

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN EXTRANET PARA LA
EMPRESA QUIRÓN SOFTWARE VERSIÓN 1.0**

**ALEXI CABALLERO ESTEBAN
CARINA LIZEBETH ORDÓÑEZ ARAQUE**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2006

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN EXTRANET PARA LA
EMPRESA QUIRÓN SOFTWARE VERSIÓN 1.0**

**ALEXI CABALLERO ESTEBAN
CARINA LIZEBETH ORDÓÑEZ ARAQUE**

**Trabajo presentado como requisito para optar al título de
Ingeniero de Sistemas**

**Director
Fernando Ruíz Díaz**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2006**

UNIVERSIDAD INDUSTRIAL DE SANTANDER

PROYECTO DE GRADO

TITULO:

**SISTEMA DE INFORMACIÓN EXTRANET PARA LA EMPRESA
QUIRÓN SOFTWARE VERSIÓN 1.0**

PRESENTADO ANTE:

Cómite de Proyectos de Grado

ESCUELA:

Ingeniería de Sistemas e Informática.

FACULTAD:

Ingenierías Físico-Mecánicas

ELABORADO POR:

Alexi Caballero Esteban

Carina Lizebeth Ordóñez Araque

DIRECTOR:

Ing. Fernando Ruíz Díaz

Profesor Titular Escuela de Ingeniería de
Sistemas e Informática.

Agradecimientos

A Dios... la fortaleza, energía y fe

A la familia... su apoyo e incondicionalidad

A cada uno de los miembros de la empresa Quirón Software

Al profesor Fernando... sus aportes, colaboración y buena disposición

A todas las demás personas que han colaborado a lo largo de la carrera y de este
proyecto.

TITULO

DESARROLLO DE UN SISTEMA DE INFORMACIÓN EXTRANET PARA LA EMPRESA QUIRON SOFTWARE VERSIÓN 1.0.**

AUTORES

Caballero Esteban, Alexi
Ordóñez Araque, Carina Lizebeth*

PALABRAS CLAVES

Lenguaje unificado de modelado, sistema de información Web, extranet, arquitectura de 3 capas, aplicaciones cliente-servidor, asp.net, gestión de proyectos.

RESUMEN

En la actualidad toda organización, no importa el tamaño ni la complejidad, debe tener un sitio Web. Contar con un sitio Web corporativo ha dejado de ser una opción para convertirse en un aspecto indispensable, y es un tema indiscutible si pensamos en realizar negocios internacionalmente. Por su bajo costo y facilidad de uso, las redes externas virtuales se constituyen en el espacio ideal de colaboración con los proveedores, socios y clientes. Todo a través del Web y las 24 horas del día, llegando a convertirse en una herramienta fundamental en el mundo de los negocios.

El sistema de información extranet, es una aplicación Web para el manejo de la información de la empresa que asegure la coherencia e integración de los participantes en proyectos e investigaciones existentes en la empresa Quirón Software.

Mediante la implementación de la Extranet, se le brinda a la empresa Quirón Software, la posibilidad de integrar en un sistema en línea la mayoría de los archivos y datos que maneja día a día, facilitando el acceso a la información, a la vez que se evita el acceso externo a los datos y se evita tener enormes cantidades de documentación impresa.

** Proyecto de Grado modalidad Investigación

* Facultad de Ingeniería Físico-Mecánicas, Escuela de Ingeniería de Sistemas e Informática, RUÍZ DÍAZ, Fernando.

TITLE

EXTRANET INFORMATION SYSTEM FOR THE QUIRON COMPANY
SOFTWARE VERSION 1.0**.

AUTHORS

Caballero Esteban, Alexi
Ordóñez Araque, Carina Lizebeth*

KEYWORDS

Web information system, applications client-server, UML, extranet, architecture from 3 covers, asp.net

ABSTRACT

On the reality all organization, never mind the size not the intricacy, debit have a site Web. Report with a site Web corporate she has left than being a refusal to convert on a aspect indispensable , and it's a theme indisputable if intended on realize trade internationally. Through his low cost and ease than use, the traps external virtual oneself constitutes on the room ideal than collaboration with the provider, partners and clients. All through of the Web and the 24 times of the day, arriving for convert on a tool fundamental on the world of the trade.

The system than information foreign, she is an application web to the running of the information of the company than insurable the coherence and integration of the participating on projects and research existent on the company Operating theater Software.

Through the implementation of the Foreign, oneself him make a toast at the company Operating theater Software, the possibility than integrate on a system online the majority of the registries and data than manageable day for day, facilitation the access at the information, at the time that oneself she avoids the access external at the age of data and oneself she avoids have thundering quantities than documentation printed.

** Project

* Physique-Mechanics Sciences Department, Computer Science Engineering, RUÍZ DÍAZ, Fernando.

GLOSARIO

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

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

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

FIREWALL. Es un dispositivo que funciona como cortafuegos entre redes, permitiendo o denegando las transmisiones de una red a la otra.

IDE. Es un entorno de desarrollo integrado compuesto por un conjunto de herramientas para un programador.

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

INTERFAZ. La idea fundamental de la interfaz es el de la mediación. La interfaz es lo que “media”, lo que facilita la comunicación, la interacción, entre dos sistemas de diferente naturaleza.

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

SCRIPT. Es una secuencia de órdenes, en un determinado lenguaje, que puede ser ejecutado por un cliente Web desde su navegador y visualizar el documento que está contenido. Actualmente los lenguajes de script más usados en páginas Web son JavaScript y VBScript.

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

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

UML. Lenguaje Unificado de Modelado. Es un lenguaje estándar para escribir planos de software, puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

XML. Es un metalenguaje extensible de etiquetas desarrollado por el W3C (World Wide Web Consortium), permite definir lenguajes para diferentes necesidades.

TABLA DE CONTENIDO

INTRODUCCIÓN	19
PARTE I: FUNDAMENTOS.....	22
DESCRIPCIÓN DEL PROYECTO.....	25
1.1 ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA	26
1.2 OBJETIVOS	27
1.2.1 Objetivo general.....	27
1.2.2 Objetivos específicos	27
1.3 DESCRIPCION DE OBJETIVOS	28
1.4 IMPACTO Y VIABILIDAD	30
1.4.1 Impacto	31
1.4.2 Viabilidad	32
MARCO TEÓRICO	33
2.1 FUNDAMENTACIÓN TEÓRICA GENERAL.....	34
2.1.1 Sistema de Información.	34
2.1.2 Modelo Cliente-Servidor	36
2.1.3 Definición del World Wide Web	39
2.1.4 Intranet.....	40
2.1.5 Extranet	41
2.1.5.1 Principios de la Extranet.....	43
2.1.5.2 Ventajas de la Extranet	44

2.1.5.3 Desventajas de la Extranet	44
2.1.6 Estructura de las aplicaciones Web	45
2.1.6.1 Modelos de organización lógica de una Aplicación Basada en Web.....	45
2.1.7 Arquitectura de aplicaciones de .net.....	46
2.1.7.1 Diseño de capas de presentación	48
2.1.7.2 Diseño de capas empresariales	50
2.1.7.3 Diseño de capas de datos.....	52
2.1.7.4 Directivas de seguridad y administración	55
2.2 FUNDAMENTACION TEORICA ESPECÍFICA	63
2.2.1 Internet Information Server (IIS)	63
2.2.2 Active server pages .NET (aspx)	64
2.2.3 SQL Server	68
2.2.4 UML	69
2.2.5 CSS (Cascading Style Sheet).....	71
MARCO METODOLÓGICO.....	73
3.1 CICLO DE VIDA DEL SOFTWARE	74
3.1.1 Tipos ciclos de vida del desarrollo software.....	75
3.1.2 Selección del ciclo de vida.....	75
PARTE II: DESARROLLO DEL SISTEMA	81
FASE DE INICIO	82
4.1 ALCANCE DEL PROYECTO	83
4.2 COMPRENDER EL CONTEXTO DEL SISTEMA.....	85
4.2.1 Modelo del Negocio	85
4.3 ANÁLISIS DE REQUERIMIENTOS.....	87
4.3.1 Casos de Uso del Negocio	90
4.3.2 Priorizar casos de uso	95
4.3.3 Atributos de la Aplicación.....	95
4.3.4 Requisitos Adicionales.....	96

4.3.5 Requisitos Visuales.....	97
4.3.6 Otros requisitos.....	98
4.4 FLUJO DE TRABAJO DE ANÁLISIS	98
4.5 ANÁLISIS DE LA ARQUITECTURA CANDIDATA	99
4.6 RIESGOS Y PLANES DE CONTINGENCIA.....	100
4.6.1 Diseño e Implementación.....	100
4.6.2 Los Datos	101
4.6.3 Seguridad.....	102
4.6.4 Los Usuarios	103
4.6.5 Técnicos de tecnologías anticuadas	104
4.6.6 Conocido de fuentes de información poco fiables.....	104
FASE DE ELABORACIÓN.....	106
5.1 DISEÑO GLOBAL	107
5.1.1 Arquitectura por capas.....	108
5.1.1.1 Capa de Datos	109
5.1.1.2 Capa de Reglas del Negocio.....	110
5.1.1.3 Capa de Presentación.....	110
5.1.2 Paquetes.....	111
5.1.3 Diagrama de clases	112
5.2 DISEÑO ARQUITECTÓNICO	115
5.2.1 Estructura del sitio	116
5.2.2 Arquitectura del sistema	117
5.3 DISEÑO DE NAVEGACIÓN.....	118
5.3.1 Diseño de Interfaz.....	119
5.3.2 Composición y distribución de las páginas	123
5.3.2 Estilo: Fuentes, Colores y Tamaños	127
5.3.3 Lineamientos de programación.....	128
5.4 DIRECTRICES DE PROGRAMACIÓN	128
5.5 MODELO DE DATOS DEL SISTEMA.....	129
FASE DE CONSTRUCCIÓN	137

6.1 PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN	138
6.2 CASOS DE USOS.....	138
6.2.1 Análisis de Casos de Usos	142
6.3 FLUJO DE PRUEBA	146
6.3.1 Resultados de las pruebas	150
CONCLUSIONES	151
RECOMENDACIONES.....	153
ANÁLISIS DE LA LITERATURA RECOPILADA	155
ANEXOS.....	160

LISTADO DE FIGURAS

Figura 1. Modelo Cliente/Servidor.....	37
Figura 2. Intranet Corporativa	41
Figura 3. Modelo Extranet.....	42
Figura 4. Componentes separados en capas según sus funciones.....	47
Figura 5. Capa de presentación.....	49
Figura 6. Capas de componentes empresariales.....	51
Figura 7. Componentes de datos.....	54
Figura 8. Efecto de las directivas organizativas en el diseño de aplicaciones ..	55
Figura 9. Aspectos de la directiva de seguridad	57
Figura 10. Aspectos de la directiva de administración operativa	59
Figura 11. Aspectos de la directiva de comunicaciones	60
Figura 12. Modelo de prototipado evolutivo.	80
Figura 13. Diagrama relación de Casos de Uso entre actores del Sistema....	88
Figura 14. Diagrama casos de uso actor Empleado	91
Figura 15. Diagrama de Casos de Uso actor Cliente.....	92
Figura 16. Diagrama de Casos de Uso actor Administrador	93
Figura 17. Diagrama de casos de uso General.....	94
Figura 18. Arquitectura General de la herramienta Extranet.....	108
Figura 19. Diagrama de paquetes utilizados en la herramienta.....	112
Figura 20. Diagrama de clases	113
Figura 21. Diagrama general de la estructura del Sitio Extranet.....	116
Figura 22. Arquitectura del Sistema.....	117
Figura 23. Diagrama de clases de interfaz de clase para un formulario de resultados	124
Figura 24. Diagrama de clase de interfaz para un formulario de ingreso/modificación y consulta.	125

Figura 25. Maqueta General	126
Figura 26. Maqueta General de Zona de contenido: Formulario general de Resultados	126
Figura 27. Maqueta General de Zona de contenido: Formulario para ingreso/modificación/consulta de información.....	127
Figura 28. Modelo General Entidad Relación	131
Figura 29. Modelo de datos de proyectos implementado en la Extranet	132
Figura 30. Modelo de Datos Personal.....	133
Figura 31. Modelo de datos módulo de seguridad (Usuarios)	135
Figura 32. Diagrama de casos de uso clasificados en paquetes	139
Figura 33. Caso de Uso Paquete Información Básica	140
Figura 34. Casos de uso paquete Clientes	140
Figura 35. Caso de uso paquete Proyectos.....	141
Figura 36. Caso de uso Empleados.....	141
Figura 37. Caso de uso Usuarios.....	142
Figura 38. Diagrama de Colaboración de actualizar proyectos	142
Figura 39. Diagrama de Colaboración de solicitar soportes	143
Figura 40. Diagrama de Secuencia de actualizar proyectos.....	144
Figura 41. Diagrama de secuencia de solicitar soportes	145

LISTADO DE TABLAS

Tabla 1. Impacto técnico, económico y social.....	31
Tabla 2. Viabilidad técnica, económica y social	32
Tabla 3. Modelo de Estructura Lógica básicos	46
Tabla 4. Capas de la Arquitectura de tres capas aplicadas a la Extranet	61
Tabla 5. Diagramas UML usados en el proyecto	71
Tabla 6. Selección del ciclo de vida	78
Tabla 7. Actores del Sistema	89
Tabla 8. Atributos de la Aplicación.....	95
Tabla 9. Factores que afectan la arquitectura.....	99
Tabla 10. Iteraciones del proyecto	138
Tabla 11. Pruebas de caja negra	149

LISTADO DE ANEXOS

ANEXO A. CASOS DE USO.....	161
ANEXO B. DICCIONARIO DE DATOS.....	177
ANEXO C. ORIENTACIONES DE PROGRAMACION	195
ANEXO D. GUIA RAPIDA DE LA HERRAMIENTA EXTRANET EN SU ESTADO FINAL.....	205

INTRODUCCIÓN

En la actualidad toda organización, no importa el tamaño ni la complejidad, debe tener un sitio Web. Contar con un sitio Web corporativo ha dejado de ser una opción para convertirse en un aspecto indispensable, y es un tema indiscutible si pensamos en realizar negocios internacionalmente. Por su bajo costo y facilidad de uso, las redes externas virtuales se constituyen en el espacio ideal de colaboración con los proveedores, socios y clientes. Todo a través del Web y las 24 horas del día, llegando a convertirse en una herramienta fundamental en el mundo de los negocios.

Quirón Software, para cumplir con una de sus metas, como es la competitividad de la corporación mediante la incorporación de nuevas tecnologías informáticas al desarrollo de sus labores, desea implementar un espacio donde se pueda llevar a cabo los procesos necesarios que conlleven al ofrecimiento de servicios al cliente, a empleados, proveedores. Por tales motivos, se hace necesario un sistema de información Extranet que integre a todos y cada uno de los miembros de la empresa, apoyados en tecnologías Web.

El amplio crecimiento de las Intranets ha permitido que las organizaciones hayan alcanzado nuevos objetivos y mejorado su funcionamiento, desde su interior. Sin embargo, queda un aspecto para evaluar. El punto de vista externo. Los sitios Web tradicionales son utilizados para desplegar información general de la compañía pero muy poco involucran a sus usuarios en los procesos e información interna, la que realmente sirve como motor de la organización. Las intranets ya soportan la comunidad interna de la organización. El nuevo modelo que permite alcanzar una interacción de los clientes con la parte interna es la Extranet.

Una Extranet es una red privada que utiliza los estándares de Internet y sistemas de telecomunicación públicos para compartir de una forma segura parte de la información (de negocios) u operaciones con clientes externos. En la práctica, una Extranet puede ser vista como una Intranet que es externa a los usuarios. Por lo general, el sistema telecomunicaciones utilizado en Internet, es público, de bajo costo y de fácil adquisición.

Las organizaciones pueden utilizar las Extranets para:

- Intercambiar grandes volúmenes de información.
- Utilizar intercambio Electrónico de Datos.
- Compartir Datos, noticias, etc.
- Colaborar con otras organizaciones.
- Generar aplicaciones y servicios de valor agregado a clientes externos.

Cada organización tiene la libertad de diseñar e implementar una gran variedad de aplicaciones para el uso de la Extranet, según necesidades internas y externas. El uso de los servicios como el WWW hacen a las Extranets sistemas flexibles, escalables, portables y de fácil integración en forma distribuida, abierta a ambientes heterogéneos y reductoras de las barreras de la red de la organización.

PARTE I: FUNDAMENTOS

El propósito de este informe es mostrar los conceptos teóricos que apoyaron el desarrollo del trabajo y explicar la forma cómo se diseñó e implementó el sistema de información extranet para la empresa Quirón software. El documento se ha organizado en dos partes así:

Parte I: Presenta los fundamentos del proyecto incluyendo los aspectos generales, el marco teórico y metodológico, que sirvieron de guía para el desarrollo del mismo.

Capítulo 1. PRESENTACIÓN DEL PROYECTO

Se muestra el ambiente, las causas y las necesidades, los objetivos, la guía de la planeación, el diseño y el desarrollo que condujeron a la realización de este proyecto como solución al problema.

Capítulo 2. MARCO TEORICO

Se describen las diferentes teorías y tecnologías aplicadas a la solución del problema, exponiendo conceptos que sirvieron de base al desarrollo del Sistema.

Capítulo 3. MARCO METODOLOGICO

Presenta una comparación entre las metodologías existentes para el desarrollo del software y explica en detalle la metodología del Prototipado Evolutivo, seleccionada como guía en este proyecto.

Parte II: Presenta el desarrollo del sistema a lo largo de cada una de las fases o etapas del proyecto.

Capítulo 4. FASE DE INICIO

Explica la fase de inicio, que permite poner en marcha el proyecto y definir el alcance del mismo.

Capitulo 5. FASE DE ELABORACIÓN

Explica la fase de diseño, que permite definir la arquitectura, el diseño y la interfaz del proyecto.

Capitulo 6. FASE DE CONSTRUCCIÓN

Explica la fase de construcción, que permite explicar la implementación del proyecto.

El informe finaliza con los aportes, conclusiones y recomendaciones de los autores sobre el proyecto.

1

DESCRIPCIÓN DEL PROYECTO

Se exponen los aspectos generales que describen la propuesta de investigación titulada “Sistema de Información Extranet para la empresa Quirón Software versión 1.0”.

El objetivo de este capítulo es ubicar al lector en el qué, para qué y cómo del trabajo realizado. Se presenta, en su orden: la descripción de la situación de interés, los objetivos y su descripción, el impacto que se prevé con el desarrollo del trabajo.

1.1 ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA

Quirón Software es una empresa de tecnología informática, dedicada al desarrollo de soluciones sistematizadas, así como la prestación de servicios de consultoría y asesoría en sistemas de información.

Desde hace algunos años, la empresa cuenta con un sitio Web estático, donde los clientes pueden consultar información de ésta, tales como la parte corporativa, portafolio, tecnologías, alianzas, entre otras. Sin embargo, no se dispone de las herramientas necesarias para la actualización de la información y la creación de nuevos servicios ofrecidos, así como de una mayor comunicación con los clientes.

La empresa ha analizado que los procesos administrativos que se ejecutan, no se llevan a cabo de una manera adecuada; frente a esta situación ha considerado el desarrollo de una plataforma Web orientada a las personas que son externas a su empresa (Extranet) pero que necesitan acceder en algún momento determinado a información correspondiente a los proyectos desarrollos y asesorías efectuados con la empresa, tales como seguimiento a los proyectos, entregables y consultorías, entre otros.

Mediante la implementación de la Extranet, se le brinda a la empresa Quirón Software, la posibilidad de integrar en un sistema en línea la mayoría de los archivos y datos que maneja día a día, facilitando el acceso a la información, a la vez que se evita el acceso externo a los datos y se evita tener enormes cantidades de documentación impresa.

1.2 OBJETIVOS

1.2.1 Objetivo general

Desarrollar y construir una Extranet para la Empresa Quirón Software soportada en tecnología Web, que asegure la coherencia e integración de los participantes en proyectos e investigaciones existentes en la empresa; permitiendo de esta manera, manejo y análisis de la información para tener un acceso oportuno y preciso, que sea de ayuda en el proceso de consulta y de toma de decisiones.

1.2.2 Objetivos específicos

- Implementar una aplicación con herramientas Web que permita:

- Organizar y proporcionar el acceso al sistema de información por el esquema rol-usuario, mediante la creación y actualización de las cuentas de cliente y los permisos de acceso a las diferentes opciones que conforman el sistema.
- La comunicación entre los diferentes usuarios, clientes y empleados por medio de la cartelera de eventos y noticias.
- El seguimiento de proyectos, cubriendo las diferentes etapas del ciclo de vida del proyecto.
- La consulta del banco de proyectos realizados anteriormente y el acceso a los documentos que se generaron durante su desarrollo.
- El registro y la consulta de información básica de las propuestas, personal de trabajo y clientes en un proyecto.
- La recolección de datos concernientes a la administración del proyecto para luego presentar informes que ayuden al proceso de evaluación y la toma de decisiones.

1.3 DESCRIPCION DE OBJETIVOS

La aplicación brindará a las personas hacia quienes va dirigida, el seguimiento de proyectos en cada una de sus fases desarrolladas por la empresa Quirón software.

La aplicación Web contará con 2 módulos, el primero de ellos consiste en los servicios orientados a empleados y clientes, que vía Web podrán efectuar:

Consultas y seguimientos de proyectos. La empresa pondrá a disposición del personal de la empresa y de clientes, una serie de servicios que servirán de apoyo a las actividades diarias e incrementarán el rendimiento laboral. Dichos servicios serán manejados a través de opciones, los cuales podrán integrarse al módulo de proyectos como nuevas piezas de este.

Consultas al banco de proyectos. Los usuarios podrán acceder a información de proyectos anteriormente desarrollados por la empresa.

Registro y consulta de propuestas, personal de trabajo y clientes en un proyecto. A través de estos servicios, el usuario podrá registrar, actualizar y consultar información personal y básica, para posterior publicación, a la que tendrán acceso el personal del trabajo y clientes vinculados a un proyecto específico.

Informes para la toma de decisiones. Se podrá generar informes de los proyectos, para la toma de decisiones.

Comunicación a través de la Cartelera de Eventos y Noticias. Es la manera que la empresa busca una comunicación e integración con los clientes y empleados, adicional al correo electrónico ya existente.

El segundo módulo, es el módulo administrador de contenidos, donde un usuario administrador podrá realizar el registro y actualización de los usuarios de la aplicación y con ello facilitar el proceso de autenticación en el momento del ingreso a ésta para asegurar la integridad del sistema, de igual forma, este módulo permitirá registrar proyectos y asignarlos a las personas realizadas. Además, permite registrar eventos, noticias de interés general.

1.4 IMPACTO Y VIABILIDAD

El beneficio del desarrollo de este proyecto así como la posibilidad de llevarlo a cabo con éxito se describe a continuación:

1.4.1 Impacto

Técnico	Económico	Social
* Se contará con una herramienta tecnológica versátil y de fácil acceso, para almacenar y organizar la información, a la vez que se protege el acceso a la misma. * El desarrollo del proyecto propone una nueva perspectiva respecto a la forma como se realiza la comunicación con clientes y empleados de la empresa Quirón Software, ofreciendo servicios que permiten la publicación de información de interés específica para cada usuario. * Se logra aprovechar recursos informáticos y telemáticos por medio de la automatización de los procesos involucrados, lo que además conlleva a un mejoramiento.	* El disponer de una herramienta que asista en el seguimiento de proyectos y actividades, le aporta al usuario (empleados, clientes), mayor organización, facilidad para la manipulación de datos esenciales para la realización de proyectos de investigación y desarrollo, lográndose una disminución de tiempo y disminución del costo de personal.	* Los clientes y empleados contarán con la posibilidad de realizar el seguimiento de los proyectos e investigaciones, accediendo desde fuera de la empresa a través de servicios de Internet disponibles. * Se logra aprovechar los servicios que se pondrán a disposición las 24 horas del día.

Tabla 1. Impacto técnico, económico y social

1.4.2 Viabilidad

Técnico	Económico	Social
* La empresa ya cuenta con los equipos y las herramientas técnicas requerida para cumplir a cabalidad el proyecto; además, se cuenta con los recursos humanos que poseen la capacidad de asimilar y compartir conocimientos de nuevas tecnologías necesarias y apropiadas para desarrollar una Extranet. * El desarrollo del proyecto se hace con herramientas que aportan numerosas ventajas en seguridad y robustez, y se cuenta con las licencias de Windows 2003 Server, de Visual Studio .Net 2003 y de SQL Server 2000.	* Dado que para el desarrollo de este proyecto se dispone de los recursos e infraestructura necesaria para su realización, se diría que es absolutamente factible puesto que el beneficio económico no solo se observa durante el proceso de evolución del proyecto sino también en su posterior utilización, mostrando una disminución de costos por la efectividad de su aplicación respectivo a la inversión realizada.	* Los efectos colaterales con la aplicación de este sistema serán ciertamente provechosos, pues además de beneficiar a empleados también afectará de manera positiva a todas las empresas y particulares que de una u otra forma están relacionados con la empresa Quirón Software, haciendo más cómoda su labor gracias al uso de este sistema.

Tabla 2. Viabilidad técnica, económica y social

MARCO TEÓRICO

Se plasman los aspectos relevantes para una buena comprensión de la teoría necesaria en el desarrollo de este proyecto. Este marco tiene el propósito de dar a la investigación un sistema coordinado y coherente de conceptos y proposiciones que permitan abordar el problema.

2.1 FUNDAMENTACIÓN TEÓRICA GENERAL

2.1.1 Sistema de Información.

Un sistema de información¹ es un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio:

- El equipo computacional: el hardware necesario para que el sistema de información pueda operar.
- El recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

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

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas

¹ OZ, Eddy. Administración de Sistemas de información. Segunda Edición. Thomson Learning. México D.F., 2001.

son datos o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfases automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáners, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecida. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace

posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo. En este caso, también existe una interfase automática de salida. Por ejemplo, el Sistema de Control de Clientes tiene una interfase automática de salida con el Sistema de Contabilidad, ya que genera las pólizas contables de los movimientos procesales de los clientes.

2.1.2 Modelo Cliente-Servidor

La **arquitectura cliente-servidor** llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realizada se efectúe con la mayor eficiencia, y permita simplificarlas.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

En el modelo CLIENTE/SERVIDOR podemos encontrar las siguientes características:

1. El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes.
2. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma.

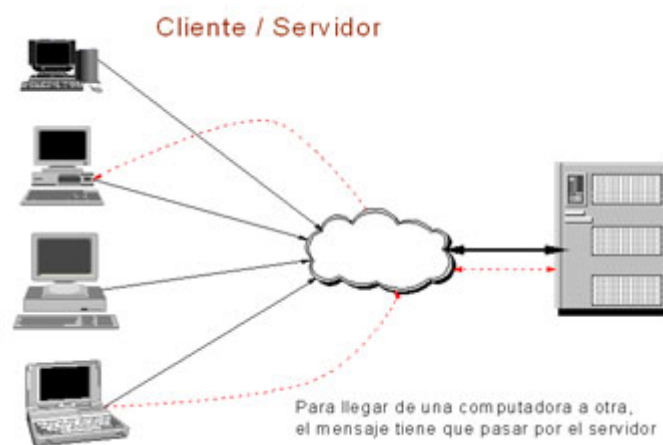


Figura 1. Modelo Cliente/Servidor

3. Un servidor da servicio a múltiples clientes en forma concurrente.
4. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por

actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final.

5. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.

6. Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un solo sistema a las estaciones Clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final.

También es importante hacer notar que las funciones Cliente/Servidor pueden ser dinámicas. Ejemplo, un servidor puede convertirse en cliente cuando realiza la solicitud de servicios a otras plataformas dentro de la red.

Su capacidad para permitir integrar los equipos ya existentes en una organización, dentro de una arquitectura informática descentralizada y heterogénea.

7. Además se constituye como el nexo de unión mas adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo.

8. Designa un modelo de construcción de sistemas informáticos de carácter distribuido:

1. Su representación típica es un centro de trabajo (PC), en donde el usuario dispone de sus propias aplicaciones de oficina y sus propias bases de datos, sin dependencia directa del sistema central de información de la organización, al tiempo que puede acceder a los
2. recursos de este host central y otros sistemas de la organización ponen a su servicio.

2.1.3 Definición del World Wide Web²

El WWW es un sistema de información y comunicación usado popularmente usado sobre Internet con comunicaciones de datos que emplean, el modelo Cliente-Servidor.

Los clientes del Web (Browsers) pueden acceder a la información que precisa multiprotocolos e hipermedios (donde las aplicaciones de ayuda están disponibles para los browsers) usando un esquema de direccionamiento.

El Web permite tanto la diseminación como la colección de información (a través de la capacidad de formatos HTML), de este modo, el Web no es sólo

² Tomado del documento electrónico Un sitio en Internet descargado del sitio <http://www.monografias.com>

un sistema de diseminación o difusión de información sino que incluye el potencial de la comunicación interactiva.

2.1.4 Intranet

Una Intranet es una red privada empresarial que utiliza los protocolos TCP/IP de Internet para su transporte básico. Los protocolos pueden ejecutar una variedad de hardware de red y pueden coexistir con otros protocolos de red, como IPX. Aquellos empleados que están dentro de una Intranet pueden acceder a los amplios recursos de Internet, pero aquellos que estén en Internet no pueden entrar a la Intranet.

Una intranet es la implantación o integración en una red local o corporativa de tecnologías avanzadas de publicación electrónica basadas en WEB en combinación con servicios de mensajería, con partición de recursos, acceso remoto y toda una serie de facilidades cliente/servidor proporcionadas por la pila de protocolos TCP/IP, diseñados inicialmente para la red global Internet.



Figura 2. Intranet Corporativa

Su propósito fundamental es optimizar el flujo de información con el objeto de lograr una importante reducción de costos en el manejo de documentos y comunicación interna. Es una herramienta de gestión que permite una potente difusión de información y mecanismos de colaboración entre el personal. Al igual que el Internet, la pieza clave es el WWW, pero de forma que la información de la empresa esté accesible sólo a los miembros de la organización.

2.1.5 Extranet³

Una extranet es una red privada que utiliza el protocolo de Internet y el sistema de comunicaciones público para compartir en forma segura información de negocios u operaciones con proveedores, vendedores, clientes u otros

³ Tomado de <http://es.wikipedia.org/wiki/Extranet>.

negocios. Una extranet puede ser vista como parte de la intranet de una compañía que se extiende a usuarios de fuera de la compañía. También puede ser descrito como un "estado del la mente" en la cual la Internet es percibida como una forma de hacer negocios con otras compañías, como por ejemplo vender productos a sus clientes. Los mismos beneficios que ofrece el HTML, SMTP y otras tecnologías de Internet ahora se diseñan para acelerar negocios entre comerciantes.

Una extranet requiere seguridad y privacidad, y hace uso de servidores firewall. Estos firewall, utilizan la certificación digital, o medios similares a la certificación, como encriptación de mensajes y el uso de redes privadas virtuales que se canalizan a través de la red pública.

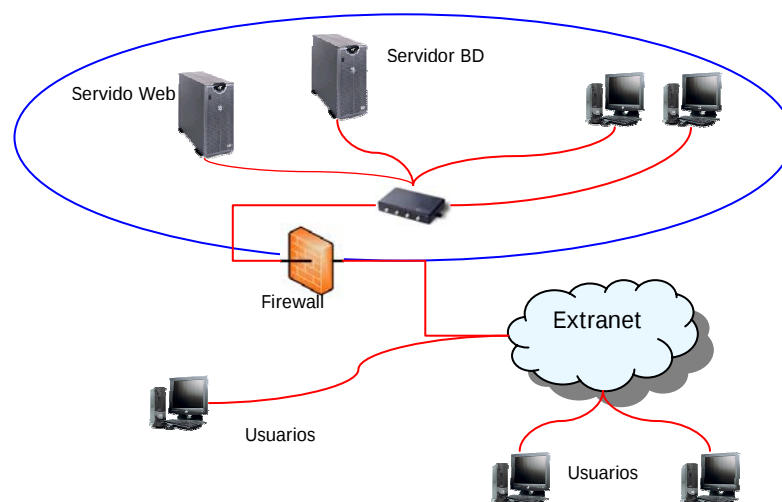


Figura 3. Modelo Extranet

2.1.5.1 Principios de la Extranet

En esto consiste y así funciona un sitio extranet:

- Un sitio extranet es un sitio Web que puede crear dentro del ámbito de su empresa para dedicarlo a la colaboración con sus socios.
- Los usuarios se conectan a un sitio extranet del mismo modo que a Internet: por medio de un explorador.
- Las extranets pueden alojar los mismos tipos de contenido que los sitios Web, como documentos, gráficos, audio, vídeo y tableros de anuncios.
- Es posible personalizar el acceso a la extranet para cada cliente.
- Puede otorgar o restringir el acceso a clientes y socios a determinados sectores de la extranet.
- Puede emplear protección antivirus y bloquear determinadas clases de archivos de la extranet a fin de proteger la red de la empresa.
- Es esencial que la extranet esté protegida con contraseña como medida de seguridad y para garantizar que sólo tengan acceso quienes usted desea que lo tengan.

2.1.5.2 Ventajas de la Extranet

- Costos asequibles, tanto de su puesta en marcha como de su uso. Es una forma muy eficiente y económica de distribuir cierta información interna, sustituyendo los medios clásicos.
- Fácil adaptación y configuración a la infraestructura tecnológica de la organización, así como gestión y manipulación. Disponible en todas las plataformas informáticas.
- Adaptación a las necesidades de diferente niveles: empresa, departamento y área de negocios, entre otros. Centraliza el acceso actualizada de la información de la empresa.
- Posibilidad de integración con las bases de datos internas de la organización.
- Acceso a la Internet, tanto al exterior, como al interior, por parte de usuarios registrados con control de acceso.

2.1.5.3 Desventajas de la Extranet

Es importante manejar el tema de seguridad y aplicar planes de control de la Extranet pues son redes expuestas a notables riesgos de esta índole.

2.1.6 Estructura de las aplicaciones Web

Hay dos tipos de estructuras en cualquier aplicación Web: la lógica y la física⁴. La estructura lógica reseña cómo se relacionan entre sí los documentos que componen la aplicación y define los vínculos entre estos. Sin embargo, la ubicación lógica de los documentos dentro del sitio no tiene por qué estar relacionada con la ubicación física real del documento, cosa que sí describe la estructura física. La estructura física muestra, por ejemplo, el camino del directorio de un documento en un servidor Web o su ubicación en una base de datos.

2.1.6.1 Modelos de organización lógica de una Aplicación Basada en Web

Existen variedades de modelos de organización lógica de una aplicación Web; sin embargo, dependiendo del tipo de contenido a presentar estos modelos son aplicables individualmente o en conjunto con otros. Algunas aplicaciones poseen una arquitectura lógica sencilla, tal como la progresión lineal de páginas, mientras que otras exhiben formas reticulares, jerárquicas o de red (Web pura). En la tabla 3 se presentan los cuatro modelos de organización lógica básicos para el diseño de aplicaciones Web.

⁴ Pese a presentar la estructura de una aplicación Web de una manera diferenciada, no se desconoce que toda aplicación basada en Web tiene una estructura arquitectónica global que va ligada a la finalidad, el contenido, la audiencia y la filosofía de la navegación de la aplicación.

Estructura Lineal	• Dispone de las páginas a presentar de manera predefinida, siguiendo una progresión lineal a través del cuerpo de la información. • Proporciona al usuario un esquema de navegación predecible y fácil de navegar.
Estructura Reticular o de rejilla	• Organiza el contenido en dos dimensiones, cada una correspondiente a una categoría, proporcionando al usuario orientación horizontal y vertical para la navegación.
Estructura jerárquica	• Dispone las páginas a lo largo de las ramas de un árbol de jerarquías, y ocasionalmente posibilita el flujo transversal atravesando la estructura principal, Es la estructura lógica más común de encontrar, • Conduce al usuario a lo largo de un árbol de navegación predefinido.
Estructura de red o Web pura	• Dispone de las páginas de manera que cada una está vinculada a todas las demás. • Permite una flexibilidad de navegación considerable, aún cuando puede resultar confuso para el usuario.

Tabla 3. Modelo de Estructura Lógica básicos

2.1.7 Arquitectura de aplicaciones de .net⁵

La arquitectura del software alude a la estructura global del mismo y a las formas en que ésta proporciona la integridad conceptual de un sistema. En su forma más simple, la arquitectura es la estructura jerárquica de los componentes del programa (módulos), la manera en que interactúan y la estructura de datos que van a utilizar los componentes.

⁵ <http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>

Debido a que la herramienta software producto del proyecto se concibió para ser manejada en un ambiente Web, la arquitectura que permite una mejor representación de su estructura global es la denominada: arquitectura de tres capas. Es importante tener en cuenta que las capas son simplemente agrupaciones lógicas de los componentes de software que conforman la aplicación o servicio. Ayudan a diferenciar entre los distintos tipos de tareas que realizan los componentes, facilitando el diseño de la reutilización en la solución. Cada capa lógica contiene un número de tipos de componentes discretos agrupados en subcapas, cada una de las cuales realiza el mismo tipo de tarea específica. En la figura 4 se muestra una visión simplificada de una aplicación y sus capas.

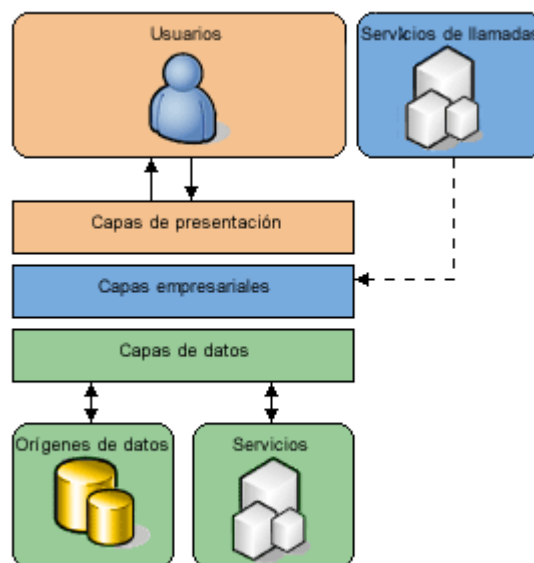


Figura 4. Componentes separados en capas según sus funciones

Una solución distribuida puede que necesite abarcar varias organizaciones o niveles físicos, en cuyo caso tendrá sus propias directivas en relación a la

seguridad, administración operativa y comunicaciones de la aplicación. Unidas, estas directivas definen reglas para el entorno en el que se implementa la aplicación y la forma en que los niveles del servicio o aplicación se comunican. Las directivas abarcan toda la aplicación y la forma en que se implementan afecta a las decisiones sobre el diseño en cada nivel. También tienen un impacto entre sí (por ejemplo, la directiva de seguridad determina algunas de las reglas en la directiva de comunicación y viceversa).

2.1.7.1 Diseño de capas de presentación

La capa de presentación contiene los componentes necesarios para habilitar la interacción del usuario con la aplicación. La capas de presentación más simples contienen componentes de interfaz, como formularios de Windows Forms o formularios Web de ASP.NET. Las interacciones más complejas conllevan el diseño de componentes de proceso de usuario que permiten organizar los elementos de la interfaz y controlar la interacción con el usuario. Los componentes de proceso de usuario resultan especialmente útiles cuando la interacción del usuario sigue una serie de pasos predecibles, como al utilizar un asistente para realizar una tarea determinada. En la figura 5 se muestran los tipos de componentes presentes en la capa de presentación.

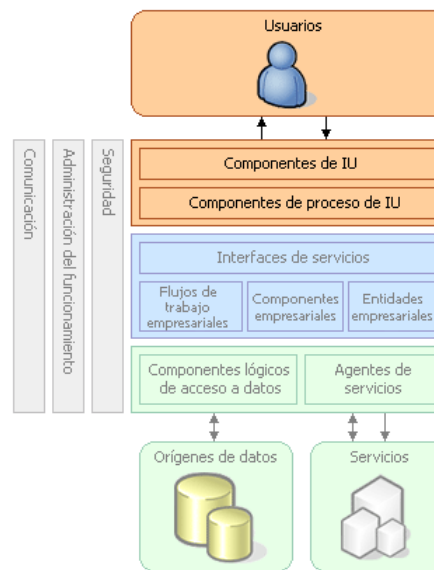


Figura 5. Capa de presentación

Componentes de interfaz de usuario (IU). La mayor parte de las soluciones necesitan ofrecer al usuario un modo de interactuar con la aplicación. En el caso de la Extranet, un sitio Web permite al cliente ver proyectos y realizar propuestas de proyectos, y a los participantes de un proyecto escribir las actividades, bitácoras de los proyectos de los clientes que han contratado los servicios de la empresa. Las interfaces de usuario se implementan utilizando páginas Microsoft ASP.NET, controles u otro tipo de tecnología que permita procesar y dar formato a los datos de los usuarios, así como adquirir y validar los datos entrantes procedentes de éstos.

Componentes de proceso de usuario. En un gran número de casos, la interacción del usuario con el sistema se realiza de acuerdo a un proceso

predecible. Por ejemplo, en la aplicación extranet, podríamos implementar un procedimiento que permita ver los datos del proyecto. De este modo, el usuario puede seleccionar un proyecto de una lista de proyectos en desarrollo y, a continuación, elegir un plan de trabajo de los proyectos seleccionados para ver los detalles correspondientes. Del mismo modo, cuando el usuario registra un proyecto, la interacción sigue un proceso predecible de recolección de datos por parte del usuario, por el cual éste en primer lugar proporciona los detalles de las actividades que desea implementar, a continuación los usuarios que van a participar en esas actividades y, por último, la bitácora. Para facilitar la sincronización y organización de las interacciones con el usuario, resulta útil utilizar componentes de proceso de usuario individuales. De este modo, el flujo del proceso y la lógica de administración de estado no se incluyen en el código de los elementos de la interfaz de usuario, por lo que varias interfaces podrán utilizar el mismo "motor" de interacción básica.

2.1.7.2 Diseño de capas empresariales

La parte más importante de la aplicación es la funcionalidad que proporciona. Una aplicación realiza un proceso empresarial que consta de una o varias tareas.

La parte principal de la lógica empresarial se suele denominar lógica de *dominio*. Los componentes empresariales también pueden realizar solicitudes

de servicios externos, en cuyo caso tal vez sea preciso implementar agentes de servicios para administrar la conversación requerida para la tarea empresarial específica realizada por cada uno de los servicios que necesita utilizar.

En la figura 6 se muestran las capas empresariales de una aplicación.

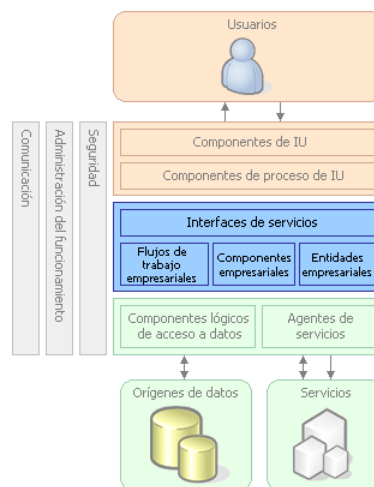


Figura 6. Capas de componentes empresariales

Flujos de trabajo empresariales. Una vez que el proceso de usuario ha recopilado los datos necesarios, éstos se pueden utilizar para realizar un proceso empresarial. Gran parte de los procesos empresariales conllevan la realización de varios pasos, los cuales se deben organizar y llevar a acabo en un orden determinado. Por ejemplo, el sistema empresarial necesita calcular el valor total, el porcentaje de cumplimiento de las actividades, validar ésta información con el valor total del proyecto, procesar el pago de la misma y preparar el envío del avance del proyecto de acuerdo al porcentaje de las actividades cumplidas. El tiempo que este proceso puede tardar en

completarse es indeterminado, por lo que sería preciso administrar las tareas necesarias, así como los datos requeridos para llevarlas a cabo. Los flujos de trabajo empresariales definen y coordinan los procesos empresariales de varios pasos de ejecución larga.

Componentes empresariales. Independientemente de si el proceso empresarial consta de un único paso o de un flujo de trabajo organizado, la aplicación requerirá del uso de componentes que implementen reglas empresariales y realicen tareas empresariales. Por ejemplo, en la extranet, se deberá implementar una funcionalidad que calcule el tiempo total de desarrollo del proyecto y agregue el tiempo adicional correspondiente. Los componentes empresariales implementan la lógica empresarial de la aplicación.

2.1.7.3 Diseño de capas de datos

Casi todas las aplicaciones y servicios necesitan almacenar y obtener acceso a un determinado tipo de datos.

Al trabajar con datos se debe determinar:

- El almacén de datos que se utiliza.

- El diseño de los componentes utilizados para obtener acceso al almacén de datos.
- El formato de los datos pasados entre componentes y el modelo de programación necesario para ello.

La aplicación o servicio puede disponer de uno o varios orígenes de datos, los cuales pueden ser de tipos diferentes. La lógica utilizada para obtener acceso a los datos de un origen de datos se encapsulará en *componentes lógicos de acceso a datos* que proporcionan los métodos necesarios para la consulta y actualización de datos. Los datos con los que la lógica de la aplicación debe trabajar están relacionados con *entidades* del mundo empresarial que forman parte de la empresa.

En la figura 7 se muestra cómo la capa de datos lógicos de una aplicación consta de uno o varios almacenes de datos y describe una capa de componentes lógicos de acceso a datos utilizados para recuperar y manipular los datos en dichos almacenes.

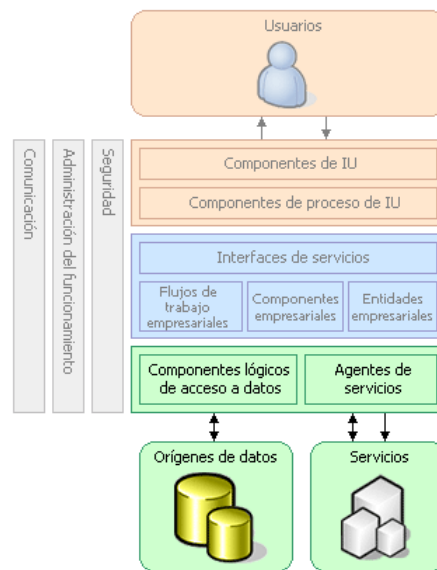


Figura 7. Componentes de datos

Componentes lógicos de acceso a datos. La mayoría de las aplicaciones y servicios necesitan obtener acceso a un almacén de datos en un momento determinado del proceso empresarial. Por ejemplo, la aplicación necesita recuperar los datos de los proyectos de una base de datos para mostrar al usuario los detalles de los mismos, así como insertar dicha información en la base de datos cuando un usuario inscribe una propuesta de proyecto. Por tanto, es razonable abstraer la lógica necesaria para obtener acceso a los datos en una capa independiente de componentes lógicos de acceso a datos, ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.

Agentes de servicios. Cuando un componente empresarial requiere el uso de la funcionalidad proporcionada por un servicio externo, tal vez sea necesario hacer uso de código para administrar la semántica de la comunicación con dicho servicio.

2.1.7.4 Directivas de seguridad y administración

Las directivas de organización definen las reglas que determinan la forma en que se protege una aplicación, el modo en que se administra, así como la forma en que los distintos componentes de una aplicación se comunican entre sí y con los servicios externos. Las directivas afectan al diseño de cada capa de la aplicación o servicio, tal como se muestra en la figura 8.

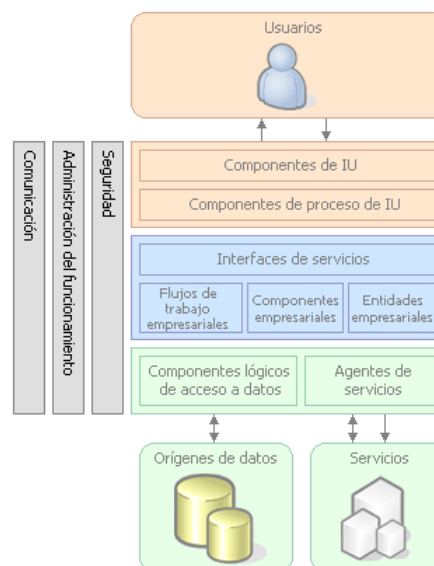


Figura 8. Efecto de las directivas organizativas en el diseño de aplicaciones

Las directivas no sólo se determinan a nivel organizativo, sino que también se pueden establecer dentro de las organizaciones. En ciertos casos, resulta útil recurrir al concepto de *zonas*: todas las aplicaciones, servicios o incluso niveles de la aplicación se encuentran en la misma zona si comparten un subconjunto de las directivas. Por ejemplo, un centro de datos de Internet puede tener un conjunto de directivas distinto al del resto de infraestructura de una compañía y definir una zona especial con unas restricciones de seguridad más estrictas que las de otras partes de la aplicación. De este modo, las aplicaciones y servicios de este centro de datos se encontrarán en una zona diferente a la de las aplicaciones y servicios de la extranet.

El conocimiento de las directivas de cada componente, lo que permite definir las zonas en las que éste se ejecutará, constituye un aspecto fundamental a la hora de determinar dónde se implementarán los componentes.

2.1.7.4.1 Diseño de la directiva de seguridad

La directiva de seguridad se ocupa de la autenticación, autorización, comunicación segura, auditoría y administración de perfiles, tal como muestra la figura 9.

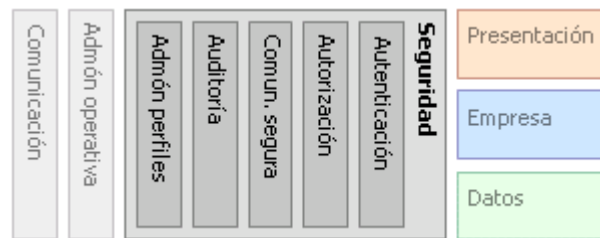


Figura 9. Aspectos de la directiva de seguridad

Autenticación. La autenticación se define como identificación segura, que básicamente quiere decir que dispone de un mecanismo para identificar con seguridad a los usuarios que se adecua a los requisitos de seguridad de su aplicación. La autenticación se debe implementar en la capa de la interfaz de usuario para proporcionar funciones de autorización, auditoria y personalización. Esto generalmente requiere que el usuario escriba sus credenciales (como por ejemplo, nombre y contraseña) para demostrar su identidad.

Autorización. En ciertos casos, la aplicación deberá realizar comprobaciones de autorización. Por ejemplo, considere el siguiente conjunto de condiciones: "Permitir que se realice una actividad si el usuario se encuentra en la función de Administrador. Estos tipos de comprobaciones de autorización se realizan mejor en el código de la aplicación como comprobaciones mediante programación y requieren un desarrollo considerable para separarlas como reglas puras. En otros escenarios más sencillos, se puede implementar la

lógica de la autorización de forma declarativa mediante el uso de atributos o parámetros de configuración.

Comunicación segura. Además de autenticar usuarios y autorizar solicitudes, debe asegurarse de que la comunicación entre los niveles de la aplicación es segura a fin de evitar ataques en los que los datos se "sondeen" o alteren mientras se transmiten o se almacenan en una cola. Las comunicaciones seguras implican proteger las transferencias de datos entre componentes y servicios remotos.

Auditoria. En muchos casos, se necesitará implementar la funcionalidad de auditoria para realizar el seguimiento del usuario y la actividad empresarial en la aplicación por motivos de seguridad. Para realizar la auditoria de las actividades empresariales, se necesita una ubicación de almacenamiento segura; de hecho, la auditoria se puede considerar como un "registro seguro". El mecanismo de auditoria puede utilizar la firma de documentos, la autenticación de plataforma y la seguridad de acceso al código para asegurarse de que no haya código malintencionado que registre entradas falsas.

Administración de perfiles. Los perfiles de usuario consisten en información sobre el usuario que puede utilizar la aplicación para personalizar su comportamiento.

2.1.7.4.2 Diseño de la directiva de administración operativa

La directiva de administración operativa se ocupa de la ejecución constante y diaria de la aplicación y abarca aspectos como la administración de excepciones, la supervisión, la supervisión empresarial, los metadatos, la configuración y la ubicación del servicio, tal como se muestra en la figura 10.



Figura 10. Aspectos de la directiva de administración operativa

Administración de excepciones. La administración de excepciones incluye la detección y generación de excepciones, el diseño de éstas, el flujo de información de las mismas y la publicación de información de las excepciones a diversos usuarios.

Todas las aplicaciones deben implementar algún tipo de control de las excepciones para detectar errores en tiempo de ejecución. Las excepciones se

deben detectar y resolver si es posible. Si no se puede resolver un estado de error, la aplicación deberá mostrar un mensaje descriptivo para el usuario y proporcionar algún medio para el registro o publicación de la información de la excepción para la depuración.

2.1.7.4.3 Diseño de la directiva de comunicaciones

La directiva de comunicaciones define la forma en que los componentes de la aplicación se comunicarán entre sí. Esta directiva trata cuestiones como la sincronización de la comunicación, el formato y el protocolo, tal como se muestra en la figura 11.



Figura 11. Aspectos de la directiva de comunicaciones

La relación de cada una de las capas de esta arquitectura, aplicadas a la aplicación Extranet se describe en la siguiente tabla:

CAPA DE PRESENTACIÓN	CAPA DE EMPRESARIAL	CAPA DE DATOS
- Navegador Web con soporte de CSS - WebForms .aspx	se almacena los componentes de datos y lógica, entre otros.	Motor de datos SQL Server, componente Motor.

Tabla 4. Capas de la Arquitectura de tres capas aplicadas a la Extranet

Entre las ventajas que presenta esta arquitectura se cuentan:

- Las llamadas de la interfaz del usuario en la estación de trabajo al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación sólo necesita transferir parámetros a la capa intermedia.
- La interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- El código de la capa empresarial puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. Esto puede reducir los esfuerzos de desarrollo, mantenimiento y costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes. Separando la aplicación de la base de datos, hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas.

- Separando la interfaz del usuario de la aplicación, libera de gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones.

2.2 FUNDAMENTACION TEORICA ESPECÍFICA

TECNOLOGÍA SOFTWARE EMPLEADA

2.2.1 Internet Information Server (IIS)

Es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un computador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET.

2.2.2 Active server pages .NET (aspx)⁶

ASP.NET es un marco de trabajo de programación generado en Common Language Runtime que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- **Mejor rendimiento.** ASP.NET es un código de Common Language Runtime compilado que se ejecuta en el servidor. A diferencia de sus predecesores, ASP.NET puede aprovechar las ventajas del enlace anticipado, la compilación just-in-time, la optimización nativa y los servicios de caché desde el primer momento. Esto supone un incremento espectacular del rendimiento antes de siquiera escribir una línea de código.
- **Compatibilidad con herramientas de primer nivel.** El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (Integrated Development Environment, IDE) de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y

⁶ Tomado del libro electrónico, Una introducción a asp.net.

colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta.

- **Eficacia y flexibilidad.** Debido a que ASP.NET se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de .NET Framework, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme. ASP.NET es también independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación o dividir la aplicación en varios lenguajes. Además, la interoperabilidad de Common Language Runtime garantiza que la inversión existente en programación basada en COM se conserva al migrar a ASP.NET.
- **Simplicidad.** ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios. Por ejemplo, el marco de trabajo de página de ASP.NET permite generar interfaces de usuario, que separan claramente la lógica de aplicación del código de presentación, y controlar eventos en un sencillo modelo de

procesamiento de formularios de tipo Visual Basic. Además, Common Language Runtime simplifica la programación, con servicios de código administrado como el recuento de referencia automático y el recolector de elementos no utilizados.

- **Facilidad de uso.** ASP.NET emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web. Debido a que la información de configuración se almacena como texto sin formato, se puede aplicar la nueva configuración sin la ayuda de herramientas de administración local. Esta filosofía de "administración local cero" se extiende asimismo a la implementación de las aplicaciones ASP.NET Framework. Una aplicación ASP.NET Framework se implementa en un servidor sencillamente mediante la copia de los archivos necesarios al servidor. No se requiere el reinicio del servidor, ni siquiera para implementar o reemplazar el código compilado en ejecución.
- **Escalabilidad y disponibilidad.** ASP.NET se ha diseñado teniendo en cuenta la escalabilidad, con características diseñadas específicamente a medida, con el fin de mejorar el rendimiento en

entornos agrupados y de múltiples procesadores. Además, el motor de tiempo de ejecución de ASP.NET controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.

- **Posibilidad de personalización y extensibilidad.** ASP.NET presenta una arquitectura bien diseñada que permite a los programadores insertar su código en el nivel adecuado. De hecho, es posible extender o reemplazar cualquier subcomponente del motor de tiempo de ejecución de ASP.NET con su propio componente escrito personalizado. La implementación de la autenticación personalizada o de los servicios de estado nunca ha sido más fácil.
- **Seguridad.** Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.

2.2.3 SQL Server

SQL Server es un Sistema de Gestión de Bases de Datos Relacionales (SGBDR), desarrollado por Microsoft, que permite, como su propio nombre indica, la gestión de un entorno de bases de datos relacional. SQL Server abarca, tanto el área de diseño, como la de administración, proporcionando un interfaz bastante amigable con el usuario.

¿Por qué se llama SQL Server? Pues bien, se llama *SQL* porque utiliza este lenguaje para la definición y manejo de los datos, y se llama *Server* porque dispone de una parte servidora que se encarga de atender a los procesos clientes, que son los que realizan las peticiones a éste; es decir, sigue una arquitectura cliente/servidor.

SQL Server utiliza una extensión al SQL estándar, que se denomina *Transact SQL*. Esto quiere decir que soporta el SQL de ANSI, pero además se le han añadido ciertas funciones adicionales, no contempladas en el estándar, y que son específicas para este producto, es decir, si ejecutamos una sentencia del conjunto adicional (Transact SQL) en otro SGBRD, éste no la entendería.

El Transact SQL, soporta la definición, modificación y eliminación de bases de datos, tablas, atributos, índices, etc., es decir, el lenguaje de definición de datos

(DDL), así como la consulta, actualización y borrado de tuplas de tablas, es decir, el lenguaje de manipulación de datos (DML).

2.2.4 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 de software. Según su definición, UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos del 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 centra en la representación conceptual y física de un sistema, y que indican cómo crear y leer modelos bien elaborados. Sin embargo, no dice qué modelos crear ni cuándo se deberían crear, esta es la tarea del proceso de desarrollo del software:

- Es un lenguaje para visualizar.
- Es un lenguaje para especificar.
- Es un lenguaje para construir.
- Es un lenguaje para documentar.

Los diagramas básicos de UML son los diagramas de clases, diagramas de objetos, diagramas de casos de usos, diagramas de secuencia, diagramas de colaboración, diagramas de estados, diagramas de actividades, diagramas de componente y diagramas de despliegue.

UML se utilizó como lenguaje de modelo en desarrollo de este proyecto, permitiendo especificar, visualizar, construir y documentar el sistema. Los diagramas de UML empleados para el modelado de este proyecto fueron: diagramas de casos de uso, diagramas de actividades, y diagramas de clases, cuyas características se describen en la tabla 5.

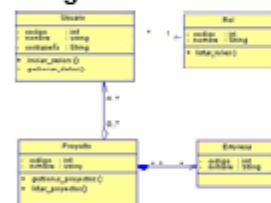
Diagrama UML	Descripción ³¹
Diagrama de Casos de Uso 	Diagrama que muestra las relaciones existentes entre actores y casos de uso dentro de un sistema. Un actor es una abstracción de las entidades externas a un sistema, subsistemas o clases que interactúan con el sistema. Un caso de uso es una unidad coherente de funcionalidad, es una interacción típica entre un usuario y un sistema.
Diagrama de Clases 	Presentación gráfica de la vista estática, que muestra una colección de elementos del modelo y sus relaciones.
Diagrama de Actividades 	Diagrama que sirve fundamentalmente para modelar el flujo de control entre actividades. Los diagramas de actividades facilitan la descripción del comportamiento que tienen una cantidad de procesos propios de un sistema.

Tabla 5. Diagramas UML usados en el proyecto

2.2.5 CSS (Cascading Style Sheet)

Las hojas de estilo en cascada (CSS), son un lenguaje formal usado para definir la presentación de un documento cuya estructura ha sido definida como HTML, XHTML ó XML. El W3C ideó esta tecnología con el fin de separar la estructura de una página de su apariencia, dejando la definición de fuentes,

colores y otros aspectos visuales a CSS y reintegrando HTML a su original propósito estructural.

CSS se basa en la definición y uso de reglas que definen las propiedades a aplicar a las etiquetas de lenguaje con el que se estructura la página Web. La mayoría de estas propiedades tienen que ver con aspectos visuales, pero también hay algunas que tocan aspectos de distribución de las páginas. Las reglas pueden aplicarse directamente como atributos de las etiquetas HTML o creando archivos CSS independientes que definan los estilos o reglas a aplicar, y que se importen en las páginas Web.

Con el uso de hojas de estilo se favorece la separación de capas en las aplicaciones Web, y por lo tanto facilita su mantenimiento. Sin embargo, como todo lenguaje y con mayor razón con toda tecnología Web, CSS está sujeto al problema de variación y versión entre navegadores, a la buena definición de etiquetas o elementos HTML y XML, y a la continua agregación de nuevas funciones CSS que generan especificaciones o versiones distintas de este lenguaje.

MARCO METODOLÓGICO

Se presenta la fundamentación teórica necesaria para la comprensión de la elección de la metodología de software a utilizar en el desarrollo del proyecto.

La temática presentada no pretende abarcar detalles específicos de cada tópico tratado, sino realizar una conceptualización general de las metodologías de software.

3.1 CICLO DE VIDA DEL SOFTWARE

Un ciclo de vida del software es un modelo que describe todo el proceso y las actividades que conlleva la creación de un producto software. “La función principal de un modelo de ciclo de vida es establecer el orden en que se especifica, se realizan los prototipos, se diseña, se implementa, revisa, prueba y se realizan otras actividades en un proyecto. Establecen los criterios que se utilizan para determinar el paso de una actividad a otra”⁷

Los estilos de desarrollo del producto varían entre las diferentes clases de proyectos, se realizan diferentes clases de tareas y distintos órdenes de las mismas. La elección equivocada de un ciclo de vida puede ocasionar emisión de tareas o una secuenciación inapropiada de las mismas, lo cual va en contra de la planificación y eficiencia del proyecto. Una buena elección, por el contrario, es una garantía de que el esfuerzo se utiliza eficientemente.

Los estilos de desarrollo del producto varían entre las diferentes clases de proyectos, se realizan diferentes clases de tareas y distintos órdenes de las mismas. La elección equivocada de un ciclo de vida puede ocasionar emisión

⁷ McConell, Steve. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. España: Mc Graw Hill, 1997.

de tareas o una secuenciación inapropiada de las mismas, lo cual va en contra de la planificación y eficiencia del proyecto. Una buena elección, por el contrario, es una garantía de que el esfuerzo se utiliza eficientemente.

Existen muchos modelos con diferentes enfoques, cada uno con sus ventajas e inconvenientes.

3.1.1 Tipos ciclos de vida del desarrollo software⁸

- **CASCADAS** (Cascada pura, cascada con fases solapadas, cascada con subproyectos y cascada con reducción con riesgos).
- **DRA** (DESARROLLO RÁPIDO DE APLICACIONES).
- **PROTOTIPADO** (Prototipado simple y evolutivo),
- **ESPIRAL**
- **PROCESO UNIFICADO**

3.1.2 Selección del ciclo de vida

Distintos proyectos tienen necesidades diferentes, incluso si todos necesitan ser desarrollados lo más rápido posible. No existe «un modelo de ciclo de vida de desarrollo rápido», debido a que el modelo más efectivo depende del

⁸ Tomado del documento CICLOS DE VIDA DE DESARROLLO SOFTWARE-Parte II. Grupo de Investigación de Sistemas y Tecnología de la Información. Profesor Luis Carlos Gómez Sarmiento.

contexto en el que se utilice. Determinados modelos de ciclo de vida son considerados más rápidos que otros, pero cada uno de ellos será más rápido en determinadas situaciones, y más lento en otras.

La siguiente tabla⁹, proporciona una guía para decidir qué ciclo de vida utilizar. La efectividad de cualquier modelo dependerá de cómo se implemente. Por otro lado, si se conoce que el modelo es débil en un área particular, se puede controlar este inconveniente al principio de su planificación y compensarlo creando una mezcla entre uno o más modelos descritos.

Siguiendo la guía de la Tabla 1. Guía para la selección del ciclo de vida a utilizar en un proyecto y analizando las características presentadas por el sistema de información Extranet, se eligió el modelo prototipado evolutivo el cual se acomoda o acopla a los siguientes aspectos:

- Algunos de los requerimientos pueden estar sujetos a cambios continuos, por ejemplo los cambios en los formatos de propuestas de proyectos, reportes, modificación de información básica de clientes y personal de la empresa, entre otros.

⁹ Se refiere a los defectos que pueden aparecer en un sistema desarrollado con un molde de ciclo de vida cuando empieza a funcionar.

- Permite a los clientes, y en especial, al personal de la empresa, mantener un constante seguimiento del la Extranet durante su desarrollo.

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

Tabla 6. Selección del ciclo de vida

- Existe una gran demanda de velocidad de desarrollo y visibilidad de avance por parte del usuario.

El prototipado evolutivo (desarrollo incremental con mejora iterativa) es un modelo de ciclo de vida donde el sistema se desarrolla en incrementos, de forma que puede modificarse de manera inmediata en respuesta a la realimentación del cliente y del usuario final. El prototipado es un enfoque de desarrollo donde se desarrolla primero los aspectos más visibles del sistema, se presenta la parte del sistema al cliente y luego desarrolla el resto del sistema a partir de esa parte.

Se centra en la idea de ayudar a comprender los requisitos que plantea el usuario. Esta versión temprana de lo que será el producto, con una funcionalidad reducida, en principio, podrá incrementarse paulatinamente a través de refinamientos sucesivos de las especificaciones del sistema, evolucionando hasta llegar al sistema final.



Figura 12. Modelo de prototipado evolutivo.

Fuente: Sotomayor Humberto. Método y técnicas para un desarrollo de software de calidad.

Al aplicar esta metodología en el desarrollo del software fue necesario llevar a cabo algunas modificaciones al diseño inicial conforme avanzaba el proyecto, comenzando con un modelo relativamente sencillo, implementando los requisitos más importantes e ir agregando los demás módulos hasta obtener el sistema requerido.

Cabe resaltar que el mayor problema de esta metodología es que al principio del proyecto no se conoce cuánto se tardará en crear un producto aceptable, por lo tanto no se conoce cuántas iteraciones serán necesarias. Estos inconvenientes se cubrieron realizando un diseño y prototipo inicial robustos, estableciendo así un periodo de tiempo aceptable en la entrega del producto, el cual debería cumplir con los objetivos específicos del proyecto, para lo cual se estableció el desarrollo de tres prototipos.

PARTE II: DESARROLLO DEL SISTEMA

Esta segunda parte del documento presenta un seguimiento detallado del trabajo que se realizó para la creación de la herramienta software, de acuerdo con la metodología sugerida.

La labor desarrollada efectuada, abarca las cuatro fases planteadas por la metodología, siendo éstas: la fase de inicio, elaboración, construcción y transición.

Los capítulos que describen cada una de las fases tratadas se han organizado en el siguiente esquema. Inicialmente, se realiza una planificación de las iteraciones y actividades que conformarán la fase y seguidamente se irán mostrando los resultados obtenidos en los diferentes flujos de trabajo (requisitos, análisis, diseño, implementación y pruebas). Por último, se hace una evaluación de los productos generados en cada fase.



4

FASE DE INICIO

Durante esta fase se definió el alcance del sistema propuesto y se limitó el área que se cubriría en el proyecto. Por otra parte, se plantearon los objetivos del sistema, que optimizarían los requisitos solicitados por el usuario.

Se pretendió en esta fase, dar el esbozo de una arquitectura candidata que pudiera ser el ámbito del sistema. Por último se analizaron los costos y cronograma para el proyecto. Cerrando el capítulo, se ilustran los productos que se obtienen al finalizar la fase de inicio.

4.1 ALCANCE DEL PROYECTO

La herramienta de software implementada propenderá por la prestación del servicio de consulta y toma de decisiones de la más alta calidad y la proyección de una imagen corporativa organizada que cuenta con actividades controladas. Esta herramienta, asiste los procesos realizados en el desarrollo de un proyecto.

Entre los principales alcances se distinguen:

a) Clientes potenciales

- Junta Directiva
- Trabajadores Profesionales de cualquier departamento de Trabajo de ésta.
- Usuarios finales.

b) Sistema de Información, para lo cual se destaca:

- Formularios diseñados para consultar y desplegar la información almacenada en la Base de datos.

- Control de accesos correspondientes a cada módulo mediante login y contraseña.
- Validación en la entrada de información.
- Automatización del proceso de la toma de Decisiones generando reportes de la información asociada con cada uno de los módulos consultados.
- Interfaz personalizada.
- Portabilidad.
- Seguridad del Sistema.

El proyecto está definido como un instrumento que proporciona:

- Información general de las actividades administrativas y operativas. Presentada a través de informes que surgen del seguimiento continuo de éstas.
- Información específica sobre el seguimiento de proyectos, a nivel de: plan de trabajo, personal de trabajo, seguimiento, soportes, documentos, bitácoras.

La herramienta de software implementada se ve en el futuro, como una herramienta apoyada en la tecnología informática y en los Sistemas de Información distribuidos, que asiste los procesos de mejoramiento, innovación y modernización, en las consultas de la información y la toma de decisiones en

las actividades de la empresa, cumpliendo con cada uno de los requisitos definidas en las fases de análisis correspondientes.

Restricciones

- El sistema sólo permite la inclusión de datos a los procesos de los módulos a cargo del usuario que se autenticó en la sesión activa.

4.2 COMPRENDER EL CONTEXTO DEL SISTEMA

Para comprender el contexto del sistema de Información Extranet para la empresa Quirón Software, se ha hecho uso de un modelo del negocio por considerarse que este nos ayude a comprender los procesos de negocio de la empresa.

4.2.1 Modelo del Negocio

En las diferentes actividades que se llevan a cabo en la empresa Quirón Software, encontramos los siguientes actores principales: Los empleados, los clientes y los directivos.

A continuación se detallan los diferentes procesos que se realizan. Estos se dan origen cuando un usuario (“cliente”), contrata los servicios de la empresa

ya sea para el desarrollo de un proyecto, soporte o consultoría, por primera vez. En este momento el usuario encargado (“administrador”) del sistema, ingresa la información básica correspondiente a la empresa o usuario cliente; si no es la primera vez que el usuario cliente contrata los servicios de la empresa, se puede actualizar la información básica de éste.

Una vez celebrado el contrato entre las dos partes, se programan el plan de trabajo y las actividades a desarrollar en el proyecto. Además, de asignarse el personal de trabajo. El ingreso del personal de trabajo, tiene el mismo procedimiento que el usuario cliente, si es un empleado nuevo, el administrador registrará su hoja de vida en el sistema, de lo contrario, se podrá actualizar la hoja de vida cuando así se requiera.

Durante el desarrollo del proyecto, se generan documentos o conclusiones de las reuniones realizadas entre las dos partes, y se lleva una agenda, donde se registran la programación de reuniones, entregas de prototipos, entre otros.

Adicionalmente, el usuario cliente, podrá solicitar soportes de mantenimiento durante la fase de pruebas, donde se le asignará un usuario empleado quien será el encargado de darle el soporte solicitado al cliente; el usuario empleado podrá notificar las dificultades que haya tenido durante éste, o manifestar el cumplimiento del soporte; si es necesario, se genera un acta donde se especifica el estado del proyecto.

4.3 ANÁLISIS DE REQUERIMIENTOS

El análisis de requerimientos de software permite especificar la función, los datos y el rendimiento del software, igualmente establece las restricciones que éste debe cumplir.

Para llevar a cabo el análisis de requerimientos de software, es importante revisar el plan de proyecto de software y revisar el ámbito del software; la meta que se busca es describir lo que el cliente requiere, establecer una base para el diseño del software, y definir un conjunto de requisitos que se puedan verificar una vez que el desarrollo culmine.

En el análisis, definición y especificación de los requerimientos del prototipo, se establecieron sus funcionalidades, restricciones, suposiciones y dependencias a las que está sujeto su desarrollo y funcionamiento. Esto permitió definir el producto final que se desea obtener.

Para el desarrollo de esta primera etapa se emplearon los diagramas de casos de uso del lenguaje unificado de modelado (UML) como herramienta esencial para la captura de los requerimientos del sistema. El diagrama de casos de uso representado en la figura 13, describe de manera general las funcionalidades características del sistema y facilita la descripción de cómo se

usará. En el anexo A, se encuentra la documentación de los casos de uso que se describen en la figura mencionada.

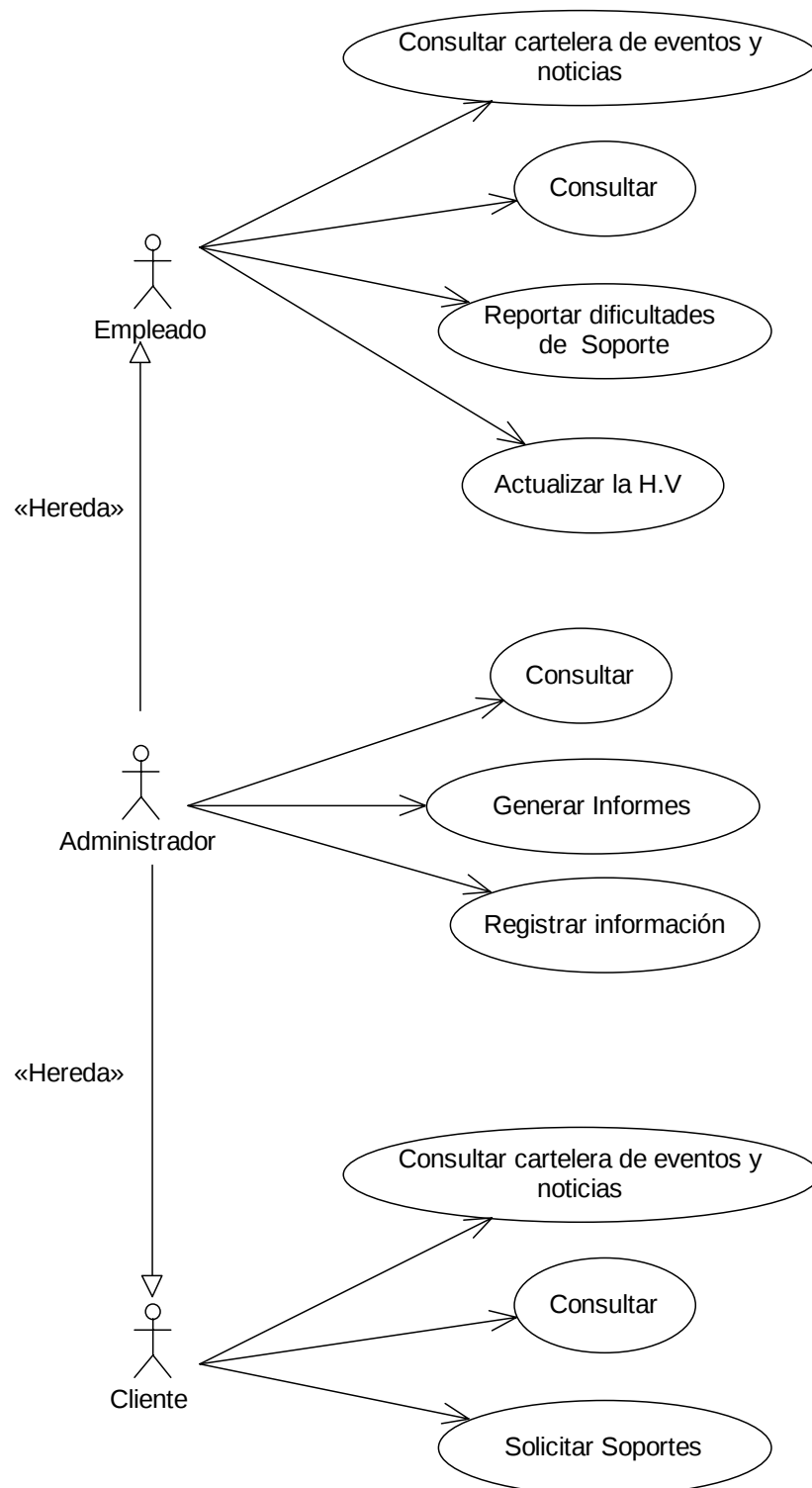


Figura 13. Diagrama relación de Casos de Uso entre actores del Sistema

En esta figura se identifican 3 actores, usuarios que desempeñan un rol o papel con respecto al sistema, en la tabla 5 se describen cada uno de los actores que participarán en el proyecto.

ACTOR	DESCRIPCIÓN	RESPONSABILIDADES (PAPEL QUE JUEGA)	NECESIDADES (PARA QUE UTILIZA EL SISTEMA)
Empleado	Representa a la(s) persona(s) que trabajan en cualquiera de los proyectos ofrecidos por la empresa.	Suministrar información personal. Actualizar su información general cuando ésta haya cambiado. Cumplir con los compromisos laborales que se derivan al estar vinculado a un proyecto en las diferentes actividades.	Registrarse en el sistema. Modificar algunos datos personales en el sistema. Consultar su hoja de vida, actividades del proyecto, soportes a realizar, así como documentos generados durante el desarrollo del proyecto.
Cliente	Representan a las diferentes personas que se encuentran vinculadas con la empresa, para el desarrollo de un proyecto.	Mantener actualizada información básica correspondiente a la empresa. Registrar solicitudes de soportes de mantenimiento.	Consultar las actividades a realizar en el proyecto. Consultar los documentos generados en el desarrollo del proyecto. Revisar actas.
Administrador	Representa a lo(s) socios que reuniendo los requisitos contemplados en el reglamento interno de la empresa ha sido elegido para ocupar ese cargo.	Aprobar el ingreso de usuarios al sistema.	Consultar información de los proyectos a cargo de la empresa, para la toma de decisiones.

Tabla 7. Actores del Sistema

4.3.1 Casos de Uso del Negocio

Actor Empleado

Consultar. El empleado puede consultar su hoja de vida personal, proyectos, actividades, solicitudes de soportes, documentos, bitácoras, entre otros.

Actualizar HV. El empleado podrá actualizar su hoja de vida en el caso de que algún dato personal, académico y/o profesional haya cambiado.

Reportar Dificultades a Soportes. Los empleados podrán dejar registros de las dificultades que se presentaron al momento de dar Soporte a un cliente en un proyecto específico.

Consultar Eventos y Noticias. El empleado podrá consultar los eventos y/o noticias de interés general.

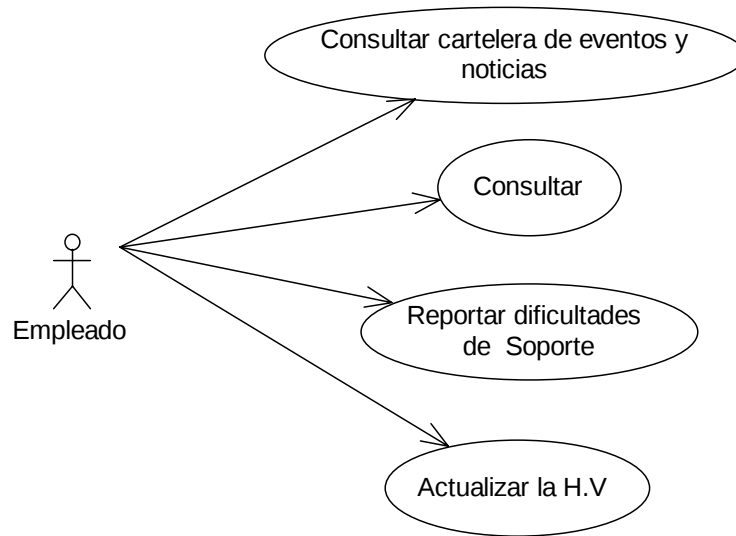


Figura 14. Diagrama casos de uso actor Empleado

Actor Cliente

Consultar. El Cliente puede consultar proyectos, actividades, bitácoras, personal vinculado al proyecto, entre otros.

Solicitar Soportes. Los clientes podrán solicitar soporte de un proyecto durante el desarrollo de éste.

Consultar cartelera de eventos y noticias. El cliente podrá consultar los eventos y/o noticias de interés general.

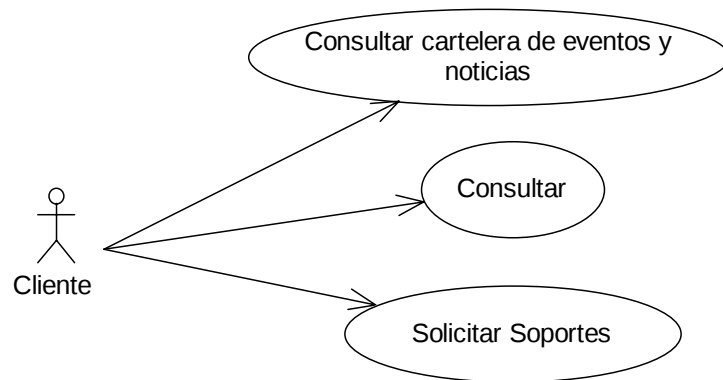


Figura 15. Diagrama de Casos de Uso actor Cliente

Actor Administrador (Director General)

Consultar. El Director puede consultar su hoja de vida personal, proyectos, actividades, solicitudes de soportes, documentos, bitácoras, entre otros.

Generar Informes de Proyectos. El Director puede generar a manera de informe información específica de proyectos, de tal manera que sirva como herramienta para la toma de decisiones.

Registrar Información. El administrador puede registrar información relacionada con clientes, personal de trabajo y proyectos.

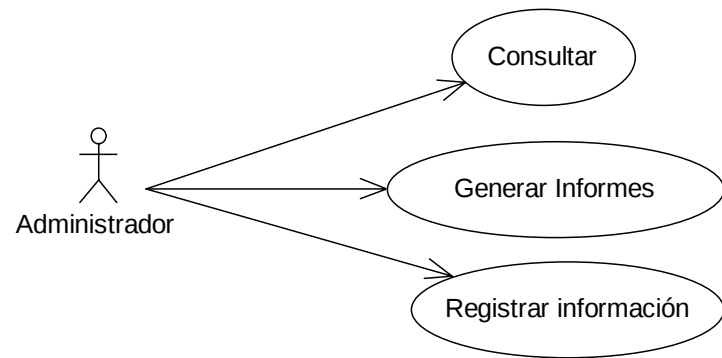


Figura 16. Diagrama de Casos de Uso actor Administrador

Modelo General de Casos de Uso

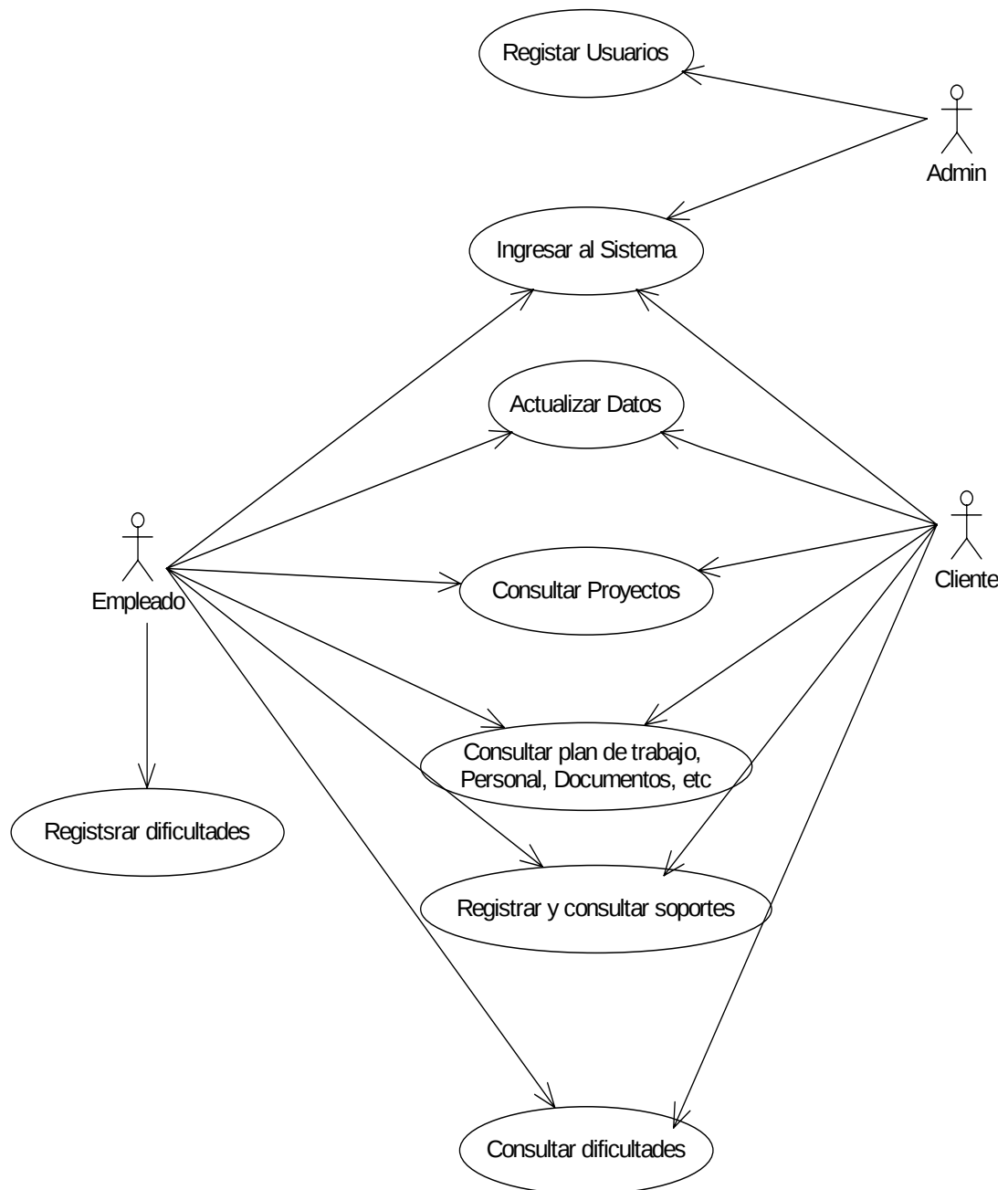


Figura 17. Diagrama de casos de uso General

4.3.2 Priorizar casos de uso

En esta actividad se establecen los casos de usos más relevantes para el sistema. Para esto, se determinó que los casos de uso necesarios son los relacionados con las consultas de proyectos y actividades asignadas en cada una de las fases.

4.3.3 Atributos de la Aplicación

ATRIBUTO	DEFINICION
FIABILIDAD	Se garantiza con las pruebas que se realicen y la corrección de los errores detectados durante la ejecución de estas.
EFICIENCIA	Hace referencia al rendimiento del tiempo de respuesta, está sujeta al tipo de conexión empleada por los usuarios para el acceso a Internet.
INTEGRIDAD	Este atributo se maneja gracias a la integridad referencial que permite Sql Server, manejo de índices, procedimientos almacenados y triggers que garantiza la integridad y consistencia en los datos.
USABILIDAD	Se realizarán manuales de usuario describiendo el funcionamiento y uso del sistema.
MANTENIBILIDAD	Codificación de la aplicación Web se hará mediante clases y objetos en lenguaje .NET, lo que garantiza facilidad en cuanto al mantenimiento del software y además escalabilidad del sistema.
NAVEGABILIDAD	El diseño gráfico de la aplicación sigue los requisitos de navegabilidad sugeridos por los desarrolladores implementando menús que facilitan a los usuarios la navegabilidad por la aplicación.
SEGURIDAD	Se usarán técnicas de encriptación de datos (Hash) para proteger la aplicación de acceso malévolo o accidental.

Tabla 8. Atributos de la Aplicación

4.3.4 Requisitos Adicionales

La infraestructura tecnológica necesaria para el desarrollo del proyecto, es la siguiente:

Hardware

- Equipo Servidor Dell Precision 670 Intel® Xeon™ 3.60 Ghz, Bus frontal balanceado de 800 MHz con hasta un máximo de 16GB de memoria de alta velocidad ECC SDRAM DDR2 de canal dual de 400 Mhz. Máximo desempeño del Procesador Dual y Confiabilidad en una Estación de Trabajo con Chasis Altamente Expansible.
- Equipo cliente Dell 720 Intel® Xeon™ Pentium 4 de 2500 MHz., Memoria RAM de 512 MB, Disco Duro de 80 Gb.

Software

- Microsoft Visual Studio 2005
- Microsoft SQL Server 2000 Developer Edition
- Microsoft Windows XP Professional, 2000 Server o 2003 Server (para el desarrollo)

4.3.5 Requisitos Visuales

Durante el diseño detallado de las interfaces a utilizar debe tenerse en cuenta:

- Procurar una interfaz amigable, sencilla y con orientaciones que facilite la navegación; el paso entre pantallas debe ir acorde al trabajo que el usuario desea realizar y el sistema debe mostrar mensajes de alertas entre las acciones de los usuarios que lo requieran, en especial cuando sus entradas no cumple las reglas de validación de formularios.
- Proporcionar mecanismos que informen al usuario en todo momento del lugar del sitio que se encuentra.
- Manejar colores suaves.
- Racionalizar el uso de imágenes que disminuyan la velocidad de descarga de las páginas.
- Las imágenes, texto y controles gráficos que aparecen en las pantallas deben tener una distribución consistente, y en conjunto deben lograr una apariencia acorde con los objetivos del sitio.

4.3.6 Otros requisitos

- Capacidad máxima para carga de archivos al sistema de 2MB.
- Mecanismos de seguridad que eviten acciones como:
 - o Acceso no autorizado a la base de datos y a recursos almacenados en el servidor.
 - o Pérdida de la integridad de la información en la base de datos y los recursos almacenados en el servidor.
 - o Entrada a páginas y opciones no autorizadas para el usuario en sesión.

4.4 FLUJO DE TRABAJO DE ANÁLISIS

En la fase de inicio, el flujo de análisis es un modelo que se utilizará para definir el 10% de los casos de uso y servirá de guía en el establecimiento de la arquitectura candidata, pero solo en una pequeña parte, ya que es tarea de la fase de elaboración dicha labor.

4.5 ANÁLISIS DE LA ARQUITECTURA CANDIDATA

El desarrollo del sistema de información Extranet para la empresa Quirón software, estará basado en tecnología .NET (ASP.NET Active Server Pages). Esta tecnología, interpretada de alto nivel, y ejecutada en el servidor; permitirá recuperar en formularios, la información que los usuarios soliciten a manera de consulta, así como, registrarla según necesario, con el fin de ser almacenada en la base de datos.

Para operar la base de datos se utilizará el manejador de base de datos SQL SERVER, el cual es utilizado por la empresa. En la tabla 9, se muestran los factores que afectan la arquitectura.

	Tipo de Factor	Descripción
1	Software del sistema	Sistema operativo Microsoft Windows 2003 Server Base de Datos SQL Server 2000 Servidor Web Information Internet Server
2	Capa Intermedia	Lenguaje ASP.NET Javascript Css
3	Necesidades de distribución	Arquitectura 3 capas

Tabla 9. Factores que afectan la arquitectura

Para poder entender un poco mejor la arquitectura candidata se muestra a continuación una descripción de la vista de análisis del modelo de casos de

uso. En esta descripción se muestran modelos que son significativos para la línea base de la arquitectura.

4.6 RIESGOS Y PLANES DE CONTINGENCIA

Los riesgos constituyen uno de los pilares fundamentales en el buen desempeño de un sistema. A continuación, se exponen algunos riesgos que a consideración de los desarrolladores, pondrían en peligro el éxito del proyecto. Es posible plantear un plan de contingencia para responder a algunos de los riesgos que se den como hechos, pero hay otros que se encuentran fuera del alcance de los desarrolladores.

4.6.1 Diseño e Implementación

Riesgos

- No se realiza un riguroso análisis de los sistemas previos, las necesidades actuales y los requerimientos a futuro.
- Definir mediocrementemente la estructura y esquema.

Plan de contingencia

- Evaluar la realización de un análisis serio de las necesidades del negocio.
- Verificar las herramientas de análisis para que sean acordes con los diferentes niveles de requerimientos.

4.6.2 Los Datos

Riesgos

- Ineficiencia y/o errores en el proceso de recuperación y manipulación de datos históricos.
- Poca calidad de los procesos de conversión a esquemas actuales.
- Manejo de ausencias, inconsistencia y/o duplicación de datos.
- Pérdida o no extracción de datos relevantes para el negocio.
- Poca efectividad y eficiencia en el proceso.
- Falla en los procesos de extracción (caída del sistema, inaccesibilidad de los datos, etc.)
- Falta de capacitación a usuarios sobre el manejo de los datos.

Plan de contingencia

- Verificar que las herramientas que se usan en esta fase son las correctas y se utilizan adecuadamente.
- Los cambios a programas que generen información deben ser autorizados y revisados antes de ponerlos en marcha.
- Crear normas y estándares para la definición de los objetos en la Base de datos.

4.6.3 Seguridad

Riesgos

- Falta de documentación sobre políticas, definiciones de responsabilidades, directrices, normas y procedimientos para que se apliquen las medidas de seguridad, los mecanismos de evaluación de riesgos y planes de seguridad.
- Acceso al sistema desde elementos externos sin autorización (aplicaciones, personas, etc.).

Plan de contingencia

- Implantar un sistema de autorización y control de acceso con el fin de restringir la posibilidad de los usuarios para leer, escribir, modificar, crear, o borrar datos importantes. Estos privilegios deben definirse de una manera consistente con las funciones que desempeña cada usuario.
- La información de la Compañía clasificada como confidencial o de uso restringido, deberá ser guardada y transmitida en forma cifrada, utilizando herramientas de encriptado robustas y que hayan sido aprobadas por la Gerencia de Informática.

4.6.4 Los Usuarios

Plan de contingencia

- Capacitar a los usuarios para que la información entregada por los sistemas se interprete correctamente y sea destinada a generar y apoyar acciones.

4.6.5 Técnicos de tecnologías anticuadas

Plan de contingencia

- Tener claro cuales son las necesidades de la empresa en el desarrollo de la aplicación y con base a esto, seleccionar diferentes tecnologías que se adapten a las características del sistema.
- Escoger las tecnologías que se ajusten al presupuesto y a las necesidades del proyecto.
- Analizar que tecnología puede llegar a ser anticuada.
- Cotizar actualizaciones y/o cambios a nuevas tecnologías.

4.6.6 Conocido de fuentes de información poco fiables

Plan de contingencia

- Consultar a los expertos sobre la credibilidad y vigencia que poseen las fuentes de información que se tienen.
- Revisar los procesos que se han construido bajo la información de dichas fuentes y realizar los cambios necesarios para que el sistema no presente problemas en etapas posteriores.
- Recibir asesoría de las directrices del proyecto.

- Buscar diferentes fuentes de información que nos permitan encontrar diversos planteamientos y escoger la adecuada a nuestro trabajo.



5

FASE DE ELABORACIÓN

En la fase de elaboración, el objetivo principal es establecer la línea base de la arquitectura que nos proporcione la guía del sistema en las fases de construcción y transición, como de sus versiones futuras.

También debemos recopilar la mayor parte de los requisitos que no se tomaron en cuenta en la fase de inicio, haciendo que los requisitos funcionales se formulen como casos de usos. Además, continuar la observación y control de los riesgos que aún queden e identificar nuevos riesgos que aparezcan.

5.1 DISEÑO GLOBAL

El proceso se inicia con la configuración y registro de los usuarios y de los proyectos en el sistema, por parte del administrador, posteriormente los usuarios habilitados pueden entrar a usar la herramienta que los apoyará en el desarrollo del proyecto o proyectos a los que fueron asignados; la información necesaria para el desarrollo y la generación de los proyectos finales, será recopilada mediante el uso de formularios y valiéndose de la orientación y documentación que guiará a los usuarios en el proceso.

Una de las áreas en las que se centra esta etapa de elaboración, es en el diseño de datos, para el cual se tomó como referencia el diagrama de clases que se construyó, que ofrece una representación de la vista de diseño estática de un sistema, muestra una colección de elementos declarativos (estáticos) del modelo, como clases, tipos y sus contenidos y relaciones¹⁰, de gran ayuda para implementar el prototipo con programación orientada a objetos.

¹⁰ BOOCH, Grady. RUMBAUCH, James. JACOBSON, Ivar. El lenguaje unificado de modelado manual de referencia. Addison Wesley, 2000. Capítulo 13.

5.1.1 Arquitectura por capas

El Sistema de Información Web, está desarrollado bajo una arquitectura DNA de tres capas, como se explica en los siguientes numerales.

La arquitectura de tres capas segmenta los componentes de una aplicación en tres capas de servicios. Estas capas no se encuentran necesariamente ubicadas en distintas maquinas o computadores de una red, ya que hacen referencia a los niveles lógicos de la aplicación.

A través de la vida de una aplicación, esta arquitectura provee beneficios como reusabilidad, flexibilidad, manejabilidad, mantenimiento, y escalabilidad. Se pueden dividir proyectos largos y complejos en proyectos o módulos más simples para facilitar su desarrollo aprovechando la independencia entre estos.

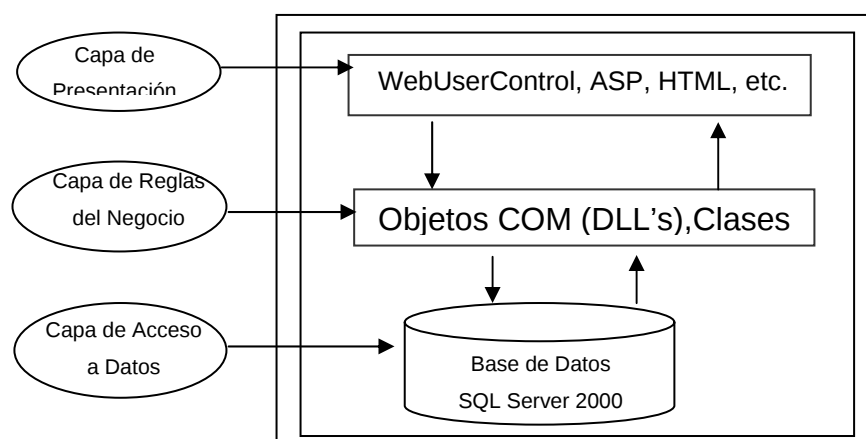


Figura 18. Arquitectura General de la herramienta Extranet

5.1.1.1 Capa de Datos

Esta capa es la encargada de acceder a la base de datos y puede ser accedida desde la capa de reglas del negocio y en ocasiones desde la capa de presentación. Provee una separación de los componentes de acceso a la base de datos de los demás componentes, permitiendo que sea fácil cambiar de manejador de base de datos sin ocasionar cambios traumáticos para el resto de la aplicación.

Desarrollada sobre el motor de Base de Datos Microsoft SQL Server 2000, está compuesta por Tablas, vistas, Funciones, Procedimientos Almacenados y Tareas Programadas o JOB's. Para efectos de conexión con la Base de Datos, se guardan los datos correspondientes a ***Nombre del Servidor de BD, Nombre de la BD, Nombre de Usuario y Clave de Usuario*** en el archivo de configuración Web.config que se encuentra en la capa de presentación, con lo cual se evita el acceso público a esta información.

5.1.1.2 Capa de Reglas del Negocio

En esta capa se resuelven problemas críticos del negocio y se logran grandes ventajas de productividad. Sirve para incrementar la seguridad de la aplicación o asegurar la integridad de los datos si se utilizan varias maquinas, y puede estar ubicada en distintos computadores según se requiera.

Esta capa está compuesta por una serie de Clases, desarrolladas en Visual C#. La función de estos objetos es servir de intermediarios para la comunicación entre las capas de datos y presentación de la aplicación, así como aplicar reglas o políticas específicas del Sistema.

5.1.1.3 Capa de Presentación

Esta capa es la encargada de presentar información al usuario navegante y ocasionalmente permitir que este manipule esta información.

En esta capa se encuentran todos los objetos que constituyen las interfaces gráficas de la solución, tanto para la Administración del Sistema como para los usuarios navegantes. Está compuesta por una serie de usercontrols, páginas html, páginas ASP y archivos de configuración.

A continuación se describe como se diferencian los componentes de cada módulo. Cada módulo funcional del sistema se diferencia en la capa de presentación mediante las carpetas físicas que se describen a continuación. Todos los archivos que se encuentran en el interior de cada carpeta pertenecen al mismo módulo de la carpeta.

5.1.2 Paquetes

Visualizar, especificar, construir y documentar grandes sistemas conlleva manejar una cantidad de clases, interfaces, componentes, nodos, diagramas y otros elementos que puede ser muy elevada. A medida que el sistema va creciendo hasta alcanzar un gran tamaño, se hace necesario organizar estos elementos en bloques mayores. En UML el paquete es un mecanismo de propósito general para organizar elementos de modelado en grupos.

La visibilidad de los elementos dentro de un paquete se puede controlar para que algunos elementos sean visibles fuera del paquete mientras que otros permanezcan ocultos. Los paquetes también se pueden emplear para presentar las diferentes vistas de la arquitectura de un sistema. Todos los grandes sistemas se jerarquizan en niveles para facilitar su comprensión. En la figura 20 se muestra los paquetes utilizados en la aplicación

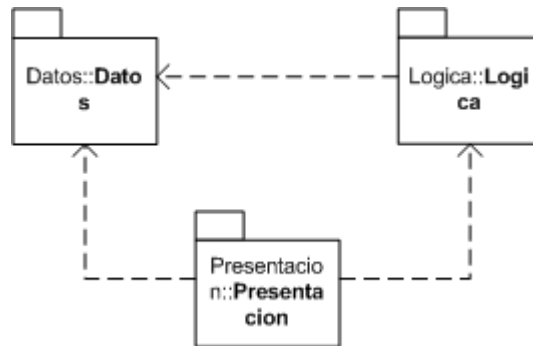


Figura 19. Diagrama de paquetes utilizados en la herramienta

5.1.3 Diagrama de clases

El diagrama de clases que se muestra en la figura 20, abarca una vista general del sistema, asociado a los datos. El diagrama lo conforman 11 clases que se describen brevemente a continuación.

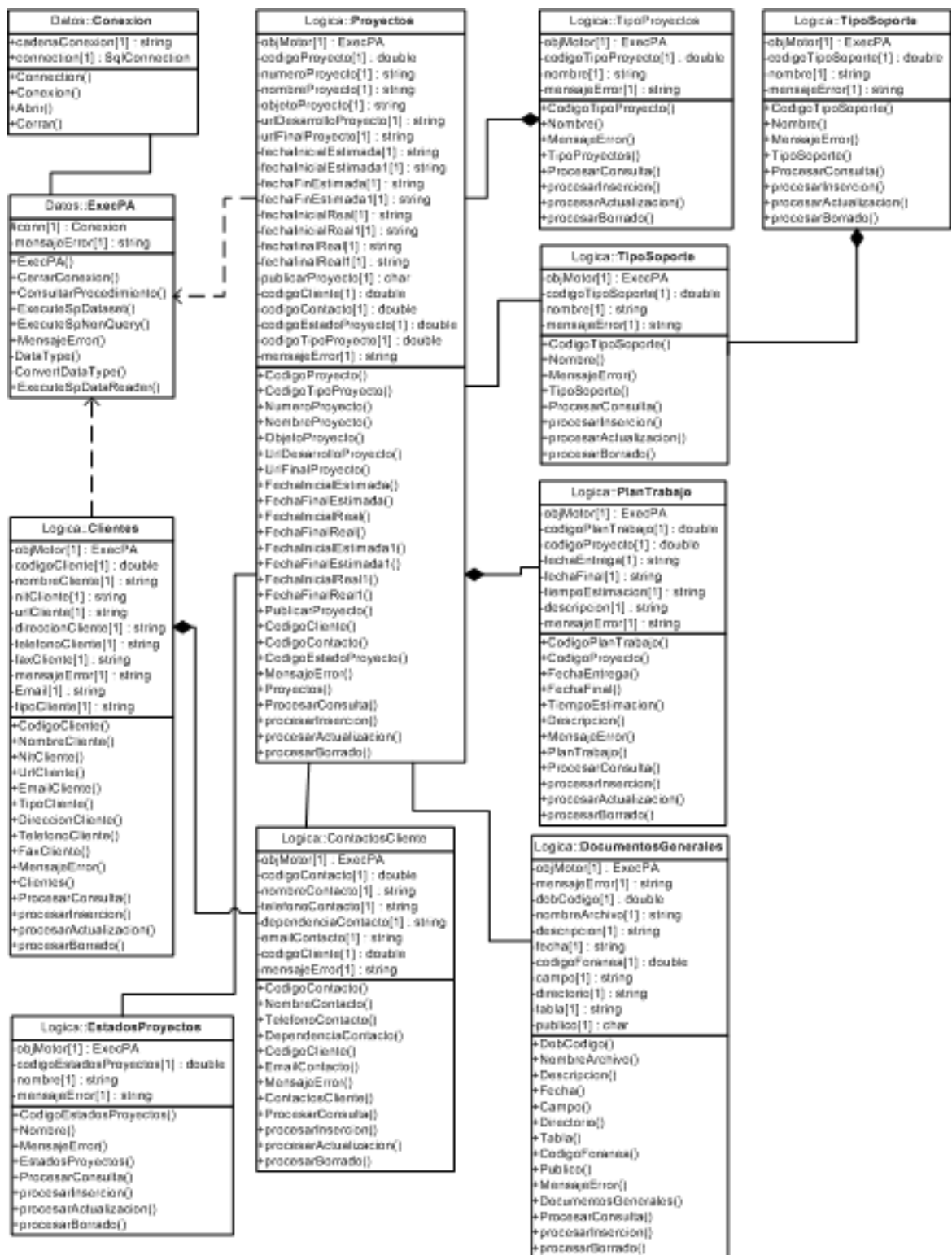


Figura 20. Diagrama de clases

- **Tipo proyectos.** La clase tipo proyectos se relaciona con la clase proyectos mediante una composición, lo que significa que los objetos de esta clase serían el todo y los objetos de la clase proyectos las partes de todo, de tal manera que vivirán y morirán con ella.
- **Tipo Soporte.** Esta clase se relaciona con la clase proyectos mediante una asociación, lo que significa que los objetos de esta clase se relacionan con todos los objetos de la clase usuario.
- **Plan de Trabajo.** La clase plan de trabajo se relacionan con proyectos por medio de una composición, siendo los objetos de la clase plan de trabajo las partes de un todo.
- **Documentos Generales.** Esta clase se relaciona con la clase proyectos por medio de una asociación, relacionándose todos los objetos de las dos clases.
- **Estados Proyectos.** Los objetos de esta clase se esta relacionando con los objetos de la clase proyectos, mediante una asociación.
- **Contactos Clientes.** Esta clase además de tener una asociación con la clase proyectos, establece una relación de composición con la clase clientes, esta clase de relación representa una asociación todo-parte,

para este caso el todo es la clase Clientes y la parte la clase contactos clientes.

- **Proyectos.** Tiene una relación de:
 - o **Composición.** Tipo Proyectos, Plan de trabajo.
 - o **Asociación.** Tipo Soporte, Documentos generales, Contactos clientes, Estados proyectos.
 - o **Dependencia.** Establece una relación de uso con los objetos de la clase ExecPA, lo que significa que un cambio en objeto de la clase ExecPA puede afectar la semántica de la clase Proyectos.

5.2 DISEÑO ARQUITECTÓNICO

La arquitectura de una aplicación basada en Web se centra en la estructura global hipermedia para el sitio, es decir, la forma como se relacionan por medio de enlaces y jerarquías las diferentes páginas que conforman la aplicación (la estructura del sitio); sin embargo, las páginas son tan sólo una parte de un sistema con varios componentes, esto es de la arquitectura de un sistema. A continuación se presenta estas dos vistas.

5.2.1 Estructura del sitio

La estructura arquitectónica se basa en el modelo de organización lógica *estructura de red*, razón por la cual los componentes arquitectónicos (en este caso las páginas Web) están diseñadas de forma que las más importantes puedan pasar el control, mediante enlaces a otros componentes del sistema. En la figura 21 se muestra el diagrama de la estructura.



Figura 21. Diagrama general de la estructura del Sitio Extranet

5.2.2 Arquitectura del sistema

La arquitectura se describe en su totalidad por el modelo de datos, la organización del software en componentes modulares y, finalmente, la distribución física de estos elementos en conjunto con elementos hardware y de conexión.

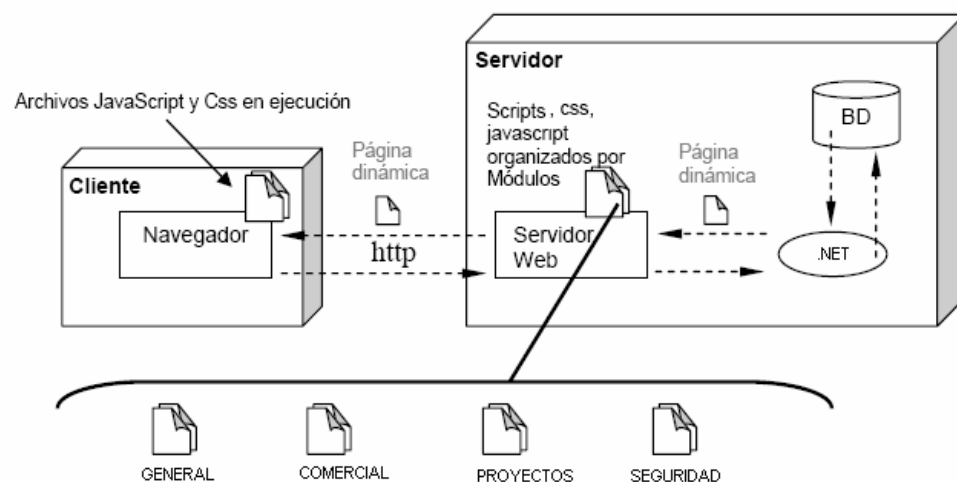


Figura 22. Arquitectura del Sistema

Como elementos de la arquitectura, en primer lugar se identifican tres componentes básicos: servidor Web + conexión de red + cliente (esto por tratarse de un sistema basado en Web y por lo tanto en arquitectura cliente/servidor). La existencia de una base de datos en el sistema hace un sitio dinámico y por lo tanto aparece el intérprete de .NET como un nuevo elemento para tener en cuenta (software de comunicación entre el servidor

Web y la base de datos). Por último, en la parte de software de la aplicación se encuentra las páginas del sitio, organizadas a la vez en módulos de código que se ejecutan en el cliente y/o servidor, (ver Figura 21).

A continuación se presentan algunos lineamientos para la modularización y distribución física de elementos en la arquitectura del sistema, dejando el detalle de modelo de datos para una sección posterior del diseño global.

- Se tomará como arquitectura base del sistema el modelo cliente/servidor, orientando el diseño hacia la arquitectura de tres capas.
- El sistema será ejecutado sobre un nodo servidor, donde residen la capa de datos y la capa lógica o de aplicaciones, y varios nodos cliente donde se ejecuta la capa de presentación. Los usuarios acceden al sistema mediante nodos clientes conectados a Internet.

5.3 DISEÑO DE NAVEGACIÓN

Con base en la arquitectura establecida es posible definir las rutas de navegación que permitirán al usuario acceder al contenido y a los servicios del sitio, teniendo en cuenta la semántica de navegación del sitio (existen diferentes roles o perfiles de usuarios y cada uno tiene asociados unos

servicios y niveles de acceso a contenido particular) y la mecánica de cada enlace de navegación¹¹.

- La opción de navegación estarán centralizadas en un menú principal que estará disponible en todas las páginas que se desplieguen en la ventana principal, y que se configura cuando el usuario entra al sitio e inicia sesión en él.
- La mecánica de navegación, tanto en el menú principal como al interior de las páginas, estará basada en enlaces de texto con orientaciones sobre la acción que desencadenan visualizables a través de la propiedad title de los objetos, de manera que cuando el usuario se ubique sobre el enlace se mostrará la ayuda especificada. También se usarán botones, sobre todo para acciones comunes como son la de ingreso de datos, y la cancelación de operaciones.
- Se contará con mecanismos de búsqueda para la localización de registros, como mecanismos auxiliares para facilitar la navegación.

5.3.1 Diseño de Interfaz

Dentro de la etapa de diseño, un área que debe cubrirse es el diseño de la interfaz. El fin de esta fase fue definir un conjunto de acciones de la interfaz

¹¹ PRESSMAN, 2000. página 530-531

que posibiliten al usuario llevar a cabo todas las tareas definidas de forma que cumplan todos los objetivos del sistema.

De acuerdo con la arquitectura seleccionada para el desarrollo del proyecto, el diseño de la interfaz corresponde a la capa de presentación del prototipo.

Una característica que debe tenerse muy en cuenta al realizar este diseño es la usabilidad, que se define como:

Definición ISO 9621 ó 9241

Es la efectividad, eficiencia y satisfacción con la que los usuarios específicos pueden alcanzar objetivos específicos en un medio particular.

Efectividad: La exactitud y finalización en la consecución de objetivos específicos.

Eficiencia: Los recursos usados, en relación con la exactitud y finalización en la consecución de objetivos.

Satisfacción: El confort y la aceptabilidad del trabajo de un sistema percibido por usuarios.

Definición IEEE'90

Es la facilidad con la que un usuario puede aprender a operar, realizar entradas e interpretar salidas del sistema.

Para el diseño de la interfaz del prototipo, se tuvieron en cuenta las reglas dadas por Jakob Nielsen¹² para garantizar una alta usabilidad, enumeradas a continuación:

- **Visibilidad del estado del sistema:** El usuario debe saber exactamente qué es lo que esta llevando a cabo, proporcionándole para ello indicadores como por ejemplo íconos, títulos de ventanas, codificaciones en colores consecuentes.
- **Prevención de errores:** Aún mejor que desarrollar buenos mensajes de errores es tener un diseño cuidadoso que eviten la ocurrencia de errores. Hay que asegurarse de que las instrucciones estén escritas de una manera clara y que éstas sean desplegadas de manera conveniente, evitando cualquier tipo de contaminación visual. Si se requiere el llenado de un formulario con campos obligatorios, destáquelos por sobre el resto de las entradas. Es muy conveniente hacer una validación de la forma antes de enviarla al servidor.
- **Preferencia al reconocimiento frente a la memorización:** Las instrucciones de uso del sistema deben ser visibles y accesibles cuando el usuario lo considere necesario.

¹² www.dcc.uchile.cl/usabilidad.htm/ Fecha de visita: 7 de septiembre de 2006.

- **Opciones de navegación obvias**, incluso para el usuario casual.
- **Ayuda para que el usuario reconozca, diagnostique y se recupere de los errores:** Los mensajes de error deben ser expresados en un lenguaje claro (sin ambivalencias) indicando el problema y proveyendo constructivamente de una solución.
- **Ayuda y documentación:** El usuario debe contar con ayuda para el manejo del sitio Web.

Para conseguir una interfaz optimizada, se siguieron estas recomendaciones de diseño¹³

- Evite el uso de marcos (frames).
- Utilice hojas de estilo CSS (Cascading Style Sheets). Esto brinda mayor control en el diseño y facilitan el automantenimiento, Además disminuye el tiempo de descarga.
- Usar enlaces con texto significativo.
- Colocar texto alternativo en todas las imágenes.

¹³ www.acceso.uv.es/accesibilidad/estudio/recomendaciones.htm/. Fecha de visita: 7 de septiembre de 2006.

- Haga uso de la mnemónica para acceder a los campos dentro de un formulario, evitando el uso continuo del Mouse, facilitando el empleo de combinaciones de acceso rápido.

Habiendo identificado los escenarios de uso del prototipo en la fase de Inicio, se diseñaron las plantillas que definen la estructura de las páginas.

5.3.2 Composición y distribución de las páginas

Las secciones de la que consta la interfaz diseñada se detallan a continuación:

- Cabecera: Mostrará el logo de la empresa, el slogan, el nombre del usuario y el perfil que posee.
- Menú: Contendrá las funcionalidades generales de navegación.
- SubMenú: Si alguna de las opciones anteriores, contiene otras funcionalidades, están se desplegarán en forma de menús.

Habiendo identificado los objetos de la que consta la interfaz se definió el diagrama de clases correspondiente, que constituye la base para la fase de construcción; uno de los diagramas se ilustran en las figuras 23 y 24:

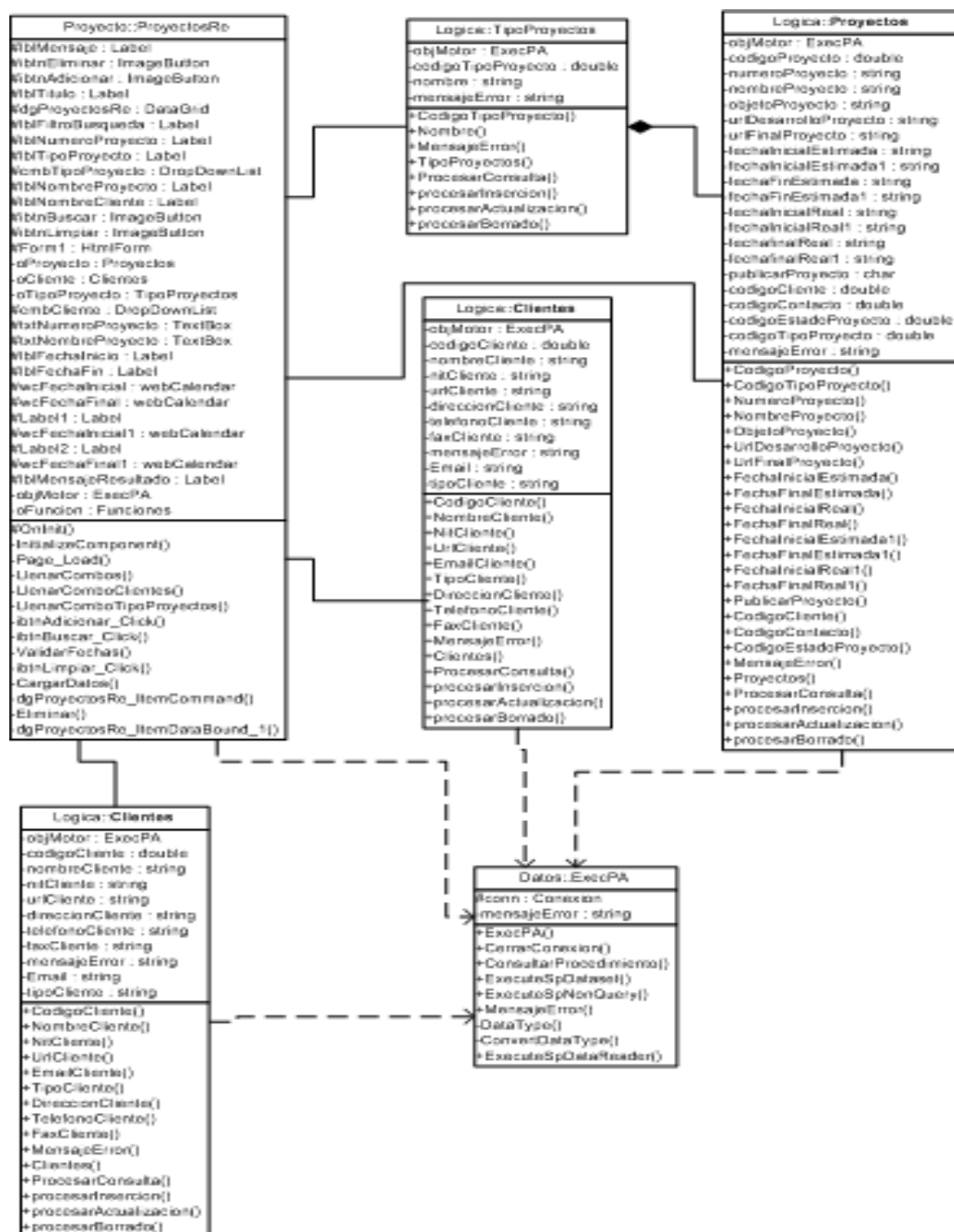


Figura 23. Diagrama de clases de interfaz de clase para un formulario de resultados

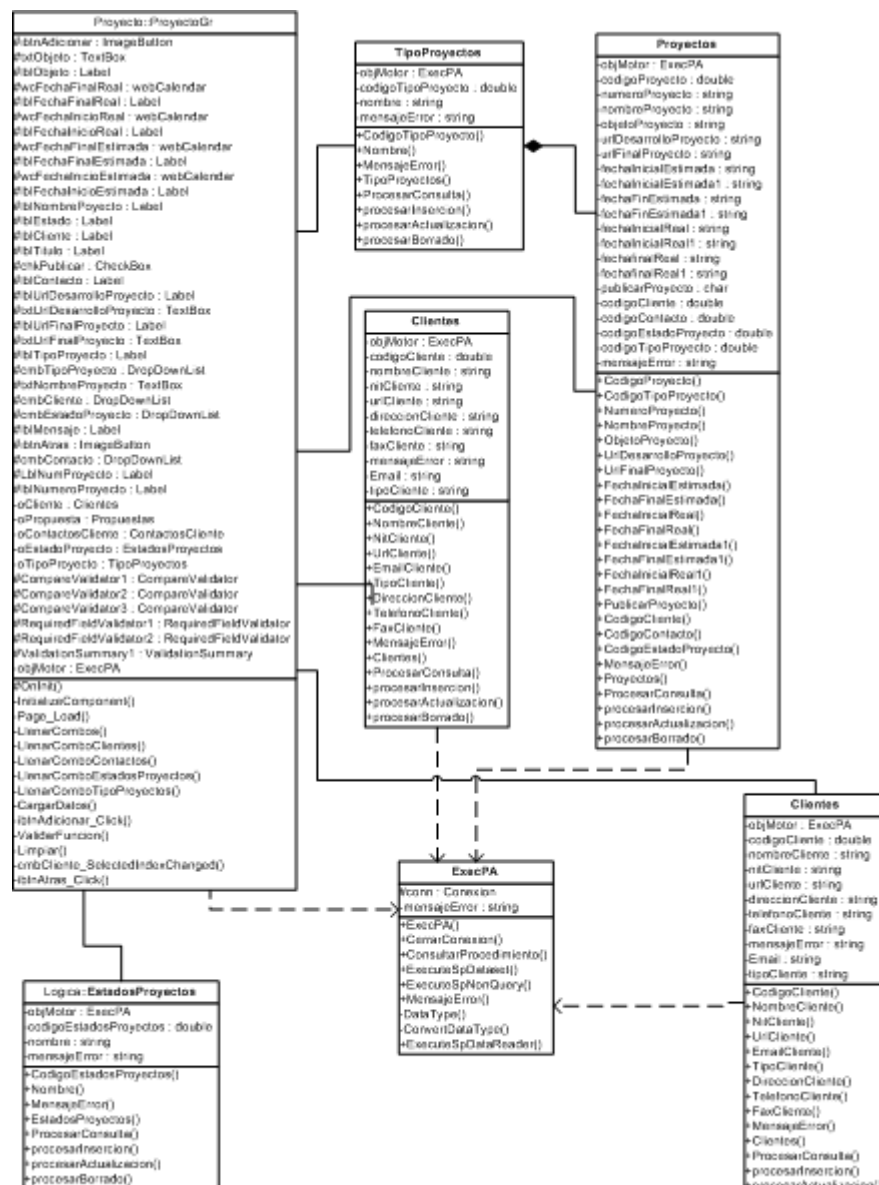


Figura 24. Diagrama de clase de interfaz para un formulario de ingreso/modificación y consulta.

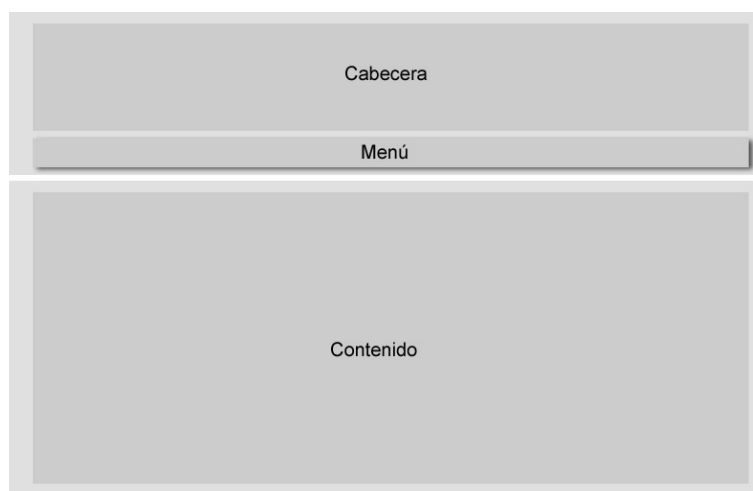


Figura 25. Maqueta General

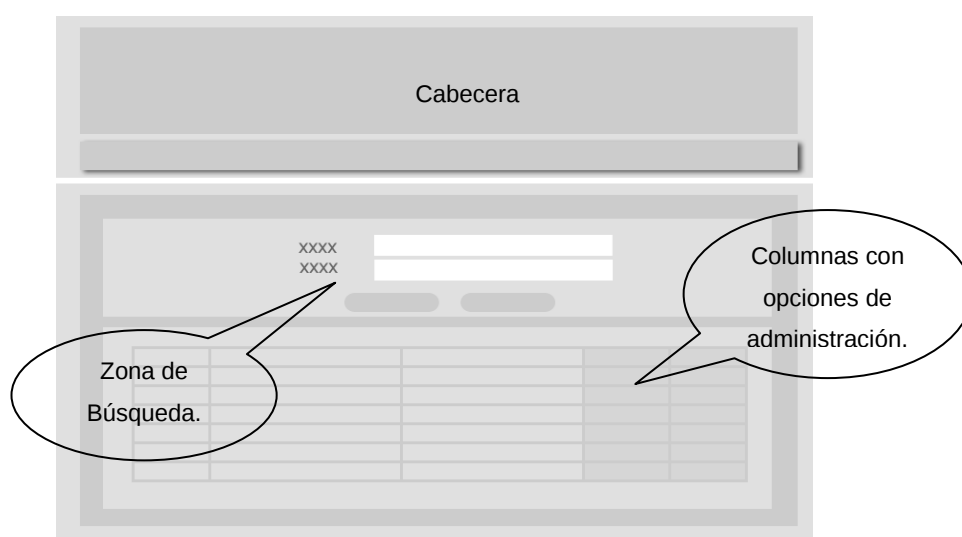


Figura 26. Maqueta General de Zona de contenido: Formulario general de Resultados



Figura 27. Maqueta General de Zona de contenido: Formulario para ingreso/modificación/consulta de información.

5.3.2 Estilo: Fuentes, Colores y Tamaños

La definición del estilo del sitio se hizo utilizando hojas de estilo en Cascada CSS, esto con el fin de facilitar las labores de mantenimiento y desarrollo, y lograr una apariencia común en las diferentes páginas que conforman la aplicación. Se establecieron las pautas básicas sobre tamaño, color y formato de textos y tablas utilizadas en las interfaces de usuarios.

5.3.3 Lineamientos de programación

- Por cuestiones de resolución en pantalla, el contenido de una página se incluirá en una tabla externa de longitud inferior a 800px y centrada en el documento. Dentro de esta tabla se crearán tres filas principales y cada una de ellas contendrá en su orden: el encabezado de la página, una tabla para el contenido y una última fila para un pie de página.
- Para que los colores visualizados en las páginas no dependan de la paleta de colores de los navegadores, los colores que se utilicen deberán especificarse por medio de su código hexadecimal.
- Para agilizar el tiempo de descarga de las páginas, el tamaño de las imágenes que se utilicen deberá especificarse en las etiquetas html, haciendo uso de las propiedades height y width, respectivamente.

5.4 DIRECTRICES DE PROGRAMACIÓN

Para facilitar las labores de mantenimiento de la herramienta, conservando la estandarización en la estructura y código del sistema se recomienda seguir las orientaciones de programación proporcionadas en el Anexo C.

5.5 MODELO DE DATOS DEL SISTEMA

Una de las actividades importantes en la etapa de diseño, es la definición del modelo de datos, el diseño de las estructuras de datos es la parte esencial en la creación de aplicaciones de alta calidad; teniendo en cuenta el análisis de requisitos del software y los diagramas realizados anteriormente, de clases, de usos, entre otros, se diseñó el modelo de datos del prototipo, que abarcó todos los módulos, igualmente se elaboró el diccionario de datos, el cual consta de una lista de elementos que componen el sistema y su descripción.

El diccionario de datos es definido según Yourdon como un listado organizado de todos los elementos de datos que son pertinentes al sistema, y que ayudan a tener una mejor comprensión de los componentes del almacén de datos¹⁴.

El modelo de datos se compone de tres piezas información interrelacionadas: El objeto de datos, los atributos que describen el objeto de datos y la relación que conecta objetos de datos entre sí; la notación gráfica de estas tres piezas se representa mediante un diagrama entidad-relación que se ilustra en la figura 28, comprendiendo el sistema en su totalidad; este modelo es el compendio de los modelos de datos que se diseñaron en cada uno de los módulos.

¹⁴ GRECH, Pablo. Ingeniería del Software: Un enfoque práctico. Prentice Hall, Bogotá, 2001. Capítulo 12.

El diccionario de datos que recogió las características de los datos asociados al software y proporcionó un enfoque adecuado para la implementación del sistema, se encuentra en el Anexo B.

El diseño del modelo entidad-relación se realizó teniendo en cuenta parámetros de normalización, un conjunto de reglas que sirven para ayudar a los diseñadores a desarrollar un esquema que minimice los problemas de lógica y redundancia de datos, cada regla está basada en la que le antecede. Dentro de las ventajas de la normalización se encuentran:

- Ayuda a prevenir errores lógicos en la manipulación de datos.
- Provee un marco de referencia para simplificar la estructura, haciendo las cosas más fáciles de entender.

Disminuye el consumo de espacio, una base de datos normalizada puede ocupar menos espacio que una no normalizada, hay menos repetición de datos, lo que tiene como consecuencia un mucho menor uso de espacio en disco.

Las figuras 29, 30, 31 y 32 presentan el modelo de datos de la base de datos Extranet, organizado según los módulos principales o aplicaciones generales desarrollados.

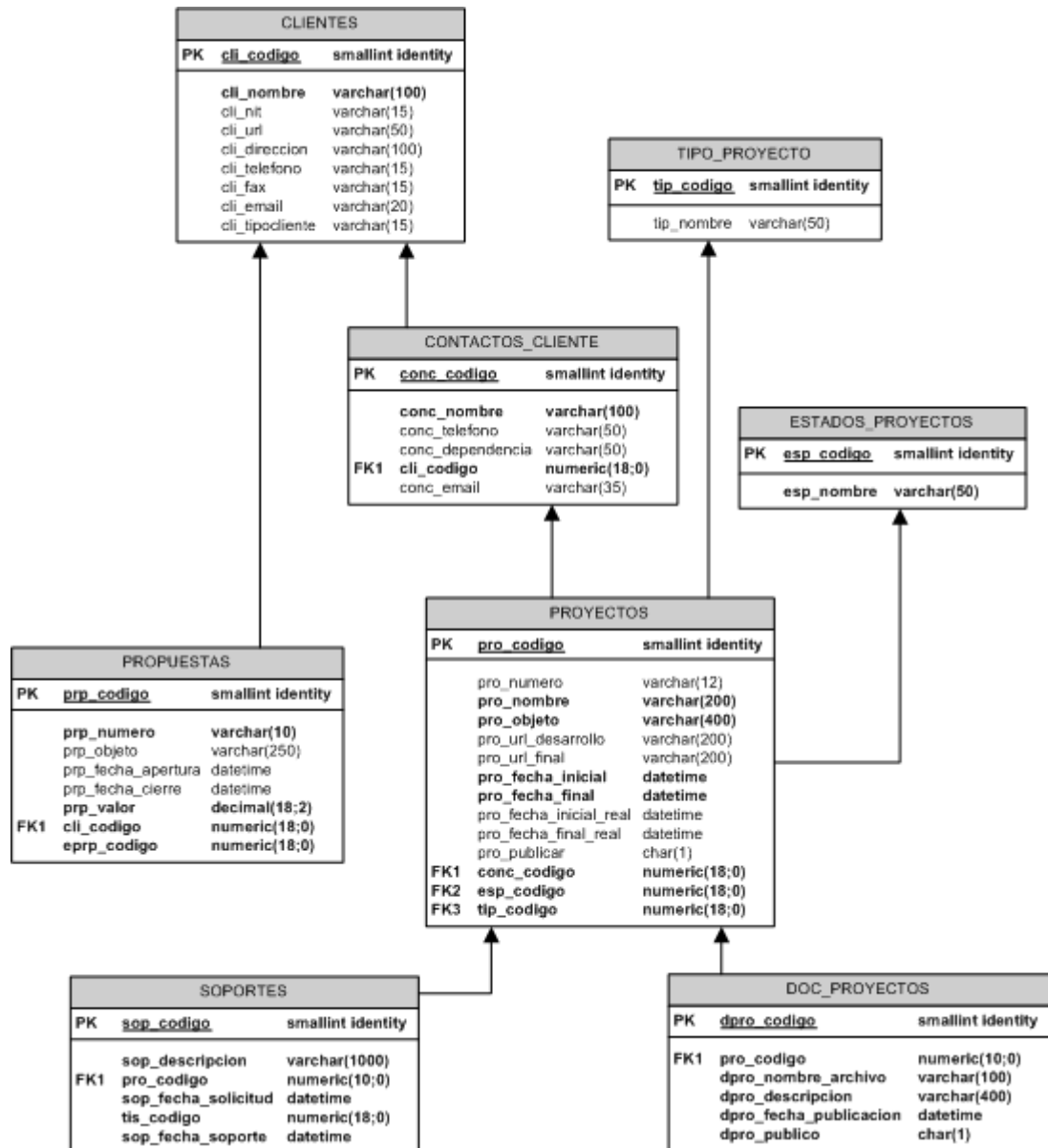


Figura 29. Modelo de datos de proyectos implementado en la Extranet

El módulo de proyectos, de la figura 29, está compuesto por cerca de 8 tablas. La tabla proyectos, es la que almacena la información correspondiente a todos los proyectos del sistema. La tabla contactos_clientes, es la encargada de manejar las personas o usuarios encargadas de los proyectos en las respectivas empresas clientes. La tabla clientes, se almacena la información

básica de las empresas clientes. Las tablas estados_proyectos, tipo_proyecto son tablas consideradas básicas y almacenan la información base para el manejo y seguimiento de proyectos. La tabla doc_proyecto almacena los documentos que surgen del desarrollo de un proyecto específico. La tabla propuestas almacena la información concerniente al contrato o convenio de respaldo al proyecto.

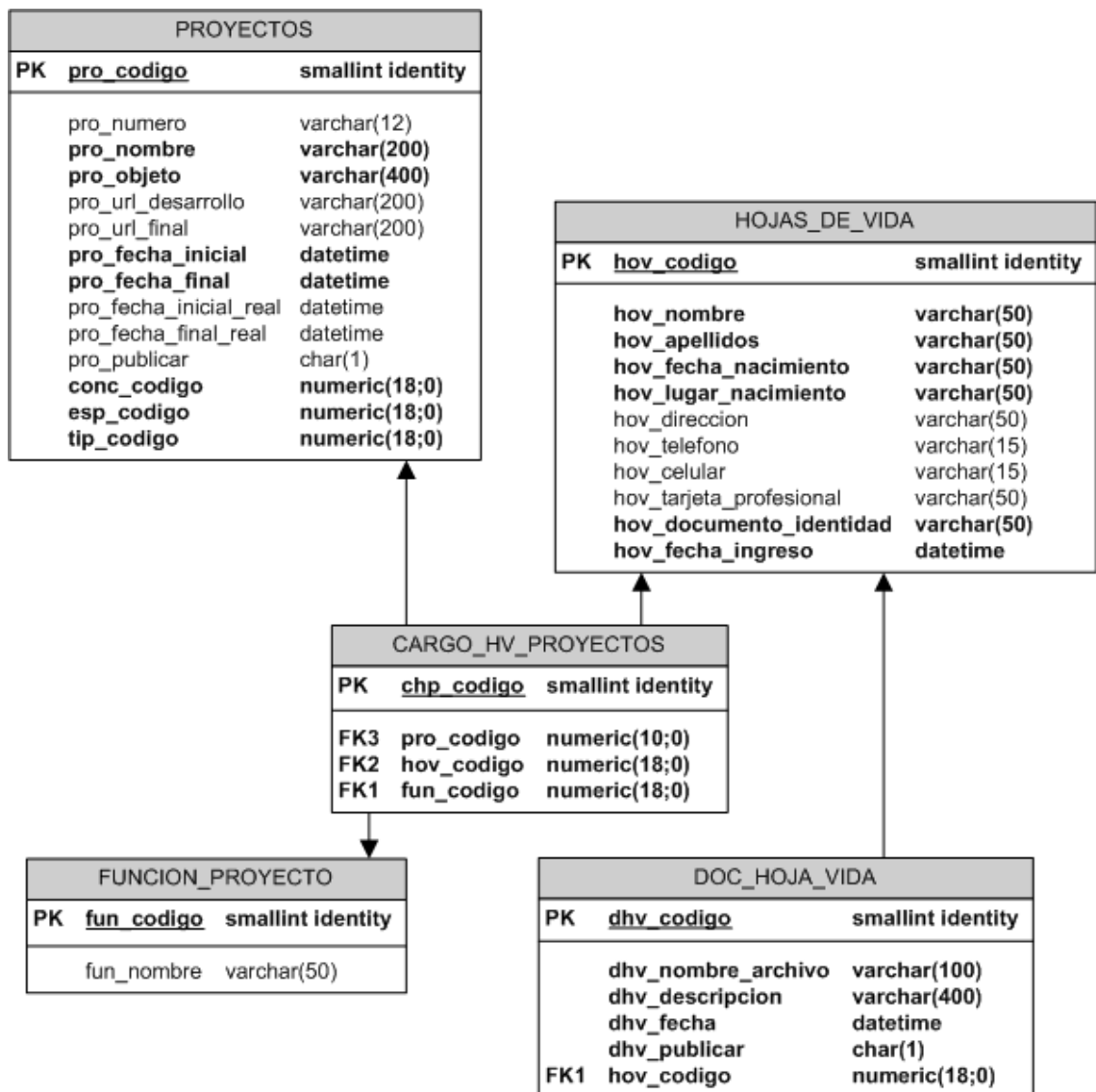


Figura 30. Modelo de Datos Personal

El módulo de personal, de la figura 30, está compuesto por cerca de 5 tablas.

La tabla Hojas de vida, es la que almacena la información correspondiente a todos la información del personal de trabajo pertenecientes al sistema. La tabla cargo_hv_proyectos, es la encargada de manejar las personas asignadas a los proyectos y es la intermediaria entre las tablas hoja de vida y proyecto. La tabla clientes, se almacena la información básica de las empresas clientes. La tabla función proyecto es una tabla básica donde se almacena las funciones o cargos que se pueden realizar en un proyecto, y la tabla doc_hoja_vida, se almacena algún archivo digital que sirva de soporte para la hoja de vida de cada empleado.

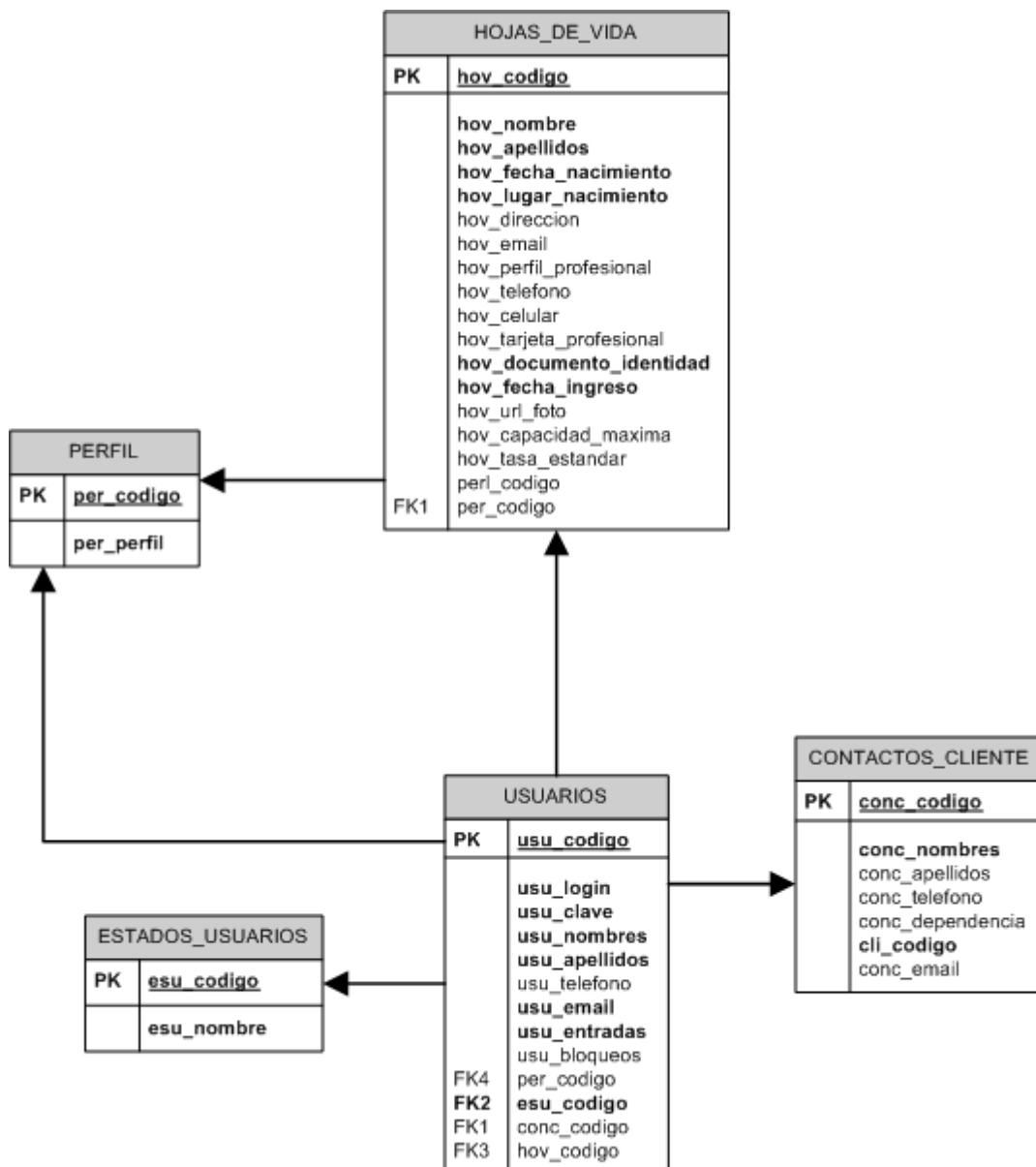
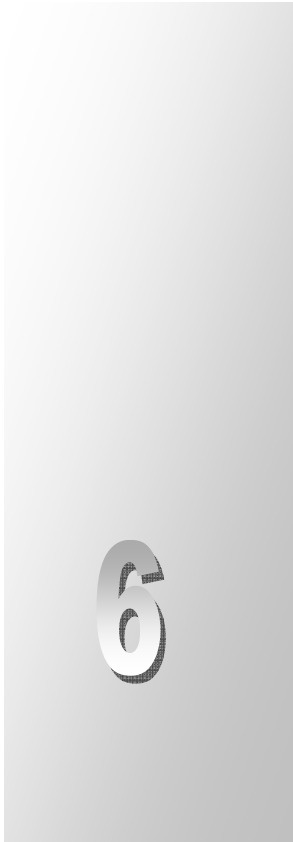


Figura 31. Modelo de datos módulo de seguridad (Usuarios)

El módulo de seguridad, de la figura 31, está compuesto por 5 tablas. La tabla usuarios, es la que almacena la información correspondiente a todos los usuarios pertenecientes al sistema. La tabla perfiles almacena información correspondiente al perfil o rol que desempeña cada usuario. La tabla

estados_usuarios, se almacena la información básica del estado de un usuario específico.



6

FASE DE CONSTRUCCIÓN

En esta fase, se tiene como propósito llevar a un producto software, que en este caso el sistema de información Extranet para la empresa Quirón Software, hacia una versión operativa inicial enmarcada dentro de los requisitos y objetivos planteados inicialmente.

El desarrollo del sistema se hace mediante interacciones. Estas se realizan a través de los flujos de requisitos análisis, diseño, implementación y pruebas. Cabe anotar que en esta fase el trabajo se concentra sobre los dos últimos flujos, ya que estos permiten lograr la consecución de los objetivos trazados.

6.1 PLANIFICACIÓN DE LA FASE DE CONSTRUCCIÓN

Esta fase se llevó a cabo en tres iteraciones, las cuales se muestran en la tabla 10. En la primera de ellas se identifican los restantes casos de usos concernientes a información base en la aplicación y se implementan lo que no se detallaron en la fase anterior. Para la segunda iteración se desarrolla la construcción del módulo de proyectos y las consultas respectivas a los planes de trabajos y actividades. Para la tercera y última iteración se desarrollan los informes referentes a los proyectos.

Iteración	Subsistemas
1	Registro de información básica
	Ingresos de usuarios al sistema
	Registro de empleados (Hojas de vida)
	Registro de Clientes
2	Inscripción de Proyectos
	Registro de plan de trabajo
	Ingreso de actividades
3	Solicitud de Soportes
	Registro de dificultades de soportes
	Generar informes proyecto

Tabla 10. Iteraciones del proyecto

6.2 CASOS DE USOS

En la figura 32, se muestran los casos de usos generales que se identificaron hasta el momento y se agrupan en paquetes correspondientes.

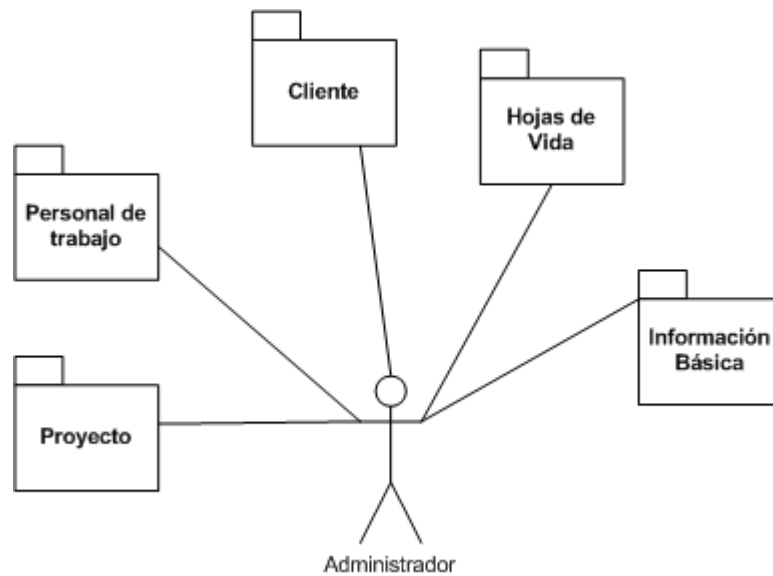


Figura 32. Diagrama de casos de uso clasificados en paquetes

Los paquetes que se muestran en la figura 32, agrupan casos de usos que se detallan en las gráficas No. 33, 34, 35, 36, 37.

En la figura 33, se ilustran los casos de uso que contiene el paquete Información Básica, el cual se definió en el diagrama de casos de usos global como un caso de uso asociado al módulo administrador.

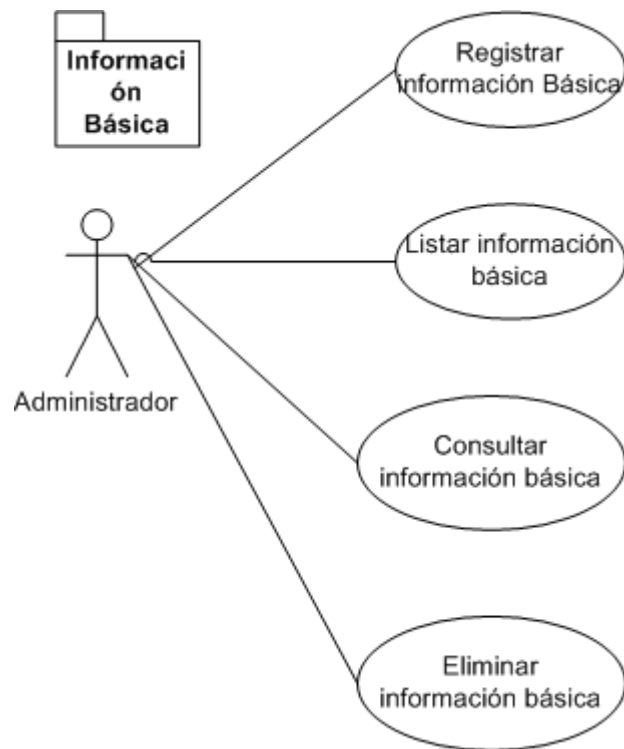


Figura 33. Caso de Uso Paquete Información Básica

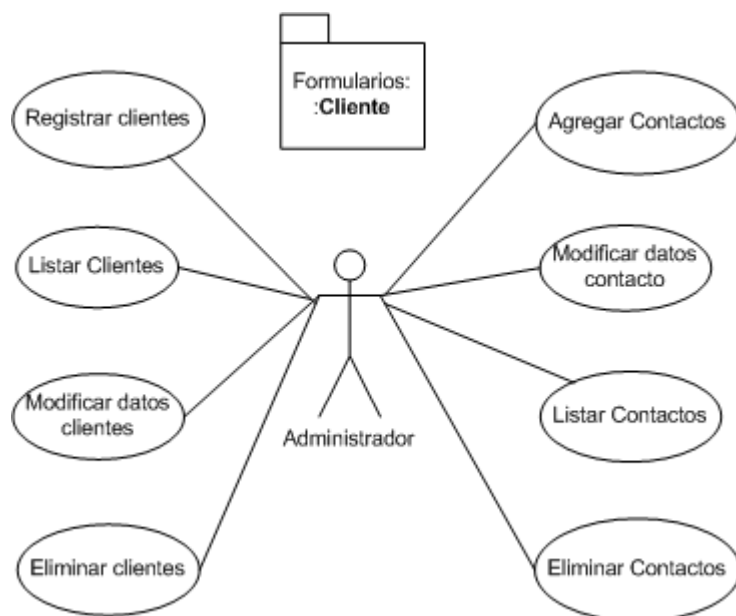


Figura 34. Casos de uso paquete Clientes

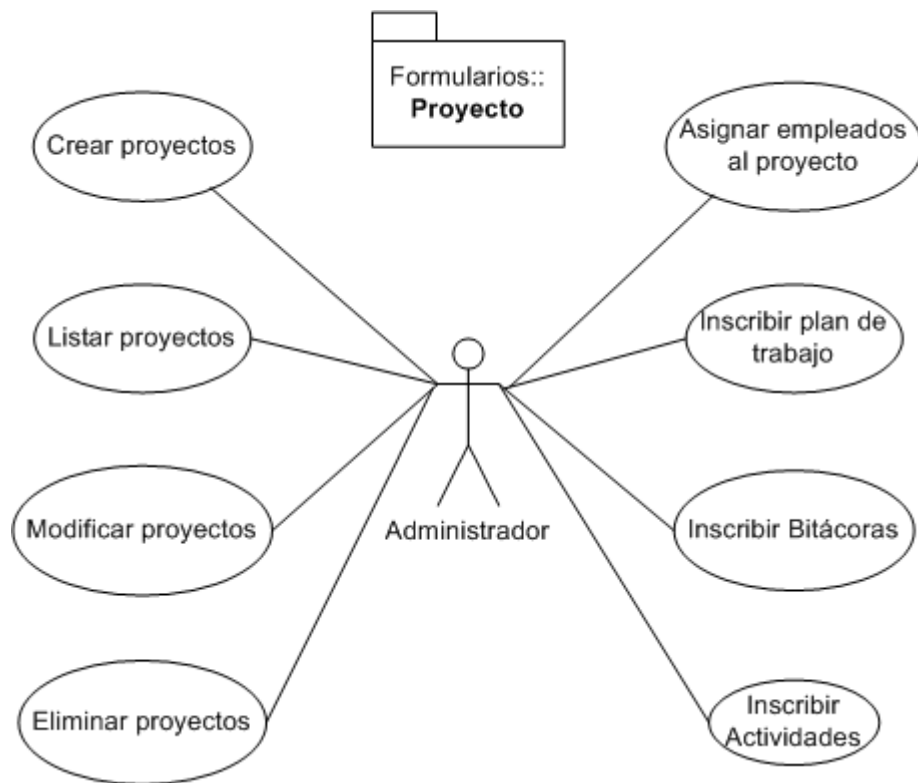


Figura 35. Caso de uso paquete Proyectos

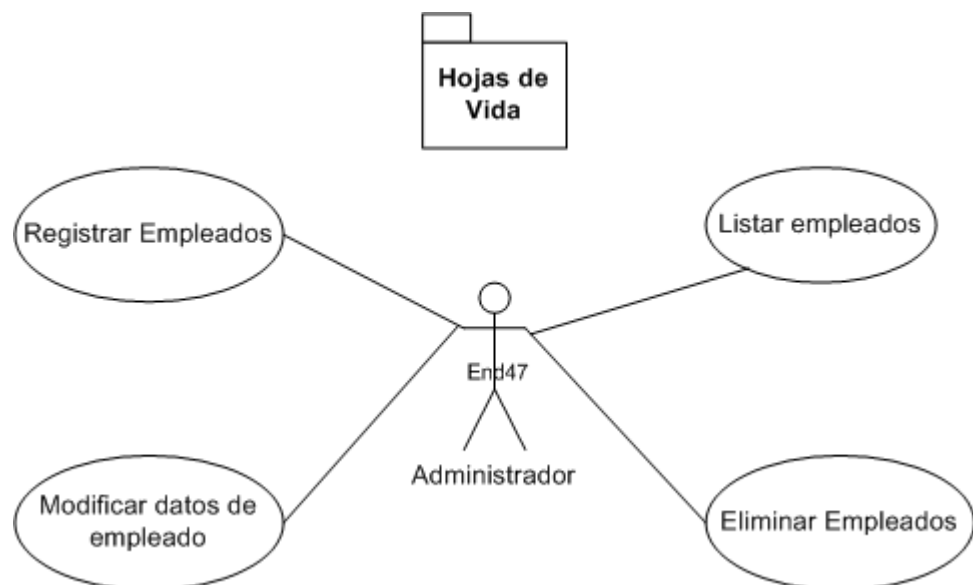


Figura 36. Caso de uso Empleados

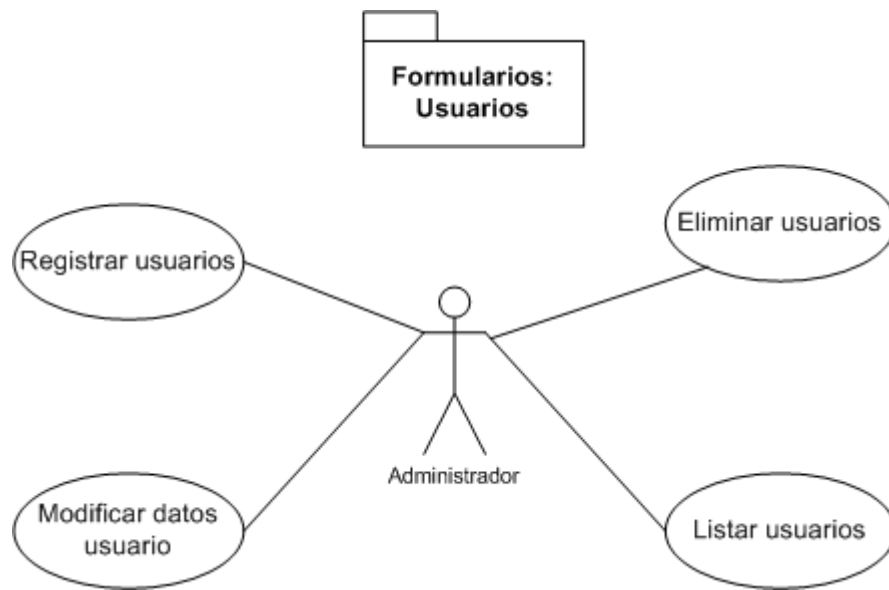


Figura 37. Caso de uso Usuarios

6.2.1 Análisis de Casos de Usos

- **Actualizar proyecto.** En la figura 38 muestra como se desarrolla el caso de uso a través de los de los análisis.

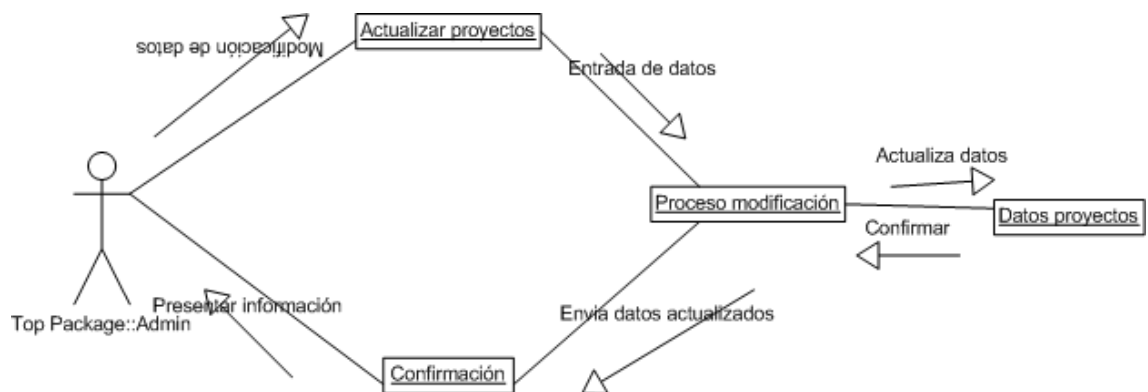


Figura 38. Diagrama de Colaboración de actualizar proyectos

Descripción. El objeto “Actualizar proyectos” proporciona el formulario donde se muestran los datos del proyecto que se puede modificar. Este objeto también puede validar los campos debidamente diligenciados en el formulario, que luego serán enviados a la clase de modificación. Esta clase se encarga de actualizar los datos correspondientes en la base de datos. La clase “modificación” recibe la confirmación y muestra los datos actualizados en la clase de interfaz proyectos. Si el usuario estando en el paso inicial no decide continuar puede cancelar la modificación de los datos.

- **Solicitar Soportes.** La figura 39, muestra como se desarrolla al caso de uso a través de las clases de los análisis.

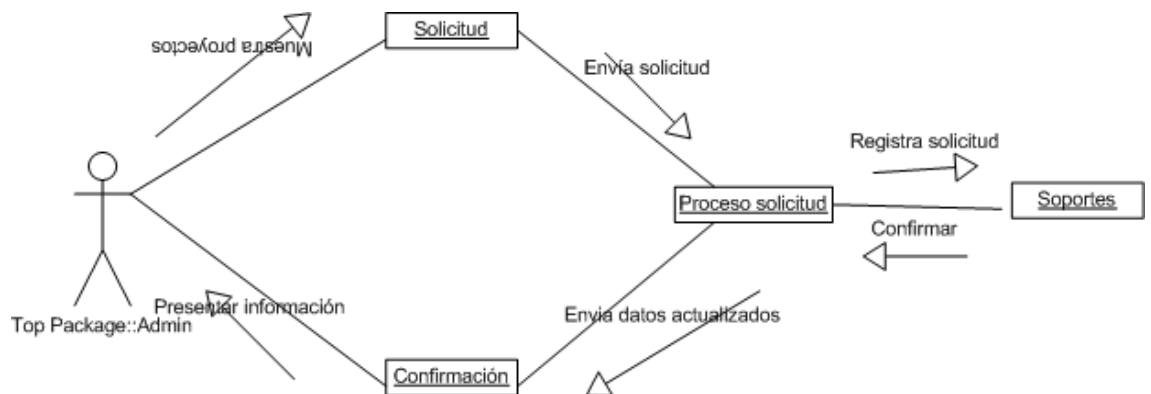


Figura 39. Diagrama de Colaboración de solicitar soportes

Descripción. El actor ingresa al objeto “Solicitud” donde se presentan los proyectos. Se envía los datos necesarios a la clase de control “Proceso de solicitud”. Esta clase manejará el registro de la información en la respectiva tabla de la base de datos en el sistema. La clase solicitud, recibe la confirmación de la acción realizada y presenta la información actualizada.

A continuación en la figura 40 se muestra el diagrama de secuencia.

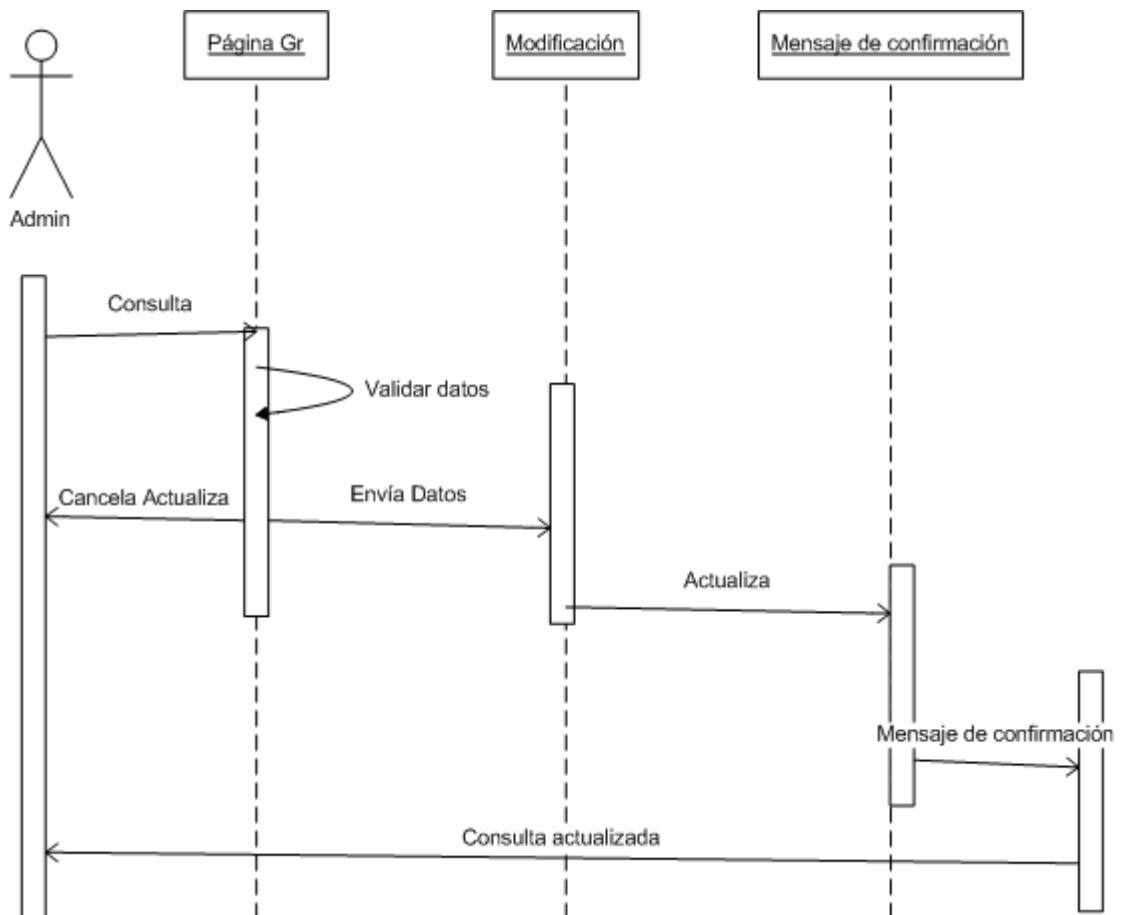


Figura 40. Diagrama de Secuencia de actualizar proyectos

El caso de uso inicia cuando el objeto “Página ProyectosGr” es invocado por el actor. En este momento, se muestra por defecto en la interfaz la información del proyecto que se desea actualizar, a las cuales puede realizar el registro de notas, se comprueba que los datos sean correctos y se envían al objeto de Modificación, el cual se encarga de ingresarlos. Cuando la acción es realizada se prepara un mensaje según el éxito de la operación que es visualizado sobre el mensaje de confirmación y se muestra al mensaje.

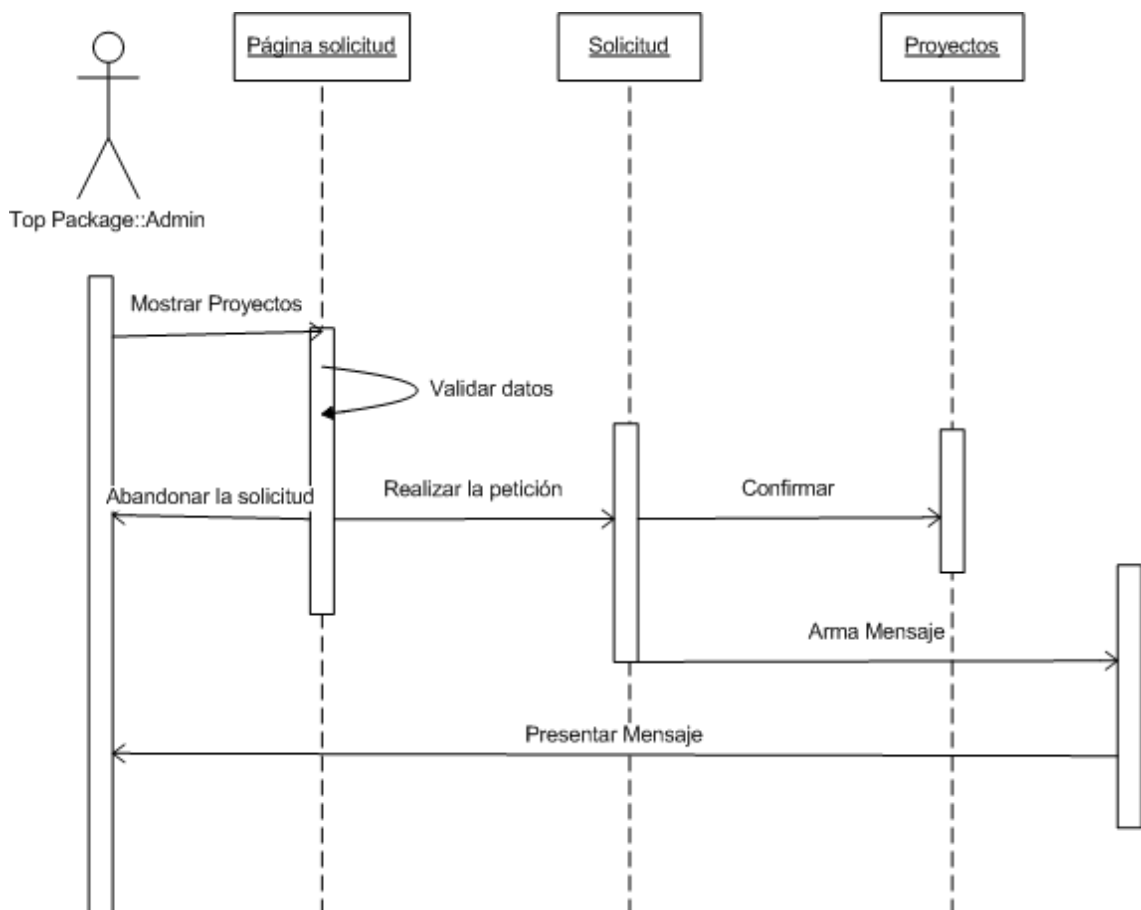


Figura 41. Diagrama de secuencia de solicitar soportes

El caso de uso inicia cuando la página de solicitud, es invocada por el actor. En este momento, se muestra por defecto en la interfaz los proyectos a los que puede realizar el registro de soportes.

El actor debe seleccionar un proyecto, luego de esto el objeto “Página solicitud” envía los datos del proyecto al objeto solicitud, quien es el encargado de realizar la operación de inserción en la base de datos. Así mismo se verifica que los resultados de la operación sean correctos y con esto se genera la solicitud de soportes. Según los resultados de la operación anterior, arma un mensaje de confirmación.

El actor puede abandonar la acción de registro cuando considere necesario.

6.3 FLUJO DE PRUEBA

Durante este flujo se busca las posibilidades de fallas y errores en la funcionalidad de cada uno de los casos de uso.

En la fase de inicio se realizaron pruebas con el fin de establecer una arquitectura candidata para el sistema. Para ello se implementaron algunas interfaces preliminares en el lenguaje HTML, con el propósito de capturar datos para un posterior uso en el sistema.

Para la fase de elaboración, se implementaron los casos de uso de relevancia para la arquitectura y las pruebas se dirigieron hacia el funcionamiento de la misma.

Las interfaces se convirtieron a ASP.NET, se incluyó código de script (JavaScript) para validar los datos capturados en estas interfaces y se crearon las clases para los casos de uso referente a consultas de información básica, proyectos, cliente y empleado.

Para la fase de construcción, las pruebas realizadas a los incrementos funcionales objetos de cada entrega planeada se orientaron a evaluar tanto los requisitos funcionales a cubrir (pruebas de caja negra¹⁵), como el funcionamiento interno de los programas (pruebas de caja blanca¹⁶), con el fin de descubrir y corregir el mayor número de errores del software antes de las pruebas con los usuarios finales y la implantación final del sistema. Estas pruebas fueron ejecutadas por los desarrolladores del proyecto durante el desarrollo y al final de cada entrega.

¹⁵ JACOBSON, Ivar, Booch, Grady. Rumbaugh, James. El proceso unificado de desarrollo de software.

¹⁶ Las pruebas de caja blanca, se centran en la estructura de control de programas, a nivel de código. Las pruebas de caja negra, o pruebas de comportamiento, se centran en los requisitos funcionales del software, orientándose a validar su funcionalidad, rendimiento y comportamiento. (PRESSMAN,2000).

Para evaluar el funcionamiento interno de los programas se elaboraron pruebas diseñadas con el fin de descubrir defectos del software a nivel de: estructura lógica de los programas y almacenamiento y recuperación de la información en la base de datos. En la realización de estas pruebas se hizo mayor énfasis en los siguientes aspectos:

- Validación de entradas en el formulario
- Carga y descarga de archivos del servidor
- Verificación de que no exista pérdida de datos en el envío de un formulario.
- Almacenamiento y actualización de datos en la base de datos.
- Resultados de las consultas realizadas a la base de datos.
- Conservación de la integridad referencial en la base de datos.
- Establecimiento y mantenimiento de una sesión de usuario.
- Filtrado de opciones según el tipo de usuario en sesión.

Para evaluar los requisitos funcionales se realizaron pruebas independientes a cada módulo y aplicación desarrollada como parte del incremento a entregar en cada iteración, y se finalizó con la integración de estos componentes y la exploración del incremento así obtenido como base en todos los casos de uso.

A continuación, en la tabla 11 se muestra algunas de las pruebas de caja negra.

CASOS DE USO	Pruebas	Observaciones
Registrar usuarios	• Ingresar registro • Abandonar registro	• El ingreso de los datos del usuario fue realizado correctamente por el sistema • Al abandonar el registro el sistema se deshabilitó la validación del usuario anterior.
Ingresar Proyectos	• Validar usuario • Abandonar registro	• La validación de los datos del usuario fue realizada correctamente por el sistema. • Al abandonar el registro del sistema deshabilitó la validación del usuario realizada anteriormente.
	• Validar datos ingresados a la página. • Realizar la verificación de la fecha. • Realizar el registro de la solicitud de cancelación de semestre,	• La información que el usuario ingreso a la página se validó correctamente y se visualizan mensajes de error en los datos. • Se realizó con éxito el ingreso de la información • El sistema validó las fechas, y se le informó al usuario si estas no son coherentes.
Solicitar soportes	• Validar usuario • Abandonar solicitud	• Similar a la prueba realizada en Ingresar Proyectos.
	• Ingresar a la página de soportes. • Seleccionar el proyecto a realizar soporte. • Realizar el registro de la solicitud de soportes.	• La información correspondiente a los proyectos apareció correctamente en pantalla. • La información corresponde con la almacenada en el servidor de Base de Datos. • El registro del proyecto seleccionado se realizó correctamente.

Tabla 11. Pruebas de caja negra

Tras la entrega final se desarrolló una experiencia orientada a evaluar principalmente dos aspectos importantes del software: validación o aceptación de la Extranet por parte de los usuarios y funcionamiento del sistema con todos los componentes software y hardware integrados. En el conjunto de pruebas

diseñadas para la experiencia se tocó aspectos con los cuales se puede identificar errores y mejoras en las características más relevantes, y pertinentes a la experiencia, para evaluar la calidad de los sistemas basados en Web, como son la usabilidad, funcionalidad, fiabilidad y eficiencia¹⁷.

6.3.1 Resultados de las pruebas

Las pruebas realizadas arrojaron como resultado:

- Validación o aceptación del sitio por parte de los usuarios:
 - o Es visualmente cómodo, y de fácil navegación.
 - o La distribución de los elementos en las páginas facilita la interacción con el sitio.
 - o Existe coherencia entre las opciones de usuario y las acciones a las que conduce.
- Funcionamiento del sistema:
 - o Se probaron las funcionalidades implementadas; no se detectaron fallas graves que impidieran la ejecución normal de la aplicación.
 - o En la estructura de datos no se detectaron fallas o inconsistencias, las operaciones de creación, modificación y eliminación arrojaron los resultados esperados, y la propagación de los datos se dio correctamente.

¹⁷ Estos son requisitos de calidad mencionados en Presuman 2000, capítulo 29, página 524.

CONCLUSIONES

- El sistema de información Extranet, entra a la empresa como herramienta de apoyo en el proceso de administración de proyectos. además, se convierte en una herramienta de soporte que mejora el seguimiento de proyectos llevados a cabo por la empresa Quirón Software, una vez que ofrece un entorno amigable para nuevos servicios a través del entorno Web.
- Complementar la metodología de autenticación del usuario dentro de un esquema rol-usuario, con el esquema de autenticación garantiza un alto nivel de seguridad en el acceso al sistema.
- El hecho de usar la tecnología orientada a objetos favoreció la aplicación de la metodología del proceso unificado de desarrollo de software, haciendo que el software que se produzca sea de calidad y satisfaga las necesidades del usuario de forma eficiente.
- A pesar de que la curva de aprendizaje de ASP.NET 2005 es lenta, es un lenguaje que presenta ventajas sobre los demás lenguajes orientados

hacia la programación para la Web como PHP, JSP, ya que es portable, más flexible y estructurado.

- La aplicación de estándares de codificación mejora el mantenimiento del software durante todas las etapas del desarrollo del mismo y se presenta como una opción para dar calidad al código de un producto de software.
- Se dio cumplimiento a los objetivos planteados al inicio del proyecto, lo cual se puede demostrar a través de la ejecución de las funcionalidades del sistema en cada uno de sus módulos.

RECOMENDACIONES

- Es importante que se continúe con el desarrollo de módulos que permitan ampliar el sistema de información Extranet hacia:
 - o Empresas aliadas.
 - o Seguimiento a indicadores, y de esta manera contar con una aplicación completa que apoye desde el diseño, seguimiento y control de los proyectos realizados en la empresa.

Ofreciendo facilidades para compartir documentación acerca de proyectos y temas relacionados con éstos, logrando crear además un portal de conocimiento que congregue a quienes se encuentran interesados en estos temas, y con ello colaborar en el crecimiento del conocimiento.

- La tendencia de la industria del software actual está orientada al desarrollo de aplicaciones sobre dispositivos móviles, tendencia mundial que crece vertiginosamente y tiene una fuerte presencia en las organizaciones. Una posible mejora sería la de implementar los módulos del sistema en esta plataforma de desarrollo.

Se propone para posteriores versiones:

- Agregar un módulo de lenguaje con el fin de que se presente la disponibilidad de varios idiomas en la presentación del portal a los usuarios.
- Mejorar los módulos y aplicaciones de la versión actual, atendiendo las limitaciones que surjan como resultado de la operación del sistema.
- Mejorar la compatibilidad del sitio con navegadores diferentes a Internet Explorer, en especial con un navegador gratuito que sea ejecutable en Linux.
- Implementar una bitácora de seguridad que registre las acciones realizadas por los usuarios en el sistema.
- Crear alertas para un mejor análisis de la estimación del tiempo de desarrollo de proyectos, considerando factores que ocasionan retrasos durante la ejecución del mismo.

ANÁLISIS DE LA LITERATURA RECOPIADA

Se presenta la referencia bibliográfica de la información recopilada en la fase de investigación, organizada según su aporte al desarrollo del presente proyecto; teniendo en cuenta que la búsqueda y soporte de fuentes bibliográficas es una actividad que continúa a lo largo del desarrollo del proyecto, con el fin de profundizar y refinar la solución a las situaciones problemas que se presentan en el transcurso de su implementación.

8.1 MARCO DESCRIPTIVO

Enumera todas las referencias que permiten realizar una adecuada descripción de la situación problema, comprender los tópicos involucrados en su identificación y estudiar otros proyectos relacionados con el tipo de problema que se está abordando.

- Sistema de información intranet-extranet para la seccional de la universidad industrial de Santander en Barrancabermeja / Edwin Ramón Suárez Álvarez.

En este proyecto se implementó un Sistema Intranet-Extranet para la gestión académica en la Universidad Industrial de Santander Seccional Barrancabermeja.

- GÓMEZ FLOREZ, Luis C. Guía para el desarrollo de proyectos de grado

Es una guía que sirve de apoyo para la realización del proyecto de grado, donde se especifican algunos parámetros o pautas a seguir en el desarrollo del mismo.

MARCO METODOLÓGICO

Son los textos que sirven de guía para la correcta documentación de la investigación y que presentan guías metodológicas de trabajo.

Libros

- Pressman, Roger S., Ingeniería del Software un enfoque práctico, edición No. 5, España, Mc Graw Hill, 2002.

Este es un libro que ofrece importante información acerca de la Ingeniería de Software de cualquier proyecto de este tipo, tratando importantes aspectos como la gestión de los proyectos, su planificación,

la metodología a usar, la gestión de los riesgos y la calidad, y las estrategias para realizar pruebas. Se considera que estos son aspectos que en cualquier proyecto de software se deben tener en cuenta.

- McConnell, Steve, Desarrollo y gestión de proyectos informáticos, Primera Edición, España, McGrawHill, 1997.

Este libro ofrece temas para ayudar a realizar la identificación de la metodología que se va a usar para el desarrollo del proyecto, y las diferentes etapas y actividades que se deben aplicar para la evolución del mismo. Contiene además, información sobre el uso del prototipado evolutivo en el desarrollo de software, lo cual será necesario consultar para realizar una buena planeación del sistema a desarrollar.

MARCO RESOLUTIVO

En él se presentan las fuentes bibliográficas que aportan la información requerida para dar solución a la situación problema. Documentos dirigidos al conocimiento de las herramientas tanto teóricas como prácticas que ayudan a construir la solución al problema planteado.

Libros

- Aprenda ya Microsoft asp.net / g. Andrew duthie, 2002

Este libro es una completa guía de referencia a Asp.Net, sirviendo como apoyo para el conocimiento de las herramientas que se van a emplear para la implementación del Sistema de Información.

- Programación en Xml para Microsoft.net. Esposito, Dino
- Diseño de aplicaciones con Microsoft Asp.net. Reyly, Douglas J.
- Superutilidades para Asp.net. Buczek, Greg

Páginas Web

- Tutorial de ASP .Net, <http://es.gotdotnet.com/quickstart/aspplus/doc/quickstart.aspx>, Microsoft Corporation, 2001.

Este tutorial explica en forma concisa y mediante ejemplos los aspectos que tienen que ver con la sintaxis de ASP .NET, la arquitectura y características de esta tecnología y los mejores procedimientos para usar la eficacia de ASP .NET. Esta guía será muy útil para el conocimiento y correcto uso de las herramientas para el desarrollo del Sistema.

- Curso de ASP .Net, [http://www.willydev.net/Descargas /Cursos/AspNet/](http://www.willydev.net/Descargas/Cursos/AspNet/), Microsoft Corporation, 2004.

Este es un curso completo de ASP .Net con ejemplos, laboratorios y prácticas, que servirá para adquirir las destrezas necesarias para el manejo de esta herramienta.

- Curso de SQL Server 2000, [http://www.willydev.net/descargas/Cursos /SQLServer2000/index.html](http://www.willydev.net/descargas/Cursos/SQLServer2000/index.html), Microsoft Corporation, 2004.

Este es un curso completo de SQL Server 2000 con ejemplos, laboratorios y prácticas, que será de utilidad para adquirir los conocimientos necesarios para el manejo de esta herramienta.

ANEXOS

ANEXO A
CASOS DE USO

DOCUMENTACIÓN CASOS DE USO

1. INTRODUCCIÓN

En el presente documento, se realiza la descripción de los casos de uso identificados para el desarrollo del proyecto, y que describen los escenarios de la forma como el software va a ser usado en determinada situación.

Un caso de uso es, una iteración típica entre un usuario y un sistema de cómputo. Los casos de uso están definidos desde el punto de vista de un actor. Un actor es un papel que las personas (usuarias) o dispositivos juegan cuando interactúan con el software [1].

2. CASOS DE USOS GENERALES

Antes de determinar los casos de uso generales que capturan la funcionalidad del sistema, se identifican los diferentes tipos de personas o actores del mismo.

ACTOR	DESCRIPCIÓN
Administrador	Encargado de gestionar la información relativa a usuarios y proyectos.
Cliente	Persona interesada en información relacionada con proyectos.
Empleado	Encargado de ingresar parte de la información necesaria en el desarrollo del proyecto al que fue asignado.

Los casos de usos generales identificados, se presentan en la figura 1, a continuación se describen estos.

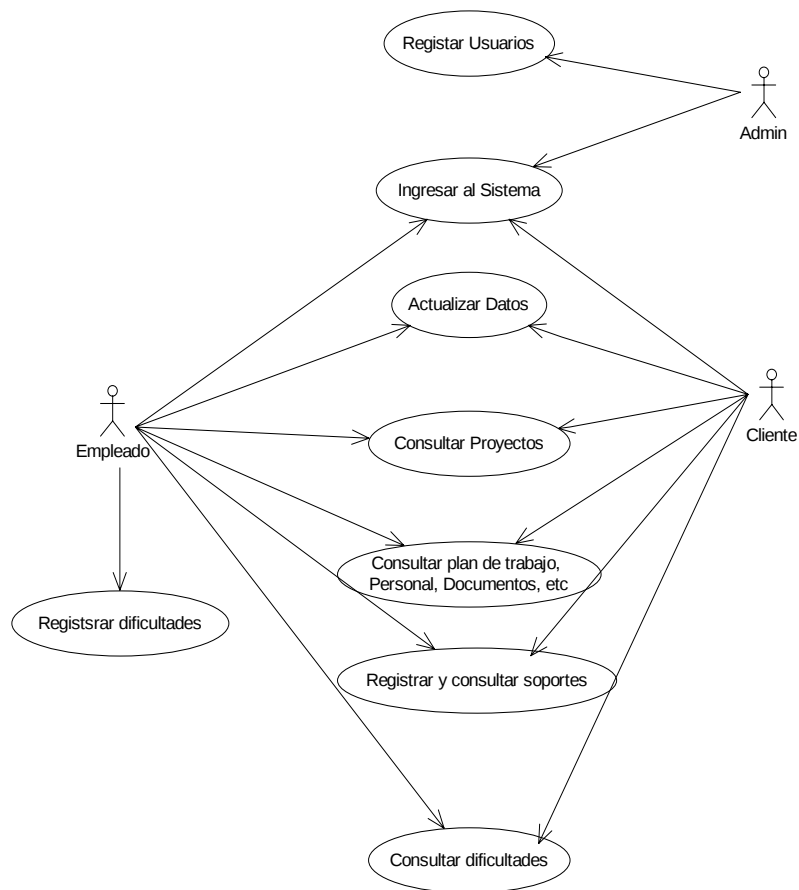


Figura 1. Diagrama de Casos de Usos Generales del Prototipo

INGRESAR AL SISTEMA		
OBJETIVO	Permitir que el usuario tenga acceso al sistema.	
ACCIONES	PASO	ACCIÓN
	1	Usuario Ingresa su nombre de usuario y contraseña de ingreso al sistema.
	2	Sistema Confronta la información con la existente en la base de datos y procede a permitir o negar la entrada al sistema.

ADMINISTRAR USUARIOS		
OBJETIVO	Permitir que el usuario gestione la información de usuarios.	
ACCIONES	PASO	ACCIÓN
	1	Usuario Selecciona opción de gestionar usuarios.
	2	Sistema Despliega formulario para gestión de usuarios.
	3	Usuario Ingresa la información y selecciona la opción
	4	Sistema Registra la información

ADMINISTRAR PROYECTOS		
OBJETIVO	Permitir que el usuario gestione la información de proyectos.	
ACCIONES	PASO	ACCIÓN
	1	Usuario Selecciona opción de gestionar proyectos.
	2	Sistema Despliega formulario para gestión de proyectos.
	3	Usuario Ingresa la información y selecciona la opción
	4	Sistema Registra la información

ADMINISTRAR DATOS DE PERSONAL		
OBJETIVO	Permitir que el usuario gestione la información de usuarios.	
ACCIONES	PASO	ACCIÓN
	1	Usuario Selecciona opción de gestionar datos de personal.
	2	Sistema Despliega formulario para gestión de información de personal.
	3	Usuario Ingresa la información y selecciona la opción para enviar
	4	Sistema Registra la información

ADMINISTRAR CLIENTES		
OBJETIVO	Permitir que el usuario gestione la información de clientes.	
ACCIONES	PASO	ACCIÓN
	1	Usuario Selecciona opción de gestionar clientes.
	2	Sistema Despliega formulario para gestión de clientes.
	3	Usuario Ingresa la información y selecciona la opción
	4	Sistema Registra la información

GENERAR INFORMES		
OBJETIVO	Permitir que el usuario genere los informes necesarios de proyectos.	
ACCIONES	PASO	ACCIÓN
	1	Usuario Selecciona opción de generar informes.
	2	Sistema Despliega formulario para gestión de informes.
	3	Usuario Selecciona la información
	4	Sistema Registra la información

3. CASOS DE USOS ESPECÍFICOS

En esta sección se presenta la documentación de los casos de usos específicos, fue elaborada a la par con el desarrollo de cada una de las etapas del proyecto, y teniendo como base los casos que se identificaron anteriormente.

En la figura 2, podemos observar cinco paquetes: usuarios, proyectos, clientes, informes, personal de trabajo.

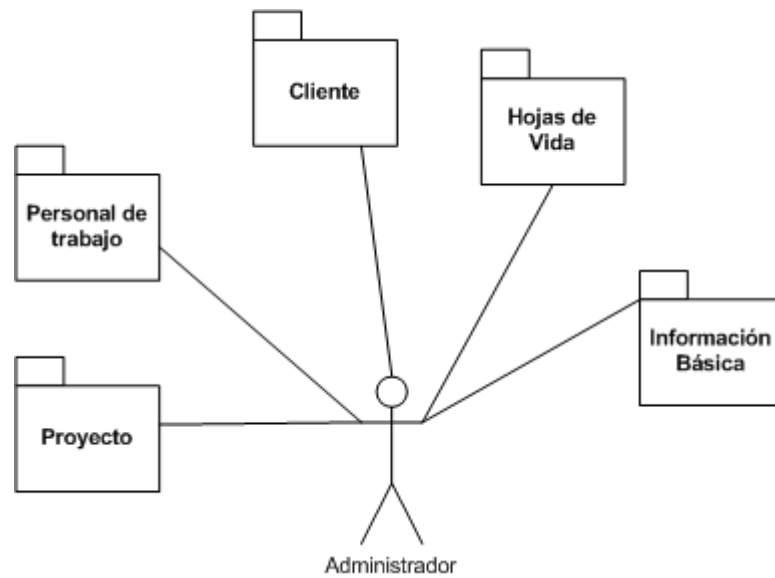


Figura 2. Casos de uso administrador del sistema organizados en paquetes

La gráfica 4, ilustra los casos del paquete usuarios

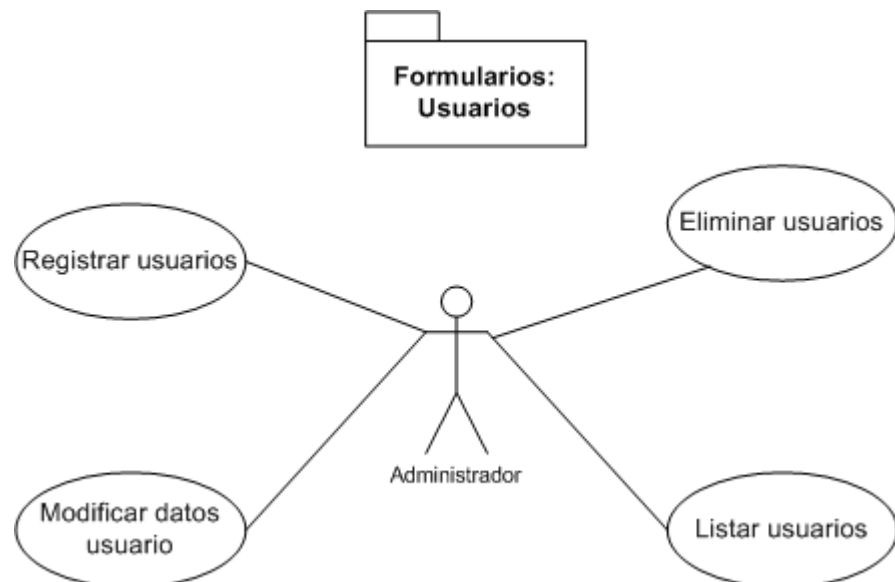


Figura 4. Casos de uso del paquete usuarios

REGISTRAR USUARIOS			
OBJETIVO	Permitir que el usuario registre nuevos usuarios en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Selecciona opción de registro para nuevos contactos.	
	2	Sistema Despliega formulario de ingreso de datos para nuevo contactos.	
	3	Usuario Ingresar los datos personales del nuevo usuario que son solicitados por el sistema y selecciona la operación ADICIONAR.	
	4	Sistema Registra del nuevo usuario en el sistema	
EXEPCIONES	PASO	EXECPCIÓN	ACCIÓN
	1	El usuario no ingresa correctamente los datos	Mostrar el mensaje con causa de error de ingreso.
	2	Ingresar caracteres no permitidos en la información solicitada.	Mostrar mensajes con las indicaciones adecuadas.
	3	La acción no es exitosa.	Mostrar mensaje con la causa de error.

LISTAR USUARIOS			
OBJETIVO	Permitir que el usuario liste los usuarios registrados en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Selecciona la opción para listar usuarios.	
	2	Sistema Despliega los usuarios registrados en el sistema.	
EXEPCIONES	PASO	EXECPCIÓN	ACCIÓN
	1	La acción no es exitosa	Mostrar el mensaje con causa de error.

MODIFICAR USUARIOS			
OBJETIVO	Permitir que el usuario modifique los datos de un usuario en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Hace petición de LISTAR USUARIOS.	
	2	Sistema Listar los usuarios registrados en el sistema.	
	3	Usuario Selecciona el usuario que quiere modificar.	
	4	Sistema Genera el formulario de actualización de datos del usuario con la información personal que se encuentra en el sistema.	
	5	Usuario Actualiza la información que desea y selecciona la opción Adicionar.	
	6	Sistema Registra los nuevos datos del usuario en el sistema.	
EXEPCIONES	PASO	EXECPCIÓN	ACCIÓN
	1	Actualizar campos que son únicos con información que atente contra la integridad de los datos	Mostrar el mensaje con causa de error de ingreso.
	5	Ingresar caracteres no permitidos en la información solicitada.	Mostrar mensajes con las indicaciones adecuadas.
	2,4,6	La acción no es exitosa.	Mostrar mensaje con la causa de error.

ELIMINAR USUARIOS			
OBJETIVO	Permitir que el usuario elimine un usuario del sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Hace petición de LISTAR USUARIOS.	
	2	Sistema Listar los usuarios registrados en el sistema.	
	3	Usuario Selecciona la opción Eliminar para el usuario que desee suprimir del sistema.	
	4	Sistema Genera una ventana de información para ratificarle al usuario la acción que desea realizar.	
	5	Usuario Selecciona la opción de Aceptar.	
	6	Sistema Elimina el usuario del sistema. Presenta un mensaje de confirmación.	
EXEPCIONES	PASO	EXECPCIÓN	ACCIÓN
	2,6	La acción no es exitosa.	Mostrar mensaje con la causa de error.

En la figura 5, se ilustra el paquete proyectos junto con los casos de uso que lo conforman.

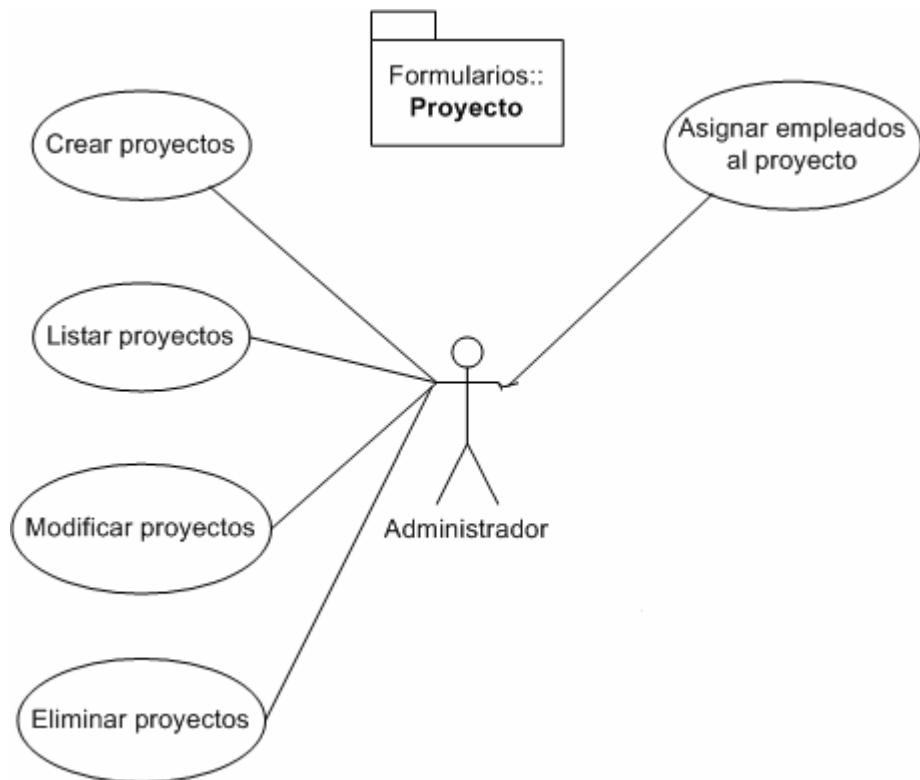


Figura 5. Caso de uso paquete Proyectos

La descripción de cada uno de los casos de uso del paquete, se consignan a continuación.

CREAR PROYECTO			
OBJETIVO	Permitir que el usuario cree un proyecto en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Selecciona la opción para CREAR proyectos.	
	2	Sistema Despliega formulario de ingreso de datos para la creación de proyectos.	
	3	Usuario Ingresa la información necesaria para la creación de proyectos y selecciona la opción Adicionar los datos.	
	4	Sistema Registra el nuevo proyecto en el sistema.	
	5	Usuario Actualiza la información que desea y selecciona la opción Adicionar.	
	6	Sistema Registra los nuevos datos del usuario en el sistema.	
EXEPCIONES	PASO	EXCEPCIÓN	ACCIÓN
	1	Ingresa información que rompa la integridad de los datos	Mostrar el mensaje con causa de error de ingreso.
	3	Ingresa caracteres no permitidos en la información solicitada.	Mostrar mensajes con las indicaciones adecuadas.
	2,4	La acción no es exitosa.	Mostrar mensaje con la causa de error.

LISTAR PROYECTOS			
OBJETIVO	Permitir que el usuario liste los proyectos del sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Selecciona la opción para listar proyectos.	
	2	Sistema Despliega los proyectos existentes del sistema.	
EXEPCIONES	PASO	EXCEPCIÓN	ACCIÓN
	2	La acción no es exitosa.	Mostrar mensaje con la causa de error.

MODIFICAR DATOS PROYECTO			
OBJETIVO	Permitir que el usuario modifique un proyecto en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Hace la petición de listar proyectos.	
	2	Sistema Lista los proyectos.	
	3	Usuario Selecciona el proyecto que desea modificar.	
	4	Sistema Genera el formulario de actualización.	
	5	Usuario Actualiza la información que desea y selecciona la opción Adicionar.	
	6	Sistema Registra los nuevos datos del usuario en el sistema.	
EXEPCIONES	PASO	EXCEPCIÓN	ACCIÓN
	3	Ingresar información que rompa la integridad de los datos	Mostrar el mensaje con causa de error de ingreso.
	3	Ingresar caracteres no permitidos en la información solicitada.	Mostrar mensajes con las indicaciones adecuadas.
	2,4	La acción no es exitosa.	Mostrar mensaje con la causa de error.

ELIMINAR PROYECTO			
OBJETIVO	Permitir que el usuario ELIMINE un proyecto en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Selecciona la opción para ELIMINAR proyectos.	
	2	Sistema Genera una ventana de confirmación.	
	3	Usuario Selecciona la opción de aceptar.	
	4	Sistema Elimina del sistema el registro del proyecto seleccionado.	
EXEPCIONES	PASO	EXCEPCIÓN	ACCIÓN
	2,4	La acción no es exitosa.	Mostrar mensaje con la causa de error.

ASIGNAR USUARIOS AL PROYECTO			
OBJETIVO	Permitir que el usuario ASIGNE usuarios a un proyecto registrado en el sistema.		
ACCIONES	PASO	ACCIÓN	
	1	Usuario Hace la petición de listar proyectos.	
	2	Sistema Lista los proyectos registrados en el sistema.	
	3	Usuario Selecciona el proyecto que desea asignar.	
	4	Sistema Lista los usuarios registrados en el sistema.	
	5	Usuario Selecciona el usuario que desee asignar al proyecto y la función a desarrollar.	
	6	Sistema Registra al usuario al proyecto seleccionado.	
EXEPCIONES	PASO	EXCEPCIÓN	ACCIÓN
	2,4,6	La acción no es exitosa.	Mostrar mensaje con la causa de error.

REFERENCIAS

[1] Ingeniería del Software. Un enfoque práctico. GRECH, Pablo. Prentice Hall, Bogotá, 2001. Capítulo 11

ANEXO B
DICCIONARIO DE DATOS

DICCIONARIO DE DATOS

1. INTRODUCCIÓN

Este anexo describe brevemente una parte del diccionario de datos correspondiente al prototipo, buscando una mejor comprensión de los datos que hacen parte del sistema y contribuya a facilitar la implementación del sistema.

El diccionario de datos del prototipo está orientado hacia la descripción de los objetos de datos¹⁸ del sistema y atributos.

¹⁸ Un objeto de datos es una representación de cualquier composición de información compuesta que debe comprender el software.

2. ESTRUCTURA DE LAS TABLAS

Tabla 'ACTIVIDAD_BITACORA'

Nombre tabla	ACTIVIDAD_BITACORA
Objeto	Almacena toda la información correspondiente a las actividades que puede presentar una Bitácora.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	acb_codigo	NUMERIC (18,0)	SI	Llave primaria de la Actividad Bitácora
	acb_nombre	STR (100)	SI	Nombre de la Actividad de la Bitácora

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_ACTIVIDAD_BITACORA_BITACORA_ACTIVIDAD	ACTIVIDAD_BITACORA	BITACORA_ACTIVIDAD	1:N
FK_BITACORA_PROYECTO_BITACORA_ACTIVIDAD	ACTIVIDAD_BITACORA	BITACORA_PROYECTO	1:N
FK_BITACORA_SOPORTE_BITACORA_ACTIVIDAD	ACTIVIDAD_BITACORA	BITACORA_SOPORTE	1:N

Tabla 'ACTIVIDADES_PLAN'

Nombre tabla	ACTIVIDADES_PLAN
Objeto	Almacenar toda la información correspondiente a las actividades realizadas durante una fase o etapa del proyecto.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	acp_codigo	NUMERIC (18,0)	SI	Llave primaria.
	acp_nombre	STR (100)	SI	Nombre de la actividad por plan.
FK	fas_codigo	NUMERIC (18,0)	SI	Llave de la fase del proyecto.

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_ACTIVXPLANXPROYECTO_ACTIVIDADES_PLAN	ACTIVIDADES_PLAN	ACTIVIDAD_PLAN_PROYECTO	1:N
FK_BITACORA_ACTIVIDAD_ACTIVIDADES_PLAN	ACTIVIDADES_PLAN	BITACORA_ACTIVIDAD	1:N
FK_TIEMPO_PROYECTO_ACTIVIDADES_PLAN	ACTIVIDADES_PLAN	TIEMPO_PROYECTO	1:N
FK_ACTIVIDADES_PLAN_FASE_PROYECTOS	FASE_PROYECTO	ACTIVIDADES_PLAN	1:N

Tabla 'ACTIVIDAD_PLAN_PROYECTO'

Nombre tabla	ACTIVIDAD_PLAN_PROYECTO
Objeto	Registrar toda la información correspondiente a las actividades realizadas en el proyecto.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	acpl_codigo	NUMERIC (18,0)	SI	Llave primaria
FK	plt_codigo	NUMERIC (18,0)	SI	Llave foránea
FK	acp_codigo	NUMERIC (18,0)	SI	Llave foránea
	acpl_fecha_inicio	DBTIMESTAMP	SI	Fecha inicio de la actividad por plan por proyecto.
	acpl_fecha_fin	DBTIMESTAMP	SI	Fecha final de la actividad por plan por proyecto.
	acpl_descripcion	DBTIMESTAMP	SI	Observación de la actividad.
	acpl_fecha_inicio_real	DBTIMESTAMP	SI	Fecha inicio real de la actividad.
	acpl_fecha_fin_real	DBTIMESTAMP	SI	Fecha fin real de la actividad.
	acpl_prioridad	NUMERIC (18,0)	SI	Prioridad de la actividad.
FK	esa_codigo	NUMERIC (18,0)	SI	Estado de la actividad.

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_ACTIVIDA_PLAN_PROYECTO_ACTIVIDADES_PLAN	ACTIVIDADES_PLAN	ACTIVIDAD_PLAN_PROYECTO	1:N
FK_ACTIVIDAD_PLAN_PROYECTO_ESTADOS_ACTIVIDADES	ESTADOS_ACTIVIDADES	ACTIVIDAD_PLAN_PROYECTO	1:N
FK_ACTIVIDAD_PLAN_PROYECTO_PLAN_TRABAJO	PLAN_TRABAJO	ACTIVIDAD_PLAN_PROYECTO	1:N

Tabla 'CARGO_HV_PROYECTO'

Nombre tabla	CARGO_HV_PROYECTO
Objeto	Almacenar toda la información correspondiente a los cargos ejercidos de los participantes durante un proyecto.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	chp_codigo	NUMERIC (18,0)	SI	Llave primaria
FK	pro_codigo	NUMERIC (10,0)	SI	Llave foránea
FK	hov_codigo	NUMERIC (18,0)	SI	Llave foránea
FK	fun_codigo	NUMERIC (18,0)	SI	Llave foránea

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_CARGO_HV_PROYECTO_FUNCION_PROYECTO	FUNCION_PROYECTO	CARGO_HV_PROYECTO	1:N
FK_CARGO_HV_PROYECTO_HOJAS_DE_VIDA	HOJAS_DE_VIDA	CARGO_HV_PROYECTO	1:N
FK_CARGO_HV_PROYECTO_PROYECTOS	PROYECTOS	CARGO_HV_PROYECTO	1:N

Tabla 'CLIENTES'

Nombre tabla	CLIENTES
Objeto	Almacenar toda la información correspondiente a los clientes de la empresa.

Columnas

Cla ve	Nombre columna	Tipo dato	NN	Descripción
PK	cli_codigo	NUMERIC (18,0)	SI	Llave Primaria
	cli_nombre	STR (100)	SI	Nombre del cliente
	cli_nit	NUMERIC (18,0)	SI	Nit o cédula del cliente
	cli_url	NUMERIC (18,0)	SI	Página web o email
	cli_direccion	STR (100)	NO	Dirección de domicilio del cliente
	cli_telefono	STR (15)	NO	Teléfono del cliente
	cli_fax	STR (15)	NO	Fax

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_CONTACTOS_CLIENTE_CLIENTES	CLIENTES	CONTACTOS_CLIENTES	1:N
FK_PROPUUESTAS_CLIENTES	CLIENTES	PROPUESTAS	1:N
FK_PROYECTOS_CLIENTES	CLIENTES	PROYECTOS	1:N
FK_USUARIOS_CLIENTES	CLIENTES	USUARIOS	1:N

Tabla 'CONTACTO_CLIENTES'

Nombre tabla	CONTACTO_CLIENTES
Objeto	Almacenar toda la información correspondiente a los contactos de la empresa.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	Conc_codigo	Numeric (18,0)	SI	Llave principal
	Conc_nombres	VarChar (100)	SI	Nombres del contacto
	Conc_apellidos	VarChar (100)	NO	Apellidos del contacto
	Conc_telefono	VarChar (50)	NO	Telefono
	Conc_dependencia	VarChar (50)	NO	Dependencia en la que labora.
FK	Cli_codigo	Numeric (18,0)	SI	Código del cliente
	Conc_email	VarChar (50)	NO	Email contacto

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_CONTACTO_CLIENTES	ACTIVIDADES_PLAN	CONTACTOS_CLIENTE	1:N
FK_USUARIOS_CONTACTOS_CLIENTE	CONTACTOS_CLIENTE	USUARIOS	1:N
FK_PROYECTOS_CONTACTOS_CLIENTE	CONTACTOS_CLIENTE	PROYECTOS	1:N

Tabla 'DIFICULTADES_SOPORTES'

Nombre tabla	DIFICULTADES_SOPORTES
Objeto	Almacenar toda la información correspondiente a las dificultades presentadas en el momento de realizar un soporte específico.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	difs_codigo	NUMERIC (18,0)	SI	Llave primaria
FK	dif_codigo	NUMERIC (18,0)	SI	Código de la dificultad
	Difs_observacion	STR (100)	SI	Observación de la dificultad por soporte
FK	sod_codigo	NUMERIC (18,0)	SI	Código del soporte

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card
FK_DIFICULTADES_SOPORTES_DIFICULTADES	DIFICULTADES	DIFICULTADES_SOPORTES	1:N
FK_DIFICULTADES_SOPORTES_SOPORTE_DETALLES	SOPORTE_DETALLES	DIFICULTADES_SOPORTES	1:N

Tabla 'EVENTOS_NOTICIAS'

Clave principal 'PK_EVENTOS'

Nombre columna	Tipo	Tabla padre
evn_codigo	PK	-----

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	evn_codigo	Numeric (18,0)	SI	Llave primaria
	evn_titulo	VarChar (100)	SI	Titulo del evento y/o noticia
	evn_descripcion	VarChar (2000)	SI	Descripción del evento y/o noticia
	evn_detalle	Text	NO	Detalle del evento y/o noticia
	evn_fecha	DateTime	SI	Fecha del evento y/o publicación de la noticia
	evn_duracion	DateTime	NO	Duración de la publicación
	evn_publicar	Char (1)	SI	Publicar S/N
	evn_activar	Char (1)	NO	Activar S/N
	evn_tipo	VarChar (50)	NO	Tipo de evento y/o noticia

Tabla 'HOJAS_DE_VIDA'

Nombre tabla	HOJAS_DE_VIDA
Objeto	Almacenar toda la información de la hoja de vida correspondientes a los participantes del proyecto.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	hov_codigo	NUMERIC (18,0)	SI	Llave primaria
	Hov_nombre	STR (50)	SI	Nombre del participante del proyecto
	Hov_apellidos	BYTES (50)	SI	Apellidos del participante del proyecto
	Hov_fecha_nacimiento	STR (50)	SI	Fecha de nacimiento
	Hov_lugar_nacimiento	STR (50)	SI	Lugar de nacimiento
	Hov_direccion	STR (50)	SI	Dirección
	Hov_telefono	STR (15)	SI	Teléfono
	Hov_celular	STR (15)	SI	Celular
	Hov_tarjeta_profesional	STR (50)	SI	No. Tarjeta profesional
	Hov_documento_identidad	STR (50)	SI	Documento de identidad
	Hov_fecha_ingreso	DBTIME STAMP	SI	Fecha de ingreso
	Pep_codigo	NUMERIC (18,0)	SI	Asignar a proyecto
FK	Esr_codigo	NUMERIC (18,0)	SI	Código de estudios realizados
	Hov_perfil	STR (8000)	SI	Perfil profesional

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_HOJAS_DE_VIDA_ESTUDIOS_REALIZADOS	ESTUDIOS_REALIZADOS	HOJAS_DE_VIDA	1:N
FK_CARGO_HV_PROYECTO_HOJAS_DE_VIDA	HOJAS_DE_VIDA	CARGO_HV_PROYECTOS	1:N
FK_SOPORTE_DETALLE_S_HOJAS_DE_VIDA	HOJAS_DE_VIDA	SOPORTE_DETALLES	1:N
FK_TIEMPO_PROYECTO_HOJAS_DE_VIDA	HOJAS_DE_VIDA	TIEMPO_PROYECTO	1:N
FK_USUARIOS_HOJAS_DE_VIDA	HOJAS_DE_VIDA	USUARIOS	1:N

Tabla 'PROPUESTAS'

Nombre tabla	PROPUESTAS
Objetivo	Almacenar toda la información correspondiente a las propuestas de un proyecto.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	Prp_codigo	NUMERIC (18,0)	SI	Llave primaria
	Prp_numero	STR (10)	SI	Número de la propuesta
	Prp_objeto	STR (250)	NO	Objetivo de la propuesta
	Prp_fecha_apertura	DBTIMESTAMP	SI	Fecha de apertura
	Prp_fecha_cierre	DBTIMESTAMP	SI	Fecha de cierre
	Prp_valor	NUMERIC (18,2)	SI	Valor de la propuesta
FK	Cli_codigo	NUMERIC (18,0)	SI	Código del cliente
FK	Eprpr_codigo	NUMERIC (18,0)	SI	Código del estado de la propuesta

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card
FK_PROPUESTAS_CLIENTES	CLIENTES	PROPUESTAS	1:N
FK_PROPUESTAS_ESTADOS_PROPUESTAS	ESTADOS_PROPUESTAS	PROPUESTAS	1:N
FK_BITACORA_PROPUESTAS_PROPUESTAS	PROPUESTAS	BITACORA_PROPUESTAS	1:N
KK_DOC_PROPUESTAS	PROPUESTAS	DOC_PROPUESTAS	1:N
FK_PROPUESTAS_PROYECTOS	PROPUESTAS	PROYECTOS	1:N

Tabla 'PROYECTOS'

Nombre tabla	PROYECTOS
Objetivo	Almacenar toda la información correspondiente a los proyectos.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	Pro_codigo	NUMERIC (10,0)	SI	Llave primaria
	Pro_numero	STR (20)	SI	Número del proyecto
	Pro_nombre	STR (200)	SI	Nombre del proyecto
	Pro_objeto	STR (400)	SI	Objetivo del proyecto
	Pro_url_desarrollo	STR (200)	NO	Dirección web de desarrollo del proyecto
	Pro_url_final	STR (200)	NO	Dirección electrónica final del proyecto
	Pro_fecha_inicial	DBTIMESTAMP	SI	Fecha inicial del proyecto
	Pro_fecha_fin	DBTIMESTAMP	SI	Fecha final del proyecto
	Pro_fecha_inicial_real	DBTIMESTAMP	SI	Fecha inicial real del proyecto
	Pro_fecha_fin_real	DBTIMESTAMP	SI	Fecha final real del proyecto
	Pro_publicar	STR (1)	SI	Proyecto publicado
FK	Cli_codigo	NUMERIC (18,0)	SI	Código del cliente
FK	Esp_codigo	NUMERIC (18,0)	SI	Código del estado del proyecto
FK	Prp_codigo	NUMERIC (18,0)	SI	Código de la propuesta

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_PROYECTOS_CLIENTES	CLIENTES	PROYECTOS	1:N
FK_ESTADOS_PROYECTOS_PROYECTOS	ESTADOS_PROYECTOS	PROYECTOS	1:N
FK_PROPUUESTAS_PROYECTOS	PROPUESTAS	PROYECTOS	1:N
FK_BITACORA_PROYECTO_PROYECTOS	PROYECTOS	BITACORA_PROYECTOS	1:N
FK_CARGO_HV_PROYECTO_PROYECTOS	PROYECTOS	CARGO_HV_PROYECTOS	1:N
FK_DOC_PROYECTOS_PROYECTOS	PROYECTOS	DOC_PROYECTOS	1:N
FK_PROYECTOS_PLAN_TRABAJO	PROYECTOS	PLAN_TRABAJO	1:N
FK_PROYECTOS_SOPORTES	PROYECTOS	SOPORTES	1:N
FK_TIEMPO_PROYECTO_PROYECTOS	PROYECTOS	TIEMPO_PROYECTO	1:N

Tabla 'SOPORTES'

Nombre tabla	SOPORTES
Objeto	Almacenar toda la información de soportes solicitados por el usuario.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	Sop_codigo	NUMERIC (18,0)	SI	Llave primaria
	Sop_descripcion	STR (1000)	SI	Descripción del soporte
FK	Pro_codigo	NUMERIC (10,0)	SI	Código del proyecto
FK	Usu_codigo	NUMERIC (18,0)	SI	Código del usuario que solicitó el proyecto
	Sop_fecha_solicitud	DBTIMESTAMP	SI	Fecha de la solicitud del soporte
FK	Tis_codigo	NUMERIC (18,0)	SI	Código del tipo de soporte
	Sop_fecha_soporte	DBTIMESTAMP	SI	Fecha del soporte

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_PROYECTOS_SOPORTES	PROYECTOS	SOPORTES	1:N
FK_SOPORTES_BITACORA_SOPORTE	SOPORTES	BITACORA_SOPORTE	1:N
FK_SOPORTES_DOC_SOPORTE	SOPORTES	DOC_SOPORTE	1:N
FK_SOPORTES_SOPORTE_DETALLES	SOPORTES	SOPORTE_DETALLES	1:N
FK_TIPO_SOPORTE_SOPORTES	TIPO_SOPORTE	SOPORTES	1:N
FK_USUARIOS_SOPORTES	USUARIOS	SOPORTES	1:N

Tabla 'USUARIOS'

Nombre tabla	USUARIOS
Objeto	Almacenar toda la información correspondiente a los usuarios del sistema.

Columnas

Clave	Nombre columna	Tipo dato	NN	Descripción
PK	Usu_codigo	NUMERIC (18,0)	SI	Llave primaria
	Usu_login	STR (20)	SI	Login de entrada al sistema
	Usu_clave	STR (20)	SI	Clave para acceder al sistema
	Usu_nombres	STR (50)	SI	Nombre de usuario
	Usu_apellidos	STR (50)	SI	Apellido de usuario
	Usu_telefonos	STR (15)	NO	Teléfono de usuario
	Usu_email	STR (20)	SI	Email usuario
	Usu_entradas	NUMERIC (18,0)	SI	Entradas del usuario al sistema
FK	Cli_codigo	NUMERIC (18,0)	NO	Código de cliente
FK	Hov_codigo	NUMERIC (18,0)	NO	Código de la hoja de vida
FK	Per_codigo	NUMERIC (18,0)	SI	Código del perfil asignado al usuario
FK	Esu_codigo	NUMERIC (18,0)	SI	Estado del usuario

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_CLIENTES_USUARIOS	CLIENTES	USUARIOS	1:N
FK_ESTADOS_USUARIOS_USUARIOS	ESTADOS_USUARIOS	USUARIOS	1:N
FK_HOJAS_DE_VIDA_USUARIOS	HOJAS_DE_VIDA	USUARIOS	1:N
FK_PERFIL_USUARIOS	PERFIL	USUARIOS	1:N
FK_USUARIOS_SOPORTE	USUARIOS	SOPORTES	1:N

Tabla 'MENUP_MENUPPAL'

Nombre tabla	MNUP_MENUPPAL
Objeto	Almacenar información correspondiente al cargue de menús y submenús de acuerdo al perfil del usuario.

Columnas

Cla ve	Nombre columna	Tipo dato	NN	Descripción
PK	MENUP_CODI GO	I4	SI	Llave primaria
	MENUP_LINK	WSTR (250)	SI	Link del menú
	MENU_DESCR IPCION	WSTR (250)	NO	Descripción del menú
	MENUP_NIVE L	I4	SI	Nivel del menú
	MENUP_ORD EN MENUP_ACTI VO	NUMERI C (18,0) CHAR(1)	SI SI	Orden del menú Menu activo o inactivo

Relaciones

Nombre de restricción	Tabla padre	Tabla hija	Card.
FK_OPCIONESXPERFIL_MN UP_PRINCIPAL	MNUP_MENUPPAL	EXP_OPCION ES_PERFIL	1:N

ANEXO C
ORIENTACIONES DE PROGRAMACION

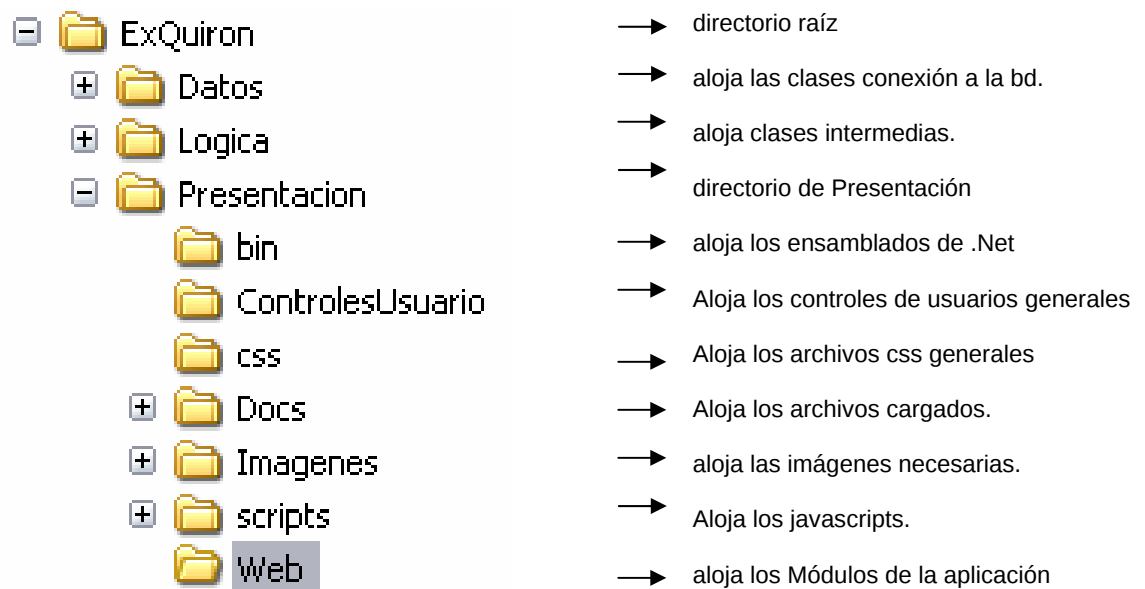
ORIENTACIONES DE PROGRAMACIÓN

3. INTRODUCCIÓN

Este anexo describe brevemente algunas consideraciones especiales a nivel de la programación de la Extranet para la empresa Quirón Software, con el objetivo de brindar al administrador del sistema y a futuros desarrolladores de la herramienta una ayuda para realizar el mantenimiento del sistema y adicionar funcionalidades, conservando estándares en su estructura y código.

4. ORGANIZACIÓN DEL SISTEMA EN EL SERVIDOR

Espacio disponible en el sistema para la instalación de la aplicación
aproximadamente 5 Mb.



5. CONVENCIONES DE PROGRAMACIÓN

3.1. Nomenclatura

- Las tablas en la base de datos se nombran de manera plural a la entidad, o el nombre compuesto por la abreviatura de las entidades registradas en las tablas, en caso de que la tabla sea una relación.
- Los campos que corresponden a identificadores de registros (llave primaria en la tabla) utilizan el formato Prefijo de la tabla_ código, en donde el prefijo del nombre de la entidad corresponden a las 3 primeras letras de ésta, o a las 4 primeras si es el caso de una tabla de relación; los nombres de los demás campos llevarán este prefijo y la descripción del dato que almacenan y en caso de que se necesite describirlo como un nombre compuesto iniciarán en minúscula y las palabras serán separadas entre sí con _. (ej: pro_nombre_proyecto).
- Los nombres de los procedimientos almacenados llevan el prefijo PA_ seguido del nombre de la tabla y un prefijo descriptivo si el procedimiento almacenado corresponde a consulta, actualización, ingreso o eliminación. (ej: PA_PROYECTOS_CONSULTAR).
- Los nombres de las vistas llevan el nombre de la tabla anteceditos por el prefijo V_.

6. Modularización del código

Adicionalmente a la modularización sugerida por el sistema de archivos Extranet se debe tener en cuenta:

- Se implementa el uso de funciones para reducir el código redundante, colocándolas ya sea en una clase aparte o en el mismo formulario si no se van a reutilizar. Lo mismo aplica para constantes y variables.
- Los nombres de las clases a utilizar en la aplicación llevan el mismo nombre de la entidad a la que hacen relación.
- De igual manera, los nombres de los formularios llevan el nombre de la entidad a la que corresponden, seguido de Re ó Gr, según el formulario sea de consulta o modificación e ingreso de información.
- Los nombres de las funciones que realizan consultas a la base de datos empiezan con el prefijo procesar, seguido de la operación a realizar (Consulta, Inserción, Modificación, Eliminación).

- Los scripts de funciones de calendario js se ubican en el directorio calendar, con nombres descriptivos a la aplicación a la que pertenecen.
- Los scripts de estilos css utilizados en el sistema se encuentran ubicados en el directorio, con nombres descriptivos del módulo o aplicación al que pertenecen css.

4.1. Otras convenciones de programación

- El código debe documentarse a lo largo de su programación, de manera que se dejen claro las consideraciones y procedimientos importantes en la tarea o el proceso que implementan.
- El envío de datos entre formularios se envían a través del método POST, que permite ocultar el contenido de las variables, y proporciona mayor capacidad para el envío de información ó variables de sesión según sea conveniente,

7. PROGRAMACIÓN DE MÓDULOS Y APLICACIONES

5.1. Orientaciones generales

Como mecanismo de ayuda rápida para orientar la entrada de datos y la realización de algunas operaciones, se proporcionan textos de ayuda cortos escritos en los formularios y visualizados de manera tooltip. Estos textos, van a nivel de formularios y proporcionan orientaciones sobre el contenido y uso de la información que se pide o se suministra.

5.2. Gestión de Sesiones

Autenticación de usuarios en el sistema. Cuando un usuario ingresa a la Extranet de la empresa, el sistema establece una variable de sesión que identifica el tipo de usuario en sesión durante toda su permanencia en la aplicación. Cuando un usuario se identifica ante el sistema, este verifica que los datos correspondan a los almacenados en la base de datos. En caso de error en la identificación, el sistema muestra el mensaje correspondiente, en caso contrario, se establece otra variable de sesión en la que se mantiene la identificación de ese miembro en la red.

8. Seguridad

A nivel de seguridad se usa un modelo sencillo de seguridad basado en perfiles de usuario. Estos permisos son los que se dan por defecto en la aplicación y pueden ser modificados desde la opción de Configuración, Información Básica.

6.1. Administrador

Tendrá acceso a todas las pantallas de la aplicación y podrá realizar cualquier acción en las mismas.

6.2. Responsable de proyecto

Solo verán aquellos proyectos de los cuales es responsable. Desde esos proyectos hacia abajo podrán realizar cualquier acción. Podrán **ver** el listado de proyectos, pero solo editar aquellos proyectos de los cuales sea responsable directo.

No podrán **aprobar** nuevas propuestas (esto solo lo podrán hacer los administradores), ni desde el árbol, ni desde el detalle de grupos de proyectos, ni desde el listado de grupos de proyectos.

Si podrán **editar** la información de los proyectos de los que sean responsables para modificar el estado, el grado de avance, etc.

6.3. Responsable de actividad/ Usuario General

Solo verán aquellos proyectos en los cuales existan actividades de las cuales sean responsables y solo tendrán permisos a partir del nivel del cual son responsables hacia abajo. Desde esas actividades hacia abajo podrán realizar cualquier acción. Podrán **editar** la información de las tareas de las que sean responsables para modificar el estado, el grado de avance, etc.

No podrán **borrar** sus tareas (esto solo lo podrán hacer los administradores y los responsables de proyecto).

6.4. Cliente

Solo verán aquellos proyectos en los cuales el sea responsable de la empresa cliente y solo tendrán permisos a partir del nivel del cual son responsables hacia abajo. Podrán **crear** nuevas propuestas, solicitudes de soportes técnicos, etc.

9. Mensajes del Sistema

A lo largo de la aplicación se utiliza dos métodos para proporcionar al usuario información sobre el resultado de sus acciones: alertas de javascript, y mensajes de error en las mismas páginas que se quieren enviar para proceso.

7.1. Alertas de Javascript

Este tipo de mensaje se utiliza principalmente en la validación de operaciones de los formularios del lado del cliente, y consiste en arrojar un cuadro de diálogo con el mensaje y un botón aceptar, utilizando para ello el método `alert()` del objeto `window javascript`. De igual manera, se hace uso del método `confirm()` (del mismo obeto javascript), que abre una caja de diálogo con los botones aceptar y cancelar para que el usuario decida sobre efectos importantes de sus acciones, como son la modificación y eliminación de información.

7.2. Resultados de operación en la misma página de proceso

Este tipo de mensaje se utiliza principalmente en la validación y como respuesta a los procesos de los diferentes formularios de la aplicación, cuando los resultados ameritan ser informados al usuario, y pueden ser fácilmente corregidos por él, razón por la cual se conserva los datos ingresados en el formulario que se desea enviar. Cuando se utiliza este tipo de mensajes, el error aparece escrito con color rojo al final del formulario con los campos a corregir por el usuario.

ANEXO D

**GUIA RAPIDA DE LA HERRAMIENTA EXTRANET EN SU
ESTADO FINAL**

GENERALIDADES

INTRODUCCIÓN

En este anexo se describe de manera general la usabilidad y funcionalidad de la herramienta en su estado final.

Durante muchos años el proceso de desarrollo de software ha sido considerado como un arte dejado a la improvisación del jefe del proyecto. Los proyectos se dirigían más bajo consideraciones técnicas, que de gestión. Las actividades de estimación y de planificación quedaban relegadas a un mero acto protocolario al comienzo del proyecto. Posteriormente, el seguimiento y control se realizaban sin un mínimo de rigor, dada la baja calidad de la estimación y la planificación realizada.

Esta herramienta Extranet ayuda a monitorizar proyectos, actividades, personas, asignaciones de personas, documentación de proyecto; a través de la Web, independientemente de cuáles sean las actividades y lugares en los que se desarrollen las actividades, logrando una visión permanentemente

actualizada de la situación de los procesos del negocio; y proporcionando una eficiente organización de proyectos y actividades, reduciendo drásticamente los costos y el tiempo invertido en sacar adelante los proyectos de la empresa.

Ventajas y características de la herramienta

Esta herramienta presenta las siguientes características:

- Planifica tiempos, costos, recursos humanos, esfuerzos, trabajos...
- Controla proyectos y actividades. Gestiona avances, esfuerzos, recursos (personas), e informes que se necesita sobre la gestión de proyectos.
- Ofrece la información que se necesita sobre cada elemento. Toda la información que aporta valor a los proyectos y actividades.
- Gestiona las actividades fácil y eficazmente. Planifica los recursos y esfuerzos de cada actividad en la gestión de proyectos.
- Controla los cambios que se producen en los proyectos.
- Modifica fácilmente los recursos y esfuerzos de los proyectos y actividades. Actualiza la información de las actividades, reasígnalas y controla su avance con el tiempo, o al ir imputándose las horas de esfuerzo mediante los partes horarios.

Ofrece importantes ventajas:

- Es online. Es el software de gestión de proyectos perfecto para entornos distribuidos, en los que varias personas necesitan acceder a la información en tiempo real.
- Es flexible. Se adapta perfectamente a la empresa, independientemente de la localización geográfica, permitiendo que las personas desde diferentes lugares puedan trabajar juntos.
- Es multiplataforma. Tan sólo se necesita una conexión a Internet e Internet Explorer 6.0 o superior como navegador Web para utilizar la herramienta Extranet.

Módulos de la Herramienta

Al ingresar a la aplicación, aparecen una serie de diferentes módulos que ayudarán a realizar las diferentes acciones:

1. Eventos y Noticias. Gestiona y controla los eventos y noticias de interés.

2. Recursos. Es el módulo que permite controlar la información relevante de aquellos usuarios de la aplicación, diferenciado usuarios internos de los externos a la empresa.

3. Proyectos. Es el módulo principal y permite planificar y controlar los proyectos y actividades al máximo, imputar horas, organizar el trabajo en equipo.

4. Soportes. Gestiona y controla las solicitudes de soporte de los clientes.

5. Configuración. Es el módulo que permite configurar las opciones base de la aplicación, así como la seguridad necesaria para ésta.

6. Informes. Genera informes de seguimiento para poder presentar fácilmente la información detallada de un proyecto.



Nombre Cliente

Fecha Inicio entre y Fecha Fin entre y

[Buscar](#) [Limpiar](#)

Resultado búsqueda proyectos

Numero Proyecto	Nombre del proyecto	Cliente	Director Proyecto	% Avance	Plan de Trabajo	Bitácora	Eliminar
B2007-00001	proyecto 0001	Laboratorios Velez	Fredy Chía Peña				
B2007-00003	proyecto de quiron	Laboratorios Velez	Fredy Chía Peña				
D2007-00006	Octubre 13/-31	Crow	Raúl Zambrano				

[Adicionar](#)

Resultados Cliente

Nombre	Eliminar
Laboratorios Velez	
Cabletec	
Corpoestetica	
Emigdio Martinez	
Presidencia de la republica	
Ministerio de Minas	
Crow	
Departamento Administrativo de Bienestar Social	
Empresa de Energía de Bogotá	
Metrovivienda	

[Adicionar](#)

Filtro de búsqueda ppersonal

Nombre empleado

[Buscar](#) [Limpiar](#)

Resultado búsqueda de Personal

Empleado	Ocupación en el periodo	Horas pendientes por planificar	Horas disponible por calendario	Eliminar
Fredy Chía Peña	80%	4	8	
Olga Lucía Fernández Pulido	100%	2	10	

[Atrás](#) [Adicionar](#)