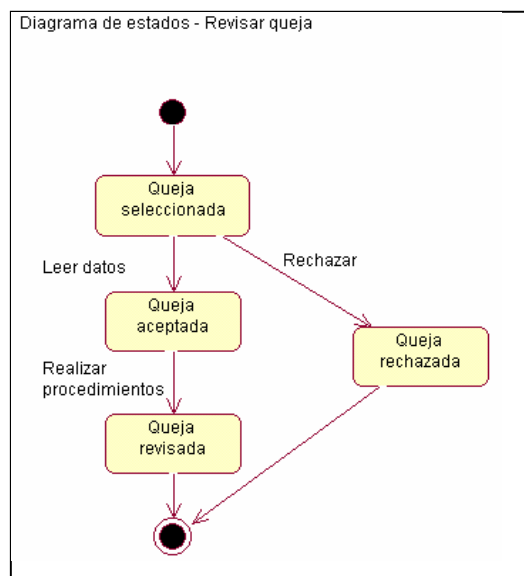


Tabla 17. Detalle de caso de uso Consultar agenda

Caso de uso	Consultar agenda
Precondición	El personero debe haber creado todas las actividades que va a realizar en la semana o mes (introducidas por día), asignándole a cada una (si así se establece por el personero); una prioridad, tiempo aproximado de duración; o haber creado para cada actividad un mensaje recordatorio.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El Personero invoca este caso de uso para revisar que actividades debe realizar en el transcurso de un día o a lo largo de una semana laboral. 2. El sistema muestra, por día, semana o mes (de acuerdo a la vista que se escoja), una lista de las actividades a realizar por el funcionario (Personero). Esta lista puede mostrarse organizadas por prioridad o por orden de ejecución. 3. La instancia del caso de uso finaliza
Caminos Alternativos	En el paso 2, si no existen actividades almacenadas, el sistema muestra un texto en el lugar del listado de actividades comunicándole al funcionario la no existencia de actividades para mostrar.
Poscondición	El caso de uso termina cuando el funcionario decide cerrar la agenda o cuando llama a otro caso de uso que hace parte de la agenda del sistema.

Figura 15. Diagrama de estados del caso de uso consultar agenda.

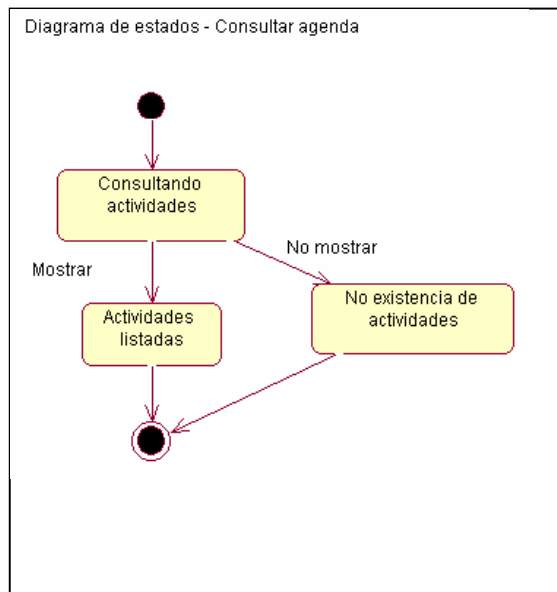


Tabla 18. Detalle de caso de uso Consultar Boletín

Caso de Uso	CONSULTAR BOLETÍN
Precondición	Debe haberse creado y subido al sistema, con la información elaborada por la oficina de divulgación y prensa de la Personería Distrital de Cartagena.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El funcionario de la personería (personero) o el quejoso (navegante), llaman a este caso de uso para ver la información contenida en el boletín informativo. Este ingreso se hace a través de Internet. 2. Específicamente en el sitio web se muestra un listado donde se encuentran los nombres de los diferentes boletines a los que puede acceder. 3. Cuando el navegante escoge el boletín que desea consultar, el sistema muestra la información contenida en este. 4. Finaliza el caso de uso.
Caminos Alternativos	En el paso 2, si el boletín que se busca no aparece en el listado, se puede seleccionar el año del listado de boletines que se desea que aparezca.
Poscondiciones	El caso de uso termina cuando es exitosa la consulta de un boletín o cuando se el usuario sale de la sección.
Requisitos especiales	Los usuarios externos (quejoso – navegante) deben contar con un acceso a Internet que les permita acceder a la información contenida en la pagina Web de la personería.

Figura 16. Diagrama de estados del caso de uso consultar Boletín

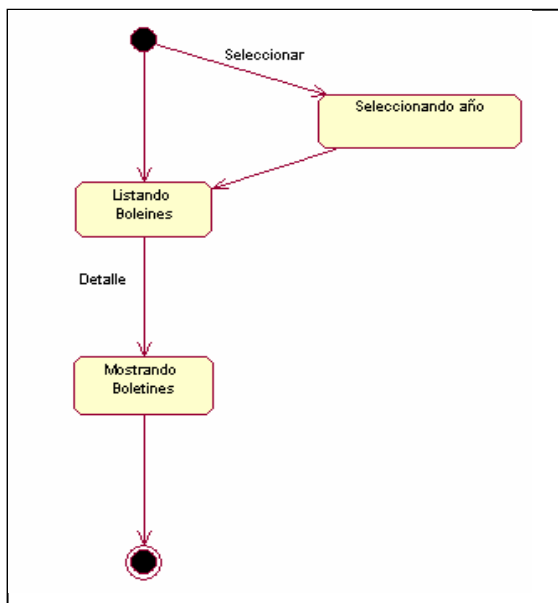


Tabla 19. Detalle de caso de uso Crear Usuario

Caso de uso	Crear Usuario
Precondición	El administrador debe conocer la información necesaria para la creación de un nuevo usuario en el sistema.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El administrador llama a este caso de uso para crear un usuario del sistema. 2. Debe suministrar información personal, para el sistema(clave, perfil, correo Intranet etc.) y laboral (Cargo, dependencia, extensión) del nuevo usuario. 3. Luego decide si activarlo o no. 4. El sistema cambiara el estado de la cuenta a activa o inactiva, dependiendo de la información suministrada por el administrador en este campo. 5. Finaliza el caso de uso.
Caminos Alternativos	Si el usuario ya se encuentra registrado, el sistema enviará un mensaje de error y no permite la creación del nuevo usuario.
Poscondición	El caso de uso termina cuando se crea satisfactoriamente la cuenta de un nuevo usuario y se le asigna el estado de activa o no activa a dicha cuenta.

Figura 17. Diagrama de estados del caso de uso Crear usuario

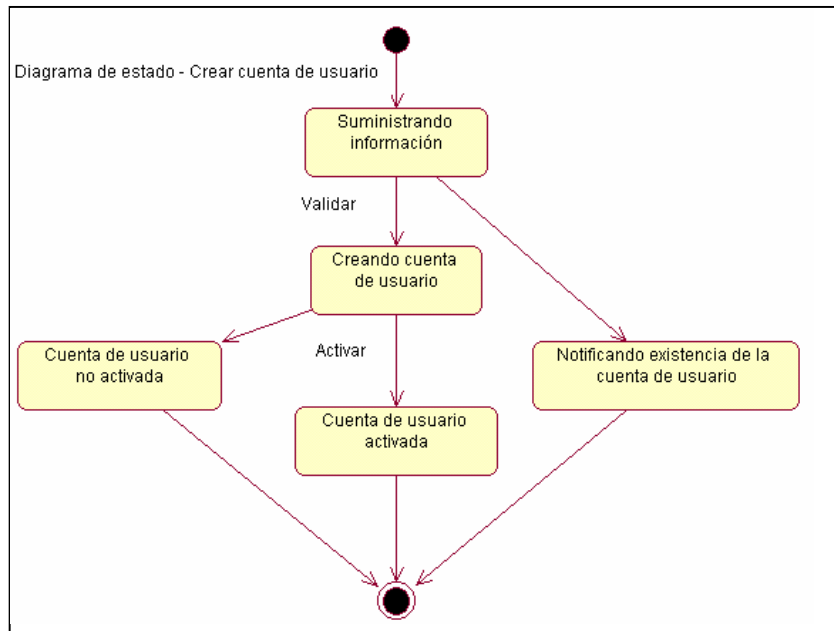


Tabla 20. Detalle de caso de uso Incluir evento en cartelera

Caso de Uso	Publicar evento en cartelera.
Precondición	Debe aparecer como opción del perfil del usuario que entra.
Descripción o Flujo de Sucesos	<p>1. El sistema le presenta un formato, el cual debe llenar con:</p> <ul style="list-style-type: none"> • Titulo • Contenido 1 • Contenido 2 • Fecha de publicación <p>2. Se pulsa el botón aceptar para incluir el evento en la lista de la cartelera.</p>
Poscondiciones	Un evento permanecerá 30 días en cartelera.

4.3. ANÁLISIS

Básicamente en el flujo de análisis se analizan los requisitos que se describieron en la captura de requisitos como casos de uso, refinándolos y estructurándolos, para conseguir una comprensión más precisa y una descripción que sea factible de mantener durante todo el desarrollo, ayudándonos a estructurar el sistema entero.

El resultado final de este flujo es la descripción de la arquitectura a través de una primera versión del **modelo de análisis**, conformado por los paquetes de análisis identificados, las clases del análisis y la realización de los casos de uso.

En la fase de inicio generalmente se realizan dos actividades en el flujo de análisis, como son: Análisis de la arquitectura candidata y analizar un caso de uso. Podría analizarse una clase y un paquete pero de forma mínima.

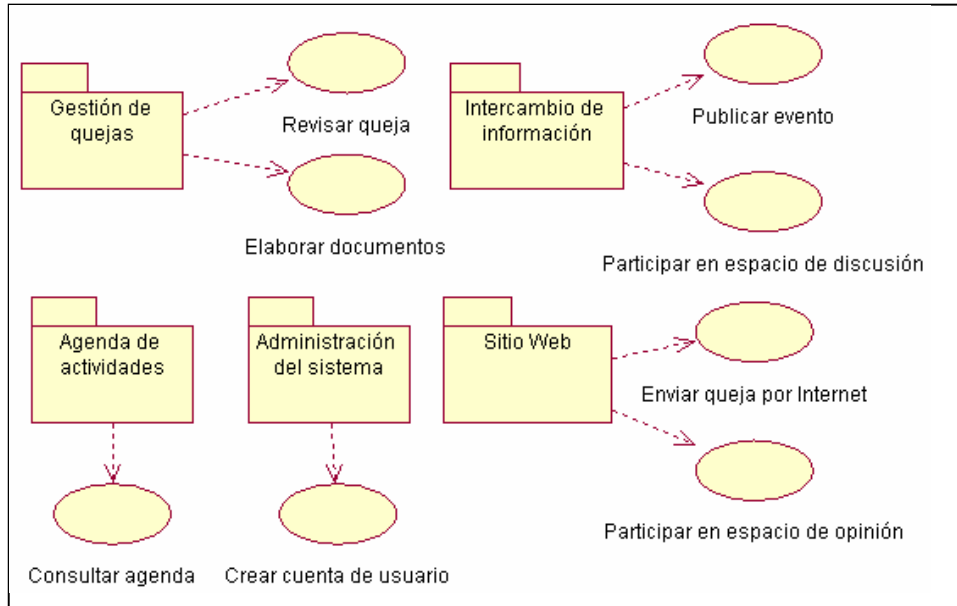
4.3.1. Análisis De La Arquitectura

El objetivo primordial es esbozar el modelo de análisis y la arquitectura mediante la identificación de paquetes de análisis y clases de análisis evidentes.

Identificación de paquetes de análisis: Estos paquetes se utilizan para dividir el modelo en partes más fáciles de manejar, esta división se hace teniendo en cuenta los requisitos funcionales con los cuales venimos trabajando desde la obtención de la lista de características, que luego algunos de ellos se transformaron en casos de uso. También podemos tener en cuenta el modelo de objetos del negocio.

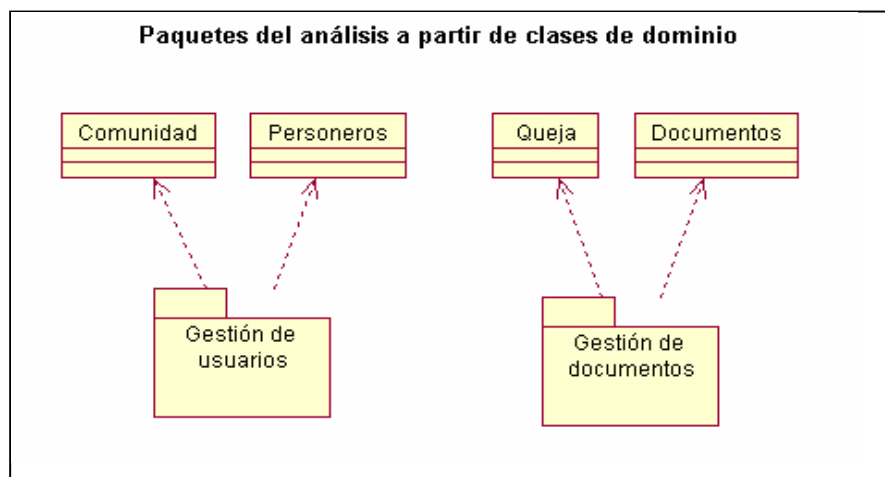
El modelo de casos de uso planteado deja ver la existencia de cinco zonas diferentes dentro del sistema las cuales agrupan un número de casos de uso, esto nos ayuda a identificar cinco paquetes de análisis, como lo muestra la Figura 17, el paquete de gestión de quejas, el paquete que agrupa los mecanismos de intercambio de información, el paquete de agenda de actividades, el paquete de administración del sistema que agrupa todos los casos de uso utilizados por el administrador del sistema y el paquete del sitio web.

Figura 18. Identificación de paquetes de análisis a partir de los casos de uso.



También se pueden identificar paquetes de análisis más generales partiendo del modelo de objetos del negocio, como se muestra en la Figura 18.

Figura 19. Paquetes de análisis a partir del modelo de objeto del negocio

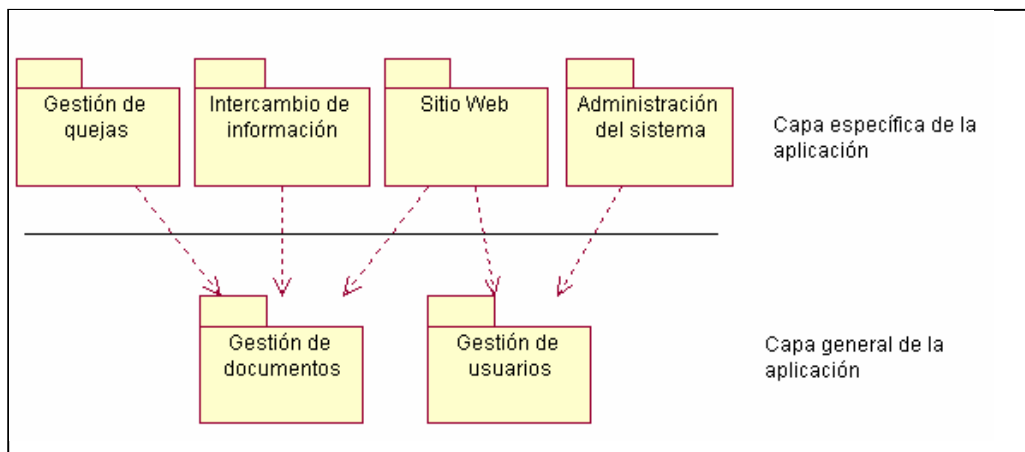


Todos estos paquetes evolucionarán y se reestructurarán a medida que se analicen mas casos de uso y se determinen las clases de análisis, probablemente se convertirán en subsistemas en el modelo de diseño.

El diagrama de la Figura 19 muestra las dependencias entre los paquetes obtenidos anteriormente. Se busca identificar paquetes que sean relativamente independientes debido a que se pueden presentar cambios o modificaciones en las clases que los conforman; procurando que estos cambios solo afecten a un paquete especifico y no al sistema completo. Se pretende que las clases que conforman un paquete estén altamente relacionadas y que estos (paquetes) tengan la menor dependencia entre si.

Una manera de representar estas dependencias es haciendo que los paquetes específicos del sistema queden en una capa superior y los paquetes generales en una capa inferior.

Figura 20. Dependencia de paquetes de análisis



Durante el diseño y la implementación, se refinarán estas capas y se agregarán más capas de bajo nivel a medida que tomemos en consideración el entorno de la implementación y los requerimientos no funcionales.








4.3.2. Análisis De Caso De Uso

Al analizar un caso de uso se busca:

- Identificar clases de análisis cuyos objetos son necesarios para llevar a cabo el flujo de sucesos de los casos de uso.
- Describir el comportamiento del caso de uso entre los objetos del análisis con los que interactúa.
- Capturar requisitos especiales específicos para la realización del caso de uso.

La Tabla 21 muestra las clases identificadas para la realización del caso de uso *Revisar Queja*, especificando su tipo, nombre y responsabilidad.

Tabla 21. Clases de análisis para el caso de uso Revisar queja.

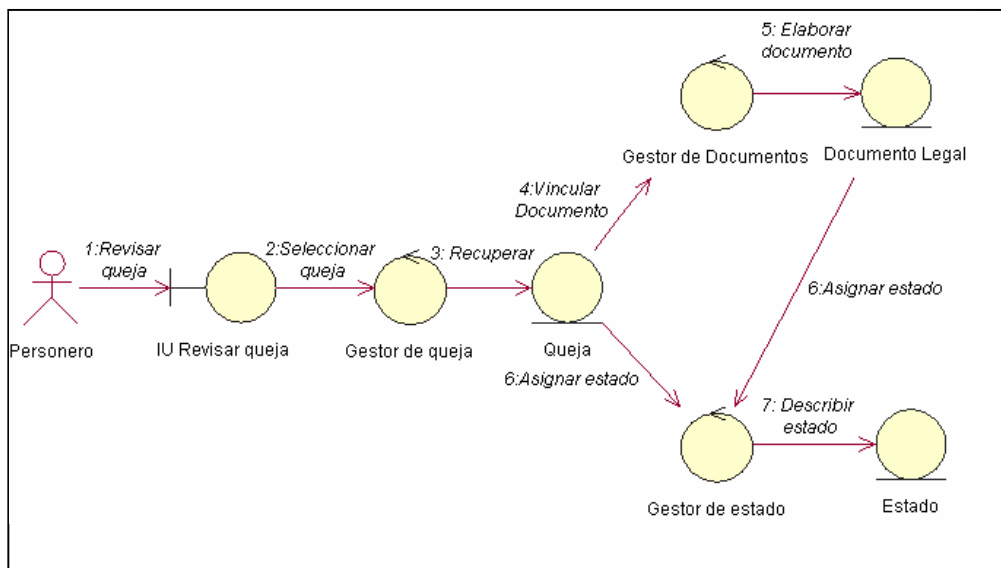
CLASES DE ANÁLISIS PARA EL CASO DE USO REVISAR QUEJA			
Diagrama	Tipo	Nombre	Responsabilidad
	Interfaz	IU Revisión queja	Mostrar la queja seleccionada por el personero para ser revisada.
	Entidad	Queja	Mostrar detalladamente la información que constituye una queja; como fecha, quien interpone la queja, personero encargado, etc.
	Entidad	Estado	Almacenar y mostrar información concerniente al estado en que se encuentra una queja y que permite describir dicho estado.
	Entidad	Documento legal	Crear y mostrar los documentos legales que corresponden a una queja en particular.
	Control	Gestor de estado	Evaluar si el estado de la queja es válido y permite la descripción correspondiente asignándola a la queja respectiva.
	Control	Gestor de quejas	Recuperar la queja dado el criterio de selección.
	Control	Gestor de Documentos	Permite la elaboración y la vinculación de los documentos legales que se utilizan en la solución de una queja.

Ahora que tenemos un esbozo de las clases necesarias para realizar el caso de uso, es necesario describir como interactúan sus correspondientes objetos del análisis, pero no de forma detallada ni organizada cronológicamente; sino

mostrando sus responsabilidades creando enlaces entre ellos y añadiendo mensajes a esos enlaces que indican el propósito de estos objetos.

Utilizaremos para ello un diagrama de colaboración que contiene las instancias de actores participantes, los objetos del análisis y sus enlaces, como lo muestra la Figura 20.

Figura 21. Diagrama de colaboración de la realización del caso de uso Revisar Queja



4.4. DISEÑO

En esta fase del Proceso Unificado de Desarrollo de Software, el objetivo principal del flujo de trabajo de diseño es lograr un esbozo del modelo de diseño de la arquitectura candidata, esto con el fin de poderlo incluir en la descripción de la arquitectura preliminar. Para esto desarrollaremos los artefactos de diseño tales como el modelo de despliegue y subsistemas que nos permiten hacer una descripción del diseño de la arquitectura.

4.4.1. Diseño De La Arquitectura

Para el diseño de la arquitectura se realizará un esbozo de los modelos de diseño y despliegue identificando los nodos y la disposición física de la red.

Identificación de nodos y configuraciones físicas de red: La disposición física de una red tiene mucha influencia en como está arquitectónicamente diseñado el software. En el desarrollo de este proyecto, se utilizara la arquitectura de múltiples capas o DNA proporcionando así que el mantenimiento y administración del sistema se convierta en una labor sencilla de realizar. El sistema desarrollado estará dividido en tres capas por lo cual la interacción con el cliente se deja en una capa, la funcionalidad de la base de datos en otra y la lógica del negocio o de la aplicación en una tercera capa.

SIPEDCAR se ejecutará sobre un nodo servidor y varios nodos clientes (Terminales asignadas a los funcionarios al interior de la personería). En el Nodo servidor se encontrará la capa de datos y la capa lógica o aplicaciones. Es aquí en donde estarán almacenados los datos del sistema. En los nodos clientes se ejecutará la capa de presentación. La comunicación entre el nodo del cliente y del servidor se realiza mediante el protocolo TCP/IP dentro de la infraestructura de red que debe montar la Personería Distrital de Cartagena para la implantación del sistema. También a través de Internet por parte de los usuarios (Comunidad) al exterior de la personería.

En la Figura 22 se ilustra el modelo de despliegue del sistema SIPEDCAR, donde se muestra la configuración física de la red así como su correspondencia con la arquitectura del software, es decir los nodos que participan en la ejecución del sistema y los componentes que residen en ellos. En la Figura 23 se muestra el modelo de despliegue para el Sitio Web de SIPEDCAR.

Figura 22. Diagrama de despliegue del sistema.

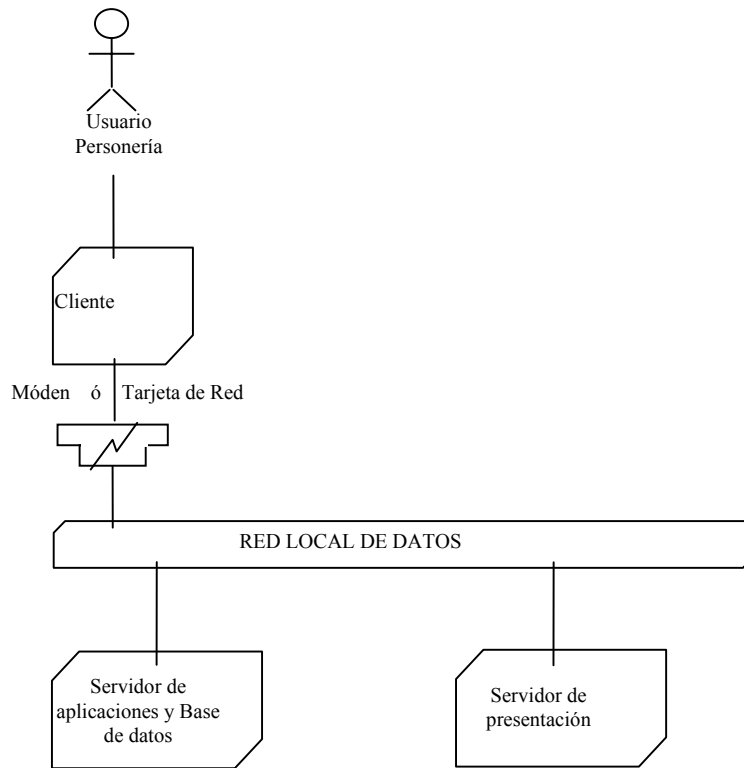
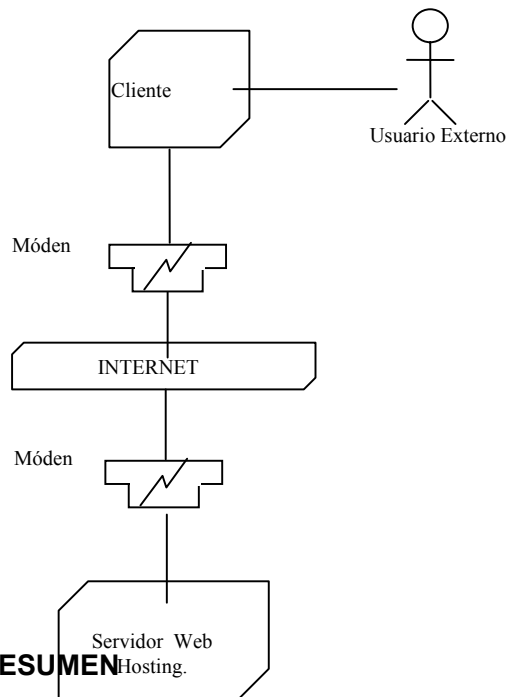


Figura 23. Diagrama de despliegue del sistema interno.



4.5. RESUMEN

Si se quiere conocer completamente los beneficios de la forma de trabajo iterativo, es necesario evaluar los avances al final de cada iteración o fase. En la fase de inicio el resultado de esta evaluación es la decisión de si seguir adelante o abandonar el proyecto.

4.5.1. Vista de la Arquitectura.

Como se ha dicho desde que se presentó el Proceso Unificado para el desarrollo de este proyecto, esta metodología está centrada en la arquitectura la cual abarca la organización del sistema software, los elementos estructurales que componen el sistema y sus interfaces, así como su comportamiento y colaboraciones entre elementos. Al describir la arquitectura se obtiene, por lo tanto, una mayor comprensión del sistema, se organiza el desarrollo y se fomenta la reutilización. Por tanto debemos construir una arquitectura que permita implementar los casos de uso del sistema de una forma económica, además se debe tener en cuenta otros factores como el sistema operativo y de bases de datos a escoger.

En esta fase, esta vista está limitada solo a la búsqueda de una arquitectura candidata, la cual se soporta solo sobre los casos de uso más importantes para el cliente, con mayor riesgo o que afectan la viabilidad del proyecto, descrita a través de los diagramas que se desarrollaron en esta fase.

4.5.2. Cumplimiento De Criterios

El desarrollo de esta fase estuvo orientado a visualizar el alcance del proyecto y a determinar la viabilidad del mismo. Para ello, se realizó un estudio básico del negocio, a través de los diagramas de casos de uso y de objetos, de la misión de la organización. Este estudio permitió tener una visión global del ámbito del sistema: de los actores y los casos de uso principales que deberían ser tomados en cuenta por el mismo; y de esta manera lograr una mejor comprensión de sus principales requisitos.

Además se limitó el alcance del sistema. Se determinó que este proyecto soportara principalmente el proceso de las quejas, creando espacios para compartir información, para la realización de discusiones, y brindando un canal para que la

Personería y la comunidad puedan hacerle un seguimiento a los procesos de manera que no queden archivados sin ninguna respuesta o excedan los tiempos requeridos por ley.

En esta fase se determinó una arquitectura candidata que sea viable para el desarrollo del proyecto dadas las limitaciones de tiempo y dinero, pero sin sacrificar la robustez y seguridad del sistema. Se determinó realizar el sistema en un ambiente Web para que sea de fácil acceso por cualquier usuario sin necesidad de demasiados requerimientos en su equipo y sin la necesidad de realizar una instalación de software adicional en su sistema.

Además se obtiene una independencia total con el software escogido en el servidor y los usuarios podrán contar con una interfaz conocida que les permita una fácil adaptación al sistema.

PARTE III

FASE DE ELABORACION

Al comienzo de la fase de elaboración, se recibe un modelo de casos de uso parcialmente completo y una descripción de la arquitectura candidata. También unos esbozos de un modelo de análisis y un modelo de diseño. Estos artefactos nos servirán de guía y punto de partida para iniciar la búsqueda de una arquitectura estable que sea consistente durante la construcción del sistema.

Para cumplir con esta meta se adoptará un punto de vista general del sistema. En algunos casos, en los que los riesgos técnicos predominen, o sean los más significativos, se necesitará profundizar para establecer una arquitectura sólida. Por tanto se tomarán decisiones de la arquitectura con base en la comprensión del sistema en su totalidad: su ámbito, sus requisitos funcionales y no funcionales. Además, al final de esta fase, se habrá acumulado la información necesaria para planificar la fase de construcción.

CAPITULO 5

PLANEACION Y ESTABLECIMIENTOS DE CRITERIOS DE EVALUACIÓN EN LA FASE DE ELABORACION

El tiempo necesario para la culminación de esta fase se estimó en 60 días, y se desarrolló en una sola iteración.

5.1. CRITERIOS DE EVALUACIÓN

Los criterios específicos a alcanzar en esta fase estarán de acuerdo con los objetivos planteados al comienzo de la misma. Estos son:

Objetivo 1: Extender los requisitos.

- ¿Se han identificado los requisitos, actores y casos de uso necesarios para diseñar la línea base de la arquitectura?
- ¿Se han detallado lo suficiente como para lograr los objetivos de esta fase?

Objetivo 2: Definir la línea base de la arquitectura

- ¿Satisface la línea base de la arquitectura no solo los requisitos recopilados formalmente hasta ahora, sino también las necesidades de todos los usuarios?
- ¿Parece la línea base de la arquitectura lo suficientemente robusta como para resistir la fase de construcción y la adición de características que puedan ser necesarios en posteriores versiones del sistema?

Objetivo 3: Mitigar los riesgos significativos

- ¿Se han mitigado de forma adecuada los riesgos críticos, ya sea eliminándolos o preparando un plan de emergencia?
- ¿Se han identificado todos los riesgos significativos?

5.2. PRODUCTOS A ENTREGAR

Como resultado de esta fase se entregara una nueva versión de todos los modelos: casos de uso, análisis, diseño, despliegue e implementación. También, se obtendrá una descripción de la arquitectura y la lista de riesgos actualizada.

CAPITULO 6

LINEA BASE DE LA ARQUITECTURA

6.1. Introducción

A través de los diferentes flujos de trabajo en la fase de elaboración se pretende buscar una base arquitectónica sólida y eliminar los elementos de más alto riesgo del proyecto. Las decisiones arquitectónicas deben tomarse con una comprensión del sistema global. Esto implica que se deben describir la mayoría de los requisitos del sistema.

Como se ha dicho, en la fase de elaboración se busca una arquitectura estable que soporte el desarrollo posterior en la fase de construcción. Para lograr este fin se deben obtener nuevos casos de uso y detallar los necesarios para entender la arquitectura. Además realizar un análisis de los paquetes que componen el sistema y el posterior diseño de los subsistemas que lo componen. Hacia el final de la fase se deben definir el diseño de la base de datos e interfaz.

Además se examina el alcance y los objetivos del sistema, la elección de la arquitectura y la resolución de los riesgos más grandes, y se decide si se debe pasar a la construcción.

6.2. REQUISITOS

Durante este flujo se trabajara sobre los requisitos identificados en la fase de inicio, se estructurará el modelo de caso de uso, identificando y detallando nuevos casos de usos determinando la prioridad de algunos de ellos.

6.2.1. Casos de uso

Se han identificado nuevos casos de uso dentro de los paquetes de análisis para completar el modelo parcial de casos de uso preparado en la fase de inicio. Estos han sido agrupados en los paquetes de análisis que venimos trabajando. (Sitio Web,

Gestión de quejas, Agenda de actividades, Mecanismos de intercambio de información y Administración del sistema).

A continuación se muestran los casos de uso de cada paquete y una descripción de los mismos con el fin de conocer que le permiten hacer a los diferentes actores del sistema.

6.2.1.1. Paquete de casos de uso – Gestión de quejas

Agrupar los casos de uso y actores involucrados en el proceso de dar solución a una queja que llega a la entidad. Ver Figura 24. En este diagrama se puede observar las instancias del actor Personero (Personero Delegado y Personero Distrital).

Figura 24. Casos de uso del paquete Gestión de quejas

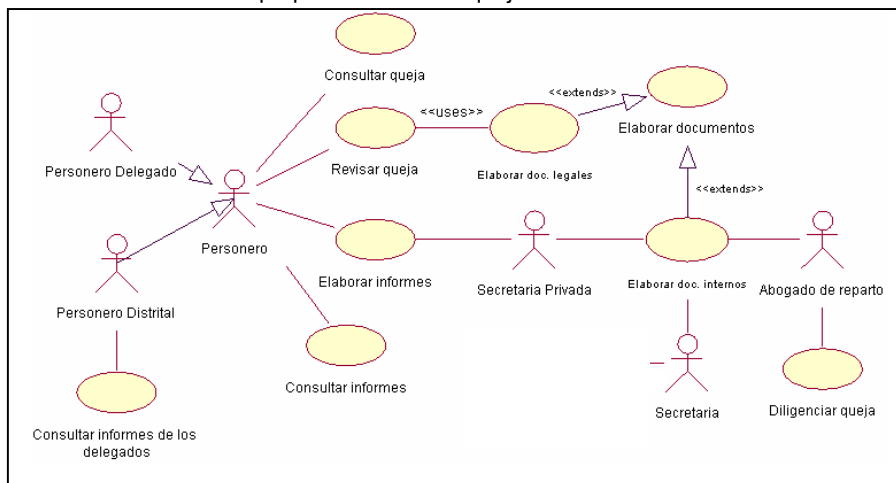


Tabla 22. Descripción de los casos de uso del paquete sitio web

Caso de Uso	Descripción
Registrar usuario	Este caso de uso permite al visitante registrarse como usuario anónimo o no anónimo, mediante este registro suministrará algunos datos incluido una clave y un nombre de usuario (para el caso de ser no anónimo) que le servirá para enviar una queja y consultar su evolución. En el caso del anónimo el sistema generará automáticamente el nombre de usuario (de esta forma A1000000) y el visitante debe suministrar una clave, entre otros datos.
Enviar queja por Internet	Permite al visitante diligenciar y enviar una queja a través del sitio Web de la Personería. Debe ingresar el nombre de usuario y la clave suministradas en el registro de usuario, luego tendrá acceso al formulario de diligenciamiento de la queja para su próximo envío.
Consultar queja por Internet	Permite al visitante consultar el estado de una queja específica. Para realizar esta consulta el visitante debe proporcionar el nombre de usuario y la clave, luego aparecerán todas las quejas interpuestas por ese usuario en el sistema y allí seleccionará cual desea consultar. En el caso de que la queja haya sido interpuesta personalmente la consulta se hará a través del número radicado

	suministrado por la oficina en el momento de interponer la queja sin necesidad de registro de usuario.
Consultar noticias	Mediante este caso de uso el visitante puede consultar las noticias de interés relacionadas con las labores de la Personería.
Consultar Boletín	Permite consultar información generada por la oficina de divulgación y prensa de la Personería Distrital de Cartagena.
Participar en espacios de opinión	Este caso de uso permite al visitante opinar sobre temas de interés común y de actualidad, a través de foros y encuestas.
Consultar Mecanismos de Denuncia	Mediante este caso de uso el visitante obtiene información acerca de los mecanismos constitucionales para hacer valer sus derechos o resolver conflictos por medio de la Personería. Cuales son, para que sirven, cuando se interponen, que se necesita para interponerlos, etc.
Consultar información institucional	Se podrá obtener información institucional tal como: misión, visión, funciones, organigrama, localización física y electrónica, teléfonos y fax, horarios de trabajo, normas básicas que regulan la actividad de la entidad y demás indicaciones que sean necesarias para que las personas puedan acceder a sus servicios.

6.2.1.2. Paquete de casos de uso - Sitio Web

Contiene los casos de uso que se presentan en el momento en que un visitante entra al Sitio Web de la Personería. Ver Figura 25

Figura 25. Casos de uso del paquete Sitio Web

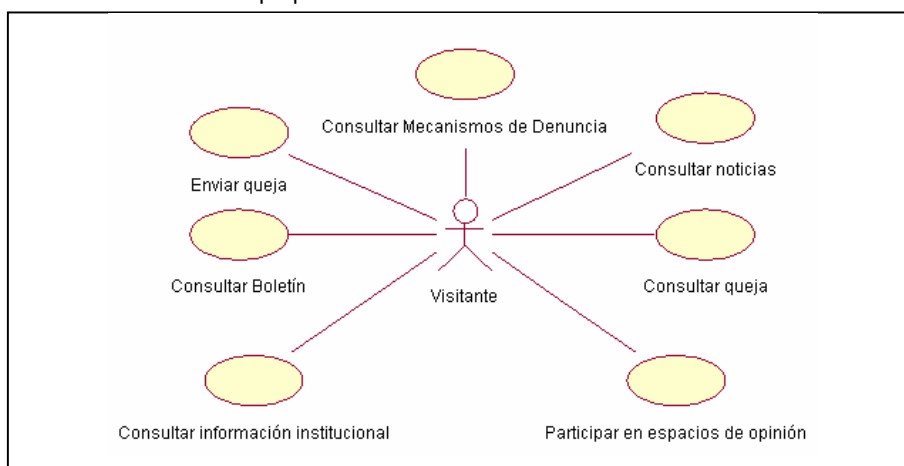


Tabla 23. Descripción de los casos de uso del paquete Gestión de quejas

Caso de Uso	Descripción
Consultar queja	Permite a los Personeros buscar y listar las quejas que se encuentran almacenadas en el sistema para su futura revisión. Esta búsqueda puede ser general (Quejas nuevas y revisadas) o avanzada (número radicado, cédula o nombre del quejoso).
Revisar queja	Permite al Personero ver los datos generales de la queja, como el nombre de la persona que la interpuso, contra quien, la fecha y la declaración de la queja, para luego tomar la decisión del procedimiento a seguir.
Elaborar informes	Por medio de este caso de uso tanto el Personero Distrital como los Delegados, podrán elaborar los informes periódicos de su gestión y enviarlo a la persona correspondiente.
Consultar informe	Este caso de uso permite a los Personeros revisar los informes que deben presentar.
Consultar informes	Este caso de uso permite al Personero Distrital consultar y revisar los informes

de los delegados	enviados por cada uno de los Personeros Delegados.
Elaborar documento	Este caso de uso permite la elaboración de documentos tanto legales como de manejo interno. Es una generalización de estos dos.
Elaborar documentos legales	Este caso de uso permite a los Personeros elaborar los documentos legales vinculados en la solución de la queja. Presentando el sistema las plantillas correspondientes. Como puede verse en la figura este caso de uso se utiliza a partir de otro caso de uso como es <i>revisar queja</i> .
Elaborar documentos internos	Este caso de uso es utilizado por las secretarias, el abogado de reparto o la secretaria privada para la elaboración de documentos de manejo interno.
Diligenciar queja	Por medio de este caso de uso el abogado de reparto toma la declaración de la queja, diligenciando el formato destinado para esto y luego enviarlo al Personero Delegado a cuyo juicio de corresponde darle solución a la queja. En este momento la queja es cargada en el sistema.

6.2.1.3. Paquete de casos de uso – Agenda de actividades

Contiene los casos de uso disponibles para los actores en el momento de interactuar con la agenda de actividades. Ver Figura 26.

Figura 26 - Casos de uso del paquete Agenda de actividades

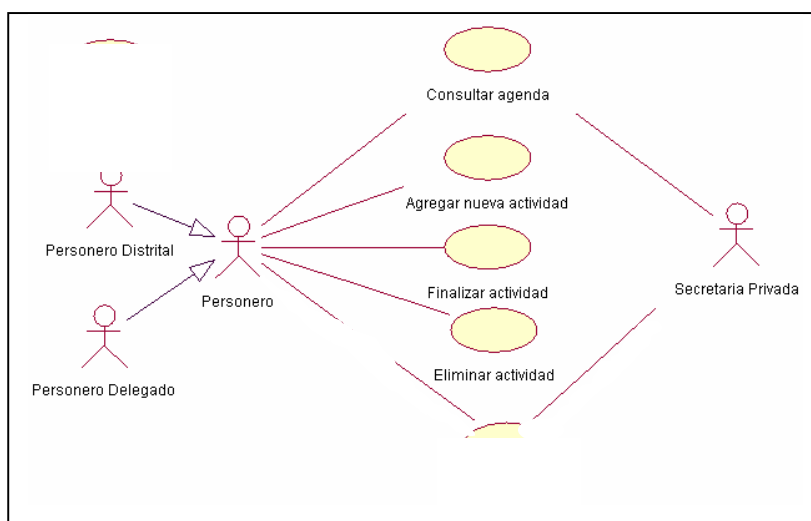


Tabla 24. Descripción de los casos de uso del paquete Agenda de actividades

Caso de Uso	Descripción
Consultar agenda	Permite al personero verificar las actividades almacenadas en el sistema, para conocer cuales debe realizar en el día, la semana o en el mes especificado.
Agregar nueva actividad	Este caso de uso permite ingresar una nueva actividad a la agenda de actividades.
Finalizar actividad	Este caso de uso permite marcar las actividades seleccionadas como completadas.
Eliminar actividad	Este caso de uso permite eliminar una actividad incluida en la agenda de actividades.

6.2.1.4. Paquete de casos de uso – Mecanismos de Intercambio de información

Agrupar los casos de uso relacionados con los mecanismos disponibles que tienen los actores para trabajar en grupo. Ver Figura 27.

Figura 27. Casos de uso del paquete Mecanismos de intercambio de información.

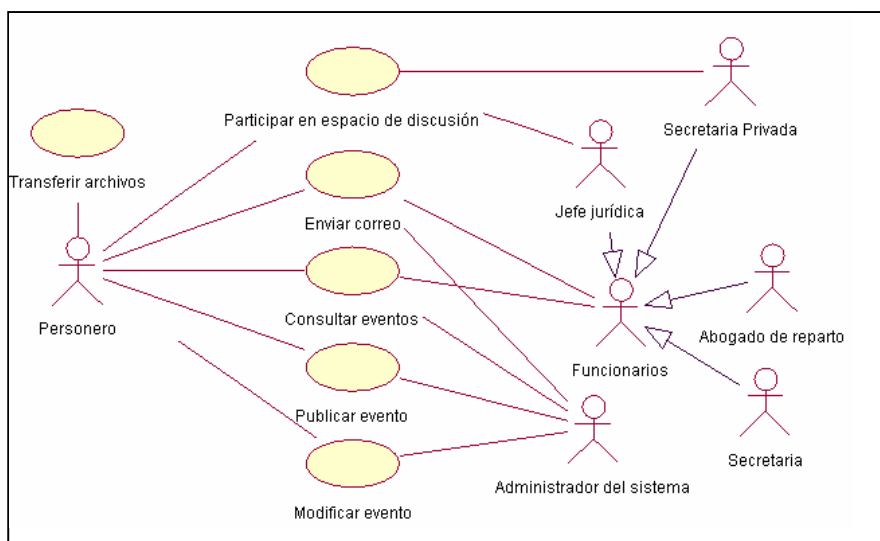


Tabla 25. Descripción de los casos de uso del paquete Intercambio de información

Caso de Uso	Descripción
Publicar noticias	Permite al administrador o al Personero Distrital incluir las noticias en la cartelera que son de interés para los funcionarios de la personería Distrital de Cartagena.
Consultar evento	Permite consultar los eventos planeados por la Personería Distrital.
Publicar evento	Permite al Personero o al administrador publicar un evento nuevo en el sistema, suministrando los datos respectivos.
Modificar evento	Permite al Personero o al administrador modificar los datos de un evento que se encuentra publicado en la cartelera.
Borrar evento	Permite al administrador eliminar un evento que se encuentra en la cartelera.
Enviar correo	Permite a los funcionarios de la entidad que tengan acceso al sistema enviar correo electrónico a un funcionario de la entidad siempre y cuando cuente con una cuenta de correo.
Participar en espacios de discusión	Por medio de este caso de uso el Personero Distrital, los Personeros Delegados, la secretaria privada y alguno de los funcionarios que se consideren indispensables podrán opinar y dar sus aportes respecto a algún tema de interés Este espacio estará dividido en salones reservando la entrada a alguno de ellos cuando fuere necesario.

Transferir archivos	Este caso de uso permite al Personero Distrital y a los Delegados enviar y recibir archivos cuando debido a su tamaño no puedan ser enviados por correo electrónico.
----------------------------	--

6.2.1.5. Paquete de casos de uso – Administración del sistema

En este paquete se encuentran los casos de usos que puede utilizar el administrador del sistema. Figura 28

Figura 28. Casos de uso del paquete Administración del sistema

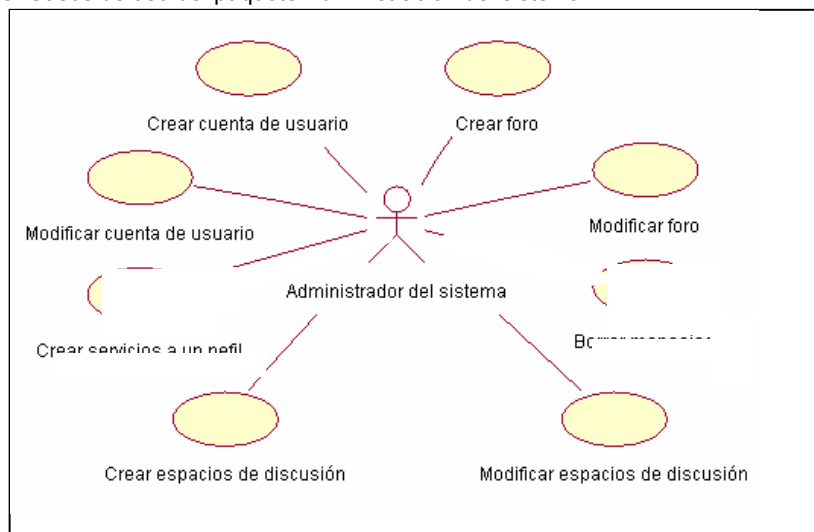


Tabla 26. Descripción de los casos de uso del paquete Administración del sistema

Caso de Uso	Descripción
Crear cuenta de usuario	Mediante este caso de uso el administrador tiene la facultad de crear, activar o desactivar cuentas de un usuario (Personero o funcionario de la Personería Distrital de Cartagena de Indias).
Modificar cuenta de usuario	Las cuentas de usuario pueden ser modificadas por el administrador a través del sistema, en el evento de que sea necesario.
Crear Foro	Permite crear un nuevo foro
Modificar Foro	Permite modificar la configuración de un foro existente.
Borrar Foro	Permite borrar un foro existente.
Borrar Mensaje de Foro	Permite borrar mensajes que ya han sido publicados en un foro.
Moderar Foro	Permite administrar un foro, decidiendo sobre la publicación o no; de cada uno de los mensajes enviados por los participantes.
Modificar espacios de discusión	Permite modificar la configuración de los salones de discusión existente.
Crear encuesta	Este caso de uso le permite al administrador crear la encuesta del sitio web, configurando la pregunta y las opciones respectivas.
Modificar	Permite cambiar la configuración de una encuesta.

<i>encuesta</i>	
-----------------	--

6.2.2. Casos de uso en detalle.

En esta sección se detallan los C. U. del modelo de casos de uso; que se consideran críticos en busca de una arquitectura estable (El proceso unificado es centrado en la arquitectura y esta debe mantenerse estable a lo largo del desarrollo del proyecto). El contenido del detalle de los casos de uso es el mismo que se presenta en la Tabla 27.

Tabla 27. Detalle del caso uso consultar queja por Internet

Caso de Uso	Consultar queja por Internet
Precondición	La persona debe haber ingresado al sitio web y seleccionado el link de consultar.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El visitante debe estar registrado. 2. El sistema le muestra una interfaz en donde se le solicita al visitante el nombre de usuario y la clave . 3. Se realiza la consulta a la base de datos, de las quejas registradas en el sistema. 4. El sistema le muestra al visitante la información obtenida, que consta de datos tales como el nombre del delegado que lleva el proceso, estado del proceso y una descripción del mismo. 5. Cuando el visitante cierra la pagina que muestra la información obtenida de la consulta, el caso de uso termina.
Caminos Alternativos	En el paso 2, si no se encuentra la queja, el sistema notifica al visitante de este hecho.
Poscondiciones	El caso de uso termina cuando el visitante cierra la pagina que muestra la información de la queja o cuando el sistema notifica al usuario que la queja no se encontró en la base de datos.
Requisitos especiales	Se necesita conexión a Internet para poder acceder al sitio de la entidad.

Tabla 28. Detalle del caso uso Participar en espacios de opinión

Caso de Uso	Participar en espacios de opinión
REQUISITOS ESPECIALES	Se requiere conexión a Internet para poder acceder al sitio de la entidad.
Precondición	La persona debe haber ingresado al sitio web y ubicado el foro y/o la encuesta existente.
Descripción o Flujo de Sucesos	<p>Si el visitante decide participar en el foro el flujo de sucesos es el siguiente:</p> <ol style="list-style-type: none"> 1. Se muestra al visitante los foros que actualmente se encuentran disponibles y las participaciones que otros usuarios han enviado. 2. El visitante selecciona el foro 3. El visitante decide hacer un aporte. 4. Ingresa el mensaje y lo envía. 5. Espera confirmación de envío y el caso de uso termina. <p>Ahora si el visitante decide participar en la encuesta los pasos a seguir son:</p> <ol style="list-style-type: none"> 1. Se muestra al visitante la encuesta que se está haciendo. 2. El visitante puede escoger entre participar o ver los resultados de la encuesta hasta el momento. 3. Si decide participar debe escoger entre las opciones y votar. 4. Para ver los resultados de la encuesta debe seleccionar esta opción. 5. Espera la finalización de la operación de envío y el caso de uso termina.
Caminos Alternativos	En el caso del foro (numeral 3) si el participante decide no enviar su aporte puede seleccionar la opción borrar.
Poscondiciones	El caso de uso termina cuando el visitante envía su mensaje (foro) o su voto.

Tabla 29. Detalle del caso uso Consultar Queja

Caso de Uso	Consultar queja
Precondición	La queja debe estar cargada en el sistema y haber sido asumida por algún Personero Delegado
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. Se debe escoger el parámetro de consulta para listar las quejas como Documento de identidad, nombre o apellido, fecha o número de radicado. 2. El sistema listara las quejas que correspondan con la consulta, mostrando datos como la fecha, el nombre del quejoso y contra quien va dirigida.
Caminos Alternativos	En el paso 2, el sistema le mostrará los datos específicos de la queja. Si la queja no aparece en el sistema, este le notifica al personero sobre este hecho.
Poscondiciones	El caso de uso termina cuando se obtiene la queja requerida y se pasa a su revisión o cuando no se encuentra la queja que se buscaba.

Tabla 30. Detalle de caso uso Diligenciar Queja

Caso de Uso	Diligenciar queja
Precondición	La queja debe haber sido recibida por la oficina de quejas y reclamos, es decir por el abogado de reparto.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. El sistema asigna un número de radicación. 2. El abogado de reparto llena los datos respectivos del formato. 3. Luego determina a que Personero Delegado se le debe enviar. 4. El caso de uso termina cuando envía la queja al Personero Delegado.
Caminos Alternativos	En el paso 1 antes de cargar la queja en el sistema, el abogado de reparto debe determinar si la queja es o no de competencia de la entidad. De no serlo le informa a la persona a donde debe dirigirse.
Poscondiciones	La queja debe estar asignada a algún Personero Delegado para que pueda ser consultada y revisada.

Tabla 31. Detalle de caso uso Agregar nueva actividad a la agenda

Caso de Uso REQUISITOS	Agregar nueva actividad a la agenda
Precondición ESPECIALES	Haber iniciado una sesión de usuario con el perfil para agregar una actividad.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. Luego de seleccionar agregar nueva actividad se llenan los datos respectivos del formulario tales como: <ol style="list-style-type: none"> a. Título de la actividad. b. Fecha. c. Hora de inicio. d. Duración. e. Prioridad. f. Descripción de la actividad. 1. El funcionario debe escoger entre guardar, guardar e introducir una nueva actividad o cancelar el ingreso. 2. El caso de uso termina.
Caminos Alternativos	En el paso 2, si la actividad ingresada está en conflicto con otra previamente almacenada el sistema notifica de este hecho al funcionario. El funcionario puede decidir entre modificar la actividad a guardar o cancelar el ingreso.
Poscondiciones	El caso de uso termina al ingresar una nueva actividad o al cancelar el ingreso.

Tabla 32. Detalle de caso uso Participar en espacios de discusión.

Caso de Uso REQUISITOS	Se debe planear con antelación por parte del administrador del sistema y los Participar en espacios de discusión
Precondición ESPECIALES	El administrador ha enviado (a los personeros que participarán en la actividad) vía e-mail, las claves de acceso al chat. El administrador ha enviado (a los personeros que participarán en la actividad) vía e-mail, las claves de acceso al chat.
Descripción o Flujo de Sucesos	<ol style="list-style-type: none"> 1. Los participantes en la actividad designada escogen el salón de discusión al cual previamente se les ha notificado pueden acceder (clave de acceso). 2. El sistema pregunta por la clave de acceso para poder acceder al salón de discusión. 3. El usuario, una vez validado su acceso, puede participar activamente en la discusión. 4. El caso de uso termina cuando el usuario sale del salón de discusión.
Caminos Alternativos	En el paso 2, si el usuario no está autorizado para entrar en el salón, el sistema notifica al usuario de este hecho y termina el caso de uso.
Poscondiciones	El caso de uso termina cuando el usuario sale del salón de discusión.

6.3. ANÁLISIS

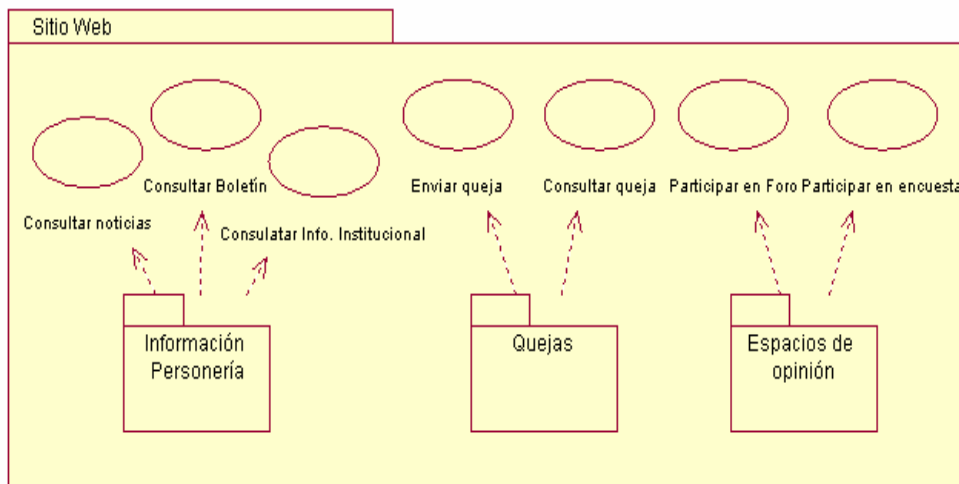
En este flujo abordamos las actividades de análisis de la arquitectura, tales como mejorar o detallar los paquetes de análisis identificados y analizar los casos de uso significativos desde el punto de vista de la arquitectura.

6.3.1. Análisis De La Arquitectura

Identificación de paquetes de análisis: Se estructuraron y detallaron los paquetes de análisis que se venían trabajando, convirtiéndose en un punto de partida importante para el diseño del sistema, como se muestra a continuación:

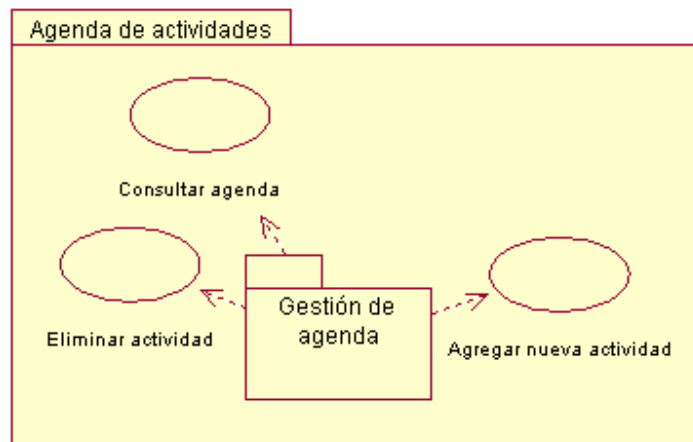
El paquete *Sitio Web*, *Figura 29*, presenta los casos de usos relacionados con el sitio web agrupados en subpaquetes que representan los tres aspectos que encuentra un visitante al entrar a la página, como: Consultar Información de la personería, Quejas y Participar en espacios de opinión.

Figura 29. Paquete de análisis Sitio Web



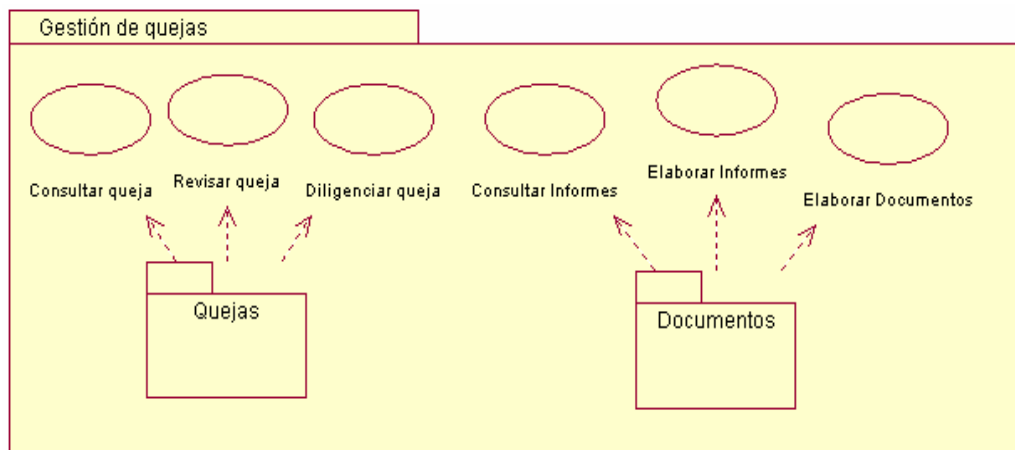
El paquete *Agenda de actividades*, *Figura 30*, en el cual se encuentran los casos de uso relacionados con la elaboración y gestión de la agenda.

Figura 30. Paquete de análisis Agenda de actividades



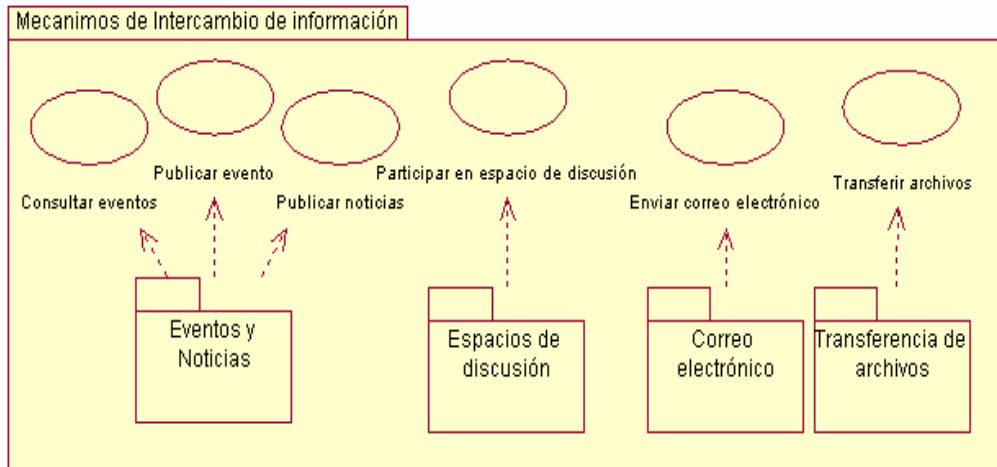
El paquete *Gestión de quejas* Figura 31, presenta los casos de uso que intervienen en la solución de una queja, desde que es interpuesta hasta los documentos e informes relacionados con ella. Se encuentra además un paquete de servicio que contiene el caso de uso de enviar alertas cuando un documento presenta vencimiento.

Figura 31. Paquete de análisis Gestión de quejas



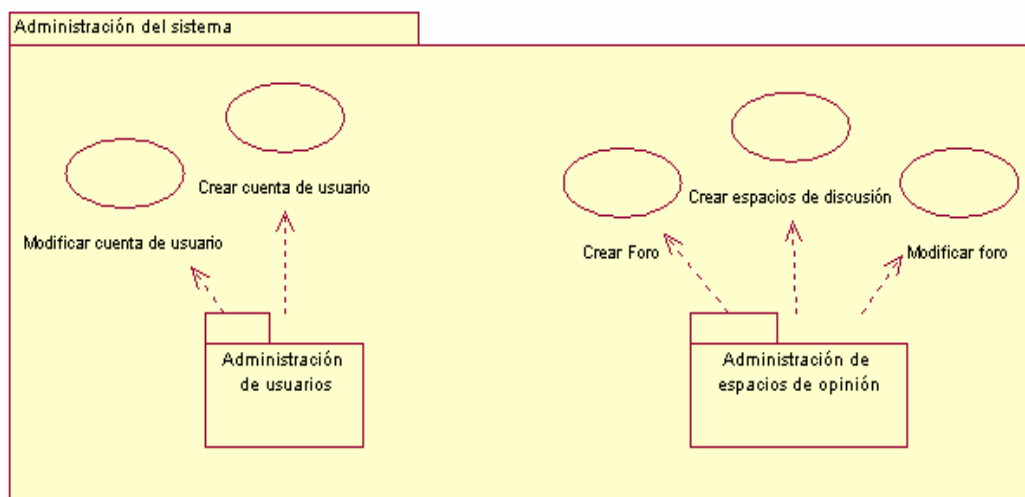
El Paquete *Mecanismos de intercambio de Información* Figura 32, contiene los casos de uso relacionados con los eventos, correo electrónico, espacios de discusión y transferencia de archivos.

Figura 32. Paquete de análisis Mecanismos de intercambio de información



En paquete *Administración del sistema* Figura 33, encontramos los casos de uso agrupados en las carpetas de administración de usuario y administración de los espacios de opinión del sistema en general.

Figura 33. Paquete de análisis Administración del sistema



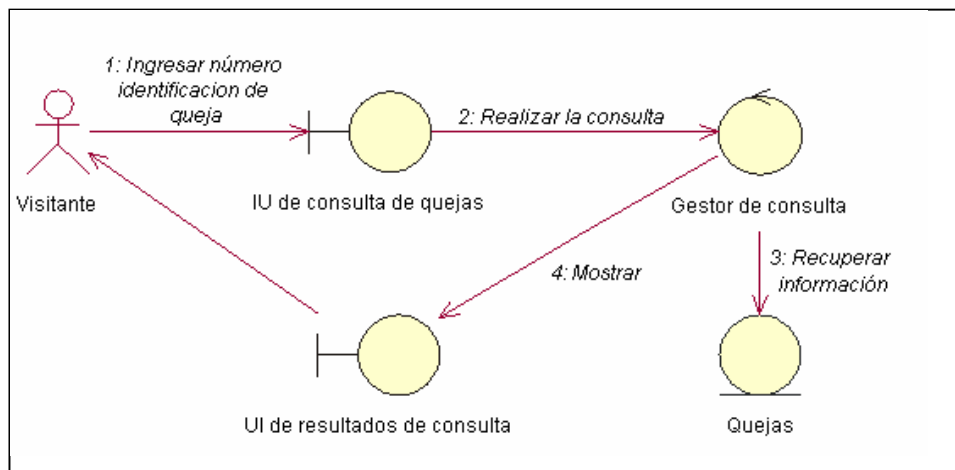
6.3.2. Análisis De Casos De Uso

En esta sección el análisis de los casos de uso se limita a los casos de uso más representativos e importantes para la arquitectura del sistema de cada paquete en los que se han agrupado.

Los siguientes son los diagramas de colaboración de los casos de uso y su respectiva descripción.

- **Caso de uso Revisar queja**

Figura 34. Diagrama de colaboración del caso de uso Consultar queja por Internet.



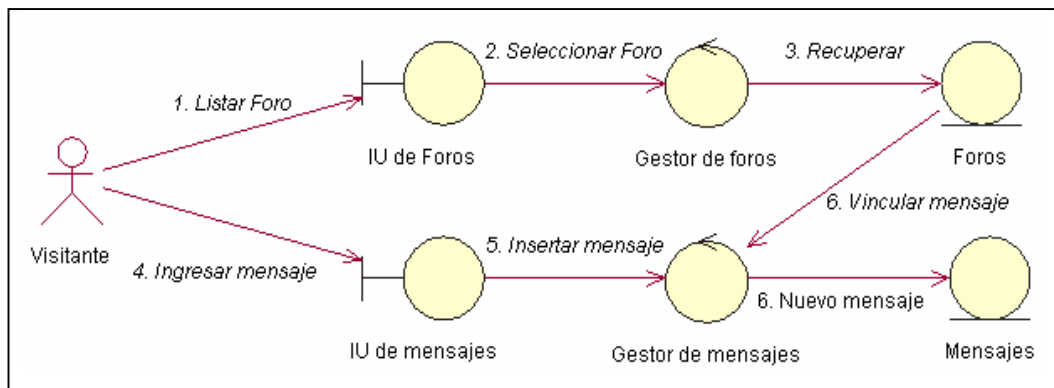
La IU de consulta de quejas que se encargan de mostrar al visitante el formato para la consulta de su queja y le permite ingresar los datos correspondientes para la búsqueda como por ejemplo el número radicado. Esta clase hace uso de la clase Gestor de Consulta a través de la cual se obtienen los datos de la clase de entidad Queja y visualizar los resultados por medio de la IU de resultados de Consulta.

- **Caso de uso Participar en Foro**

La IU de foros que se encargan de mostrar al participante los foros que actualmente se encuentran disponibles y ver las participaciones que otros usuarios han enviado a estos. A través de la segunda clase de interfaz, IU de mensajes, el actor puede enviar un nuevo mensaje y de esta manera lograr su participación en el foro. Estas dos clases hacen uso de las clases de control Gestor de Foros y Gestor de

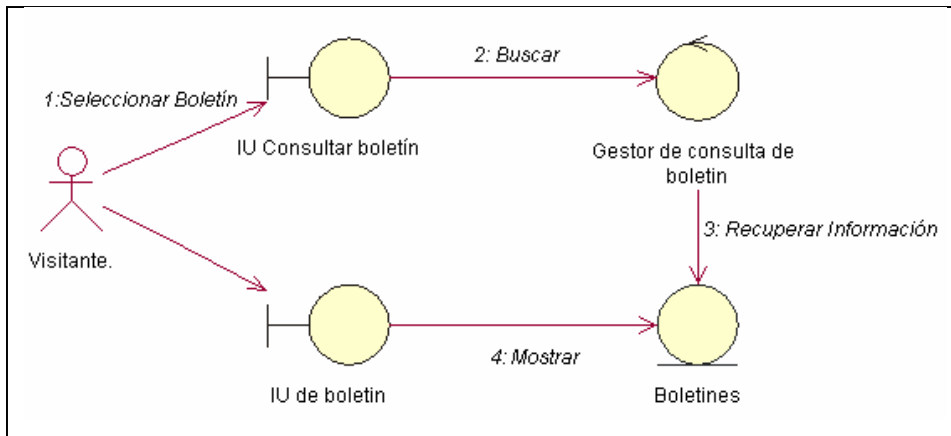
Mensajes para obtener la información necesaria de la clase entidad Foros y para enviar su participación en el mismo, respectivamente.

Figura 35. Diagrama de colaboración del caso de uso Participar en espacio de opinión (foro)



- **Caso de uso Consultar Boletín**

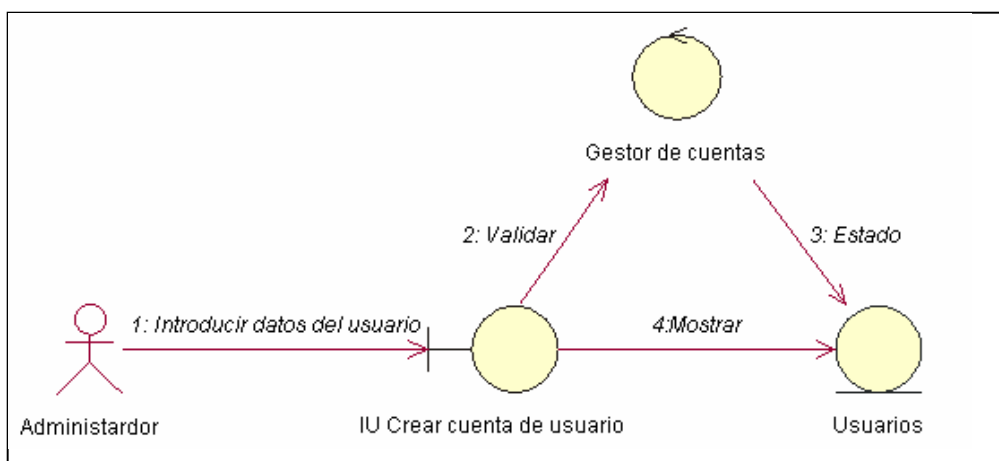
Figura 36. Diagrama de colaboración de la realización del caso de uso Consultar Boletín



La IU de consultar se encarga de mostrar al navegante los criterios para la búsqueda de un boletín específico, la clase de control Gestor de consulta de boletín recupera la información que corresponde (Boletín) almacenada en la clase de entidad Boletines; para que esta información se muestre a través de la IU de Boletín.

- **Caso de uso Crear cuenta de usuario**

Figura 37. Diagrama de colaboración de la realización del caso de uso Crear cuenta de Usuario.



La IU Crear cuenta de usuario muestra los datos necesarios para la creación de un usuario del sistema, por medio de la clase de control Gestor de cuentas se validan los campos asegurando que no se omita ninguno, así como también el verificar que no se encuentre ya registrado el usuario en el sistema. Esta clase también modifica el estado del usuario como activo o no activo.

6.4. DISEÑO

Este flujo es el centro de atención de la fase de elaboración ya que modelamos el sistema y encontramos su forma, para que soporte todos los requisitos incluyendo los no funcionales y otras restricciones, contribuyendo a una arquitectura estable y sólida y a crear un modelo físico que sirva como plano de la implementación.

6.4.1. Diseño de la Arquitectura

El objetivo del diseño de la arquitectura es esbozar los modelos de diseño y despliegue y su arquitectura mediante la identificación de los nodos, configuraciones de red, subsistemas, interfaces, software del sistema y capa intermedia.

6.4.1.1. Identificación de nodos y configuraciones de red.

Un modelo de despliegue se desarrollo en el capítulo 4 ; donde pudimos identificar dos tipos o clases de nodos con los que el sistema funcionará. Uno es el nodo servidor en donde se alojarán las capas de datos y lógica del negocio y los nodos cliente que se encargará de la capa de presentación en un modelo de arquitectura de tres capas, como el que se utilizará para la distribución del sistema.

Además, dado que el sistema será desarrollado con tecnología web se utilizaran los protocolos http y TCP/IP para la comunicación entre los nodos. Los requerimientos de hardware de cada uno de estos nodos se describe a continuación:

- ***Nodo Servidor***

Para la puesta en marcha del sistema de información se requiere un equipo servidor cuyas características serán definidas en el plan preliminar de la fase de Transición que será elaborado al terminar la fase de construcción sin embargo como servidor para la fase de desarrollo se utilizará un PC con las siguientes características:

- Procesador Pentium III de 500MHz o superior
- Memoria RAM de 128MB
- Disco Duro de 10G
- Tarjeta de Red

Para el uso del sistema de información debe implantarse una infraestructura de red, disponiendo en un comienzo de un total de puntos de acceso correspondientes a las oficinas de los personeros delegados, personero distrital, oficinas administrativas y recepción.

- **Nodo Cliente**

Los requerimientos para los equipos clientes son:

- Procesador Pentium de 133MHz o superior
- Memoria RAM de 32MB
- Tarjeta de Red

Como se planteo anteriormente en la fase de inicio debe contratarse un servicio de web hosting para el alojamiento del sitio cuyo servidor debe tener las siguientes características mínimas:

- Hosting 50 Mb de espacio para el sitio Web o más.
- Linux Red Hat con Apache
- Base de Datos MySQL / phpMyAdmin
- Soporte PHP4
- FTP ilimitado para actualizar el sitio
- Cuentas email suficiente para los funcionarios de la entidad.
- Webmail, acceso a email via web
- Shockwave Flash

6.4.1.2. Modelo de arquitectura de tres capas

El sistema SIPEDCAR se desarrolla a partir del concepto de *arquitectura en tres capas*. Así en la capa de datos encapsulamos todas las funciones de base de datos, las funciones de interfaz de usuario en la capa de Presentación y finalmente toda la lógica de operación en la capa Lógica.

Este hecho permite mayor flexibilidad y reutilización de partes de código. Además facilita modificaciones sencillas y rápidas a cualquiera de estas capas sin afectar el funcionamiento de las demás. Esto permite el mantenimiento y evolución del sistema, ya que este puede ser actualizado, sin tener que reconstruirlo en su totalidad.

La capa de datos esta determinada por la base de datos y las funciones para el manejo de la misma. Las interfaces de inserción, modificación y eliminación de registros a la misma son llevadas a cabo mediante scripts desarrollados en PHP y por la aplicación PHPMyadmin.

La capa lógica o de negocio contendrá todos los scripts y funciones que llevan a cabo la lógica del sistema teniendo en cuenta los casos de uso del sistema.

Por ultimo y para aprovechar la ventaja de la independencia entre capas que expone el desarrollo de software utilizando esta arquitectura, se implementa la capa de presentación por medio de templates o plantillas que mantendrán las interfaces independientes de la lógica del negocio, lo cual permitirá una mayor independencia de esta con respecto a la de aplicaciones.

6.4.1.3. Diseño a partir de plantillas (Templates)

Para la finalidad del proyecto es importante separar la lógica de la aplicación de su diseño grafico. Para este fin utilizaremos la clase FastTemplate. Mediante esta clase y diferentes plantillas html se realizará esta separación.

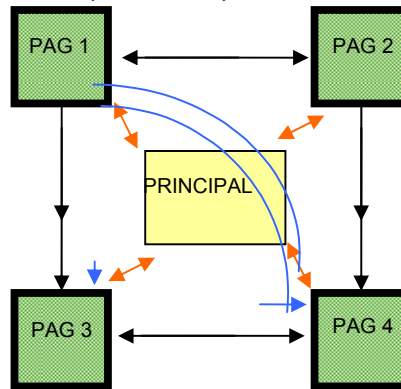
Un template o plantilla consiste en un archivo que contiene un trozo de código HTML, por ejemplo el cuerpo principal del Web. Entre este código se encuentran insertadas varias palabras entre llaves {} que harán de variables de la plantilla. Lo que se hacen entonces es asignar valores a estas variables y generar el código resultante. Más adelante en la sección software de la aplicación se explicara con más detalle la clase FastTemplate.

La ventaja principal de utilizar plantillas es la separación de los datos que se quieren presentar, del código. Así el diseñador no tendrá que preocuparse por corromper el código y a su vez el programador no necesitara interactuar con HTML.

6.4.2. Diseño De La Interfaz

El diseño de la interfaz del sistema ha sido desarrollado teniendo en cuenta los paquetes de análisis identificados. Para el diseño de la interfaz del **sitio Web** se tomaron como guía los casos de uso pertinentes y siguiendo algunos lineamientos generales.

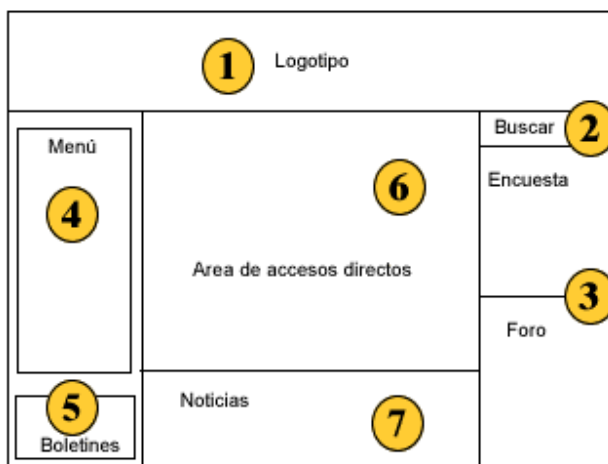
Figura 38. Estructura de malla completa o Web pura.



El sitio Web fue diseñado a una resolución (en píxeles) de 800 x 600 por considerarla la más adecuada. Por otra fue estructurado mediante una jerarquía de malla completa (red o Web pura), en la cual todas las páginas del sitio están enlazadas entre sí teniendo definida una página de inicio, como lo muestra la Figura 38.

Dentro del sitio encontramos una página principal o inicial, que se convierte en la página de entrada y varias páginas interiores o subpáginas donde se desarrollan los contenidos propuestos en la página principal, también encontramos aunque en menos proporción las llamadas “ventanas flotantes” que brindan al visitante algún tipo de ayuda o información necesaria para hacer uso correcto de un servicio de la aplicación. En la Figura 39 encontramos la página inicial del sitio y lo describiremos a continuación.

Figura 39. Interfaz de la página inicial del sitio web



(1)Cabezote: En esta parte se encuentra ubicado el logotipo de la Personería y en las páginas interiores además de éste se encuentran los elementos permanentes de la barra de navegación.

(2)Herramientas de búsqueda: El buscador presente en esta página da la oportunidad al visitante de seleccionar los temas y servicios que desea consultar o hacer uso de ellos respectivamente, además podrá acceder a algunos temas incluidos en estas páginas que no cuentan con un enlace directo, pero que podrían ser importantes para los intereses del visitante.

(4) Menú de Navegación: Este menú de navegación contiene los vínculos de las secciones y servicios principales del sitio., como son:

Tabla 33. Contenido del menú de navegación de la página inicial del sitio web

Información Institucional	En esta sección se ha organizado información referente a la normatividad, función, estructura organizacional, funcionarios, servicios ofrecidos, los horarios de atención, la trayectoria del personero y la reseña histórica.
Defienda sus derechos	En esta sección se ha incluido la información referente a los mecanismos con que cuenta Personería para ayudar o asesorar a la comunidad en la defensa de sus derechos. En algunos se encuentra la plantilla de los documentos correspondientes y la posibilidad de descargarlos.
Proyectos en ejecución	En esta sección encontrará vínculos de los principales proyectos que tenga la Personería. En la actualidad el sitio cuenta con información referente al proyecto del Personero Estudiantil, el cronograma de actividades y los centros educativos incluidos en el proyecto.

Presupuestos	Se podrá revisar o descargar los presupuestos de los diferentes años.
Informe de gestión	Se podrá revisar o descargar los informes de gestión de los diferentes años.
Noticias	Como su nombre lo indica este enlace nos lleva a la página de noticias generadas por la oficina de divulgación y prensa.
Contratación	En este enlace se podrá revisar las licitaciones realizadas por la personería antes y después de la fecha actual. Su objeto, fecha de apertura, fecha de cierre y presupuesto, además se podrá descargar el documento completo.

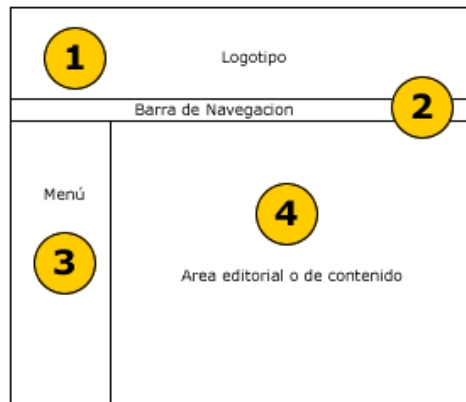
(5)Boletines: Mediante este script se visualizarán los Boletines de Prensa generados por la oficina de Divulgación y Prensa de la Personería. Aparecerá publicado la fecha, el número del Boletín y un titular. Al dar clic sobre uno de ellos los llevará al documento completo.

(3)Sección espacios de opinión: En esta sección de la página principal el visitante podrá participar de la encuesta que se realice en ese momento, votando por la opción deseada y visualizando inmediatamente los resultados. Por otra parte podrá opinar en el foro activo (sección opine), al dar clic sobre la pregunta del foro, el visitante es llevado a otra página donde dispone de la opción de insertar un nuevo mensaje, además observará los mensajes de los otros participantes.

Además en el área central se encuentran enlaces directos **(6)** para interponer las quejas, consultar sus estados y las noticias más recientes **(7)**, al dar click sobre una de ellas nos llevará al desarrollo completo de la noticia.

En la figura observamos la interfaz de las páginas interiores en la cual encontramos El cabezote **(1)**, el menú de navegación **(2)** y el área editorial o de contenido **(4)**, que puede ser estático o generado por consultas a la base de datos y el menú principal **(3)** el cual contiene todas las opciones del menú principal de la página inicial.

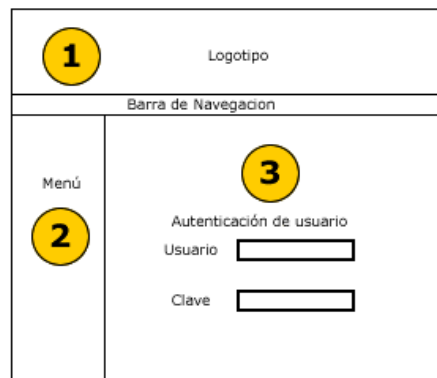
Figura 40. Interfaz de las páginas interiores del sitio web



Para el sistema de información interno, que corresponde a los otros paquetes de análisis identificados se ha diseñado la interfaz de la siguiente manera.

La página inicial Figura 41 donde encontramos los siguientes elementos:

Figura 41. Diseño de la página de inicio del sistema web de información interno



(1) Cabezote: donde encontramos el logotipo de la entidad, la fecha y hora actual.

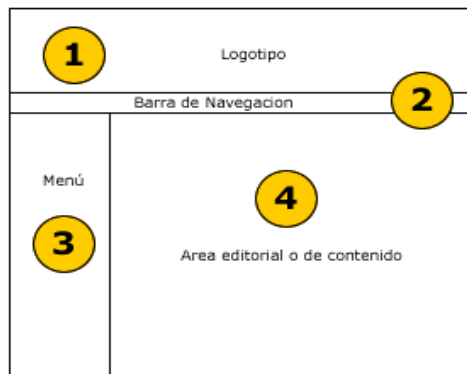
(2) Menú Informativo: En esta sección encontramos las opciones generales a cada usuario en el sistema; información como la misión y visión de la organización, así como una breve explicación sobre que es SIPEDCAR.

(3) Área de autenticación de usuario: Esta área controla el ingreso al sistema, solicitando un nombre de usuario y una clave, a través de las cuales se determina si

el usuario puede ingresar al sistema y cuales son sus permisos. Es decir se determina el perfil del usuario.

Las páginas interiores fueron diseñadas utilizando frames (marcos) para mayor comodidad cuando el contenido de las páginas lo ameritan, es decir con su altura son mayores a una pantalla. Se diseño de la siguiente manera Figura 42.

Figura 42. Interfaz de las páginas interiores del sistema interno



(1) Cabezote: de iguales características a la inicial...

(2) Menú de zona común: En esta sección encontramos las opciones que son comunes a todos o que pueden ser consultados por todos los usuarios que se encuentran registrados en el sistema.

(3) Menú del perfil: En esta sección encontramos las opciones particulares a cada perfil de usuario en el sistema. Esto se puede observar en la Tabla 34 *Servicios del sistema*.

(4) Área editorial o de contenido: Esta sección contiene el contenido correspondiente a la opción o al vínculo seleccionado.

Tabla 34. Servicios del sistema de información web según el perfil de usuario

	Personero Distrital	PersoneroD elegado	Secretaria	Abogado Reparto	Admin.. Sistema
Quejas					
Consultar queja	X	X			
Revisar queja	X	X			
Elaborar informes	X	X			
Consultar informe	X	X			
Consultar informes de los delegados	X				

Diligenciar queja	X			X	
Elaborar documento					
Elaborar documentos legales	X	X			
Elaborar documentos internos	X	X	X	X	
Consultar documentos internos	X	X	X	X	
Agenda					
Consultar agenda	X	X	X	X	X
Agregar nueva actividad	X	X	X	X	X
Eliminar actividad	X	X	X	X	X
Agregar nuevo aviso	X	X	X	X	X
Eventos					
Publicar noticias					X
Consultar evento y noticia	X	X	X	X	X
Publicar evento	X	X	X	X	X
Modificar evento					X
Borrar evento					X
Comunicación					
Enviar correo	X	X	X	X	X
Participar en espacios de discusión	X	X	X	X	X
Transferir archivos	X	X	X	X	X
Administración Sistema					
Crear cuenta de usuario					X
Modificar cuenta de usuario					X
Crear Foro					X
Modificar Foro					X
Borrar Foro					X
Borrar Mensaje de Foro					X
Administrar espacios de discusión					X
Crear encuesta					X
Modificar encuesta					X

6.4.3. Diseño De La Base De Datos

Para una mayor claridad y comprensión del sistema se ha decidido dividir en dos el diagrama Entidad-relación, una parte correspondiente al sitio web y la otra al sistema de gestión interno. A continuación en la tabla 35 se presenta un resumen general de las entidades presentes en el diagrama del sitio web.

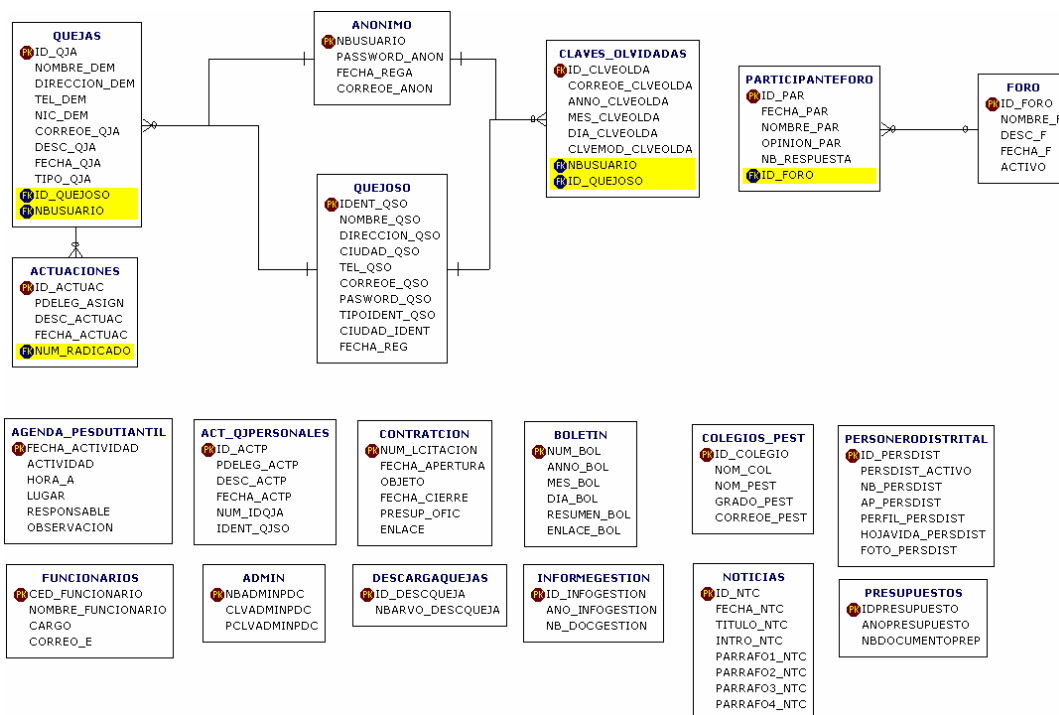
Tabla 35. Descripción de las tablas en el Diseño de la Base de Datos.

Tabla	Descripción
quejoso	Datos de las personas que colocan una queja por Internet y se registran como no anónimo.
anonimo	Datos correspondientes a una persona que se registra como anónimo en el momento de presentar su queja por Internet
Queja	Como su nombre lo indica, son las quejas interpuestas por los visitantes en la página.
actuaciones	Son las diferentes actuaciones de los personeros delegados a lo largo del proceso de solución de la queja.
Foro	Foros en los que actualmente el visitante puede intervenir. Aunque en realidad solo estará activo uno solo.
participanteforo	Contiene los datos del participante en el foro, tales como su nombre, mensaje y si va dirigido a alguien en especial.

act_qjpersonales	Son las diferentes actuaciones de los personeros delegados a lo largo del proceso de solución de las quejas que fueron interpuestas directamente en las oficinas.
admin	Contiene los datos del administrador del sistema.
agenda_pestudiantil	Contiene los datos referentes a las actividades planeadas por la Personería con relación a los personeros estudiantiles.
claves_olvidadas	
colegios_pest	Datos de los colegios incluidos en el proyecto del personero estudiantil.
contratacion	Información sobre las licitaciones abiertas y cerradas de la Personería.
informegestion	Contiene la dirección de los archivos correspondientes a los informes de gestión.
noticias	Contiene las noticias generadas por la Personería.
personerodistrital	Contiene información general sobre el personero actual y una dirección del archivo de su hoja de vida.
presupuestos	Contiene la dirección de los archivos correspondientes a los presupuestos.
boletin	Los boletines generados por la oficina de Divulgación y Prensa de la Personería.
funcionarios	En esta tabla se encuentran los funcionarios principales de entidad, su nombre y el cargo que ocupan.

6.4.4. Diagrama Entidad Relacion De La Base De Datos Del Sitio Web

Figura 43. Diagrama E/R de la base de datos del sitio web



En la tabla 35 se presenta un resumen general de las entidades presentes en el diagrama del sistema interno.

Esta base de datos fue desarrollada en Posgresql; a diferencia de la del sitio web creada en MySQL. Para mayor detalle debe consultar el Anexo E - Diccionario de datos.

6.4.5. Diagrama Entidad Relación De La Base De Datos Del Sistema Interno

Figura 44. Diagrama E/R de la base de datos del sistema interno

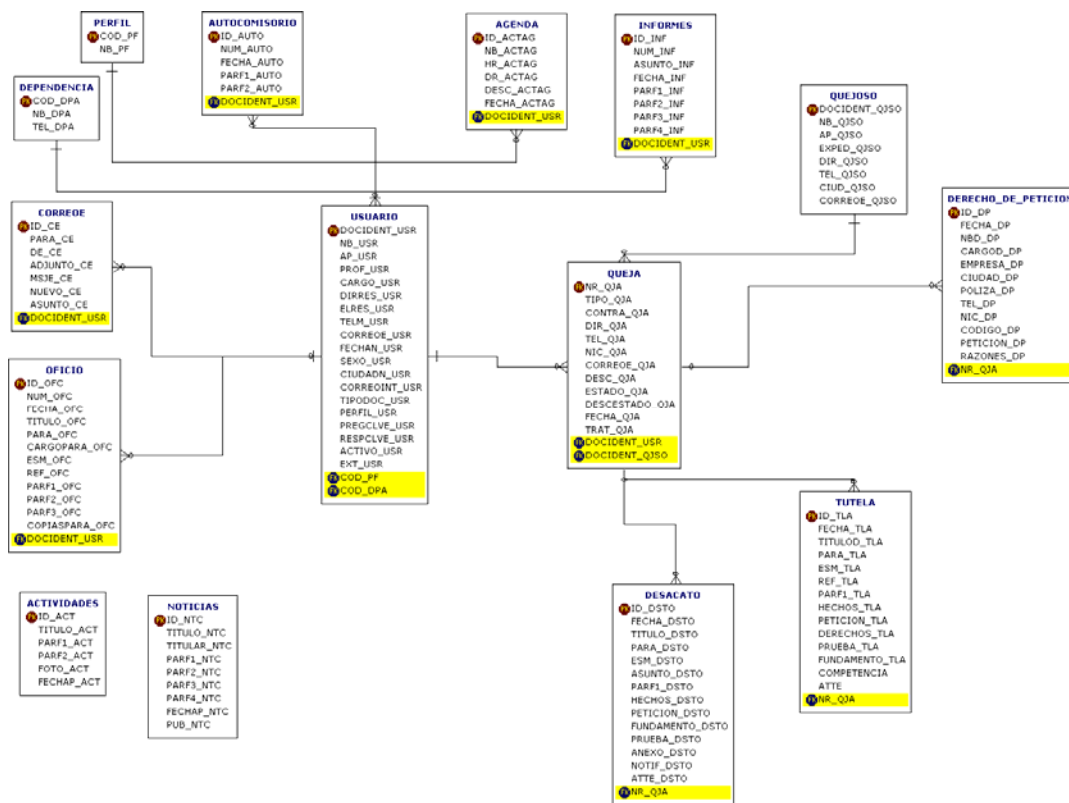


Tabla 36. Descripción de las tablas de la base de datos del sistema interno

<i>Tabla</i>	Descripción
usuario	Datos de las personas que se encuentran registradas como usuarios del sistema
actividades	Contiene los datos relacionados con los eventos de la personería como: el título, la actividad, una foto etc.
dependencia	Datos de la dependencia a la cual pertenece la queja, tiene un código, un nombre y teléfono.
correoe	Contiene los datos del correo electrónico de los usuarios.
noticias	Contiene las noticias generadas por la Personería, que son de interés para la planta de personal.
queja	Como su nombre lo indica, son las quejas diligenciadas y asignadas a los personeros.
quejoso	Datos de las personas que colocan la queja en las oficinas.
derechodepeticion	Contiene la información pertinente al documento derecho de petición.
desacato	Contiene la información pertinente al documento derecho de petición.
perfiles	Contiene el código y nombre del perfil, importantes para la personalización de los menús.
tutela	Contiene la información pertinente al documento acción de tutela.
oficio	Contiene la información pertinente al documento derecho de petición.
autocomisorio	Contiene la información pertinente al documento autocomisorio.
informes	Se encuentran datos de los diferentes informes elaborados por los personeros.
agenda	Contiene los datos de la agenda de los personeros.

6.5. IMPLEMENTACION

Este flujo de trabajo implementa y prueba los componentes arquitectónicamente significativos a partir de los elementos de diseño obtenidos. El resultado es la línea base de la arquitectura, implementada normalmente a partir de menos del 10 por ciento de los casos de uso.

La mayor parte de la arquitectura del sistema es capturada durante el diseño. La implementación tiene como propósito desarrollar la arquitectura y el sistema como un todo. Dado el enfoque incremental del proceso unificado la implementación se dará a través de una sucesión de pasos pequeños y manejables.

En esta fase, la implementación se enfoca solo a instalar y configurar el software del sistema así como de asignar los componentes necesarios a cada uno de los nodos detallados en el diagrama de despliegue.

6.5.1. Software Del Sistema Y La Capa Intermedia

El software del sistema y la capa intermedia constituyen los cimientos de un sistema. Ya que toda la funcionalidad descansa sobre software como sistemas operativos, sistemas de gestión de bases de datos, software de comunicaciones etc.

La elección de productos software es objetivo de la fase de inicio y elaboración.

Como se planteo en los objetivos del proyecto el Sistema de información Web para la Personería Distrital de Cartagena se desarrolla bajo el sistema operativo Linux .

Para el manejo de la base de datos se utilizarán dos sistemas, **MySQL** para la base de datos del sitio web y por otra parte **POSTGRES** único sistema gratuito en el mundo que soporta base de datos objeto y SQL, para el manejo de la base de datos del sistema interno, con la ayuda de la aplicación **PHPMyadmin** y **Pgadmin** para la manipulación y mantenimiento respectivamente.

Se utilizará el servidor para páginas **Web Apache**, muy utilizado en la actualidad por su eficiencia, funcionalidad y velocidad y **PHP (Hypertext Preprocessor)**, que es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor como lenguaje de la capa de la lógica de la aplicación.

Todas las herramientas mencionadas anteriormente son productos “hechos en Internet”, lo cual se traduce en costo nulo y con un grupo de desarrolladores de excelente calidad, logrando una solución económica sin que el sistema pierda su robustez y seguridad.

En la siguiente tabla se presenta el software utilizado y la manera en que fueron configurados.

Tabla 37. Software del sistema

Software
Sistema Operativo Linux Red Hat 7.x
Servidor web Apache 1.3
Lenguaje PHP 4.0.6
Postgres
MySQL 3.23.51
PHPMyadmin 2.3.2
Class FastTemplate 1.0.1

La clase **FastTemplate** fue instalada como software de la capa intermedia, muy importante para la utilización de las plantillas y por lo tanto la implementación de la arquitectura de tres capas. Esta clase presenta varios métodos de los cuales los más utilizados los encontramos en la siguiente tabla, si desea más información puede dirigirse al Anexo C.

Tabla 38. Principales métodos de la clase FastTemplate.

	Método	Descripción
define(array(key,value pairs))	define(array(crear =>"crear.tpl", curso =>"cursos.tpl"));	Define un vector asociativo con las plantillas y los nombres dados a estas. Según el ejemplo, asigna a a crear la plantilla crear.tpl
assign((key,value pair)	\$tpl->assign(TITULO =>"Sieduis");	Sustituye las etiquetas del template con el valor pasado a esta función. En el ejemplo a la etiqueta o variable TITULO le asigna la palabra Sieduis
parse(RETURN, FileHandle(s))	\$tpl->parse(FINAL, "crear");	Asigna a la variable FINAL el contenido del template.
FastPrint(HANDLE)	\$tpl->FastPrint("FINAL");	Imprime el contenido del template dado.

6.5.2. Manejo De Sesiones

En esta sección se describe el uso de "SESIONES" en el lenguaje PHP, como mecanismo de seguridad en el manejo de la información que se almacena y consulta en la base de datos del sitio y del sistema interno.

Al contar con una serie de páginas que en alguna u otra medida, guardan relación entre si (por ejemplo: el interponer o consultar una queja por parte de un usuario

específico), se hace necesario para el sitio web; implementar un mecanismo que permita almacenar y compartir información entre las distintas páginas. Esto se hace por medio de una *SESSION*; una sesión inicia cuando el usuario ingresa a la aplicación, para nuestro caso en particular cuando ingresa a registrarse, interponer o consultar una queja y cuando el administrador inicia su sesión de trabajo; y termina cuando la abandona.

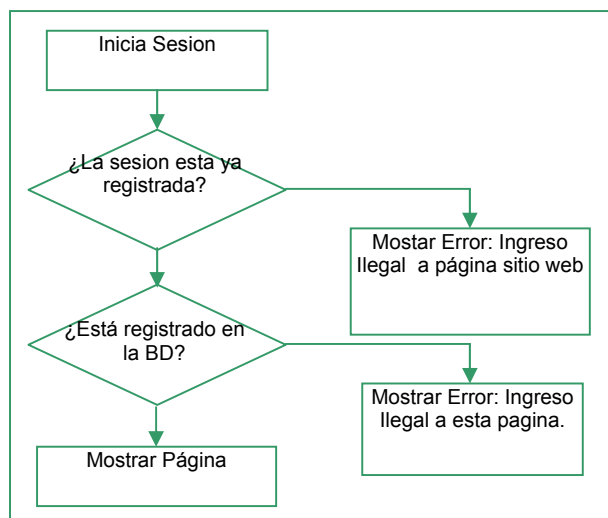
Cuando se inicia una sesión se registra información que es indispensable para acceder a la siguiente página relacionada a esta (la que inicia la sesión), solo así puede tener acceso a la información generada dinámicamente por una o varias páginas.

El ejemplo claro de esto consiste en la verificación de ingreso de un usuario (Autenticación de usuario); cuando el usuario, desea hacer uso de un proceso para llevar a cabo un caso de uso (interponer una queja o consultarla) implementado por el sistema (Sitio Web Personería Distrital de Cartagena), se debe verificar que dicho usuario esté registrado en la BD y pueda entonces acceder a cambiar o modificar la información que este maneja. Esto se hace solicitando al usuario un nombre de usuario y una clave de acceso, confirmando luego la existencia del registro del usuario en la BD, el gestor de sesiones envía un identificador de sesión⁴⁴ al usuario, que lo identifica; además de ciertos datos propios del usuario. Este identificador de sesión es el que le da acceso al usuario a las páginas que sólo él puede acceder. Esto es de especial importancia, dado que para la comunicación entre el cliente y servidor se utiliza tecnología Internet, exponiendo los problemas de seguridad propios de esta tecnología.

En la Figura 45. Se muestra un diagrama de flujo que describe el proceso de restricción de acceso usando sesiones.

⁴⁴ El termino Sesión para una instancia de un usuario que visita un sitio donde se mira una o mas paginas. Secuencia de paginas que un usuario visita en un sito web desde que entra hasta que lo abandona.

Figura 45 Diagrama de restricción de acceso usando sesiones



6.6. RIESGOS CRITICOS

Sumado a los riesgos críticos identificados en el Capítulo 4 Tabla 4.5 encontramos además los aspectos correspondientes a las características que debe cumplir el servidor de web hosting donde se alojará el sitio web de la entidad, el cual debe cumplir con los siguientes características:

- Hosting 50 Mb de espacio para el sitio Web o más.
- Linux Red Hat con Apache
- Base de Datos MySQL / phpMyAdmin
- Soporte PHP4
- FTP ilimitado para actualizar el sitio
- Cuentas email suficiente para los funcionarios de la entidad.
- Webmail, acceso a email via web
- Shockwave Flash

6.7. RESUMEN

6.7.1. Cumplimiento de criterios

En varias oportunidades, se ha hecho mención del objetivo de esta fase, el cual es encontrar la arquitectura estable que guíe el desarrollo del sistema. En la búsqueda de este objetivo se han enfocado las tareas de los diferentes flujos de trabajo, obteniendo la vista de la arquitectura estable.

Durante esta fase se identificaron la mayor parte de los requisitos del sistema, se verificó la importancia de los actores encontrados en la fase de inicio y se detalló gran parte de los casos de uso. Dado que la mayoría de estos casos de uso no tienen complicaciones ni excepciones mayores en su desarrollo, fue suficiente con una descripción de los mismos para ser comprendidos totalmente y dar paso a su implementación en la fase de construcción.

Sin embargo, se analizaron los casos de uso claves para tener una visión global del sistema. Para ello se escogió un caso de uso relevante de cada paquete de análisis, que reuniera los aspectos más importantes de los demás casos de uso de dicho paquete. Así se obtuvo de forma directa el diagrama Entidad-Relación y la interfaz de usuario.

También, gracias a los casos de uso y a los subsistemas de diseño obtenidos a partir de los paquetes de análisis, se obtuvo fácilmente las opciones que debería soportar el sistema y la manera como estas se organizarían para brindar al usuario una fácil visualización de las mismas.

Los diferentes perfiles del sistema surgieron de los actores del sistema, que además contribuyeron a verificar que la línea base de la arquitectura satisficiera las necesidades de todos los usuarios.

PARTE IV

FASE DE CONSTRUCCION

El objetivo general de esta fase es alcanzar un producto listo para ser distribuido como versión beta y ser sometido a pruebas en esta fase y en la transición. En este momento los requisitos y la arquitectura son estables y solamente en algunos casos se necesitarán realizar pequeños cambios a ellos.

Los objetivos específicos de esta fase son:

- La extensión de la identificación, descripción y realización de todos los casos de uso.
- El mantenimiento de la integridad de la arquitectura, modificándola solo cuando sea necesario.
- La monitorización de los riesgos críticos y significativos encontrados durante las dos primeras fases y su mitigación si se materializan.
- La creación de los manuales de usuario y de administración del sistema.

Esta parte está conformada por los capítulos 7, 8 Y 9, el primero de iguales características que las partes I y II, el segundo presenta generalidades sobre estándares que se utilizaron en la construcción del sistema , en cuanto a la documentación del código y mensajes, además se detallan dos subsistemas muy importantes para la implementación de la arquitectura. Y finalmente el capítulo 9 trata sobre la implementación de los subsistemas de la aplicación.

CAPITULO 7

PLANEACION Y ESTABLECIMIENTOS DE CRITERIOS DE EVALUACIÓN EN LA FASE DE CONSTRUCCION

Los criterios específicos de esta fase dependen de la iteración en que se encuentre. En el caso de este proyecto en particular estos deben ser los los criterios⁴⁵ a tener en cuenta para la evaluación final de la fase:

- ¿Se han identificado todos los requisitos, actores y casos de uso del sistema?
- ¿Se han completado el estudio de todos los casos de uso encontrados?
- ¿Se obtuvo al final de cada iteración una versión operativa del subsistema paquete de análisis a tratar en la iteración?
- ¿Se realizaron las pruebas de unidad con resultados satisfactorios?
- ¿Se elaboraron los manuales de usuario, técnicos y para la administración?
- ¿El sistema ha alcanzado su capacidad operativa inicial, es decir, puede ser considerado como una versión terminada del sistema?

ACTIVIDADES Y PRODUCTOS A ENTREGAR

Como se ha dicho, el principal producto a entregar en esta fase es la primera versión con capacidad operativa inicial del sistema. Además de las versiones finales de modelos de casos de uso y de la descripción de la arquitectura.

Dado que la atención principal de esta fase recae sobre los flujos de implementación y pruebas, sus actividades serán resaltados en los capítulos de esta parte del libro, los demás flujos solo se mencionarán en caso estrictamente necesario.

⁴⁵ Para ver mas indicaciones sobre la obtención de los criterios de esta fase, el lector puede remitirse a la página 369 del libro "El Proceso Unificado de Desarrollo de Software" de Ivar Jacobson, Grady Booch y James Rumbaugh. Primera edición. Addison Wesley. España, 2000.

CAPITULO 8

ESTANDARES DE IMPLEMENTACION

8.1 Introducción

Debido a la importancia que tiene el brindar mantenimiento a una aplicación específica, es importante fijar ciertos estándares para facilitar esta labor en el futuro. Depende de nosotros los desarrolladores crear aplicaciones bien documentadas tanto en su codificación, por fuera de ella (Manuales) y en la interacción con el usuario (mensajes del sistema)

8.2 CONVENCIONES DE CODIFICACIÓN

En esta sección se establecerán estándares para obtener un código fácil de entender y mantener en el sistema, y aunque algunas parecen obvias muchas veces por la ligereza o por simple descuido las olvidamos y perdemos la oportunidad de desarrollar un código fácilmente comprensible.

En cuanto a la construcción de las sentencias se tuvieron en cuenta las siguientes pautas:

- Dado que en PHP, lenguaje utilizado para el desarrollo, es sensitivo al uso de mayúsculas y minúsculas, se conserva el estándar seguido por el lenguaje, es decir, las palabras reservadas del lenguaje como funciones y variables propias de PHP se escribirán en minúscula, salvo las excepciones dadas por el mismo lenguaje.

Para la elección de nombres de variables se utiliza el siguiente estándar:

- Si la variable almacena información sobre un campo, una tabla o un registro de la base de datos, entonces la variable llevará el nombre del objeto de la base de datos al que hace referencia.
- Las variables de sesión también seguirán este estándar.
- Las demás variables, describirán el contenido de los datos almacenados en ella y que faciliten la lectura del código.

- El nombre de los controles incluidos en los formularios, es igual al campo de la tabla de la base de datos que van a actualizar. Esto evita cometer errores en el momento en el que se realiza la consulta a la base de datos con los valores enviados a través de un formulario.
- El nombre de las páginas que hacen consultas, actualizaciones o borrados sobre la base de datos van terminadas en `_bd`.
- El nombre de los campos de una tabla de la base de datos tiene que ser lo más claro posible terminados con un sufijo que denota la tabla a la cual pertenecen. Por ejemplo el campo nombre de usuario de la tabla usuario será `nb_usr`.
Para adicionar de comentarios de explicación del código se seguirá el siguiente estándar de acuerdo a los dos tipos principales de comentarios:

- *Comentarios de línea*. Se colocan directamente dentro del código y explican cuestiones que allí surjan.
- *Comentarios de cabecera* (como los comentarios en las cabeceras de un archivo, módulo o de una función). Estos comentarios se utilizaran para informar al lector sobre cosas genéricas del archivo o sobre la siguiente pieza de código.

El siguiente es un ejemplo de comentario de cabecera de archivo:

Este caso es un *comentario de cabecera de función*, que incluye una descripción del propósito de la misma, los parámetros de la función y los resultados que serán obtenidos.

Tabla 39. Comentario de cabecera de función

```
<?php
/*
*****
funciones_comunes.php -- Archivo que contiene funciones que son de
uso común a todas las paginas.
*****
Autores: Carlos M. Caraballo G. y Mario R. Pomares C.
-- Última Modificación: Octubre 04/2003
*****
Funciones:
--genlista_select: Genera una lista de opciones de un objeto select con
base a una consulta a la base de datos. Devuelve el objeto Select.

--gen_anonimo: Genera el login de un usuario registrado como anónimo.

--gen_marquesina: Genera los boletines de la marquesina en la página de
inicio.

--genfechaactual: Genera la fecha actual.

--genlista_selectadmin: Genera una lista de opciones de un objeto Select
con base a una consulta a la base de datos. Devuelve el objeto Select
para la asignación de nuevas claves a usuarios (Administración).

--cadenaleatoria: Genera una cadena aleatoria

*****
*/
```

Se Estructurará el código en grupo de funciones lógicas. Para ello se utilizarán archivos para agrupar las funciones, no solo lógica sino físicamente.

En cuanto a los Datos de Entrada y Salida, se tuvo en cuenta:

- Validar los datos de entrada antes de realizar operaciones en la base de datos.
- Realizar la comprobación de cada una de las operaciones que manejan los datos como por ejemplo las instrucciones de inserción, borrado y modificación.

8.3 CONVENCIONES DE INTERFAZ

Para la interacción con el usuario se tuvieron en cuenta los siguientes aspectos genéricos en el diseño de la interfaz según Roger Pressman en su libro “Ingeniería del Software”⁴⁶.

- Preguntar por la verificación de cualquier acción destructiva que no sea común.
- Reducir al máximo la cantidad de información que debe ser memorizada por el usuario para ejecutar cualquier operación.
- Utilizar verbos de acción simples o frases verbales cortas para definir las operaciones que realizará el usuario (Ej. Revisar, Enviar, Elaborar Doc., etc.)
- Utilizar formatos de presentación que permitan una asimilación rápida de la información.
- Producir mensajes de error significativos.
- Utilizar Mayúsculas y minúsculas, tabulaciones y agrupaciones de texto para ayudar a la comprensión.
- Usar eficientemente el espacio disponible en la pantalla.
- Minimizar el número de acciones de entrada de datos que debe realizar el usuario.
- El formato para presentación de mensajes de error debe ser suficientemente llamativo.

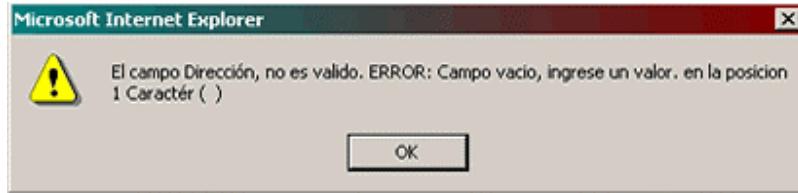
8.4 MENSAJES DEL SISTEMA

Los mensajes de error seguirán un estándar para su presentación y serán mostrados en una de las siguientes formas:

- *En una ventana propia del navegador.* Estos mensajes son generados por la capa de presentación, y se deben a validaciones o preguntas realizadas al usuario antes de enviar los datos al servidor para su procesamiento.

⁴⁶ Pressman, Roger. Ingeniería del software. Un enfoque práctico. Cuarta edición. McGraw Hill. España, 1998. Capítulo 4. Página 479-500.

Figura 46. Mensaje En una ventana propia del navegador



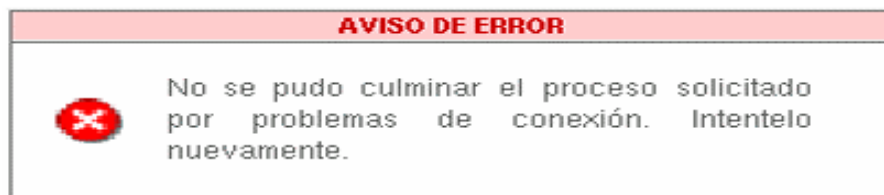
- *En la misma pagina que generó el error.* Este tipo de mensajes se muestra cuando el error generado no es critico para el sistema. Un ejemplo es cuando un usuario no introduce toda la información necesaria en un formulario o cuando la introduce incorrectamente.

Figura 47. Mensaje En la misma pagina que generó el error



- *En una pagina nueva.* Cuando sea un error o problema critico. Un ejemplo de este caso es cuando un usuario desea ingresar al sistema de manera ilegal.

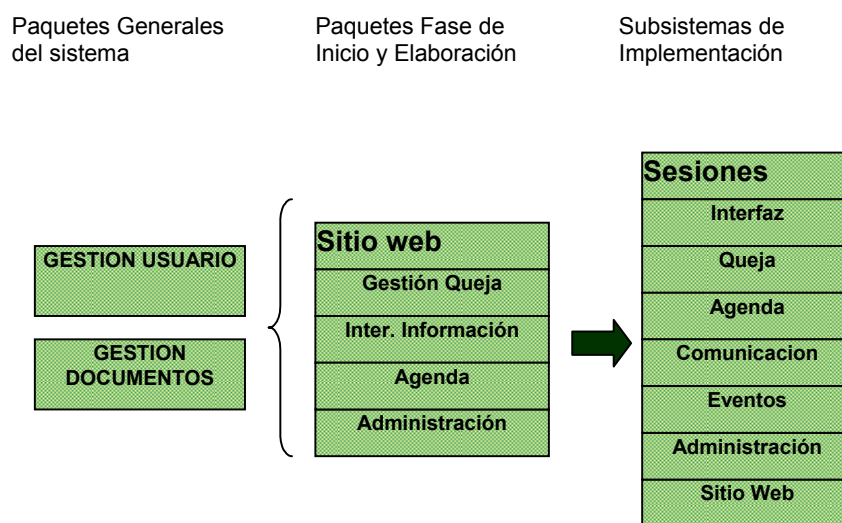
Figura48. Mensaje En una pagina nueva



8.5 IMPLEMENTACION

Durante este flujo se mostrara la forma en que se implementaron los paquetes identificados y utilizados a lo largo del desarrollo de cada flujo de trabajo de las fases de inicio y elaboración, los cuales llamaremos subsistemas de implementación, que proporcionan una forma de organizar los artefactos del modelo de implementación, como scripts y funciones, en trozos más manejables. La figura llegar a los subsistemas de implementación que explicaremos más adelante en esta sección.

Figura 49. Paquetes de las Fases de inicio y Elaboración y subsistemas de implementación



Es importante entender que un subsistema de implementación se manifiesta a través de un “mecanismo de empaquetamiento” concreto en un entorno implementación determinado, tales como un paquete en Java, un proyecto en Visual Basic o un directorio de archivos en C++ y en PHP.

El empaquetamiento a través de directorios es el caso particular de este proyecto y de trabajo. Los subsistemas de implementación están muy por ello se han estructurado una serie de directorios según los subsistemas diseñados en la fase de elaboración. La estructura general de los directorios se presenta en el Anexo D con una breve descripción de su contenido.

En cada uno de los directorios se encuentra una carpeta llamada plantillas que, como su nombre lo indica, contiene las plantillas utilizadas en dicho subsistema. De

esta forma en la implementación se mantienen independiente la capa de aplicación y la de presentación del sistema.

8.5.1 Subsistemas De Gestión De Sesiones E Interfaz

Dentro de esta sección se tratarán estos dos subsistemas debido a que son subsistemas generales de servicios y dan soporte a los demás. Ambos se han tratado ligeramente en la fase de elaboración ,describiendo algunos aspectos sobre sesiones y manejo de plantillas en php.

8.5.1.1 Subsistema De Gestión De Sesiones

En esta sección mostraremos otros aspectos importantes de este módulo en su implementación para el caso particular del desarrollo con PHP, el cual soporta en forma nativa el concepto de sesiones, básicamente es un sistema que preserva ciertos datos en una serie de accesos permitiendo construir aplicaciones más personalizadas.

En la etapa de implementación de este proyecto, estas sesiones fueron realizadas con ayuda del modulo de PHP para el manejo de las mismas, el cual soporta en forma nativa el concepto de sesiones, el cual es básicamente un sistema que preserva ciertos datos en una serie de accesos permitiendo construir aplicaciones más personalizadas.

El gestor de sesiones asigna a cada visitante registrado que acceda al sitio web un único id, el conocido id de sesión. Éste se almacena en una cookies del usuario o bien se propaga con la URL. El soporte de sesión le permite transportar tantas variables como se desee a través de las solicitudes del cliente.

Cuando un visitante registrado accede al sistema, PHP chequea automáticamente (si `session.auto_start` está puesta a 1), manualmente (si se añade el comando `session_start()`) o implícitamente (al añadir `session_register()`) si se ha establecido una sesión concreta con la llamada. Si es así, el entorno grabado es reproducido.

Como hay dos formas de propagar un id de sesión: por cookies o por parámetros URL, se ha implementado el sitio web de tal forma que de soporte a los dos métodos. Las Cookies son ideales, pero su permanencia no la controla el servidor, sino el cliente. El segundo método integra el id de sesión en las URLs. PHP es capaz de hacer esto de forma transparente cuando se ha compilado con `-enable-trans-sid`, en el archivo de configuración de php, `php.ini`. Si se activa esta opción las URLs relativas se cambiarán automáticamente para contener el id de sesión.

En el sistemas SIPEDCAR , cuando un usuario desea ingresar, envía su login y su clave de acceso. El sistema valida esta información y crea una nueva sesión registrando las variables que serán preservadas mientras se encuentre dentro del sistema. Para registrar las variables se utiliza el siguiente código.

Este segmento de código corresponde al sitio web, en el momento en que el administrador ingresa su nombre de usuario y clave para acceder a la página de administración, se realiza la siguiente secuencia:

1. Se establece una sesión concreta.
2. Las variables nombre de usuario(en este caso administrador) y clave son declaradas globales.
3. Se realiza la consulta respectiva, para ver si los datos suministrados se encuentran en la base de datos.
4. Estas variables son registradas como variables de sesión.
5. El sistema verifica que las variables de sesión hayan sido creadas, esto se hace con el fin de evitar entradas no autorizadas por el URL.
6. Finalmente accede a la página deseada, las variables de sesión se mantendrán hasta el momento en que el usuario ingrese a alguna página que contenga la instrucción `session_destroy()`.

Tabla 40. Código para el inicio de sesión y registro de las variables

```

<?php
session_start(); (1)
include "../plantillas/class.FastTemplate.php3";
include "../paginasphp/conexion.php";
include "../paginasphp/funciones_comunes.php";

function InicializarPlantillas() {
    global $tpl;
    $tpl = new FastTemplate("../plantillas");
    $tpl->define( array( paginaprincipal => "principaladmin.tpl",
        scriptreloj => "scriptreloj.tpl",
        scriptvalidacion => "valformingresoadmin.tpl",
        contenidooacceso => "contacesonegado.tpl",
        contenidoadmin => "contadmin.tpl" ) );
}

function ArmarPaginaPlantillas($nusuario,$clave) {
    global $tpl;
    global $nbadadminreg; (2)
    global $pswadminreg;
    $sw=1;
    $conex=ConectarBaseDatos();

    if((isset($nusuario))&&(isset($clave)))
    {
        $cons = "SELECT * FROM admin WHERE( nbadminpdc ='$nusuario' and
        clvadminpdc ='$clave)"; (3)
        $resultado = mysql_query($cons,$conex);
        $Num_filas = mysql_num_rows($resultado);
        if($Num_filas == 0)
        {
            $razon='Usuario no registrado en la Base de Datos';
            $tpl->assign(razon,$razon);
            $tpl->parse(contenidoprincipal,"contenidooacceso");
            $sw=0;
        }
        else {
            $nbadadminreg=$nusuario;
            $pswadminreg=$clave;
            session_register('nbadadminreg','pswadminreg'); (4)
        }
    }
    if($sw)
    {
        if(!session_is_registered('nbadadminreg') && !session_is_registered('pswadminreg'))
        (5)
        {
            $razon='Sesion de Usuario no registrada';
            $tpl->assign(razon,$razon);
            $tpl->parse(contenidoprincipal,"contenidooacceso");
        }
        else
        {
            $tpl->parse(contenidoprincipal,"contenidoadmin");
        }
    }
}

```

8.5.1.1 PRUEBAS

Tabla 41. Prueba uno subsistema de sesión

Estado del sistema:		
<i>El usuario aun no ha accedido al sistema. Por lo tanto la variable de sesión aun no ha sido inicializada y por lo tanto no se requiere ningún parámetro del sistema.</i>		
Parámetros		
Nombre	Valor	Descripción
-----	-----	-----

Tabla 42. Prueba con datos inválidos/ subsistema de sesión

Datos de entrada		
<i>Datos necesarios para validarse en el sistema al entrar a ingresoadmin.php</i>		
Parámetros		
Nombre	Valor	Descripción
<i>Nombre de Usuario</i>	<i>admin</i>	<i>válido</i>
<i>Clave de Usuario</i>	<i>xxxxxx</i>	<i>Invalido</i>
Resultado: Datos de Salida		
<i>Mensaje de Error sobre acceso al sistema. No se crea ninguna variable de sesión. Este mismo resultado se obtiene si el login y/o password son inválidos. "Usuario no registrado en la base de datos - sesión de usuario no registrada – Vuelva a la página de ingreso."</i>		

Tabla 43. Prueba con acceso ilegal / subsistema de sesión

Datos de entrada		
<i>Dirección web (que podría ser digitada a través del navegador) a la cual desea acceder En este caso administración.php</i>		
Parámetros		
Nombre	Valor	Descripción
<i>Nombre de Usuario</i>	<i>-----</i>	<i>Invalido</i>
<i>Clave de Usuario</i>	<i>-----</i>	<i>Invalido</i>
Resultado: Datos de Salida		
<i>Mensaje de Error sobre acceso al sistema. No se crea ninguna variable de sesión. "Acceso no autorizado, sesión de usuario no registrada – Vuelva a la página de ingreso administración e inicie una sesión válida."</i>		

Tabla 44. Prueba con acceso y datos válidos / subsistema de sesión

Datos de entrada		
<i>Datos necesarios para validarse en el sistema al entrar a ingresoadmin.php</i>		
Parámetros		
Nombre	Valor	Descripción
<i>Nombre de Usuario</i>	<i>admin</i>	<i>válido</i>
<i>Clave de Usuario</i>	<i>xxxxxx</i>	<i>válido</i>

Resultado: Datos de Salida
Muestra la página de entrada <i>administracion.php</i> , e inicia las variables de sesión.
identificador de sesión
sess_83485f25b964e1b9f727bed201809856

8.5.2 SUBSISTEMA DE GESTION DE INTERFAZ

El subsistema de gestión de interfaz hace uso principalmente de la clase FastTemplate. Dado que este subsistema es importante desde el punto de vista de la arquitectura, en la fase de elaboración se reseñaron algunas características de esta clase.

En la fase de construcción y siguiendo el diseño de la interfaz realizado durante la fase de elaboración se implementó el uso de plantillas en todas las páginas del sitio y del sistema interno.

Para ilustrar esto tenemos los códigos que se encuentran en la tablas 45,46 y 47 correspondientes a una página del sitio web, es importante para la comprensión del código, la Tabla 38 Métodos de la clase FastTemplate.

En la primera parte del código encontramos instrucciones “include” con la ruta y el nombre de los archivos que se utilizan. De estos archivos se utilizan las funciones que estos implementen como por ejemplo ConectarBaseDatos() es una función incluida en el archivo conexión.php. Esta función permite establecer la conexión con la base de datos del sitio.

Tabla 45. Código de la página noticias.php del sitio web.

```

<?php
include "../plantillas/class.FastTemplate.php3";
include "../paginasphp/conexion.php";

function InicializarPlantillas() {
    global $tpl;
    $tpl = new FastTemplate("../plantillas");
    $tpl->define( array( pagina => "pagina.tpl",
        tabla_noticias => "tabla_noticias.tpl",
        filas_tabla_noticias => "filas_tabla_noticias.tpl",
        tablaprincipal => "tablaprincipal.tpl",
        tb_menu_tituloimg => "tb_menu_tituloimg.tpl",
        pie_pagina => "pie_pagina.tpl",
        marcosup => "marcosup.tpl",
        pagina_nulo => "pagina_nulo.tpl",
        tabla_nulo => "tabla_nulo.tpl",
        tablaprincipal_nulo => "tablaprincipal_nulo.tpl" ));
}

function ArmarPaginaPlantillas() {
    global $tpl;
    $conex=ConectarBaseDatos();
    $cons = "SELECT * FROM noticias ORDER BY `fecha_ntc` DESC ";

    $resultado = mysql_query($cons,$conex);
    $Num_filas = mysql_num_rows($resultado);

    if($Num_filas == 0){require "../busqueda_nula.php";}else{

        $tpl->assign(barraimagen, Barra_Noticias);
        $tpl->assign(paginaregresoc, noticias);
        $tpl->parse(marcosuperior, "marcosup");
        $tpl->parse(menutitulo, "tb_menu_tituloimg");
        $tpl->parse(piepagina, "pie_pagina");

        while($fila=mysql_fetch_row($resultado)){
            $tpl->assign(cont_campo, "$fila[1]");
            $tpl->assign(cont2_campo, "$fila[2]");
            $tpl->assign(vinculo, "noticias_detalle.php");
            $tpl->assign(identconsulta, "$fila[0]");
            $tpl->parse(FILAS_TABLA, ".filas_tabla_noticias");
        }
        mysql_free_result($resultado);
        CerrarBaseDatos($conex);

        $tpl->parse(tabla_dinamica, "tabla_noticias");
        $tpl->parse(tablaprincipal, "tablaprincipal");
        $tpl->assign(titulo, Noticias);
        $tpl->parse(FINAL, "pagina");
        $tpl->FastPrint(FINAL);
    }
}

InicializarPlantillas();
ArmarPaginaPlantillas();
?>

```

En la función inicializarplantillas() se listan los archivos .tpl que van a conformar la interfaz. Un ejemplo son los archivos tabla_noticias.tpl – Tabla 46 y filas_tabla_noticias.tpl – Tabla 47. El primero forma la tabla en la que se van a presentar los datos y el segundo genera las filas de la tabla, una por cada registro de la tabla noticias de la base de datos.

Además encontramos palabras encerradas entre llaves, estas serán remplazadas por parámetros en el código de noticias.php como se muestra en la función ArmarPaginaPlantillas() en donde además se realiza la consulta a la base de datos.

Tabla 46. Código de tabla_noticias.tpl

```
<TABLE BORDER=0 WIDTH=100%>
{FILAS_TABLA}
</TABLE>
```

Utilizando los métodos de la clase FastTemplate se estructura la plantilla, finalmente se ejecutan las funciones que inicializan la plantilla y la que las arma.

Tabla 47. Código de filas_tabla_noticias.tpl

```
<TR>
<TD bgcolor="#CCCC99" height="35" width="85" ><div
align="center"><b><font face="Verdana, Arial, Helvetica, sans-serif"
size="1">{cont_campo}</font></div></TD>
<TD bgcolor="#FFE991"><div align="left"><a
href="/website/{vinculo}?id={identconsulta}"><font face="Verdana, Arial,
Helvetica, sans-serif" size="2">{cont2_campo} </font><a></div></TD>
</TR>
```

Para el caso del sistema interno, las páginas fueron diseñadas e implementadas con marcos(frames) siendo cada uno de ellos una página .php independiente agrupadas bajo una sola página .php y cada página utiliza sus plantillas (archivos .tpl) respectivas. Si desea conocer más sobre la utilización de plantillas y la clase FastTemplate se recomienda ver el Anexo C.

8.5.2.1 PRUEBAS

La Figura 50 es el resultado obtenido a partir de las plantillas descritas en la sección anterior.

Figura 50. Interfaz de noticias.php creada a partir de plantillas



CAPITULO 9

SUBSISTEMAS DE APLICACION

En este capítulo se presentarán cada uno de los subsistemas que dan origen a los módulos del sistema de información web SIPEDCAR (Incluyendo el sitio web). Para cada uno de ellos se han presentado un breve resumen con los aspectos más importantes para no saturar al lector con demasiada información, permitiéndole visualizar los aspectos más destacados en la implementación.

9.1. SITIO WEB

El sitio Web fue diseñado con una resolución (en píxeles) de 800 x 600 debido a que esta resolución es la más utilizada. Por otra parte fue estructurado mediante una jerarquía de malla completa (red o Web pura), en la cual todas las páginas del sitio están enlazadas entre sí teniendo definida una página de inicio.

Dentro del sitio encontramos una página principal o inicial, que se convierte en la página de entrada y varias páginas interiores o subpáginas donde se desarrollan los contenidos propuestos en la página principal, también encontramos las llamadas “ventanas flotantes” que brindan al visitante algún tipo de ayuda o información necesaria para hacer uso correcto de un servicio de la aplicación.

Figura 51. Interfaz de la página inicial del sitio web



Tabla 48. Zonas de la página inicial del sitio web

Zonas	Descripción
Cabezote	En esta parte se encuentra ubicado el logotipo de la Personería y en las páginas interiores además de éste se encuentran los elementos permanentes de la barra de navegación
Herramientas de búsqueda	El buscador presente en esta página da la oportunidad al visitante de seleccionar los temas y servicios que desea consultar o hacer uso de ellos respectivamente, además podrá acceder a algunos temas incluidos en estas páginas que no cuentan con un enlace directo, pero que podrían ser importantes para los intereses del visitante.
Menú de Navegación	Este menú de navegación contiene los vínculos de las secciones y servicios principales del sitio, como son: Ver Tabla 9.2
Boletines	Mediante este script se visualizarán los Boletines de Prensa generados por la oficina de Divulgación y Prensa de la Personería. Aparecerá publicado la fecha, el número del Boletín y un titular. Al dar clic sobre uno de ellos los llevará al documento completo.
Sección espacios de opinión	En esta sección de la página principal el visitante podrá participar de la encuesta que se realice en ese momento, votando por la opción deseada y visualizando inmediatamente los resultados. Por otra parte podrá opinar en el foro activo (sección opine), al dar clic sobre la pregunta del foro, el visitante es llevado a otra página donde dispone de la opción de insertar un nuevo mensaje, además observará los mensajes de los otros participantes.

Tabla 49. Opciones del menú de navegación del sitio web

Opciones del Menú de Navegación	
Información Institucional	En esta sección se ha organizado información referente a la normatividad, función, estructura organizacional, funcionarios, servicios ofrecidos, los horarios de atención, la trayectoria del personero y la reseña histórica.
Defienda sus derechos	En esta sección se ha incluido la información referente a los mecanismos con que cuenta Personería para ayudar o asesorar a la comunidad en la defensa de sus derechos. En algunos se encuentra la plantilla de los documentos correspondientes y la posibilidad de descargarlos.
Proyectos en ejecución	En esta sección encontrará vínculos de los principales proyectos que tenga la Personería. En la actualidad el sitio cuenta con información referente al proyecto del Personero Estudiantil, el cronograma de actividades y los centros educativos incluidos en el proyecto.
Presupuestos	Se podrá revisar o descargar los presupuestos de los diferentes años.
Informe de gestión	Se podrá revisar o descargar los informes de gestión de los diferentes años.
Noticias	Como su nombre lo indica este enlace nos lleva a la página de noticias generadas por la oficina de divulgación y prensa.
Contratación	En este enlace se podrá revisar las licitaciones realizadas por la personería antes y después de la fecha actual. Su objeto, fecha de apertura, fecha de cierre y presupuesto, además se podrá descargar el documento completo.

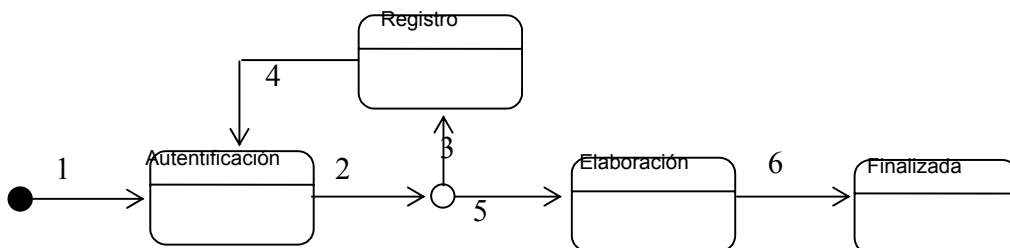
La interfaz de las páginas interiores Figura 52 encontramos, el cabezote, el menú de navegación y el área editorial o de contenido, que puede ser estático o generado por consultas a la base de datos

Figura 52. Interfaz de las páginas interiores del sitio web



El navegante puede interponer una queja a través del sitio en el link *quejese* , siguiendo el proceso descrito en el diagrama de estado de la Figura 53, para consultar el estado de su queja lo hará a través del link *consulte*.

Figura 53. Diagrama de estados interposición de queja por Internet

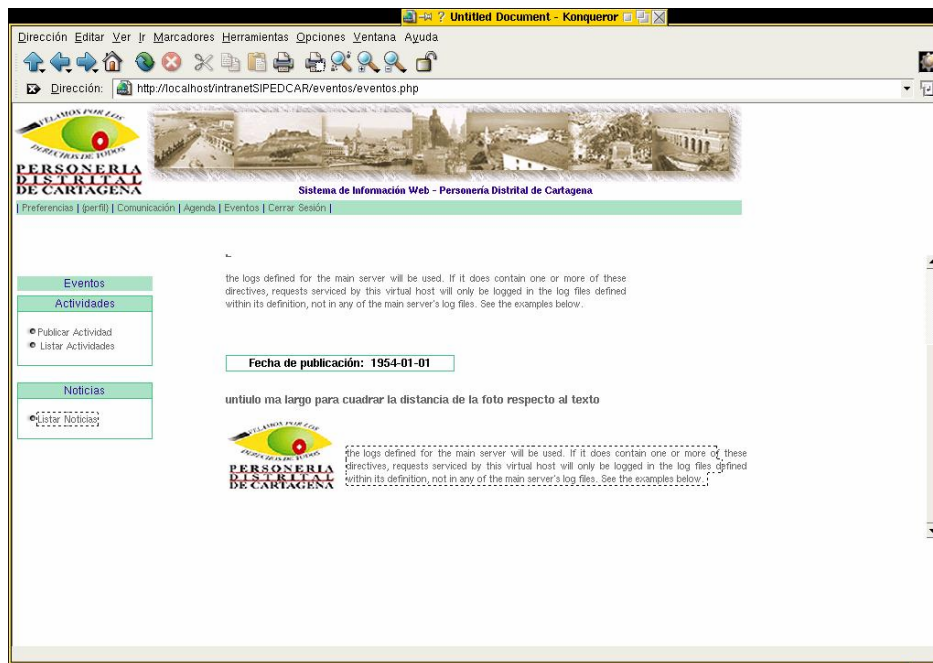


- 1: Ingresar autenticación de usuario
- 2: Verificar nombre de usuario y clave
- 3: Usuario no registrado : Registrar
- 4: Volver Autenticar
- 5: Usuario registrado: Elaborar queja
- 6: Enviar: Finalizar interposición

9.2. SUBSISTEMA DE EVENTOS

Este subsistema se encuentra en la zona común del sistema, como vemos todos los perfiles pueden listar las actividades, noticias y publicar una actividad. El administrador es el único que puede publicar las noticias generadas por la oficina de divulgación y prensa de la Personería, además de modificar y borrar una actividad.

Figura 54. Interfaz de la página del subsistema eventos



9.2.1 Pruebas

Las pruebas de este módulo están centradas en la verificación de las restricciones para los diferentes perfiles en cuanto a los casos de uso permitidos en este subsistema.

A continuación se resume las diferentes opciones a los cuales tiene acceso un usuario según su perfil.

Tabla 50. Acceso de los usuarios a las opciones de eventos, según su perfil.

Perfil	Acceso a:
--------	-----------

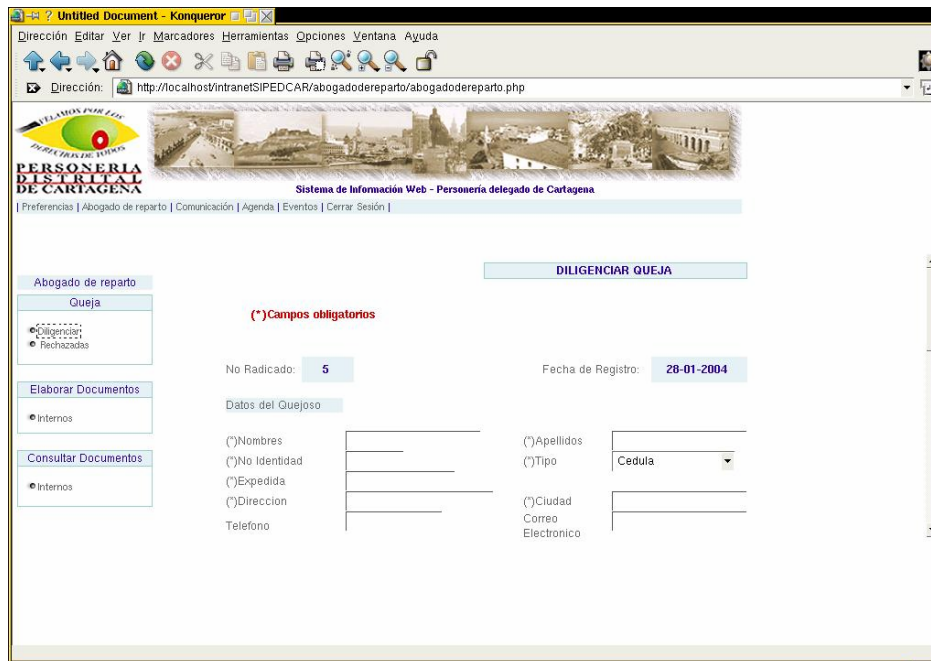
<i>PERSONERO DISTRITAL</i>	<ul style="list-style-type: none"> • Listar actividad y noticia • Publicar actividad
Personero Delegado	<ul style="list-style-type: none"> • Listar actividad y noticia • Publicar actividad
Abogado de reparto	<ul style="list-style-type: none"> • Listar actividad y noticia • Publicar actividad
Secretaria	<ul style="list-style-type: none"> • Listar actividad y noticia • Publicar actividad
Administrador	<ul style="list-style-type: none"> • Publicar noticias • Listar actividad y noticia • Publicar actividad • Modificar actividad • Borrar actividad
Funcionario	<ul style="list-style-type: none"> • Listar actividad y noticia • Publicar actividad

9.3. SUBSISTEMA DE QUEJA

Este subsistema soporta el proceso fundamental de la gestión de una queja dentro de la Personería. En el participan los Personeros, incluido el Distrital y el abogado de reparto, en este subsistema solo hay un caso de uso que es utilizado por el Personero Distrital como es el consultar los informes de los Personeros Delegados. Existen dos maneras de interponer una queja, en la oficina de quejas y reclamos o cuando se hace directamente con un Personero. En el primero de los casos la queja es diligenciada por el abogado de reparto y asignada a un Personero. El perfil Personero Distrital o Delegado debe seleccionar el caso de uso asignadas, para listar las quejas que tiene por resolver, y decide si acepta o la rechaza. Si es rechazada, el abogado de reparto la tendrá dentro de las quejas rechazadas para reasignarlas. Si la queja es aceptada, la queja se debe listar cuando el personero seleccione el caso de uso revisar. Allí podrá modificarla y darle un estado.

En el caso de que la queja sea interpuesta en forma directa a un personero, el mismo diligencia y por lo tanto queja aceptada queda aceptada.

Figura 55. Interfaz de la página del subsistema queja



9.3.1 Pruebas

En la tabla que se encuentra a continuación observamos las opciones del subsistema queja. Ver Anexo I – Escenarios de prueba.

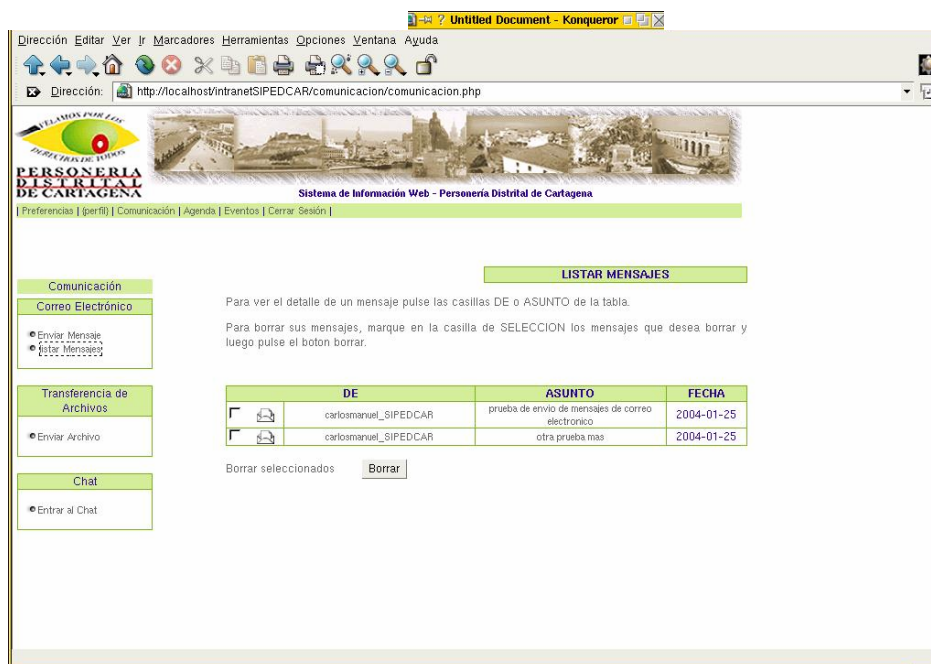
Tabla 51. Acceso de los usuarios a las opciones de quejas según su perfil.

Perfil	Acceso a:
<i>PERSONERO DISTRITAL</i>	<ul style="list-style-type: none"> • Consultar queja • Revisar queja • Elaborar informes • Consultar informe • Consultar informes de los delegados • Diligenciar queja • Elaborar documentos legales
Personero Delegado	<ul style="list-style-type: none"> • Consultar queja • Revisar queja • Elaborar informes • Consultar informe • Diligenciar queja • Elaborar documentos legales
Abogado de reparto	<ul style="list-style-type: none"> • Diligenciar queja
Secretaria	
Administrador	
Funcionario	

9.4. SUBSISTEMA DE COMUNICACIÓN

Este subsistema contiene los caos de uso relacionados con los mecanismos para el intercambio de información como es el correo electrónico, chat y ftp. Aplicaciones bastante conocidas en los ambientes web.

Figura 56. Interfaz de la página del subsistema queja



9.4.1 Pruebas

En la Tabla 52 se resume las diferentes opciones a los cuales tiene acceso un usuario según su perfil.

Tabla 52. Acceso de los usuarios a las opciones de comunicación según su perfil.

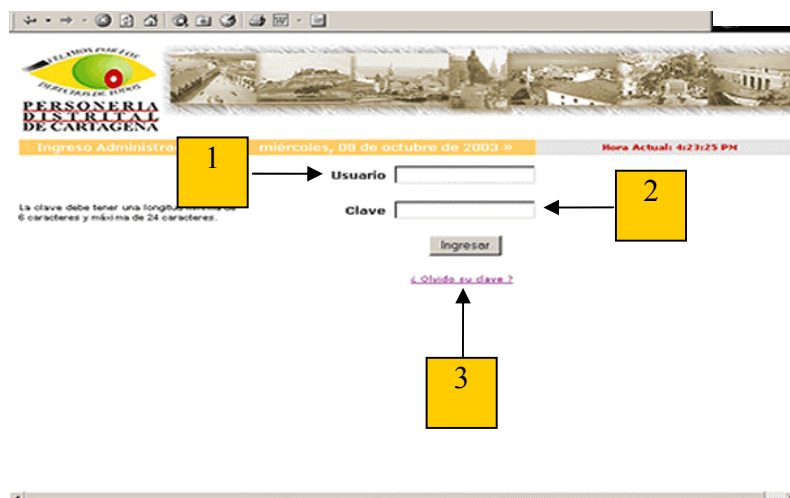
Perfil	Acceso a:
<i>PERSONERO DISTRITAL</i>	<ul style="list-style-type: none"> • Enviar correo • Chat • Transferir archivos
Personero Delegado	<ul style="list-style-type: none"> • Enviar correo • Chat • Transferir archivos
Abogado de reparto	<ul style="list-style-type: none"> • Enviar correo

	<ul style="list-style-type: none"> • Chat • Transferir archivos
Secretaria	<ul style="list-style-type: none"> • Enviar correo • Chat • Transferir archivos
Administrador	<ul style="list-style-type: none"> • Enviar correo • Chat • Transferir archivos
Funcionario	<ul style="list-style-type: none"> • Enviar correo • Chat • Participar en espacios de discusión • Transferir archivos

9.5. SUBSISTEMA ADMINISTRACIÓN

El módulo de administración para el sitio web consta de una serie de páginas que contienen enlaces a las aplicaciones que permiten añadir, borrar y actualizar; información en la base de datos.

Figura 57. Interfaz de la página de ingreso a administración



Luego de ingresar el nombre de usuario y la clave de acceso del administrador(1 y 2 de la Figura 57), ingresamos a una página que contiene las opciones de administración (Figura 58), ver Tabla 53.

Figura 58. Interfaz de la página de ingreso a administración

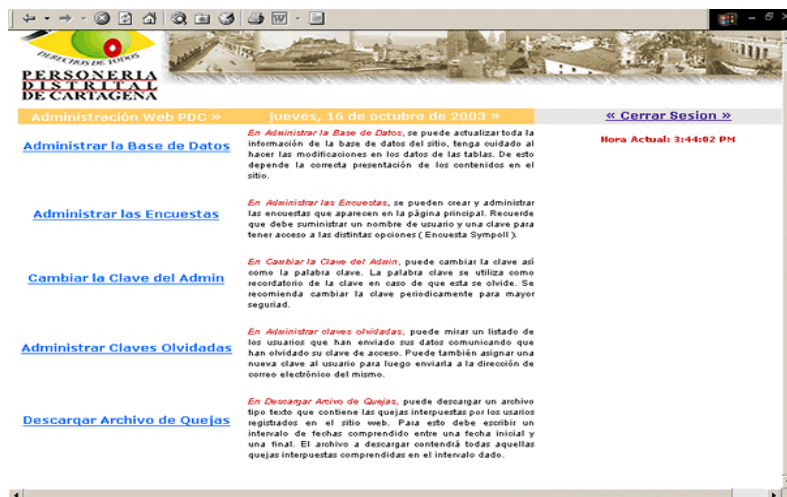


Tabla 53. Opciones de administración del sitio web

Opción	Descripción
Administrar la base de datos	Este enlace abre una nueva página a la aplicación de administración de la base de datos phpMyadmin ⁴⁷ .
Administrar las encuestas	Este enlace abre una nueva página a la aplicación de creación y administración de las encuestas que aparecen en la página principal del Sitio. Se solicita un username y un password; por defecto el valor del username es <i>admin</i> y el password es <i>personeria</i> . Ese administrador de encuestas (sympoll), brinda la posibilidad de crear, modificar, activar o desactivar una encuesta; en la creación de estas solo se dispone de máximo cuatro opciones de voto por encuesta.
Cambiar la clave del administrador	En ésta página se solicita una nueva clave así como también una palabra clave asociada a esta, para recordar la clave en caso de olvido.
Administrar claves olvidadas	Este enlace abre la página donde el administrador selecciona de un listado, los usuarios (que olvidaron su clave) a quienes asignará una nueva clave de acceso. Un listado similar aparece, donde se muestra la dirección electrónica del usuario y la nueva clave que le fue asignada .
Descargar Archivo de Quejas	Este enlace abre la página donde el administrador ingresa un intervalo de fechas, que comprende las quejas interpuestas por los usuarios registrados en el sitio web. El formato de las fechas es (aaaa-mm-dd). Luego muestra la página que contiene el archivo creado con las quejas registradas y que se puede descargar,

⁴⁷ Aplicación en PHP para la administración de bases de datos MySQL, Para mas información ver la página www.phpmyadmin.net.

haciendo clic en el link respectivo.

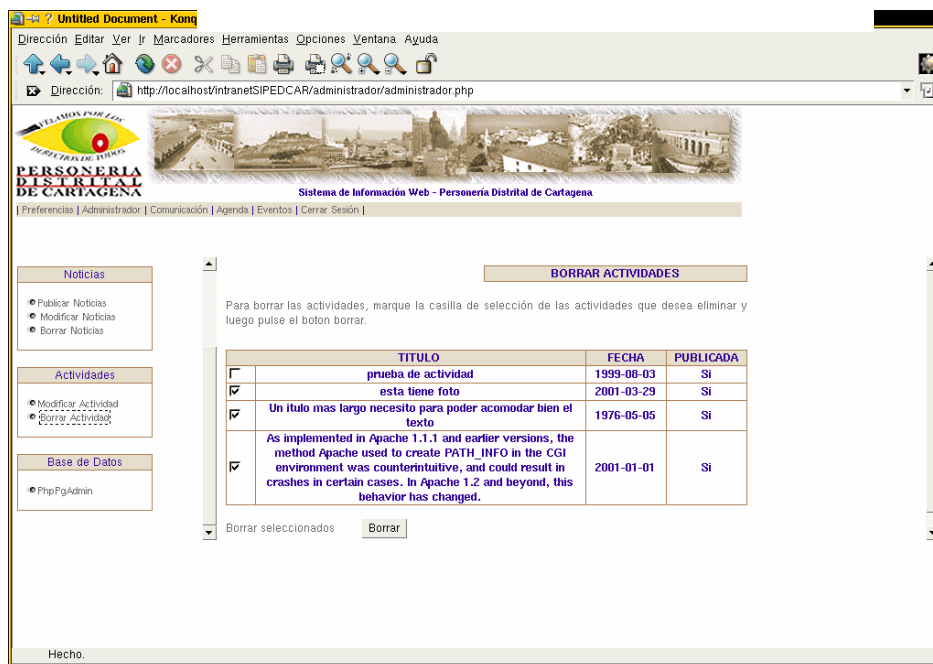
En el sistema interno el administrador tiene acceso a las opciones que aparecen en la Tabla 54, como opciones avanzadas de este perfil, se realiza un enlace desde este sistema hacia phpPgAdmin⁴⁸, aplicación para el manejo directo de la base de datos.

Tabla 54. Opciones de administración

Perfil	Acceso a:
<i>PERSONERO</i> <i>DISTRITAL</i>	
Personero Delegado	
Abogado de reparto	
Secretaria	
Administrador	<ul style="list-style-type: none">• Crear cuenta de usuario• Modificar cuenta de usuario• Crear Foro (SW)• Modificar Foro (SW)• Borrar Foro (SW)• Borrar Mensaje de Foro (SW)• Administrar chat• Crear encuesta (SW)• Modificar encuesta (SW)
Funcionario	

⁴⁸ Aplicación en PHP para la administración de bases de datos postgres.

Figura 59. Interfaz de la página del perfil administrador



9.7. RESUMEN

La evaluación de esta fase se realiza con base en los resultados de las pruebas y en la revisión final del modelo de caso de usos que permite comprobar que el producto ha alcanzado su capacidad operativa inicial. Por lo tanto, el principal producto que se obtiene de esta fase es el sistema software ejecutable.

La vista de la arquitectura durante esta fase tiene pocas variaciones ya que se definió desde la fase de elaboración y en la construcción sólo se realizaron leves modificaciones y actualizaciones a la misma.

9.7.1 Elaboración de Manuales de Usuario.

Para concluir, se elaboró el plan de la fase de transición y los manuales de usuario – *Manuales del sistema*. Estos manuales son:

- *Manual de procedimientos técnicos*: Este manual abarca la documentación sobre la estructura del sistema desde el punto de vista técnico, el cual será

obtenido a partir de la configuración del servidor y del diseño de la base y de los casos de uso involucrados en los diferentes subsistemas.

- *Manual operativo*: Necesario para que el usuario realice su actividad diaria; la documentación de este manual ha de estar orientada a los casos de uso que realizan los diferentes perfiles de usuario del sistema, de desarrolló para el sistema interno.
- *Manual de procedimientos administrativos*: Es esencialmente un manual operativo pero dirigido al administrador del sistema, contendrá las pautas que permitan el mantenimiento de la información ya sea a través del sistema o directamente en la base de datos que lo soporta. Además incluirá guías para la instalación, configuración y puesta en marcha del sistema y esquemas para el manejo de situaciones de contingencia en caso de una caída del sistema.

CONCLUSIONES

- La utilización de la metodología del Proceso Unificado para desarrollo de software fue una decisión acertada ya que permitió identificar desde un comienzo los riesgos críticos que podrían poner en peligro la ejecución del proyecto y diseñar los caminos alternativos para mitigarlos, nos dio la posibilidad de medir y cuantificar los productos resultantes a medida que se alcanzaban los hitos al final de cada fase y tomar decisiones cruciales de continuar o no el proyecto. Al establecer temprano una arquitectura candidata eliminamos el riesgo de no poder incorporar fácilmente los cambios que podrían presentarse.
- Los modelos (casos de uso, análisis, diseño, despliegue, implementación y prueba) y diagramas que proporciona la metodología son claves para la descripción de todas las perspectivas diferentes del sistema, haciéndolo claro para todos y satisfaciendo sus necesidades de información.
- Debido a que la construcción de cada subsistema se realizó a través de iteraciones incrementales, según lo recomendado por la metodología, se hizo evidente que el seguimiento de estándares de implementación para el diseño de la interfaz, para la documentación de código y para la organización de los subsistemas en el servidor, facilita la integración de un nuevo subsistema al sistema actual, así como la reutilización de código y funciones que pertenecen a los subsistemas existentes. Todo esto induce a concluir que el desarrollo basado en iteraciones incrementales junto con el seguimiento de estándares, permite obtener un sistema que puede evolucionar a través del tiempo.
- El conocimiento adquirido sobre Software Libre⁴⁹, y la experiencia obtenida gracias al desarrollo de este proyecto con herramientas creadas bajo este

⁴⁹ "Software Libre" se refiere a la libertad de los usuarios de correr, copiar, distribuir, estudiar, cambiar y mejorar el software. El termino no se refiere a que sea gratis o que no se pueda comercializar. Para mas información, consulte la página www.gnu.org/home.es.html.

criterio, permite asumir un compromiso frente a la utilización de software libre como elemento fundamental en la modernización de las entidades publicas , ya que les representa una opción viable, tanto económica como tecnologicamente, permitiendo el desarrollo de nuevos proyectos sin la inversión de grandes sumas de dinero, sin quedar atados a las restricciones de las licencias o contratos impuestos por los creadores de software propietario y por supuesto, sin comprometer la seguridad, estabilidad, escalabilidad y el desempeño de estas aplicaciones.

- Al construir el sistema utilizando plantillas en PHP a través de la clase FastTemplate se obtiene una separación del diseño y la lógica de programación, es decir el código php y las consultas a la base de datos están separadas del código html y javascript encargado de la presentación y validación de los datos del lado del cliente , esto nos presenta un sistema eficiente, con mayor facilidad y rapidez para su mantenimiento y actualización.
- Con el desarrollo de este sistema la Personería Distrital de Cartagena cumple con el compromiso de hacer parte de un estado eficiente y transparente, que hace uso intensivo de las Tecnologías de la información, para prestar servicios a los ciudadanos en la oportunidad y con la calidad debidos a través de un óptimo desempeño de sus funciones. Se adoptó la Estrategia de Gobierno en línea y la comunidad podrá contar con una extensión de los servicios de la Personería sin necesidad de desplazarse a las oficinas, haciéndolo a través del sitio web.
- La realización de este proyecto se constituye en un aporte importante al Grupo de Investigación de Sistemas y Tecnologías de Información – STI, el cual ha enfocado sus esfuerzos en la apropiación de conocimiento alrededor del desarrollo de sistemas de información basados en tecnologías Internet - Intranet y enfocado además en el campo de la Informática Jurídica.

RECOMENDACIONES

- Entre las posibles mejoras sugeridas para el sistema se encuentra la implementación de aspectos regulatorios relacionados con firma digital y entidades certificadoras, características que no se consideraron en el ámbito de este proyecto.
- La creación de un módulo que elabore estadísticas apropiadas, con base en los datos recopilados por el sistema. Estas estadísticas, por ejemplo, pueden ayudar a generar informes de gestión, a detectar problemas en el flujo y asignación de quejas a las diferentes Personerías Delegadas evidenciados en la demora en la solución de algunos procesos.
- SIPEDCAR fue desarrollada utilizando tecnologías de punta, pero debido a la continua evolución de las herramientas de desarrollo es necesario mantener un equipo no solo de administración sino también de estudio y evaluación de las nuevas tecnologías para hacer de este sistema un proyecto en continua evolución.
- El equipo de administración y desarrollo debe pensar en extender la lista de documentos a elaborar en el sistema de acuerdo a la frecuencia con que se utilicen, es decir construir las respectivas plantillas, interfaz y requerimientos en la base de datos para que los usuarios puedan contar con una lista bastante completa de todos los documentos que se elaboran.
- El sistema fue desarrollado en un pc de mediano rendimiento. Para posibilitar su acceso al público es necesario migrar el sistema a un computador que ofrezca mejores características de desempeño, por esto se debe contar a corto plazo con un equipo disponible para ello. Para el montaje del sitio web en un servidor de web hosting recomendamos seguir los requisitos establecidos en este documento para garantizar su correcto funcionamiento y no afectar su rendimiento.

- Se recomienda la realización de copias de seguridad de los datos del sistema para evitar daños o pérdidas de los mismos, estas copias deberán realizarse de acuerdo a lo planteado en el manual del administrador del sistema. La frecuencia de las mismas será determinada por la cantidad de información que maneje el sistema. Sin embargo se recomienda que sea semanal o máximo cada quince días.

BIBLIOGRAFÍA

- Bobadilla, Jesús - Alcocer Alejandro y otros. Superutilidades para Webmasters. McGraw Hill. Primera edición. España, 1999.
- C.J, Date. Introducción a los Sistemas de Bases de Datos. Prentice Hall. Séptima edición, México, 2001.
- Evans, Tim. Construya su propia Intranet. Segunda edición. Prentice Hall. México, 1997.
- FROSINI VICTORIO, INFORMÁTICA Y DERECHO, EDITORIAL TEMIS S.A.
- Gil Javier, Tejedor Jorge, Yague Agustín, Santiago Alonso Villaverde, Abraham Gutiérrez. Creación de sitios web con PHP 4 . McGraw Hill.
- Gómez F, Luis Carlos. Guía para el Desarrollo de Proyectos de Grado. Bucaramanga. UIS, 1993.
- Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Primera edición. Addison Wesley. España, 2000.
- Jordan Florez Fernando, La Informática Jurídica (teoría y práctica), Universidad Piloto de Colombia, Editora Guadalupe LTDA, Bogotá (Col), 1987.
- Lockhart, Thomas. The PostgreSQL Administrator's Guide. The PostgreSQL Global Development Group. 1998.
- Lockhart, Thomas. The PostgreSQL Programmer's Guide. The PostgreSQL Global Development Group. 1998.
- Lockhart, Thomas. The PostgreSQL Tutorial Introducion. The PostgreSQL Global Development Group. 1998.
- Lockhart, Thomas. The PostgreSQL User's Guide. The PostgreSQL Global Development Group. 1998.
- Parra Benítez Jorge, Informática Jurídica y Derecho Informático, Universidad Pontificia Bolivariana, Señal Editora, Medellín (Col) , 1994.

- Personería Distrital de Cartagena de Indias. Manual de funciones, y normatividad. Personería Distrital, 2001.
- Pressman, Roger. Ingeniería del software. Un enfoque práctico. Quinta edición. McGraw Hill. España, 1998.
- Petersen, Linux, Manual de Referencia. Segunda edición. Mc Graw - Hill. México, 2001.
- Tanenbaum, Andrew. Redes de computadores. Prentice Hall. Prentice Hall, USA, 1998
- Téllez Julio, Derecho Informático, Editorial McGraw Hill.

- Orlando Solano B., Manual de informática Jurídica, Ediciones Jurídicas Gustavo Ibáñez, Bogotá (Col.),1997.
- Thomas A. Powell. Diseño de sitios web. McGraw Hill.

- Howto – Linux
- URL:<http://www.phpbuilder.com> Sitio de Desarrolladores en PHP.
- URL:<http://www.apache.org>. Sitio oficial de Apache.
- URL:<http://www.php.net>. Sitio oficial de PHP
- URL:<http://www.postgres.com>. Sitio oficial de Postgres
- URL:<http://www.redhat.com>. Sitio oficial de RedHat
- URL:<http://www.hotscripts.com>. Sitio de superutilidades en PHP y otros lenguajes.
- URL:<http://www.mysql.com> Sitio oficial de MySQL.
- URL:<http://www.rational.com/uml> . Sitio oficial de Rational / UML
- URL:<http://www.microsoft.com/dna/>. Sitio de la arquitectura DNA de Microsoft.

En este apéndice se encuentra un resumen sobre la informática jurídica (definición, clasificación y aplicaciones) resultado del estudio y la investigación realizada por la línea de informática jurídica, perteneciente al Grupo de investigación en sistemas y tecnologías de la información S T I ⁵⁰ de la Universidad Industrial de Santander, sobre algunos textos de autores Colombianos y otros internacionales existentes en este campo. Para ampliar esta información dirigirse a la dirección de Internet que se encuentra en el pie de página.

DEFINICION

La informática, como uno de los fenómenos más significativos de los últimos tiempos, ha encontrado muchos campos de acción en los cuales deja sentir su influencia. Prácticamente todas las áreas del conocimiento humano en mayor o menor proporción han experimentado sus beneficios, dentro de las cuales el Derecho no puede ser la excepción, dando lugar a la llamada Informática Jurídica, nacida propiamente en 1959 en los Estados Unidos. Ante la diversidad de los términos utilizados para significar el ingreso de la informática al derecho, es necesario crear una mentalidad en la cual no se rehúse a considerar el impacto de la tecnología en el ámbito del derecho.

Este fue el propósito de distinguidos juristas⁵¹ en el mundo, el de cuantificar los fenómenos jurídicos o crear un nuevo método para estudiarlos, tratando de conjugar lo jurídico y lo tecnológico. Una acertada definición de informática jurídica , teniendo en cuenta lo anterior, es la siguiente:” La utilización de los diferentes conceptos, categorías, métodos y técnicas propios de la informática en el ámbito de lo jurídico. Se relaciona con la creación, flujo, clasificación, organización, sistematización y utilización de datos requeridos en la producción y/o en la administración de lo jurídico” ⁵²

⁵⁰ <http://www.uis.edu.co/investigacion/grupos/sti/STI/documentosframe.htm>
marco teórico, estado del arte y resúmenes de los proyectos).

⁵¹ Loevinger (Jurimetrics), Paul S. Hoppman (Lawtomatic), Philippe Dreyfus(Informatique juridique), Vittorio Frossini (Juritécnica), Mario G. Losano(Giuscibernética).

⁵² Fernando Jordan (Opus cit. P.49). Tomado del libro Manual de Informática Jurídica de Orlando Solano .

Otras definiciones

- ❑ **Mario Lozano:** Considera que la historia del derecho “ está condicionada por las tres revoluciones, de la escritura, de la imprenta y de la ordenación electrónica de datos”. Los grandes sistemas jurídicos, Madrid. Debate 1982.
- ❑ **Julio Téllez:** “Técnica interdisciplinaria que tiene por objeto el estudio e investigación de los conocimientos de la informática en general, aplicables a la recuperación de información jurídica, así como la elaboración y aprovechamiento de los instrumentos de análisis y tratamiento de información jurídica necesarios para lograr dicha recuperación”. Derecho Informático.
- ❑ **Edgar Salazar:** “Ciencia del tratamiento racional y automático de la información de contenido jurídico”.

CLASIFICACION

La informática jurídica se puede clasificar en tres grandes ramas dependiendo en que centran su atención cada una de ellas y la manera como fueron apareciendo.

- ❑ Informática jurídica documentaria.
- ❑ Informática jurídica de control y gestión. (Operacional y registral).
- ❑ Informática jurídica metadocumentaria o decisional.

Informática Jurídica Documentaria.

Centra su atención en el *Documento*, es decir en el almacenamiento y recuperación de información que contienen datos principalmente jurídicos, documentos como:

Leyes: Proyectos de Ley, Proyectos de actos legislativos, Constitución, Leyes, Decretos, Ordenanzas, Acuerdos, Resoluciones.

Jurisprudencia: Sentencias de la Corte Suprema de Justicia, del Consejo de Estado,
de Tribunales, de los Juzgados.

Doctrina: Doctrina de tratadistas nacionales e internacionales.

Textos jurídicos: Poderes, Contratos, Memoriales, Escrituras.

- **Principales Características**

La finalidad de la información en un sistema documentario consiste en encontrar lo más rápida y pertinentemente posible la información que ha sido almacenada. El conjunto de esas informaciones constituye el banco de datos o corpus.

La interfase de almacenamiento-recuperación de información jurídica induce los siguientes procesos:

- ❑ *Entrada de documentos:* (Leyes, reglamentos, jurisprudencia, doctrina, acuerdos, etc.) Esta entrada de documentos varía dependiendo del método que el sistema maneje, por ejemplo si el texto se toma y se le da entrada tal como es, o si se le hace un análisis previo para una condensación, donde solo contenga elementos característicos de este.
- ❑ *Búsqueda de documentos:* Este proceso se lleva a cabo para obtener información almacenada en el banco de datos. Los criterios de búsqueda varían dependiendo del sistema, sin embargo existen búsquedas comunes como, búsqueda por fecha, por tema, título.

• Esquemas De Representación Documentaria

Uno de los principales problemas que se enfrenta al construir un sistema de documentación jurídica, está en función del lenguaje jurídico, ya que este no describe al derecho sino que es parte de él. Las palabras que integran un documento no pueden ser consideradas como unidades fundamentales con sentido cada una, esto conlleva a que cada documento está caracterizado por las palabras utilizadas en él mismo. Actualmente existen dos métodos de almacenamiento de datos documentarios:

- ❑ *Método del texto integral (full text).* Como su nombre lo indica, consiste en introducir los textos tal cuales en el computador, su desventaja es la gran cantidad de espacio de memoria usado y la generación de ruido, pero su ventaja es que disminuye los riesgos de silencio informático.
- ❑ *Método de indización (key word).* Este método implica el análisis previo del documento para extraer y concretar las características fundamentales (conceptos, circunstancias, elementos de decisión, etc), es decir el *abstract*. Esto conlleva un costo por la necesidad de personal calificado, pero disminuye los riesgos de ruido.

Estos métodos se pueden utilizar conjuntamente, por ejemplo la indización puede dar un resumen que será introducido y explotado según los métodos del texto integral.

Informática Jurídica de Gestión y Control

Es el método o el sistema empleado que hace de toda actividad jurídica algo estructurado, algo más simple y efectivo. Sirve para organizar y hacer más funcionales los despachos que puedan de alguna forma tratar temas jurídicos, está dirigida a actuar dentro de un proceso fundamentalmente administrativo . Se subdivide en:

- ❑ **Informática Registral:** Se ocupa de todos los tipos de registros (públicos o privados).
- ❑ **Informática Operacional:** Permite el control de los pasos que en una clase de proceso específico se vayan cumpliendo.

Este tipo de informática jurídica se encuentra en sistemas de oficinas jurídicas: De reparto judicial, De registro Penitenciario, De notariado y registro, De contravenciones, De consultorios Jurídicos.

Según el tipo de información requerida para la administración de los asuntos públicos, se presentan diferentes tipos, tales como:

- ❑ Informática Normativa.
- ❑ Informática Diagnóstica.
- ❑ Informática Cognoscitiva.

Según el campo en que actúa:

- ❑ Según su uso en la Administración Pública.
- ❑ Según su uso en los órganos Jurisdiccionales.
- ❑ Según su uso en despachos y notarias.

Su uso en la Administración Pública: Mejoramiento sustancial de las estructuras jurídico-administrativas, agilizar tramites de asuntos jurídico-administrativos para contribuir a la disminución de la burocracia y corrupción especialmente en la administración de la justicia, permitiendo la impartición de una justicia rápida, expedita y particularizada.

Su uso en Órganos Jurisdiccionales: Originó a la informática Judicial, con gran desarrollo en la actualidad. Automatización de actividades tales como formulación agenda de jueces y magistrados, redacción automática de textos jurídicos a manera de sentencias.

Su uso en Despachos y Notarías: La Podemos conceptualizar como “Ofimática Jurídica “ (Automatización de oficinas de índole jurídico). En el campo notarial más que simple tratamiento de textos se ofrece un menú completo de las principales actividades con una organización, control y seguimiento de los mismos (agenda, estado de asuntos, registros, etc.).

En los despachos de abogados los sistemas automatizados simplifican un gran número de labores propias de dicho entorno (control de asuntos, honorarios, redacción y verificación de escritos, etc.).

- ❑ **Aplicaciones de Informática jurídica de gestión y control**⁵³

⁵³ La Informática Jurídica (teoría y práctica), Fernando Jordan Florez, Universidad Piloto de Colombia, Editora Guadalupe LTDA, Bogotá (Col), 1987

- Sistema SISJURI.
- Sistema JURISPROCE.
- Sistema OFILEX2000.

Informática Metadocumentaria o decisional.

Emplea el ordenador en la ayuda de toma de decisiones a través de la solución de casos (Inteligencia Artificial).

El señor Orlando Solano Bárcenas presenta en su libro *Manual de Informática Jurídica* la siguiente definición:

“La Informática Jurídica Decisional, actúa sobre los modelos que se pueden construir sobre la base de datos retenidos en registros previamente levantados con los términos de la Informática Jurídica Documental. Dichos modelos, puestos en funcionamiento, permiten explicar la realidad de la cual el modelo es una versión abstracta, que sirve para hacer factible una previsión de comportamiento frente a múltiples posibilidades. La Informática Jurídica Decisional utiliza una herramienta, los sistemas expertos. Estos le facilitan la toma de decisiones, por su inmensa capacidad para escoger la respuesta adecuada dentro de parámetros característicos de la situación que se ha de resolver”.

Es de notar que este autor contempla la Informática Jurídica Decisional como una parte especializada de la Informática Jurídica de Gestión

Por su parte, Carmen Cintron Ferrer expone en la conferencia titulada *La Aplicación de las Técnicas de Inteligencia Artificial a la Informática Jurídica*⁵⁴, una posible relación entre estas dos disciplinas:

“El uso de las tecnologías en los sistemas de justicia debe dirigirse por rumbos más positivos y prácticos. Los conceptos y métodos de la informática, particularmente la rama que denominamos *Inteligencia Artificial*, deben encaminarse al auxilio de los juristas en su obligación de impartir justicia en igualdad de condiciones. [...] Las decisiones emitidas por robots (máquinas inteligentes) no garantizan el que se imparta justicia. No obstante, la posibilidad de la existencia y uso de dichas máquinas, aunque distante, se acerca cada día más”.

El iusinformático Antonio Martino⁵⁵ define los sistemas expertos legales de la siguiente forma:

⁵⁴ Conferencia presentada en el I Congreso Iberoamericano de Informática Jurídica, Santo Domingo, Octubre 29 a Noviembre 2 de 1984.

⁵⁵ Tomado de *La Informática Jurídica Hoy*, en la revista de Derecho Industrial, Buenos Aires, Depalma, 1985 No.21.

“Un Sistema Experto Legal consiste en una serie de programas que consisten en obtener inferencias válidas a partir de una base de datos jurídicos estructurada, siguiendo recorridos no previstos, justificando cada recorrido en la indicación de las reglas aplicadas y poniendo a disposición una interfase hombre-máquina que facilite la introducción de nuevas reglas en los puntos en los cuales la base de conocimientos se revela incompleta o no actualizada. Un Sistema Experto Legal debería ser capaz de simular aquella parte del razonamiento jurídico (de un experto jurídico) que de las normas extrae conclusiones o de los precedentes consigue una regla general”.

Otros autores tienden a ver el objetivo de la Informática Jurídica Decisional muy similar al de la Informática Jurídica de Gestión, como es el caso de Hermilio Tomás Azpilcueta:

“Debe ser perfectamente comprendida (la Informática Jurídica Decisional) para no pensar en un automatismo de la decisión equivalente a una *iuscibernética*⁵⁶, sino en la ayuda que pueda prestar al juez en cuanto a establecer unas determinadas pautas en resoluciones muy repetitivas y facilitar la redacción de las mismas, bien al propio juez, bien al funcionario judicial que tenga que llevar a cabo la tarea de copia”.

- **Aplicación de la inteligencia artificial en la informática jurídica decisional**

Como menciona Solano, los sistemas expertos son una herramienta usada por la Informática Jurídica Decisional, siendo estos una aplicación de la Inteligencia Artificial.

La Inteligencia Artificial, como tal, se puede agrupar en tres grandes áreas o clasificaciones a saber:

1. Análisis automático y conocimiento del lenguaje (idioma natural).
2. Solución automática de problemas y prueba o demostración de teoremas.
3. Robótica.

Se descarta la aplicación de la robótica en el ámbito jurídico, porque el desarrollo de un jurista mecánico o de un sistema de justicia que dependa del uso de robots no se contempla en un futuro cercano. Las restantes categorías, por el contrario, tienen gran pertinencia y aplicabilidad en el ámbito jurídico.

Las posibles aplicaciones, según Cintron, pueden ser de dos clases, los sistemas de informática jurídica y los sistemas de análisis jurídico.

⁵⁶ La *iuscibernética* surge a raíz de la aparición y desarrollo de la cibernética. Es la idea de ver al sistema jurídico dentro de la cibernética, como ciencia general.

En los Sistemas de Informática Jurídica, la actividad inteligente hace referencia a la recuperación de la mayor cantidad de información jurídica pertinente del banco de datos utilizado. Resulta, pues, viable la implantación de un sistema automático de búsqueda conceptual. Otro aspecto es la posible utilización de las técnicas de respuesta a interrogantes en los bancos de datos jurídicos, o el establecimiento de bancos de datos que provean acceso a la información pertinente, sin que el investigador tenga que dedicarse a examinar y evaluar los documentos que resultan de la búsqueda.

El Sistema de Análisis Jurídico, es decir, un sistema jurídico experto, puede describirse como un sistema que organiza y expresa automáticamente un conjunto de conocimientos y produce para el usuario una recomendación acorde a las normas preestablecidas de la acción a seguirse. Los elementos que integran un sistema experto de esta naturaleza son los siguientes:

4. Capacidad para representar en abstracto la descripción de un caso en cuanto a los hechos.
5. Capacidad para representar en abstracto, y en forma lógica, los principios jurídicos aplicables al caso, y la capacidad para evaluar las implicaciones de la aplicación de las normas al caso.
6. La existencia de un mecanismo que efectúe el análisis jurídico; esto es, que seleccione las normas o principios pertinentes y los aplique.

Un programa inteligente aplicado al derecho contencioso, por ejemplo, trataría de simular el comportamiento de un juez cuando se enfrenta a un problema litigioso. En este caso, el programa no necesitaría pensar como un juez (y, por supuesto, tampoco sentir) sino que debe actuar como él. Sobre todo en lo relacionado con sus métodos de encontrar soluciones a determinados problemas; deducción, solución de acertijos, prueba de teoremas, combinaciones, y, sobre todo, de la teoría de las decisiones.

Naturalmente, para llegar a un “juez” mecánico óptimo se necesitaría del concurso de jueces humanos expertos que transmitan su experiencia.

- **Aplicaciones**

- Thomas Hendrik y Bruce Buchanan iniciaron en 1970 una serie de investigaciones sobre inteligencia artificial, que tiene como finalidad ayudar a los juristas a la comprensión de los criterios de razonamiento y solución de problemas de orden jurídico.
- Sistema *Taxman II* desarrollado por Mc Carty y Sridham, para títulos accionarios y derecho fiscal.
- Sistema *Hafner Legal Research System*, especializado en derecho cambiario.
- Sistema de Allen, de normalización del lenguaje jurídico.

- Sistema desarrollado por el Instituto para la Documentación Jurídica de Florida para el cálculo de montos en accidentes automovilísticos.

Hasta el momento, los autores no tienen conocimiento sobre el desarrollo de sistemas de este tipo en Colombia.

ANEXO B LENGUAJE UNIFICADO DE MODELADO - UML

A mediados de los noventa existían muchos métodos de análisis y diseño, lo que suponía que los mismos conceptos tenían distinta notación según el método de que se tratara. Ante esta situación, en 1994 Booch, Rumbaugh y Jacobson decidieron unificar sus métodos dando lugar a UML. Esta unificación fue promovida por el OMG de tal manera que UML se convirtió en la notación estándar para la descripción de métodos software. Según su definición, UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva orientada a objetos:

- Es un lenguaje: Proporciona un vocabulario y unas reglas que se centran en la representación conceptual y física de un sistema, y que indican cómo crear y leer modelos bien formados. Sin embargo, no dice qué modelos crear ni cuándo se deberían crear, ésta es la tarea del proceso de desarrollo de software.
- Es un lenguaje para visualizar: Es un lenguaje gráfico que mezcla gráficos y texto, pero es algo más que un simple montón de símbolos. De hecho, detrás de cada símbolo en la notación UML hay una semántica bien definida, de manera que un desarrollador puede escribir un modelo en UML, y otro desarrollador, o incluso otra herramienta, puede interpretar ese modelo sin ambigüedad.
- Es un lenguaje para especificar: Cubre la especificación de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema con gran cantidad de software.
- Es un lenguaje para construir: No es un lenguaje visual, pero sus modelos pueden conectarse de forma directa con una gran variedad de lenguajes de programación. Es posible establecer correspondencias desde un modelo UML a un lenguaje de programación como Java o C++, o incluso a tablas en una base de datos relacional o al almacenamiento persistente en una base de datos orientada a objetos. Permite ingeniería directa e inversa.
- Es un lenguaje para documentar: cubre toda la documentación de la arquitectura de un sistema y todos sus detalles. También proporciona un lenguaje para expresar requisitos y pruebas del software. Finalmente, UML proporciona un lenguaje para modelar las actividades de planificación de proyectos y gestión de versiones.

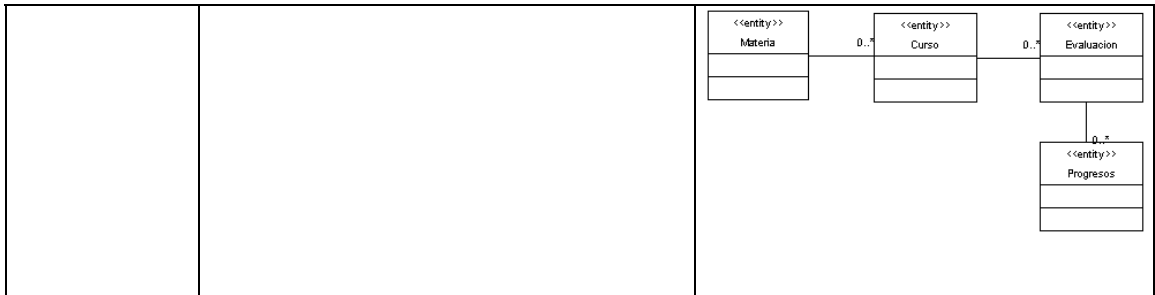
Desde la consolidación de UML como lenguaje estándar para el modelado se ha definido un buen número de procesos para el desarrollo de aplicaciones orientadas a objetos que utilizan este lenguaje como medio de expresión de los diferentes modelos que se crean durante el ciclo de vida. Las características principales deseables en cualquier proceso software basado en UML son:

- Un proceso debe ser iterativo e incremental, y debe centrarse en los aspectos críticos en
- las primeras iteraciones para minimizar riesgos.
- Debe estar guiado por los requisitos (casos de uso). Los requisitos cambian a lo largo del desarrollo del proyecto y el proceso debe estar preparado para identificar nuevos requisitos a lo largo de todo el ciclo de vida, ya que es muy difícil que puedan capturarse todos los requisitos antes de empezar la implementación.
- Debe utilizar arquitecturas basadas en componentes.
- Debe existir un control de cambios del software. La ausencia de un control de cambios hace que el proceso degenere rápidamente en un caos. Si se hace un control de cambios se solucionan parte de las dificultades principales del desarrollo de software, como la comunicación entre equipos de desarrollo, la consistencia, la interferencia entre miembros de un equipo que trabajan en paralelo, etc.

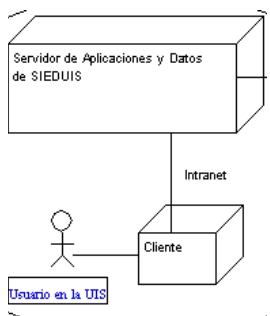
En todos los procesos basados en UML el concepto de caso de uso juega un papel primordial, ya que se emplea para definir los requisitos funcionales del sistema y en torno a ellos se articulan todas las etapas del proceso.

A continuación se presenta un resumen de los diagramas UML utilizados en la realización de este proyecto. Cabe resaltar que la conveniencia de utilizar un determinado diagrama está determinada por el aporte que este da al entendimiento del sistema. No está determinada ni por la metodología, ni por el propio lenguaje UML.

DIAGRAMA	DESCRIPCIÓN	EJEMPLO
De Casos de Uso	Representa las funciones del sistema desde el punto de vista del usuario. Se utilizan para la comunicación con los usuarios y para expresar de forma clara y sencilla los requisitos.	
De Clase	Bosquejan la arquitectura estática del sistema en función de clases y sus asociaciones.	



<p>De Secuencia</p>	<p>Representan un conjunto de elementos de un sistema que interaccionan entre ellos organizados en secuencias de tiempo.</p>	<pre> sequenceDiagram actor Student as :Estudiante participant Manager as Gestor de Sesiones : Student->>Manager: Solicitar Ingreso Manager-->>Student: Solicitar Identificacion Student->>Manager: Enviar login y password Manager->>Manager: Validar Datos Manager-->>Student: Enviar Nueva Sesión </pre>
<p>De Colaboración</p>	<p>Son parecidos a los diagramas de secuencia pero dan mayor libertad para distribuir los objetos. Representan la interacción entre elementos de un sistema y se organiza con respecto a espacio y tiempo. Es una representación espacial de los objetos, sus enlaces y sus interacciones. Junto con los de secuencias se denominan, diagramas de interacción.</p>	
<p>De Estado</p>	<p>Representan el estado de un elemento del sistema, las condiciones y respuestas que da a ciertos eventos a los que es expuesto.</p>	<pre> stateDiagram-v2 [*] --> Asignacion Pendiente Asignacion Pendiente --> Asignacion Entregada : Enviar Asignacion: Asignacion Entregada --> [*] </pre>

De Despliegue	Muestra un conjunto de nodos y sus relaciones. Representa el despliegue de los componentes y subsistemas sobre los dispositivos físicos.	 <p>El diagrama muestra un servidor de aplicaciones y datos de SIEDUIS (representado como un cubo) conectado a un cliente (representado como un cubo) a través de una línea etiquetada como 'Intranet'. Un actor humano (representado como un palo) interactúa con el cliente. Una etiqueta 'Usuario en la UIS' apunta al actor.</p>
---------------	--	--

Tomada del proyecto: SISTEMA INTRANET DE INFORMACIÓN PARA EL APOYO DE LA ACTIVIDAD ACADÉMICA EN EL PROGRAMA DE ESPECIALIZACION EN DOCENCIA UNIVERSITARIA DEL CEDEDUIS - SIEDUIS 1.0 -. Realizado por Andrea Marcela Mora Villamizar, Marzo del 2002.

ANEXO C PLANTILLAS Y CLASE FastTemplate

Este anexo contiene un artículo en donde se explica la ventaja de utilizar plantillas al momento de diseñar e implementar un sitio web. Además contiene otro artículo donde se describe con detalle la implementación y los métodos que emplea la clase `class.FastTemplate.php3`, una de las más utilizadas, y la empleada en la implementación del sitio web de la personería; en el diseño e implementación de sitios web.

TEMPLATES - WHY AND HOW TO USE THEM IN PHP3

Sascha Schumann

Are you faced with the task of creating a site which should be programmed by a coder (namely you) and designed by a designer? Do not you know how to make the life easy for both of you? I have got the answer for you: Use FastTemplate to make your site more customizable!

Ok, you probably want to know why you want to use FastTemplates.

- change the look of your entire web site in seconds
- abstract programming without dirty HTML
- the designer does not need to take care of all that "fuzzy" code
- it is amazingly fast
- easier reuse of old templates (think of common forms)

FastTemplate is derived from a similar named Perl packet (can be found on CPAN). You can download the PHP port from [its homepage](#). You just need one class file (class.FastTemplate.php3) from it.

Let me explain first that there is a difference between creating a page using templates and simply spitting it out by using echo or print. The simple echo/print approach is great for writing short scripts, but leaves you alone with better organization or customization. Templates on the other hand give you the ability to create multiple language sites by just changing one parameter. They force you to pay attention to what you do.

Do not be afraid of thinking before you start coding. It may take some time, but this time will be given back to you after your project grows in size.

So, how does one utilize FastTemplate? You need to start with a simple call to `<?php $tpl = new FastTemplate("path"); ?>` passing it the path to the template directory where all your template files live. It returns an object which you can use to assign parameters, create pages and so on.

FastTemplate is based on the assumption that a large page is built out of many small parts. Every part has a unique name. The smallest part is a normal text string which is assigned to such a unique name. This is done by `<?php $tpl->assign(NAME, "text"); ?>`. Now, FastTemplate knows what you mean, if one of your templates contains `{NAME}`.

Additionally, FastTemplate needs to know how you want to call your templates. You need to give it a hint by passing an associative array to `<?php $tpl->define(); ?>`:

```
<?php $tpl->define(array(foo => "foo.tpl", bar => "bar.tpl")); ?>
```

This assigns the name foo and bar to the respective files (namely foo.tpl and bar.tpl).

Now you want FastTemplate to substitute all your `{MACROS}` with their respective values in the template foo. Issue the command `<?php $tpl->parse(PAGECONTENT, "foo"); ?>` to do that. This command will assign the content of the template "foo" to the unique name PAGECONTENT.

Of course, we are not done with that yet since the template bar holds the main page definition and FastTemplate needs to substitute the macro `{PAGECONTENT}` in it. We also need to assign a PAGETITLE and to do that we say:

```
<?php
```

```
$tpl->assign(PAGETITLE, "FooBar test");
```

```
$tpl->parse(MAIN, "bar");
```

```
?>
```

Easy, is not it? We just need to print it out now:

```
<?php
```

```
$tpl->FastPrint(MAIN);
```

```
?>
```

The following three files show how the more verbose description looks in practice. I would not know how to live without this technique in real life - your designers will be happy and your boss will smile, because you can do more things in less time.

bar.tpl

```
<!-- bar.tpl -->
<HTML>
<HEAD><TITLE>Feature world - {PAGETITLE}</TITLE></HEAD>
<BODY BGCOLOR=BLACK TEXT=WHITE>
<H1>{PAGETITLE}</H1>
{PAGECONTENT}
</BODY>
</HTML>
```

foo.tpl

```
<!-- foo.tpl -->
This does not do anything obvious. Please look at {NAME}.
```

demo.php3

```
<?php
```

```
include "class.FastTemplate.php3";
```

```
$tpl = new FastTemplate(".");
$tpl->define(array(foo => "foo.tpl", bar => "bar.tpl"));
```

```
$tpl->assign(NAME, "me");
$tpl->assign(PAGETITLE, "Welcome!");
```

```
$tpl->parse(PAGECONTENT, "foo");
$tpl->parse(MAIN, "bar");
```

```
$tpl->FastPrint(MAIN);
```

```
?>
```

BUILDING WHOLE TABLES

I have also written a short example which demonstrates how to build whole tables with single row templates. It is effective, because you still do not need to tinker with HTML directly.

We attach the content of a template to an already defined unique name to build the HTML table. This can be done by sticking a single "." into the front of the template name when calling `$tpl->parse()`:

```
<?php
```

```
# assign the content of template foo to the unique name TPL1
$tpl->parse(TPL1, "foo");
```

```
# attach the content of template bar to the unique name TPL1
```

```

$tpl->parse(TPL1, ".bar");

?>

page.tpl

<HTML>
<HEAD><TITLE>Feature world - {PAGE_TITLE}</TITLE></HEAD>
<BODY BGCOLOR=BLACK TEXT=WHITE>
<H1>{PAGE_TITLE}</H1>
{PAGE_CONTENT}
</BODY>
</HTML>

table.tpl

<TABLE>
<TR> <TH>name</TH> <TH>size</TH> </TR>
{TABLE_ROWS}
</TABLE>

table_row.tpl

<TR>
<TD>{FILENAME}</TD>
<TD>{FILESIZE}</TD>
</TR>

yad.php3

<?php

include "class.FastTemplate.php3";

function InitializeTemplates() {
    global $tpl;

    $tpl = new FastTemplate(".");
    $tpl->define( array( page => "page.tpl",
                       table => "table.tpl",
                       table_row => "table_row.tpl" ) );
}

function ReadCurrentDirectory() {
    global $tpl;

    $handle = opendir(".");
    while($filename = readdir($handle)) {
        $tpl->assign(FILENAME, $filename);
        $tpl->assign(FILESIZE, filesize($filename));
        $tpl->parse(TABLE_ROWS, "table_row");
    }
    closedir($handle);
    $tpl->parse(PAGE_CONTENT, "table");
}

function PrintPage($title) {
    global $tpl;

    $tpl->assign(PAGE_TITLE, $title);
    $tpl->parse(FINAL, "page");
    $tpl->FastPrint(FINAL);
}

InitializeTemplates();
ReadCurrentDirectory();

```

```
Printpage("Yet Another Demo");
```

```
?>
```

SPEED DISCUSSION

"Ok," you might say, "that is all fine and nifty. But doesn't it impact the speed of my web site?"

Yes, your site will probably become faster. There is a simple reason for that: Because you as the coder are focused on designing your application and building code, your code will be more efficient, handling the same task easier and quicker. So, you might add just another reason to the above list why you should consider using FastTemplate in your next PHP project.

If you just want to convert an existing web site, the performance hit will probably not be noticed. I introduced a regex (short for "regular expression") cache into PHP 3.0.7 which helps in this special case. Because FastTemplate uses a regex for every macro, each regex will be compiled only once and the speed impact becomes negligible.

CLASS.FASTTEMPLATE.PHP3

UPDATE: July 20, 1999: First PHP4 wierdness found. PHP4 does not like the `{ $` combinations within FT's parse() routine. Here's the [diff](#):

```
--- class.FastTemplate.php3    Sun Jun 27 13:44:47 1999
+++ php4.FastTemplate.php3    Tue Jul 20 10:49:25 1999
@@ -196,8 +196,10 @@
         settype($val,"string");
     }

-         $template = ereg_replace("{ $key}", "$val", "$template");
-         // $template = str_replace("{ $key}", "$val", "$template");
+         // php4 doesn't like '{ $' combinations.
+         $key = '{ ' . "$key" . ' ';
+         $template = ereg_replace("$key", "$val", "$template");
+         // $template = str_replace("$key", "$val", "$template");
    }
}

@@ -410,7 +412,7 @@
    }
    if($send)
    {
-         $newParent .= "{$MacroName}\n";
+         $newParent .= '{ ' . "{$MacroName}\n";
    }
    // Next line please
    if($send) { $send = false; }
```

<**PUPDATE: July 19, 1999:** [PHP4/Zend](#) has finally been released into public Beta. If anyone running PHP4 runs into a problem under the new environment, let me know. I should have the majority of my code tested for PHP4 by the end of this week. If I run into any major weirdness I'll release an updated class file asap.

UPDATE: June 28 1999: Two things. **First:** I built PHP 3.0.11 this morning and they've apparently fixed the `str_replace()` bug in PHP. If you run PHP 3.0.11 you can safely edit the FT source and change it to use `str_replace()` (Just do a search for it in the code - there's only one line, just uncomment the `str_replace()` line and comment the `ereg_replace()` line) Don't do this on pre 3.0.11 builds of PHP as it'll core dump Apache when you try it). See the [CHANGELOG](#) for more details.

Second: FastTemplate 0.8 is available if you really want it. Apologies to the Freshmeat link-followers that are getting the 404 error. I didn't make the announcement on Freshmeat, and FT 0.8 does have a few bugs so it's better if you use 1.1.0 anyway. Pop me a [note](#) if you want an older version of any of the classes here. Hey, that PHP Guru [Sascha Schumann](#) wrote up a nice [FastTemplate Tutorial](#). Check it out!

Note: If you've got an anal build of PHP you might receive the following warning:

Warning: Uninitialized variable or array index or property (????) in class.FastTemplate.php3 on line 299
If this happens, make the following changes (add 1 line):

```
--- class.FastTemplate.php3~ Tue Jun 29 08:40:14 1999  
+++ class.FastTemplate.php3 Tue Jun 29 08:41:06 1999  
@@ -296,6 +296,7 @@
```

```
    if($append)  
    {  
+    if(!isset($this->$ReturnVar)) { $this->$ReturnVar = ""; }  
    $this->$ReturnVar .= $this->parse_template($this->$val,$this->PARSEVARS);  
    }  
    else
```

- [NAME](#)
- [SYNOPSIS](#)
- [DESCRIPTION](#)
- [CORE METHODS](#)
 - [define\(array\(key,value pairs \) \)](#)
 - [define_nofile\(\) alias: define_raw\(\)](#)
 - [define_dynamic\(\\$Macro, \\$ParentName\)](#)
 - [clear_dynamic\(\\$Macro\)](#)
 - [assign\(\(key,value pair\) or \(array\(key value pairs\) \)](#)
 - [parse\(RETURN, FileHandle\(s\)\)](#)
 - [strict\(\)](#)
 - [no_strict\(\)](#)
 - [FastPrint\(HANDLE\)](#)
- [OTHER METHODS](#)
 - [fetch\(HANDLE\)](#)
 - http://www.thewebmasters.net/php/FastTemplate.phtml#get_assigned_Var_Christian_Bra
 - [clear\(\)](#)
 - [clear_parse\(\)](#)
 - [clear_href\(KEY\)](#)
 - [clear_define\(\)](#)
 - [clear_tpl\(\)](#)
 - [clear_all\(\)](#)
 - [Variables](#)
 - [Variable Interpolation \(Template Parsing\)](#)
 - [FULL EXAMPLE](#)

- [VERSION](#)
- [AUTHOR](#)
- [DOCUMENTATION](#)
- [SEE ALSO](#)

NAME

FastTemplate 1.1.0 - PHP extension for managing templates and performing variable interpolation.

SYNOPSIS

```
include("class.FastTemplate.php3");
$tpl = FastTemplate("/path/to/templates");
$tpl->define( array( main => "main.tpl",
                  row  => "table_row.tpl",
                  all  => "table_all.tpl" ));
$tpl->assign(TITLE, "I am the title.");
$defaults = ( array( FONT => '<font size+=2 face=helvetica>',
                   EMAIL => 'cdi@thewebmasters.net' ));
$tpl->assign($defaults);
$tpl->parse(ROWS, ".row"); // the '.' appends to ROWS
$tpl->parse(CONTENT, array("row", "all"));
$tpl->parse(CONTENT, "main");
$tpl->FastPrint(CONTENT);
$raw = $tpl->fetch("CONTENT");
echo "$raw\n";
```

DESCRIPTION

What is a template?

A template is a text file with variables in it. When a template is parsed, the variables are interpolated to text. (The text can be a few bytes or a few hundred kilobytes.) Here is a simple template with one variable ('{NAME}')

```
Hello {NAME}. How are you?
```

When are templates useful?

Templates are very useful for CGI programming, because adding HTML to your PHP code clutters your code and forces you to do any HTML modifications. By putting all of your HTML in separate template files, you can let a graphic or interface designer change the look of your application without having to bug you, or let them muck around in your PHP code.

Why use FastTemplate?

Speed

FastTemplate parses with a single regular expression. It just does simple variable interpolation (i.e. there is no logic that you can add to templates - you keep the logic in the code). That's why it's has 'Fast' in it's name!

Flexibility

The API is robust and flexible, and allows you to build very complex HTML documents/interfaces. It is also completely written in PHP and (should) work on Unix or NT. Also, it isn't restricted to building HTML documents -- it could be used to build any ascii based document (postscript, XML, email - anything).

What are the steps to use FastTemplate?

The main steps are:

1. define
2. assign
3. parse
4. FastPrint

These are outlined in detail in CORE METHODS below.

CORE METHODS

define(array(key,value pairs))

The method define() maps a template filename to a (usually shorter) name;

```
$tpl = new FastTemplate("/path/to/templates");
$tpl->define( array( main => "main.tpl",
                  footer => "footer.tpl" ));
```

This new name is the name that you will use to refer to the templates. Filenames should not appear in any place other than a define().

(Note: This is a required step! This may seem like an annoying extra step when you are dealing with a trivial example like the one above, but when you are dealing with dozens of templates, it is very handy to refer to templates with names that are independent of filenames.)

TIP: Since `define()` does not actually load the templates, it is faster and more legible to define all the templates with one call to `define()`.

define_nofile() alias: **define_raw()**

THESE METHODS ARE NOT PORTED TO THE PHP VERSION

And probably never will be. The purpose of this class is to **eliminate** HTML from your PHP code, not to create new ways of adding it back in.

define_dynamic(\$Macro, \$ParentName)

Nino Martincevic, don@agi.de, emailed me with a question about doing something like this, and I thought it was a such a cool idea I immediately sat down and cranked it out...

You can define dynamic content within a static template. (Lists) Here's an example of `define_dynamic()`;

```
$tpl = new FastTemplate("./templates");
$tpl->define( array( main => "main.tpl",
                  table => "dynamic.tpl" ));
$tpl->define_dynamic( "row" , "table" );
```

This tells `FastTemplate` that buried in the `table` template is a dynamic block, named `row`. In older versions of `FastTemplate` (pre 0.7) this `row` template would have been defined as it's own file. Here's how a dynamic block appears within a template file;

```
<!-- NAME: dynamic.tpl -->
<table>
<!-- BEGIN DYNAMIC BLOCK: row -->
<tr>
<td>{NUMBER}</td>
<td>{BIG_NUMBER}</td>
</tr>
<!-- END DYNAMIC BLOCK: row -->
</table>
<!-- END: dynamic.tpl -->
```

The syntax of your `BEGIN` and `END` lines needs to be VERY exact. It is case sensitive. The code block begins on a new line all by itself. There cannot be ANY OTHER TEXT on the line with the `BEGIN` or `END` statement. (although you can have any amount of whitespace before or after) It must be in the format shown;

```
<!-- BEGIN DYNAMIC BLOCK: handle_name -->
```

The line must be exact, right down to the spacing of the characters. The same is true for your `END` line. The `BEGIN` and `END` lines cannot span multiple lines. Now when you call the `parse()` method, `FastTemplate` will automatically spot the dynamic block, strip it out, and use it exactly as if you had defined it as a stand-alone template. No additional work is required on your part to make it work - just define it, and `FastTemplate` will do the rest. Included with this archive should have been a file named **define_dynamic.phtml** which shows a working example of a dynamic block.

There are a few rules when using dynamic blocks - dynamic blocks should not be nested inside other dynamic blocks - strange things WILL occur. You -can- have more than one nested block of code in a page, but of course, no two blocks can share the same defined handle. The error checking for `define_dynamic()` is miniscule at best. If you define a dynamic block and `FastTemplate` fails to find it, no errors will be generated, just really weird output. (`FastTemplate` will not append the dynamic data to the returned output) Since the `BEGIN` and `END` lines are stripped out of the parsed results, if you ever see your `BEGIN` or `END` line in the parsed output, that means that `FastTemplate` failed to find that dynamic block.

clear_dynamic(\$Macro)

This provides a method to remove the dynamic block definition from the parent macro provided that you haven't already parsed the template. Using our example above:

```
$tpl->clear_dynamic("row");
```

Would completely strip all of the **unparsed** dynamic blocks named `row` from the parent template. This method won't do a thing if the template has already been parsed! (Because the required `BEGIN` and `END` lines have been removed through the parsing) This method works well when you are accessing a database, and your `rows` may or may not return anything to print to the template. If your database query doesn't return anything, you can now strip out the rows you've set up for the results. (Gee, maybe I ran into this problem myself ? :-)

assign((key,value pair) or (array(key value pairs))

The method `assign()` assigns values for variables. In order for a variable in a template to be interpolated it must be assigned. There are two forms which have some important differences. The simple form, is to accept an array and copy all the key/value pairs into an array in `FastTemplate`. There is only one array in `FastTemplate`, so assigning a value for the same key will overwrite that key.

```
$tpl->assign(TITLE => "king kong");
$tpl->assign(TITLE => "godzilla"); // overwrites "king kong"
```

parse(RETURN, FileHandle(s))

The parse function is the main function in FastTemplate. It accepts a new key value pair where the key is the TARGET and the values are the SOURCE templates. There are three forms this can be in:

```
$tpl->parse(MAIN, "main");           // regular
$tpl->parse(MAIN, array ( "table", "main" ) ); // compound
$tpl->parse(MAIN, ".row");           // append
```

In the regular version, the template named ``main" is loaded if it hasn't been already, all the variables are interpolated, and the result is then stored in FastTemplate as the value MAIN. If the variable '{MAIN}' shows up in a later template, it will be interpolated to be the value of the parsed ``main" template. This allows you to easily nest templates, which brings us to the compound style.

The compound style is designed to make it easier to nest templates. The following are equivalent:

```
$tpl->parse(MAIN, "table");
$tpl->parse(MAIN, ".main");
// is the same as:
$tpl->parse(MAIN, array("table", "main"));
// this form saves function calls and makes your code cleaner
```

It is important to note that when you are using the compound form, each template after the first, must contain the variable that you are parsing the results into. In the above example, 'main' must contain the variable '{MAIN}', as that is where the parsed results of 'table' is stored. If 'main' does not contain the variable '{MAIN}' then the parsed results of 'table' will be lost.

The append style allows you to append the parsed results to the target variable. Placing a leading dot . before a defined file handle tells FastTemplate to append the parsed results of this template to the returned results. This is most useful when building tables that have a dynamic number of rows - such as data from a database query.

strict()

When strict() is on (it is on by default) all variables found during template parsing that are unresolved have a warning printed to STDERR;

```
[FastTemplate] Warning: no value found for variable: SOME_VAR
```

Also, the variables will be left in the output document. This was done for two reasons: to allow for parsing to be done in stages (i.e. multiple passes), and to make it easier to identify undefined variables since they appear in the parsed output. If you want to replace unknown variables with an empty string, see: no_strict().

Note: STDERR output should be captured and logged by the webserver. With apache (and unix!) you can tail the error log during development to see the results as in;

```
tail -f /var/log/httpd/error_log
```

no_strict()

Turns off warning messages about unresolved template variables. A call to no_strict() is required to replace unknown variables with an empty string. By default, all instances of FastTemplate behave as if strict() was called. Also, no_strict() must be set for each instance of FastTemplate;

```
$tpl = new FastTemplate("/path/to/templates");
$tpl->no_strict();
```

FastPrint(HANDLE)

The method FastPrint() prints the contents of the named variable. If no variable is given, then it prints the last variable that was used in a call to parse() which I find is a reasonable default.

```
$tpl->FastPrint();           // continuing from the last example, would
                           // print the value of MAIN
$tpl->FastPrint("MAIN"); // ditto
```

This method is provided for convenience. If you need to print somewhere else (a socket, file handle) you would want to fetch() a reference to the data first:

```
$data = $tpl->fetch("MAIN");
fwrite($fd, $data); // save to a file
```

OTHER METHODS

fetch(HANDLE)

Returns the raw data from a parsed handle.

```
$tpl->parse(CONTENT, "main");
$content = $tpl->fetch("CONTENT");
print $content;           // print to STDOUT
fwrite($fd, $content); // write to filehandle
```

get_assigned(\$Var) Christian Brandel cbrandel@gmx.de

This method will return the value of a variable that has been set via assign(). This allows you to easily pass variables around within functions by using the FastTemplate class to handle ``globalization" of the variables. For example;

```
$tpl->assign( array( TITLE => $title,
```

```
BGCOLOR => $bgColor,  
TEXT => $textColor );  
(sometime later...)  
$bgColor = $tpl->get_assigned(BGCOLOR);
```

clear()

Note: All of the clear() functions are for use anywhere where your scripts are persistent. They generally aren't needed if you are writing CGI scripts.

clear() Clears the internal references that store data passed to parse(). clear() accepts individual references, or array references as arguments.

Often clear() is at the end of a script:

```
$tpl->FastPrint("MAIN");  
$tpl->clear("MAIN");  
or  
$tpl->FastPrint("MAIN");  
$tpl->FastPrint("CONTENT");  
$tpl->clear(array("MAIN","CONTENT"));
```

If called with no arguments, removes ALL references that have been set via parse().

clear_parse()

See: clear()

clear_href(KEY)

Removes a given reference from the list of refs that is built using:

```
$tpl->assign(KEY = val);
```

If called with no arguments, it removes all references from the array.

(Same as clear_assign())

```
$tpl->assign( array( MOVIE => "The Avengers",  
                  RATE => "Sucked" ));  
$tpl->clear_href("MOVIE");  
// Now only {RATE} exists in the assign() array
```

clear_define()

Clears the internal list that stores data passed to:

```
$tpl->define();
```

Note: The hash that holds the loaded templates is not touched with this method. (See: clear_tpl()) Accepts a single file handle, an array of file handles, or nothing as arguments. If no argument is given, it clears ALL file handles.

```
$tpl->define( array( MAIN => "main.tpl",  
                  BODY => "body.tpl",  
                  FOOT => "foot.tpl" ));  
  
// some code here  
$tpl->clear_define("MAIN");
```

clear_tpl()

Clears the internal array that stores the contents of the templates. (If they have been loaded) If you are having problems with template changes not being reflected, try adding this method to your script.

```
$tpl->define(MAIN,"main.tpl" );  
// assign(), parse() etc etc...  
$tpl->clear_tpl(MAIN); // Loaded template now unloaded.
```

clear_all()

Cleans the module of any data, except for the ROOT directory. Equivalent to:

```
$tpl->clear_define();  
$tpl->clear_href();  
$tpl->clear_tpl();  
$tpl->clear_parse();
```

In fact, that's exactly what it does.

Variables

A variable is defined as:

```
{[A-Z0-9_]+}
```

This means, that a variable must begin with a curly brace '{'. The second and remaining characters must be uppercase letters or digits 'A-Z0-9'. Remaining characters can include an underscore. The variable is terminated by a closing curly brace '}'.

For example, the following are valid variables:

```
{FOO}  
{F123F}
```

{TOP_OF_PAGE}

Variable Interpolation (Template Parsing)

If a variable cannot be resolved to anything, a warning is printed to STDERR. See `strict()` and `no_strict()` for more info.

Some examples will make this clearer.

Assume:

```
$FOO = "foo";  
$BAR = "bar";  
$ONE = "1";  
$TWO = "2";  
$UND = "_";
```

Variable Interpolated/Parsed

```
{FOO}            foo  
{FOO}-{BAR}     foo-bar  
{ONE_TWO}       {ONE_TWO} // {ONE_TWO} is undefined!  
{ONE}{UND}{TWO} 1_2  
${FOO}           $foo  
$25,000          $25,000  
{foo}           {foo}    // Ignored, it's not valid, nor will it  
                 // generate any error messages.
```

If you're thinking you could have done the same thing in a few lines of plain PHP, well yes you probably could. But, how would a graphic designer tweak the resulting HTML? How would you have a designer editing the HTML while you're editing another part of the code? How would you save the output to a file, or pipe it to another application? How would you make your application multi-lingual? How would you build an application that has options for high graphics, or text-only? FastTemplate really starts to shine when you are building mid to large scale web applications, simply because it begins to separate the application's generic logic from the specific implementation.

ANEXO D

ORGANIZACIÓN DEL DIRECTORIO DEL SISTEMA INTERNO

Directorio		Subdirectorios	Descripción
SISTEMA INTERNO	Común	agenda	Estas carpetas contienen las páginas php correspondientes a cada uno de las opciones de la zona común o de los subsistemas de implementación.
		comunicaciones	
		eventos	
		preferencias	
	Perfiles	persdistrital	Estas carpetas contienen las páginas php correspondientes a cada uno de los perfiles identificados. Son las páginas particulares a cada uno de ellos.
		persdelegado	
		abogadoreparto	
		secretaria	
		administrador	
		funcionarios	
	Gestión de Interfaz	plantillas	En esta carpeta encontramos las plantillas (archivos .tpl) para cada uno de los perfiles y cada opción de la zona común, organizadas respectivamente.
		estilos	Contiene archivos .css , los cuales se utilizan para definir los estilos (color, fuente, etc.) de la interfaz del sistema.
		ingreso	Contiene páginas php relacionadas con el acceso al sistema y páginas de información corporativa.

ORGANIZACIÓN DEL DIRECTORIO DEL SITIO WEB

Directorio		Subdirectorios	Descripción
SITIO WEB	animaciones		Animaciones utilizadas en las diferentes páginas.
	boletines		En este directorio se encuentran almacenados los boletines de prensa generados por la oficina de divulgación y prensa de la personería Distrital de Cartagena.
	contrataciones		En este directorio se encuentran almacenados las diferentes licitaciones publicas que contrata la personería Distrital de Cartagena.
	documentosLegales		En este directorio se encuentran los formatos de documentos legales como derecho de petición que se pueden descargar del sitio.
	fotoperoneroDistrital		En este directorio se encuentran las fotos de las hojas de vida del personero Distrital.
	hojasvidaperoneros		En este directorio se encuentran las hojas de vida (archivos de texto) del personero Distrital.
	imagenes		Como su nombre lo indica se alojan las imágenes utilizadas en la construcción del sitio.
	informes_gestion		Contiene los archivos de los informes de gestión de la entidad de los diferentes años.
	paginasphp		Contiene los archivos php que implementan funciones que son de uso comunes a todas las paginas. Ejemplo: Conexión a la base de datos y Funciones comunes.
	phpMyAdmin-2.3.2		Contiene todos los archivos relacionados con esta aplicación. Este subdirectorio además de encontrarse en el directorio del sitio, debe estar en el directorio de servidor de documentos del servidor web utilizado. Por ejemplo, si el servidor utilizado es Apache , esta debe estar en el directorio <i>httdocs</i> .
	plantillas		En este directorio encontramos todos los archivos con extensión .tpl, que son archivos con código html correspondientes al diseño de la interfaz y la clase class.FastTemplate.php3.
	presupuestos		Contiene los archivos de los presupuestos de la entidad de los diferentes años.
	sympolls		Contiene los archivos relacionados con la aplicación sympoll 1.0 (Encuesta y votación), incluida la licencia de uso.
	quejasdescargadas		En este directorio se encuentran los archivos que contienen las quejas interpuestas por los usuarios registrados en el sitio web de la personería.

Descripción de las páginas

Descritos los subdirectorios del sitio web, se listan las páginas PHP y archivos TPL relacionadas con una breve descripción de su funcionalidad.

Plantillas Globales: Estas plantillas son utilizadas por todas las páginas del sitio web excepto las de administración.

Estas son: pagina.tpl tablaprincipal.tpl marcosup.tpl pie_pagina.tpl tb_menu_tituloimg.tpl

	Plantillas – Archivos .TPL	<i>Páginas .PHP</i>
Esta página muestra el contenido correspondiente a la acción de cumplimiento.	tabla_accioncumplimiento.tpl , tb_menu_tituloimg.tpl , Plantillas Globales	accioncumplimiento
Esta página muestra el contenido correspondiente a la acción de tutela.	tabla_acciondetutela.tpl , tb_menu_tituloimg.tpl Plantillas Globales	acciondetutela
Esta página muestra el contenido correspondiente a la acción de grupo.	tabla_acciongrupo.tpl Plantillas Globales	acciongrupo
Esta página muestra el contenido correspondiente a la acción popular.	tabla_accionpopular.tpl Plantillas Globales	accionpopular
Esta página muestra las actuaciones de los personeros delegados en la solución de una queja interpuesta personalmente.	tabla_actuaciones.tpl filas_tabla_actuacion_detalle.tpl pagina_nulo.tpl tabla_nulo.tpl tablaprincipal_nulo.tpl, Plantillas Globales	actuacion_presencial
Esta página muestra las actuaciones de los personeros delegados en la solución de una queja.	tabla_actuaciones.tpl filas_tabla_actuacion_detalle.tpl tabla_nulo.tpl tablaprincipal_nulo.tpl, Plantillas Globales	actuaciones

Esta página contiene todas las opciones para la administración del sitio web. Solo tiene acceso autorizado el administrador del sistema; eso lo hace escribiendo su nombre de usuario y clave de acceso en la página ingresoadmin.php.	principaladmin.tpl scriptreloj.tpl valformingresoadmin.tpl contaccesonegado.tpl contadmin.tpl, Plantillas Globales	administración
Esta página muestra información sobre la ubicación física de la personería, los horarios de atención, fax y dirección.	tabla_atencion_comunidad.tpl	atencion_comunidad
Nos muestra una tabla con el número, la fecha y un encabezado de todos los boletines almacenados en el sistema	tabla_boletines.tpl filas_tabla_boletin.tpl pagina_nulo.tpl tabla_nulo.tpl tablaprincipal_nulo.tpl	boletines
Esta página permite el cambio de la clave de acceso del administrador del sitio.	principaladmin.tpl scriptreloj.tpl valformcambioswadmin.tpl contaccesonegado.tpl contcambioswadmin.tpl	cambipaswadmin
Esta página hace la actualización en la tabla admin, del campo correspondiente a la clave de acceso del administrador y cuyo nuevo valor ha sido agregado en la página cambioswadmin.php.	principaladmin.tpl scriptreloj.tpl valformingresoadmin.tpl contaccesonegado.tpl contenidocambioexitoso contcambioexitoso.tpl	cambiospwexitoso
Muestra el formulario solicita el número radicado para realizar la consulta de las actuaciones de las quejas interpuestas personalmente.	paginaval.tp tabla_conpresencial.tpl	consulta_pornumrad
Esta página realiza la consulta a la base de datos(tabla act_qjpresencial). Recupera y muestra las actuaciones de los personeros delegados de las quejas interpuestas personalmente	paginaval.tpl tabla_conpresencial.tpl	consulta_presencial
Esta página realiza la consulta a la base de datos(tabla actuaciones). Recupera y muestra las actuaciones de los personeros delegados	paginaval.tpl filas_tabla_queja.tpl tabla_resulquejas.tpl tabla_clave_invalida.tpl tabla_u_noregistrado.tpl	consultarqueja
Muestra el formulario que permite ingresar la clave y el nombre de usuario para realizar una consulta	paginaval.tpl tabla_consulte.tpl	consulte
Realiza la consulta a la tabla contratación de la base de datos y muestra las contrataciones que continúan vigentes..	tabla_contratacion.tpl filas_tabla_contratacion.tpl pagina_nulo.tpl tabla_nulo.tpl tablaprincipal_nulo.tpl	contratacion
Realiza la consulta a la tabla contratación de la base de datos y muestra las contrataciones que se han cerrado.	tabla_contratacion.tpl filas_tabla_contratacion.tpl tb_menu_tituloimg.tpl pagina_nulo.tpl	contratacion_cerradas

	tabla_nulo.tpl tablaprincipal_nulo.tpl	
Guarda los datos del anónimo en la base de datos.	tabla_bienanonimo.tpl	datosanonimo
Guarda en la base de datos el contenido de la queja interpuesta por el usuario.	paginaval.tpl tabla_quejaok.tpl tabla_registrofallido.tpl	datosqueja
Guarda los datos del usuario no anónimo en la base de datos.	tabla_bienquejoso.tpl tabla_registrofallido.tpl	datosquejoso
Esta página muestra el contenido correspondiente al derecho de petición.	tabla_derecho_de_peticion.tpl tb_menu_tituloimg.tpl	derecho_de_peticion
Esta página almacena la información de un nuevo mensaje del foro en la base de datos.	paginaval.tpl	enviaropinion
Esa página describe toda la estructura organizacional de la Personería Distrital de Cartagena (Organigrama).	No utiliza plantillas	estructuraorg
Esta página muestra los participantes en el foro activo. Su nombre, opinión, fecha y a quien va dirigido el mensaje.	pagina.tpl tabla_foro.tpl filas_tabla_foro.tpl	foro
Esta página nos muestra las funciones en general de la Personería Distrital de Cartagena.	tabla_funcion.tpl	funcion
Esta página nos muestra una lista con el nombre y el cargo de los funcionarios principales de la Personería.	tabla_funcionarios.tpl filas_tabla_funcionarios	funcionarios
Es la página inicial del sitio web	paginaprincipalindex.tpl logobuscarindex.tpl logobotonesindex.tpl encuestaindex.tpl boletinesindex.tpl enlacesindex.tpl sientetubanderaindex.tpl opineindex.tpl contenidocentralindex.tpl menuprincipalindex.tpl noticiasahoraindex.tpl filas_noticias_index.tpl	index
Esta página nos muestra en una tabla el año y el archivo a descargar del informe de gestión seleccionado.	tabla_infogestion.tpl pagina_nulo.tpl tabla_nulo.tpl filas_tabla_infogestion.tpl tablaprincipal_nulo.tpl	informegestion
En esta página el administrador digita su nombre de usuario y su clave de acceso para ingresar a la página que contiene las opciones de administración (administración.php).	principaladmin.tpl valformingresoadmin.tpl contingresoadmin.tpl scriptreloj.tpl	ingresoadmin
Muestra los datos del usuario que interpone la queja y el	paginaval.tpl	interponerqueja

formulario para diligenciar el cuerpo de la queja.	tabla_queja_noanonimo.tpl tabla_queja_anonimo.tpl tabla_u_noregistrado.tpl	tabla_clave_invalida.tpl	
Script de Java que contiene las opciones del menú principal que aparece en la página index del sitio web.			menuprincipal
Esta página nos muestra la misión y visión de la Personería.	tabla_mision_vision.tpl		mision_vision
Esta página nos muestra la normatividad por la cual fue creada y se rige en su funcionamiento la Personería Distrital de Cartagena	tabla_normatividad.tpl		normatividad
Esta página recupera de la base de datos todas las noticias almacenadas.	tabla_noticias.tpl pagina_nulo.tpl tablaprincipal_nulo.tpl	filas_tabla_noticias.tpl tabla_nulo.tpl	noticias
Está página recupera una noticia en particular, mostrando el total de su contenido.	Tabla_noticias.tpl pagina_nulo.tpl tablaprincipal_nulo.tpl	filas_tabla_noticias_detalle.tpl tabla_nulo.tpl	noticias_detalle
Esta página muestra el formulario para ingresar la información de un nuevo mensaje en el foro.	paginaval.tpl tabla_nuevomsg.tpl		nuevomsgforo
Esta página nos muestra las funciones de todas las Personerías Delegadas existentes en la entidad.	tabla_personeriasdeleg.tpl		personeriasdeleg
Esta página muestra toda la información contenida en la tabla personerodistrital y que es información sobre el perfil profesional del personero distrital.	tabla_personero_distrital.tpl pagina_nulo.tpl tabla_nulo.tpl	tablaprincipal_nulo.tpl	personero_distrital
Nos muestra el cronograma de actividades de la Personería con los personeros estudiantiles.	tabla_agenda_pest.tpl filas_tabla_agenda_pest.tpl tabla_nulo.tpl	tablaprincipal_nulo.tpl	personero_estagenda
Nos muestra generalidades del programa Personero estudiantil.	tabla_personeroest.tpl		personeroest
Nos muestra los nombres de los centros educativos y sus personeros estudiantiles.	tabla_cedu_pest.tpl filas_tabla_cedu_pest.tpl pagina_nulo.tpl tabla_nulo.tpl	tablaprincipal_nulo.tpl	personeroest_cedu
Esta página contiene información sobre el gobierno escolar, esto enmarcado dentro del programa personero estudiantil.	tabla_personeroestgob.tpl		personeroestgob
Esta página contiene información de interés para los personeros estudiantiles.	tabla_personeroestinteres.tpl.tpl		personeroestinteres
Esta página contiene información sobre quien es y las funciones del personero estudiantil.	tabla_personeroestquien.tpl		personeroestquien

Esta página nos muestra en una tabla el año y el archivo a descargar del presupuesto seleccionado.	tabla_presupuesto.tpl pagina_nulo.tpl tabla_nulo.tpl tablaprincipal_nulo.tpl	filas_tabla_presupuesto.tpl	presupuestos
Nos muestra información sobre que es una queja, un reclamo o una sugerencia.	tabla_quejasyreclamos.tpl		quejasyreclamos
Esta página muestra formulario que permite ingresar el nombre de usuario y la clave para interponer una queja, en caso de que el usuario se encuentre registrado; sino debe seleccionar la opción registro como anónimo o no anónimo.	paginaval.tpl tabla_quejese.tpl valformingresoadmin.tpl		quejese
Nos muestra información correspondiente a la querella.	tabla_querella.tpl		querella
Muestra el formulario para registro como quejoso anónimo	paginaval.tpl tabla_reganonimo.tpl valformdatosanonimo.tpl		reganonimo
Muestra el formulario para registro como quejoso no anónimo	paginaval.tpl tabla_regnoanonimo.tpl		regnoanonimo
Esta página nos muestra una reseña histórica sobre la Personerías.	tabla_resena_historica.tpl		resena_historica
Nos presenta una introducción a los servicios prestados por la personería y nos permite seleccionar que personería delegada queremos conocer.	tabla_servicios.tpl		servicios
Esta página muestra un mensaje de advertencia, indicando la no existencia del registro de un usuario en la base de datos del sitio web.	paginaval.tpl tabla_u_noregistrado.tpl		usu_noregistrado
Esta página contiene un cuadro de texto donde el administrador debe digitar la palabra clave asociada con su clave de acceso, para que el sistema le asigne una nueva clave.	principaladmin.tpl scriptreloj.tpl valformcambioclaveoldadmin.tpl contolvidoclveadmin.tpl		olvidoclveadmin
Esta página muestra los campos que debe digitar un usuario que ha olvidado su clave. Envía la notificación al administrador del sitio.	paginaval.tpl tabla_claveolvidada.tpl valformclaveolvidada.tpl		claveolvidada
Esta página muestra un mensaje de confirmación notificando el cambio exitoso realizado en la modificación de la clave de acceso de un usuario registrado.	paginaval.tpl tabla_enviocambioclavexitoso.tpl		clavemodificadausr
Esta página muestra los campos que debe digitar un usuario que desea modificar su clave de acceso.	paginaval.tpl tabla_cambioclaveusr.tpl valformcambioclaveusr.tpl		cambioclaveusr

Esta página modifica el cambio de la clave de acceso del administrador (en caso de que la haya olvidado), asignando una nueva clave generada aleatoriamente.	principaladmin.tpl scriptreloj.tpl contaccesonegado.tpl contcambioclveoldadmin.tpl	cambioclveolvidadadmin
Esta página permite al administrador, consultar y asignar nuevas claves de acceso a los usuarios que han enviado la notificación de olvido.	principaladmin.tpl scriptreloj.tpl valformcambioclveolda.tpl contaccesonegado.tpl formcambioclavesolvidadas.tpl tabla_cambioclavesolvidadas.tpl tabla_enviocambioclavesolvidadas.tpl contadminclavesolvidadas.tpl	adminclavesolvidadas
Esta página muestra un mensaje notificando la actualización exitosa de la nueva clave de acceso asignada a un usuario.	principaladmin.tpl scriptreloj.tpl contaccesonegado.tpl contrestringido.tpl contadminclveusractualizada.tpl	adminclveusractualizada
En esta página se escribe un intervalo de fechas para generar un archivo que contenga las quejas interpuestas por los usuarios registrados, y que estén comprendidas en el rango digitado. Este archivo generado se podrá descargar luego.	principaladmin.tpl scriptreloj.tpl valformfechasarchivoquejas.tpl contaccesonegado.tpl contdescquejasadmin.tpl	descargas_quejas
En esta página se muestra el archivo a descargar, que contiene las quejas interpuestas por los usuarios registrados en el sitio web de la personería distrital.	principaladmin.tpl scriptreloj.tpl contaccesonegado.tpl contrestringido.tpl contarchivoa descargaradmin.tpl	descarga_archivo_quejas

Tablas de la base de datos del sitio web.

- **TABLA ACT_QJPERSONALES**

Campo	Tipo	Atributos	Nulo	Predeterminado	Descripción
<u>id_actp</u>	int(11)		No		auto_increment
pdeleg_actp	varchar(80)		Sí	NULL	Nombre el personero delegado encargado de resolver la queja, reclamo o sugerencia interpuesta.
desc_actp	text		Sí	NULL	Descripción del estado de la queja, reclamo o sugerencia interpuesta.
fecha_actp	date		Sí	NULL	Fecha de la actuación de la queja, reclamo o sugerencia interpuesta
num_idqja	varchar(10)		Sí	NULL	Número (número de radicado por ejemplo) que identifica a la queja, reclamo o sugerencia interpuesta.
ident_qjsop	int(15)		Sí	NULL	Número identificador del usuario a quien corresponde la queja, reclamo o sugerencia interpuesta.

- **TABLA ACTUACIONES:**

Campo	Tipo	Atributos	Nulo	Predeterminado	DESCRIPCIÓN
<u>id_actuac</u>	int(11)		No		Identificador de actuación, auto_increment
pdeleg_asign	varchar(80)		No		Nombre el personero delegado encargado de resolver la queja, r
desc_actuac	text		Sí	NULL	Descripción del estado de la queja, reclamo o sugerencia interpu
fecha_actuac	date		Sí	0000-00-00	Fecha de la actuación de la queja, reclamo o sugerencia interpu
id_qja	int(11)		No	0	Número que identifica a la queja, reclamo o sugerencia interpu

- **TABLA ADMIN**

Campo	Tipo	Atributos	Nulo	Descripción
<u>nbadminpdc</u>	varchar(13)		No	Nombre de usuario del administrador de la base de datos
clvadminpdc	varchar(24)		No	Clave del administrador para tener acceso a la base de datos.
pclvadminpdc	varchar(33)		No	Palabra clave del administrador para recordar su clave

- **TABLA AGENDA_PESTUDIANTIL**

Campo	Tipo	Atributos	Nulo	Predeterminado	Descripción
<u>fecha_actividad</u>	date		No	0000-00-00	Fecha en que se realizara la actividad programada
actividad	varchar(200)		No		Actividad programada
hora_a	time		Sí	<i>NULL</i>	Hora en que realizara la actividad programada.
lugar	varchar(100)		Sí	<i>NULL</i>	Lugar donde se llevara acabo la actividad programada
responsable	varchar(100)		Sí	<i>NULL</i>	Persona o personas a cargo de la actividad programada.
observacion	varchar(200)		Sí	<i>NULL</i>	Puntos para tener en cuenta en la realización de la actividad programada.

- **TABLA ANONIMO**

Campo	Tipo	Atributos	Nulo	Predeterminado	DESCRIPCIÓN
<u>nbusuario</u>	varchar(9)		No		Nombre de usuario anónimo, es generado por el sistema en el momento del registro. Presenta el siguiente formato A1000000 (A1 seguido de 6 dígitos).
password_anon	varchar(15)		No		Es la clave suministrada por el usuario anónimo en el momento del registro.
fecha_rega	date		Sí	NULL	Es la fecha en la que se registro el usuario. Es tomada del sistema
correoe_anon	varchar(50)		Sí	NULL	Es el correo electrónico del usuario suministrado durante el registro.

- **TABLA BOLETÍN**

Campo	Tipo	Nulo	Predeterminad	Descripción
<u>num_bol</u>	varchar(4)	No		Número que identifica a el boletín de prensa
anno_bol	varchar(4)	No		Año de publicación el boletín de prensa.
mes_bol	char(2)	No		Mes de publicación el boletín de prensa.
dia_bol	char(2)	No		Día de publicación el boletín de prensa.
resumen_bol	varchar(60)	Sí	NULL	Resumen corto (60 caracteres máximo) sobre el tema principal del boletín de prensa.
enlace_bol	varchar(40)	Sí	NULL	Nombre del documento, con su extensión, y que debe estar almacenado en el directorio boletines del sitio web.

- **TABLA CLAVES_OLVIDADAS**

Campo	Tipo	Nulo	Descripción
<u>id_clveolda</u>	int(11)	No	Identificador de la clave olvidada, es un campo
usuario_clveolda	varchar(9)	No	Nombre de usuario que olvido la clave.
correoe_clveolda	varchar(50)	No	Correo electrónico del usuario que olvido la clave.
anno_clveolda	varchar(4)	No	Año en el que el usuario envía la notificación de olvido de clave.
mes_clveolda	char(2)	No	Mes en el que el usuario envía la notificación de olvido de clave.
dia_clveolda	char(2)	No	Día en el que el usuario envía la notificación de olvido de clave.
clvemod_clveolda	char(1)	No	La nueva clave que se asigna a persona que olvido la clave.

- **TABLA COLEGIOS_PEST**

<u>id_colegio</u>	int(11)	No	Identificador del colegio donde existe un personero estudiantil.
nom_col	varchar(80)	Sí <i>NULL</i>	Nombre del colegio donde existe un personero estudiantil.
nom_pest	varchar(80)	Sí <i>NULL</i>	Nombre del personero estudiantil.
grado_pest	int(2)	Sí <i>NULL</i>	Grado que cursa el personero estudiantil.
correoe_pest	varchar(50)	Sí <i>NULL</i>	Correo electrónico del personero estudiantil.

- **TABLA CONTRATACIÓN**

Campo	Tipo	Nulo	Predeterminado	DESCRIPCIÓN
<u>num_licitacion</u>	varchar(4)	No		Número que identifica la licitación.
fecha_apertura	date	No	0000-00-00	Fecha de apertura de la licitación. El formato de este campo es aaaa-mm-dd.
objeto	varchar(250)	No		Motivo por el cual se abre la licitación.
fecha_cierre	datetime	No	0000-00-00 00:00:00	Fecha de cierre de la licitación. El formato de este campo es aaaa-mm-dd hh-mm-ss
presup_ofic	int(11)	No	0	Presupuesto asignado para la ejecución de la licitación
enlace	varchar(50)	No		Nombre del documento que contiene el formato de la licitación. Este debe estar almacenado en el directorio contrataciones del sitio web

- **TABLA FORO**

Campo	Tipo	Nulo	Predeterminado	Descripción
<u>id_foro</u>	int(11)	No		identificador del foro
nombre_f	varchar(80)	No		Nombre del foro (máximo 80 caracteres).
desc_f	varchar(150)	Sí	<i>NULL</i>	Breve descripción del tema del foro, hasta 150 caracteres máximo.
fecha_f	date	No	0000-00-00	Fecha de inicio del foro.
activo	int(1)	No	0	Activa o desactiva el foro, esto hace que aparezca en la página principal o no. Solo puede estar activo un foro a la vez; este campo solo puede tomar los valores de (1) o (0).

- **TABLA FUNCIONARIOS**

Campo	Tipo	Nulo	DESCRIPCIÓN
<u>ced_funcionario</u>	int(10)	No	Cédula o documento de identidad de los funcionarios de la personería distrital de Cartagena
Nombre_funcionario	varchar(70)	No	Nombre de los funcionarios de la personería distrital de Cartagena.
cargo	varchar(70)	No	Cargo que desempeñan los funcionarios de la personería distrital de Cartagena.
correo_e	varchar(50)	No	Correo electrónico de los funcionarios de la personería distrital de Cartagena

- **TABLA INFORMEGESTION**

Campo	Tipo	Nulo	Descripción
<u>id_infogestion</u>	int(11)	No	Identificador del informe de gestión
ano_infogestion	varchar(15)	No	Año de elaboración de los informes de gestión.
nb_docgestion	varchar(35)	No	Nombre del documento (archivo con su extensión) que contiene los informes de gestión elaborados en la personería distrital de Cartagena. Estos documentos deben estar almacenados en directorio informes_gestion del sitio web.

- **TABLA NOTICIAS**

Campo	Tipo	Nulo	Predeterminado	DESCRIPCIÓN
<u>id_ntc</u>	int(4)	No		
fecha_ntc	date	No	0000-00-00	Fecha de elaboración de la noticia.
titulo_ntc	varchar(55)	Sí	NULL	Título de la noticia (máximo 55 caracteres)
intro_ntc	text	Sí	NULL	Breve introducción sobre el tema de la noticia.
parrafo1_ntc	text	Sí	NULL	Párrafos de contenido central de la noticia, se dispone de cuatro párrafos para desarrollar el tema central de las noticias.
parrafo2_ntc	text	Sí	NULL	
parrafo3_ntc	text	Sí	NULL	
parrafo4_ntc	text	Sí	NULL	

- **TABLA PARTICIPANTEFORO**

Campo	Tipo	Nulo	Predeterminado	Descripción
<u>id_par</u>	int(11)	No		identificador del participante, es una campo auto incremental y es la llave principal de la tabla.
fecha_par	date	No	0000-00-00	Es la fecha de participación en el foro.
nombre_par	varchar(15)	Sí	NULL	Nombre del participante.
opinion_par	varchar(250)	Sí	NULL	Opinión del participante
nb_respuesta	varchar(15)	Sí	NULL	Nombre de la persona a la que se le envía un mensaje.
id_foro	int(11)	Sí	NULL	identificador del foro.

- **TABLA PERSONERODISTRITAL**

Campo	Tipo	Nulo	Descripción
perdist_activo	char(1)	No	Activa o desactiva el personero distrital vigente, esto hace que aparezca el la página del personero distrital o no. Solo puede estar activo un personero a la vez; este campo solo puede tomar los valores de (1) o (0).
nb_perdist	varchar(15)	No	Nombres del personero distrital.
ap_perdist	varchar(30)	No	Apellidos del personero distrital.
perfil_perdist	text	No	Breve descripción del perfil profesional del personero distrital.
hojavidaperdist	varchar(40)	No	Nombre del documento (archivo y su extensión), que contiene la hoja de vida del personero distrital. Este archivo debe estar almacenado en el directorio hojasvidaperpersoneros del sitio web.
foto_perdist	varchar(30)	No	Nombre del archivo (archivo y su extensión), que contiene la foto del personero distrital. Este archivo debe estar almacenado en el directorio fotoperpersonerodistrital del sitio web.
<u>id_perdist</u>	int(11)	No	auto_increment

- **TABLA PRESUPUESTOS**

Campo	Tipo	Nulo	Descripción
<u>idpresupuesto</u>	int(11)	No	Identificador de presupuesto, auto incremental
anopresupuesto	varchar(4)	No	Año de los presupuestos asignados a la personería
nbdocumentoprep	varchar(35)	No	Nombre del archivo (archivo y su extensión), que contiene el presupuesto asignado a la personería distrital. Este archivo debe estar almacenado en el directorio presupuestos del sitio web.

- **TABLA QUEJA**

Campo	Tipo	Atributos Nulo Predeterminado		Descripción
<u>id_qja</u>	int(11)	No		identificador de la queja, es un campo auto incremental, auto_increment
nombre_dem	varchar(80)	Sí	NULL	Nombre del demandado
direccion_dem	varchar(60)	Sí	NULL	Dirección del demandado
tel_dem	int(15)	Sí	NULL	Teléfono del demandado.
nic_dem	varchar(15)	Sí	NULL	NIC del demandado
correoe_qja	varchar(50)	Sí	NULL	Correo electrónico del demandado.
desc_qja	text	Sí	NULL	Descripción de la queja.
fecha_qja	date	Sí	NULL	Fecha de interposición de la queja.
tipo_qja	varchar(15)	Sí	NULL	Tipo de queja: Queja, reclamo o sugerencia.
identidad_qso	varchar(15)	No	0	Número de identificación del quejoso.

- **QUEJOSO**

Campo	Tipo	Nulo Predeterminado		Descripción
nombre_qso	varchar(80)	Sí	NULL	Nombre del quejoso.
direccion_qso	varchar(50)	Sí	NULL	Dirección del quejoso.
ciudad_qso	varchar(20)	Sí	NULL	Ciudad del quejoso.
tel_qso	int(20)	Sí	NULL	Teléfono del quejoso.
correoe_qso	varchar(50)	Sí	NULL	Correo electrónico del quejoso.
password_qso	varchar(15)	No		Clave suministrada por el quejoso en el momento de su registro.
<u>ident_qso</u>	varchar(9)	No		Número de identidad del quejoso.
tipoident_qso	varchar(25)	Sí	NULL	Tipo de identificación del quejoso.
ciudad_ident	varchar(20)	Sí	NULL	Ciudad de registro del documento de identidad.
fecha_reg	date	Sí	NULL	Fecha de registro en el sistema. Es tomada del sistema.

- **TABLA DESCARGAQUEJAS**

Campo	Tipo	Atributos	Nulo	Extra
<u>id_descqueja</u>	int(11)		No	identificador del archivo de quejas a descargar, auto_increment
nbarvo_descqueja	varchar(37)		No	Nombre del archivo de quejas a descargar.

A continuación encontramos las tablas creadas al instalar la aplicación de encuestas Sympoll.

SYMPOLL_AUTH

Campo	Tipo	Atributos	Nulo	Predeterminado	Extra
<u>uid</u>	int(11)		No		auto_increment
user	varchar(32)		No		
pass	varchar(32)		No		
access	smallint(5)	UNSIGNED	No	0	
secret	varchar(32)		Sí	<i>NULL</i>	

SYMPOLL_DATA

Campo	Tipo	Atributos	Nulo
pid	int(10)	UNSIGNED	No
cid	int(10)	UNSIGNED	No
choice	varchar(250)		No
votes	int(10)	UNSIGNED	No

40SYMPOLL_IPLOG

Campo	Tipo	Atributos	Nulo	Predeterminado
vid	varchar(32)		No	
pid	int(10)	UNSIGNED	No	0
voted	int(10)	UNSIGNED	No	0

SYMPOLL_LIST

Campo	Tipo	Atributos	Nulo	Predeterminado	Extra
<u>pid</u>	int(10)	UNSIGNED	No		auto_increment
nextcid	int(10)	UNSIGNED	No	0	
question	varchar(250)		No		
timeStamp	int(10)	UNSIGNED	No	0	
cookieStamp	int(10)	UNSIGNED	No	0	
status	smallint(5)	UNSIGNED	No	0	

Este anexo presenta las principales pruebas realizadas para comprobar el correcto funcionamiento del sistema. Las pruebas se encuentran organizadas por subsistemas.

Para el desarrollo de estas pruebas la base de datos ha sido poblada con datos ficticios. Para cada uno de estos casos también se realizaron pruebas de validación de todos los datos solicitados al usuario.

En caso de utilizar un navegador Konkeror o Netscape, al insertar algunos datos y realizar consultas inmediatamente después, es necesario actualizar o recargar el navegador, ya que estos siempre cargan las páginas del historial, no mostrando los datos reales o actuales. Acción que no es necesario hacerlo con Internet Explorer 4.0 o superior. Igualmente en el momento de adjuntar un archivo en la aplicación de correo electrónico o adjuntar una fotografía para publicar una noticia o una actividad En estos navegadores mencionados anteriormente la aplicación no funciona correctamente, caso contrario ocurre para Internet Explorer.

Es de recordar que el navegador utilizado por la Personería Distrital de Cartagena es Internet Explorer.

Subsistema de Sesiones

Estado del sistema:		
<i>El usuario aun no ha accedido al sistema. Por lo tanto la variable de sesión aun no ha sido inicializada y por lo tanto no se requiere ningún parámetro del sistema.</i>		
Parámetros		
Nombre	Valor	Descripción
-----	-----	-----

Prueba uno subsistema de sesión

Datos de entrada		
<i>Datos necesarios para validarse en el sistema al entrar a ingresoadmin.php</i>		
Parámetros		

Nombre	Valor	Descripción
Nombre de Usuario	admin	valido
Clave de Usuario	xxxxxx	Invalido
Resultado: Datos de Salida		
Mensaje de Error sobre acceso al sistema. No se crea ninguna variable de sesión. Este mismo resultado se obtiene si el login y/o password son inválidos. "Usuario no registrado en la base de datos - sesión de usuario no registrada – Vuelva a la página de ingreso."		

Prueba con datos inválidos/ subsistema de sesión

Datos de entrada		
Dirección web (que podría ser digitada a través del navegador) a la cual desea acceder En este caso administración.php		
Parámetros		
Nombre	Valor	Descripción
Nombre de Usuario	-----	Invalido
Clave de Usuario	-----	Invalido
Resultado: Datos de Salida		
Mensaje de Error sobre acceso al sistema. No se crea ninguna variable de sesión. "Acceso no autorizado, sesión de usuario no registrada – Vuelva a la página de ingreso administración e inicie una sesión válida."		

Prueba con acceso ilegal / subsistema de sesión

Datos de entrada		
Datos necesarios para validarse en el sistema al entrar a ingresoadmin.php		
Parámetros		
Nombre	Valor	Descripción
Nombre de Usuario	admin	válido
Clave de Usuario	xxxxxx	válido
Resultado: Datos de Salida		
Muestra la página de entrada administracion.php, e inicia las variables de sesión.		
identificador de sesión		
sess_83485f25b964e1b9f727bed201809856		

Prueba con acceso y datos válidos / subsistema de sesión

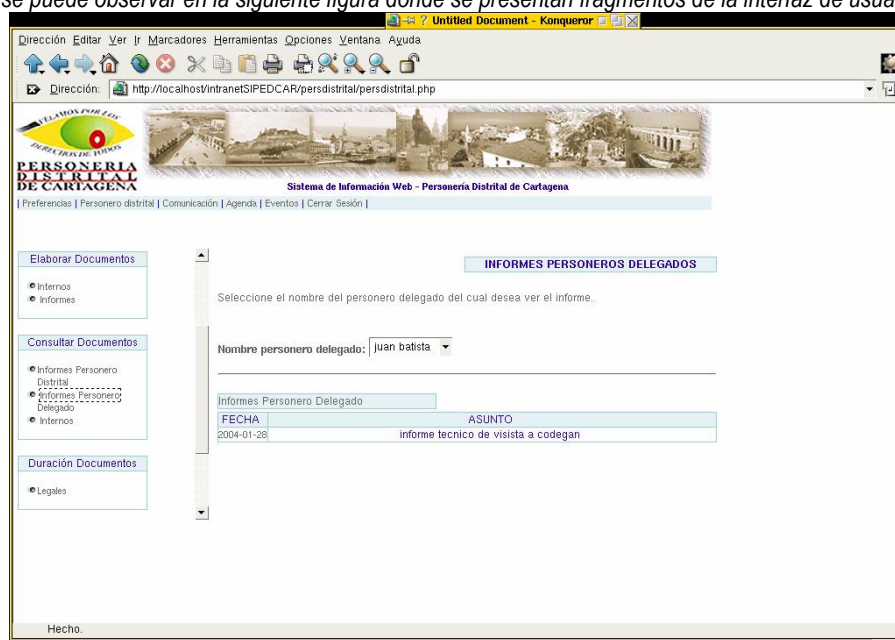
Subsistema de Interfaz

CASO 1. INTERFAZ PARA UN A ZONA DETERMINADA		
Datos de Entrada		
El usuario a través de la barra de acceso a la zona envía como dato la variable de la zona que desea ver.		
Nombre	Valor	Observación
Variable de sesión para el Perfil: \$perfil_usr	1	Identificador de perfil para el usuario actual es 1 o sea Personero Distrital

Variable de registro \$docident_usr	9097843	Documento de identidad del usuario. Este valor se mantiene mientras el usuario permanezca en el sistema.
--	---------	--

Resultado: Datos de Salida

El resultado obtenido se puede observar en la siguiente figura donde se presentan fragmentos de la interfaz de usuario.



Información	Valor Esperado	Valor Obtenido
Zonas a las que tiene acceso	Preferencias Comunicación Eventos Agenda	Preferencias Comunicación Eventos Agenda
Opciones dibujadas-	Diligenciar queja Revisar queja Asignadas Consultar documentos internos Consultar documentos legales Elaborar documentos Elaborar informes Consultar informes de delegados	Diligenciar queja Revisar queja Asignadas Consultar documentos internos Consultar documentos legales Elaborar documentos Elaborar informes Consultar informes de delegados

Subsistema de quejas

Estado del sistema:		
El usuario ya ha accedido al sistema. Por lo tanto las variables de sesión ya han sido inicializada.		
Parámetros		
Nombre	Valor	Descripción
Variable de sesión para el Perfil: \$perfil_usr	2	Identificador de perfil para el usuario actual es 2 o sea Personero Delegado

<i>Variable de sesión de la identificación del usuario: \$docident_usr</i>	33256785	<i>Identificación del personero delegado que ingresa al sistema.</i>
--	----------	--

CASO 1. REVISAR UNA QUEJA			CASO 2. DATOS VALIDOS.		
Datos de Entrada			Datos de Entrada		
<i>Se debe seleccionar una queja de las antiguas para revisar o seleccionar la opción asignadas y mirar las nuevas para determinar si se acepta o se rechaza. Luego de ser aceptada pasa a ser antigua.</i>			<i>La queja ha sido aceptada por el personero delegado para darle trámite.</i>		
Nombre	Valor	Observación	Nombre	Valor	Observación
<i>Estado queja</i>	<i>Rechazada</i>		<i>Estado queja</i>	<i>Aceptada</i>	
<i>Número radicado</i>	<i>25</i>		<i>Número radicado</i>	<i>26</i>	
			<i>Actuaciones</i>	<i>Esta es la observación que se le da a la queja en primera instancia, es la primera actuación del personero delegado</i>	
			<i>Documentos legales</i>	<i>Relaciona documentos como tutela, demanda , etc.</i>	
Resultado: Datos de Salida			Resultado: Datos de Salida		
<i>Esta queja no aparece en quejas para revisar del personero delegado, pasan a rechazadas del abogado de reparto para reasignarlas.</i>			<i>Le asigna un estado de en proceso, y la queja es almacena para futuras revisiones.</i>		

DIRECTIVA PRESIDENCIAL No. 02

PARA: Ministros del Despacho
Directores de Departamento Administrativo,
Superintendentes
Directores de Unidades Administrativas Especiales
Directores, Gerentes o Presidentes de Entidades
Descentralizadas del Orden Nacional
Gerentes o Presidentes de Empresas Industriales y
Comerciales del Estado
Gerentes o Presidentes de Sociedades de Economía
Mixta del Orden Nacional

DE: ANDRES PASTRANA ARANGO
Presidente de la Republica

REFERENCIA: Gobierno en línea

FECHA: Agosto 28 de 2000

INTRODUCCION

Las Tecnologías de la Información ofrecen una oportunidad única para que los países en vía de desarrollo den un salto en su evolución económica, política, social y cultural, disminuyendo la brecha que los separa de los países desarrollados. Estas tecnologías, y en especial Internet han trascendido los campos tecnológico y científico, constituyéndose hoy en día en herramientas que se encuentran al alcance y servicio de toda la comunidad en los ámbitos económico educativo y de salud entre otros.

Las Tecnologías de la Información son herramientas que permiten el desarrollo de una nueva economía, la construcción de un Estado más moderno y eficiente, la universalización del acceso a la información, y la adquisición y eficaz utilización del conocimiento, todos estos elementos fundamentales para el desarrollo de la sociedad moderna.

Con el fin de desarrollar este proceso, el Gobierno Nacional ha diseñado la Agenda de Conectividad, como una Política de Estado, que busca masificar el uso de las Tecnologías de la Información en Colombia y con ello aumentar la competitividad del sector productivo, modernizar las instituciones públicas y socializar el acceso a la información.

La Agenda de Conectividad es una Política de Estado presentada y aprobada por el Conpes mediante el documento 3072 del 9 de febrero de 2000 y su lanzamiento se realizó el 14 de marzo de 2000, cuya coordinación está a cargo

de la Presidencia de la República. Con ella deben estar comprometidas y es de obligatorio cumplimiento por parte de todas las entidades públicas, según se instrumenta por medio de la presente directiva.

Dentro de esta Política de Estado, el Gobierno debe ser un usuario modelo. Con este fin, se ha definido la estrategia de Gobierno en Línea de la Agenda de Conectividad, que establece como uno de sus objetivos el proveer al Estado la conectividad que facilite la gestión en línea de los organismos gubernamentales y apoye su función de servicio al ciudadano, como un complemento al esquema actual, en el que se realizan estos procesos en forma presencial en las oficinas del gobierno y se sustentan con documentos escritos en papel.

En desarrollo de esta estrategia de Gobierno, a partir del 24 de agosto de 2000, en periodo de prueba, y desde el 13 de octubre de 2000, en forma definitiva, entrará en funcionamiento el portal de [Gobiernoenlinea.gov.co.](http://Gobiernoenlinea.gov.co), cuyos principales objetivos son facilitar a los ciudadanos, empresas, funcionarios y otras entidades estatales el acceso a la

información de las entidades públicas e iniciar la integración y coordinación de los esfuerzos de las entidades en este propósito.

En consecuencia, el Presidente de la República se permite, en aras del cumplimiento de la política que ha diseñado y viene implementando la actual Administración para garantizar la modernización del Estado, impartir las siguientes instrucciones, que deberán cumplirse sin falta cubriendo los requerimientos detallados para cada una de las fases a más tardar en los plazos establecidos y cuyo seguimiento estará a cargo de la Coordinación de la Agenda de Conectividad en la Presidencia de la República y del Departamento Nacional de Planeación.

Con esta directiva es claro mi compromiso de dejar al país al finalizar mi Gobierno, el legado de un Estado eficiente y transparente, que haga uso intensivo de las Tecnologías de la información, para prestar servicios a los ciudadanos a través de un óptimo desempeño de sus funciones. Los invito a todos a unirse en forma entusiasta a este empeño.

Estrategia de Gobierno en Línea

Son responsabilidad de las entidades del Estado destinatarias de esta Directiva Presidencial el cumplimiento de estas instrucciones generales, además de las que se detallan más adelante:

1. Identificar las necesidades de tipo normativo para atenderla satisfactoriamente y en caso de ser necesario proponer o tramitar las modificaciones normativas requeridas, ante las autoridades competentes, para lo cual tendrán un plazo máximo del 31 de diciembre de 2000.
2. Incluir el cumplimiento de estas instrucciones, correspondientes a la estrategia de Gobierno en Línea de la Agenda de Conectividad, dentro del Plan Estratégico de cada entidad, con la mayor prioridad para su desarrollo eficiente y oportuno.
3. Identificar las necesidades de carácter presupuestal y hacer las provisiones correspondientes para dar cumplimiento a las instrucciones contenidas en esta directiva así como a todos los proyectos bajo su responsabilidad, dentro del marco de la Agenda de Conectividad.
4. Adecuar la infraestructura de redes y telecomunicaciones para garantizar el nivel de servicio exigido.
5. Crear en sus respectivas instituciones la cultura de trabajo utilizando tecnologías de información.
6. Adecuar la organización, los procesos y los sistemas de información internos para atender los requerimientos de información y servicios de los ciudadanos en la oportunidad y con la calidad debidos.

Las entidades del Estado deberán implantar esta política con base en las tres fases que a continuación se enumeran y se detallan más adelante. Es claro que estas fases pueden desarrollarse en forma paralela, siempre y cuando se cumpla a más tardar con las fechas límite establecidas para la puesta en marcha de cada una. Son ellas:

Fase 1: proveer información en línea a los ciudadanos, por parte de todas las entidades destinatarias de esta Directiva Presidencial, la cual deberá estar finalizada a más tardar el 31 de diciembre de 2000.

Fase 2: ofrecer servicios y trámites en línea a los ciudadanos, la cual deberá estar finalizada a más tardar el 31 de diciembre de 2001.

Fase 3: contratación en línea, la cual deberá estar finalizada a más tardar el 30 de junio de 2002.

Aquellas entidades que requieran apoyo para la implementación de esta Directiva Presidencial, deberán ponerse en contacto con la Oficina de Coordinación de la Agenda de Conectividad en la Presidencia de la República

Gobierno en Línea Fase 1: proveer información en línea a los ciudadanos

La primera fase consiste en lograr el siguiente objetivo: suministro de información en línea a los ciudadanos, por parte de todas las entidades del Estado destinatarias de esta Directiva Presidencial, en forma escalonada a partir del segundo semestre de 2000 y con fecha máxima para que la totalidad de las entidades cumplan con el 100 % del objetivo, a más tardar el 31 de diciembre de 2000. La información suministrada en línea por cada entidad incluirá, como mínimo:

- Funciones, organigrama, localización física y electrónica, teléfonos y fax, horarios de trabajo y demás indicaciones que sean necesarias para que las personas puedan cumplir sus obligaciones o ejercer sus derechos.
- Funcionarios principales.
- Normas básicas que regulan la actividad de la entidad.
- proyectos de normatividad que están en proceso de expedición .
- Regulaciones y procedimientos que le son aplicables a la entidad (incluidos los internos).
- Regulaciones, procedimientos y tramites a que están sujetas las actuaciones de los particulares, precisando de manera detallada los documentos que deben ser suministrados, dependencias responsables y plazos que se deben cumplir.
- Planes y programas en ejecución y futuros.
- Presupuesto en ejercicio y proyectado para la siguiente vigencia. Procesos de contratación a realizar y términos de referencia.
- Contratos realizados, indicando identificación del contratista, objeto, monto, plazo y demás condiciones.
- Entidades de control que vigilan a la entidad y mecanismos de control.
- Mecanismos de rendición de cuentas a la ciudadanía tales como metas e informes de gestión e indicadores de desempeño.
- Mecanismos de participación ciudadana en la gestión de la institución.
- Mecanismos de interacción ciudadana con la institución (canales de comunicación, responsables y direcciones electrónicas).

Obligaciones de las entidades:

Presentar, antes del fin de 2000, con actualizaciones semestrales, a la Oficina Coordinadora de la Agenda de Conectividad en el Ministerio de Comunicaciones y al Departamento Nacional de Planeación, la información sobre el avance en la implementación de los Programas. Dicha información deberá incluir también, el esquema y cronograma de ejecución, las fuentes de financiación y las metas a alcanzar. Estos dos organismos presentaran en forma coordinada dicho avance ante el Conpes.

7. Verificar el cumplimiento y tomar las medidas necesarias para cumplir con los indicadores definidos por la Agenda de Conectividad, para evaluar y monitorear el alcance, eficacia, oportunidad, y calidad de la información suministrada.
8. Aplicar los mecanismos que se definan por parte de la Agenda de Conectividad para el intercambio de información entre los diferentes sistemas informáticos del sector público y/o privado en forma oportuna e integral.
9. Poner a disposición de los ciudadanos como mínimo la información aquí detallada, a más tardar en la fecha indicada.
10. Mantener estos servicios y tramites en línea a disposición de los ciudadanos 7 días a la semana, 24 horas al día, garantizando disponibilidad mínima de servicio en un 95 % del tiempo.
11. Revisar y actualizar como mínimo una vez al mes la información en línea aquí detallada.
12. Colocar en línea, con una anticipación no menor a quince (15) días toda la información referente a proyectos de normatividad a expedir, planes y programas de acción futuros y términos de referencia de cualquier tipo de contratación a realizar, permitiendo el acceso abierto de los ciudadanos a dicha información y solicitando comentarios y sugerencias. La información relativa a la contratación directa, deberá colocarse en línea con una anticipación no menor a cinco (5) días, cuando así lo amerite para efectos de obtener el mayor número de ofertas.
13. Considerar y evaluar todas las propuestas que se reciban de los ciudadanos sobre los procesos mencionados anteriormente y dar respuesta a todas ellas, explicando los motivos por los cuales son o no acogidos.

Gobierno en línea Fase 2: ofrecer servicios y tramites en línea a los ciudadanos

La segunda fase consiste en lograr el siguiente objetivo: prestación de servicios, tramites y suministro información en línea adicional a aquella presentada en la fase 1 por parte de todas las entidades del Estado destinatarias de esta Directiva Presidencial, en forma escalonada a partir del segundo semestre de 2000, y con fecha máxima para que la totalidad de las entidades cumplan con el 100 % del objetivo, a más tardar el 31 de diciembre de 2001. La prestación de servicios y el suministro adicional de información en línea incluirán, como mínimo:

- Información recogida regularmente por la institución, incluidas estadísticas, estudios de competitividad, etc.
- Informes finales de estudios de interés público realizados y/o contratados por la institución.
- Trámite en línea de solicitudes, licencias, paz y salvos y certificaciones de todo tipo.
- Recepción y atención de reclamos en línea.
- Recibir todo tipo de pagos a la entidad estatal a través de transferencias electrónicas (impuestos, contribuciones, tarifas, multas, tasas, etc.
- Procesos participativos democráticos y estudios de opinión.

Para el efecto, se deberá dar especial consideración a los aspectos regulatorios relacionados con firma digital, entidades certificadoras, seguridad y protección de la privacidad de la información de los ciudadanos, que para el momento estén en vigor.

Obligaciones de las entidades:

1. Presentar, antes del fin de marzo de 2001, con actualizaciones semestrales, a la Oficina Coordinadora de la Agenda de Conectividad en el Ministerio de Comunicaciones y al Departamento Nacional de Planeación, la información sobre el avance en la implementación de los Programas. Dicha información deberá incluir también el esquema y cronograma de ejecución, las fuentes de financiación y las metas a alcanzar. Estos dos organismos presentaran en forma coordinada dicho avance ante el Conpes.
2. Poner a disposición de los ciudadanos como mínimo los servicios y tramites aquí detallados, a mas tardar en la fecha indicada.
3. Mantener estos servicios y tramites en línea a disposición de los ciudadanos 7 días a la semana, 24 horas al día. garantizando disponibilidad mínima de servicio en un 95 % del tiempo.
4. Revisar y actualizar permanentemente el tipo y contenido de tramites y servicios, para implantar en línea todos los nuevos que ofrezca la institución.

Gobierno en línea Fase 3: contratación en línea

La tercera fase consiste en lograr el siguiente objetivo: lograr que el 100 % de los procesos de contratación estén en línea por parte de todas las entidades del Estado destinatarias de esta Directiva Presidencial, en forma escalonada a partir del segundo semestre de 2000, y con fecha máxima para que la totalidad de las entidades cumplan con el 100 % del objetivo, a más tardar el 30 de junio de 2002.

Obligaciones de las entidades:

1. Presentar, antes del fin de junio de 2001, con actualizaciones semestrales, a la Oficina Coordinadora de la Agenda de Conectividad en el Ministerio de Comunicaciones y al Departamento Nacional de Planeación, la información sobre el avance en la implementación de los Programas. Dicha información deberá incluir también, el esquema y cronograma de ejecución, las fuentes de financiación y las metas a alcanzar. Estos dos organismos presentaran en forma coordinada dicho avance ante el Conpes.
2. Realizar todos sus tramites de contratación en línea, cualquiera que sea su modalidad, tiempo o lugar, a más tardar en la fecha indicada.
3. Abrir estos procesos y realizarlos en forma abierta y en línea en un proceso activo 7 días a la semana, 24 horas al día. garantizando disponibilidad mínima de servicio en un 95 % del tiempo.
4. Revisar y actualizar permanentemente el tipo y contenido de los procesos de contratación, para implantar en línea todos los nuevos que ofrezca la institución.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

*Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA*

*Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.*

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS