

**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB
QUE PERMITA RESOLVER PROBLEMAS DE FORMA
CONJUNTA EN LAS ORGANIZACIONES**

WALTER SERRANO ZAMBRANO

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

2008

**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB
QUE PERMITA RESOLVER PROBLEMAS DE FORMA
CONJUNTA EN LAS ORGANIZACIONES**

WALTER SERRANO ZAMBRANO

**Proyecto para optar el Título de
Ingeniero de Sistemas**

Director

**FERNANDO ROJAS MORALES
Ingeniero de Sistemas**

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

2008

AGRADECIMIENTOS

Quiero dar mis más sinceros agradecimientos:

A Dios quien me ha regalado una familia que en todo momento ha confiado en mí brindándome siempre ese apoyo incondicional y necesario para afrontar de la mejor manera las adversidades de la vida.

Al Ingeniero Fernando Rojas, director del proyecto quien me aportó los mejores consejos y recomendaciones para llevar este proyecto a feliz término.

A mis amigos y compañeros, que siempre estuvieron ahí dispuestos a escucharme para brindarme apoyo cuando más lo necesité.

DEDICATORIA

A *Mi madre por ser mi apoyo
incondicional cuando más la necesito.*

TABLA DE CONTENIDO

INTRODUCCION.....	11
1.PRESENTACION.....	13
1.1 OBJETIVOS.....	14
1.1.1 OBJETIVO GENERAL.....	14
1.1.2 OBJETIVOS ESPECIFICOS.....	14
1.2 JUSTIFICACIÓN.....	15
2. ANTECEDENTES.....	17
2.1 Sistema de Información Basado en Problemas.....	17
2.2 Lenguaje de Representación de Decisiones (drl).....	17
2.3 Preguntas Opciones y Criterios (qoc).....	18
3. ESTADO DEL ARTE.....	19
3.1 Acollab	19
3.2 dotProject.....	20
3.3 Streber	20
4. MARCO TEORICO.....	22
4.1 Introducción.....	22
4.2 Sistema.....	23
4.3 Ingeniería de Sistemas.....	24
4.4 Sistemas Colaborativos.....	24
4.5 Fundamentación de la Información del SSD.....	25
4.6 Componentes.....	27
4.6.1 Hardware	28
4.6.2Software.....	28
4.6.3 Recursos humanos.....	29
4.7Beneficios	29
4.8 Tipos de Usuario del Sistema.....	30
4.9 DISEÑO DE LA INTERFASE DE USUARIO.....	30

4.9.1	Diseño gráfico.....	31
4.9.2	Diagramación.....	31
4.9.3	Elementos de diagramación.....	32
4.9.4	Tipografía.....	32
4.9.5	Color.....	33
4.9.5	Contraste.....	33
4.9.6	Composición.....	34
4.9.7	Jerarquización.....	34
4.9.8	Proporción.....	35
4.9.11	Ritmo.....	36
4.9.12	Contragrafismo.....	37
4.9.13	Formatos.....	37
4.9.14	Tamaños de pantalla.....	37
4.9.15	Botones.....	38
4.9.16	Diseño centrado en el usuario (DCU).....	38
4.9.18	Arquitectura de la información.....	39
4.9.19	Accesibilidad.....	39
4.9.20	Usabilidad.....	39
4.9.21	Estándares internacionales.....	40
4.9.22	Métodos de usabilidad.....	40
4.9.23	Evaluación heurística	40
4.9.24	Test de usabilidad	41
4.9.25	Tipos de test.....	41
4.9.26	Principios de la usabilidad.....	42
4.10	Bases de Datos.....	43
4.10.1	Base de datos relacionales.....	43
4.10.2	Diseño de las bases de datos relacionales.....	44
5.	METODOLOGÍA.....	46
5.1	Etaa 1 – Desarrollar un Modelo General.....	47
5.2	Etaa 2 – Hacer una lista de Rasgos.....	47
5.3	Etaa 3 - Desarrollo de un Plan Detallado.....	47
5.4	Etaa 4 – Diseño por Rasgo.....	47
5.5	Etaa 5 – Desarrollo por Rasgo.....	48
6.1	Aplicaciones Web	49
6.2	Tecnologías de Servidor	51
6.2.1	ColdFusion.....	51
6.2.2	Jsp.....	51
6.2.3	Asp.....	51
6.2.4	Perl.....	52
6.2.5	PHP.....	52
6.2.6	Por qué se utiliza PHP [7].....	52
6.3	Comparativa de Gestores de Bases de Datos.....	54
6.3.1	PostgreSQL.....	54
6.3.2	Oracle.....	54

6.3.3 MySQL.....	54
6.3.4 Porque se escogió MySQL.....	55
6.4 Comparativa de Tecnologías de Desarrollo Web para aplicaciones Interactivas.....	55
6.4.1 AJAX (Asynchronous JavaScript And XML) [8].....	55
6.4.2 Adobe Flash [9].....	57
6.4.3 XML (eXtensible Markup Language).....	58
6.4.4 CSS.....	62
6.4.6 Porqué se decide usar Backbase (Motor AJAX).....	64
7. DIAGRAMAS UML.....	67
7.1 Diagrama de casos de uso.....	67
7.1.1 Diagrama de Contexto.....	69
7.1.2 Diagrama de Casos de Uso - Nivel General.....	70
7.1.3 Especificación de Casos de Uso de Nivel General:.....	71
7.1.4 Caso de Uso: Gestionar Proyectos.....	73
7.1.5 Caso de Uso: Ver Aportes.....	73
7.1.6 Caso de Uso: Administrar.....	74
7.1.7 Caso de Uso: Gestionar Aportes.....	74
7.2 DIAGRAMA DE CLASES: Definición.....	75
7.2.1 Diagrama de Clases.....	78
7.2.2 Diccionario de Clases.....	78
7.3 Diagrama Entidad – Relación: Definición.....	80
7.3.1 Diagrama Entidad Relación.....	81
7.4 Diccionario de Datos: Definición.....	82
7.4.1 Diccionario de Datos.....	82
7.4 Diagrama de Actividades: Definición.....	85
7.4.1 Diagrama de Actividades	86
7.5 Diagrama de Navegación: Definición.....	93
7.5.1 Diagrama de Navegación.....	94
8. INTERFACES DEL SISTEMA.....	102
CONCLUSIONES.....	120
BIBLIOGRAFÍA.....	121

TABLA DE FIGURAS

FIGURA 1. PANTALLA ACOLLAB.....	19
FIGURA 2. PANTALLA DOTPROJECT.....	20
FIGURA 3. PANTALLA STREBER.....	21
FIGURA 4. ADMINISTRACIÓN DE LA FUNDAMENTACIÓN.....	27
FIGURA 5. COMPONENTES DEL SISTEMA.....	29
FIGURA 6. BENEFICIOS DEL SISTEMA.....	30
FIGURA 7. INTERRELACIÓN ENTRE LOS TIPOS DE TEST Y LAS FASES DE DESARROLLO DE UN PRODUCTO.....	42
FIGURA 8. METODOLOGÍA.....	46
FIGURA 9. MODELO DE COMUNICACIÓN.....	56
FIGURA 10. HOJA DE ESTILOS CSS.....	62
FIGURA 11. EL FRAMEWORK DE BACKBASE.....	66
FIGURA 12. ACTOR Y CASOS DE USO.....	68
FIGURA 13. CONEXIÓN ENTRE ACTORES Y CASOS DE USO.....	68
FIGURA 14. CONEXIÓN ENTRE CASOS DE USO.....	68
FIGURA 15. DIAGRAMA DE CONTEXTO.....	69
FIGURA 16. DIAGRAMA CASOS DE USO – NIVEL GENERAL.....	70
FIGURA 17. CASO DE USO: GESTIONAR PROYECTOS.....	73
FIGURA 18. CASO DE USO: VER APORTES.....	73
FIGURA 19. CASO DE USO: ADMINISTRAR.....	74

FIGURA 20. CASO DE USO: GESTIONAR APORTES.....	74
FIGURA 21. DIAGRAMA DE CLASES.....	78
FIGURA 22. DIAGRAMA ENTIDAD-RELACIÓN.....	81
FIGURA 23. DIAGRAMA DE ACTIVIDADES: INICIO.....	87
FIGURA 24. DIAGRAMA DE ACTIVIDADES: MENU PRINCIPAL.....	88
FIGURA 25. DIAGRAMA DE ACTIVIDADES: VER PROYECTO.....	88
FIGURA 26. DIAGRAMA DE ACTIVIDADES: VER PROBLEMA.....	89
FIGURA 27. DIAGRAMA DE ACTIVIDADES: CREAR PROBLEMA ..	89
FIGURA 28. DIAGRAMA DE ACTIVIDADES: MODIFICAR PERFIL.....	90
FIGURA 29. DIAGRAMA DE ACTIVIDADES: VER ALTERNATIVAS.....	91
FIGURA 30. DIAGRAMA DE ACTIVIDADES: VER GLOSARIO.....	91
FIGURA 31. DIAGRAMA DE ACTIVIDADES: ADMINISTRACIÓN.....	92
FIGURA 32. DIAGRAMA DE ACTIVIDADES: BÚSQUEDA.....	92
FIGURA 33. DIAGRAMA DE ACTIVIDADES: E-MAILER.....	93
FIGURA 34. DIAGRAMA DE ACTIVIDADES: CERRAR SESIÓN.....	93
FIGURA 35. DIAGRAMA DE NAVEGACIÓN – INICIO.....	94
FIGURA 36. DIAGRAMA DE NAVEGACIÓN - BÚSQUEDA.....	95
FIGURA 37. DIAGRAMA DE NAVEGACIÓN – ENVIAR E-MAIL.....	95
FIGURA 38 .DIAGRAMA DE NAVEGACIÓN - ADMINISTRACIÓN.....	96
FIGURA 39. DIAGRAMA DE NAVEGACIÓN - ESTADÍSTICAS.....	97

FIGURA 40. DIAGRAMA DE NAVEGACIÓN – VER PROBLEMA.....	98
FIGURA 41. DIAGRAMA DE NAVEGACIÓN – VER PROYECTO.....	99
FIGURA 42. DIAGRAMA DE NAVEGACIÓN – NUEVO PROYECTO	100
FIGURA 43. DIAGRAMA DE NAVEGACIÓN – ALTERNATIVAS.....	101
FIGURA 44. PANTALLA DE BIENVENIDA.....	102
FIGURA 45. FORMULARIO DE INGRESO AL SISTEMA.....	103
FIGURA 46. FORMULARIO DE REGISTRO.....	104
FIGURA 47. MENÚ PRINCIPAL.....	104
FIGURA 48. PANTALLA INICIO.....	105
FIGURA 49. FORMULARIO EDITAR PERFIL DE USUARIO.....	106
FIGURA 50. FORMULARIO CREAR NUEVO PROYECTO.....	107
FIGURA 51. PANTALLA VER PROYECTO.....	108
FIGURA 52. FORMULARIO EDITAR PROYECTO.....	109
FIGURA 53. VER GLOSARIO.....	110
FIGURA 54. FORMULARIO NUEVO PROBLEMA.....	111
FIGURA 55. FORMULARIO EDITAR PROBLEMA.....	111
FIGURA 56. PANTALLA VER PROBLEMA.....	112
FIGURA 57. VER ALTERNATIVAS.....	113
FIGURA 58. VER ALTERNATIVA EN DETALLE.....	114
FIGURA 59. VER GLOSARIO.....	115

FIGURA 60. ENVIAR E-MAIL.....116

FIGURA 61. BÚSQUEDA.....117

FIGURA 62. ADMINISTRACIÓN DE USUARIOS.....118

FIGURA 63. ADMINISTRACIÓN APORTES.....119

FIGURA 64. ESTADÍSTICAS.....119

RESUMEN

TÍTULO: DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB QUE PERMITA RESOLVER PROBLEMAS DE FORMA CONJUNTA EN LAS ORGANIZACIONES

AUTOR: WALTER SERRANO ZAMBRANO

Palabras claves: Rationale, Administración, Web, Gestión, Proyectos, AJAX, Backbase.

Este proyecto nace como respuesta a una necesidad clara, en las organizaciones no se tiene un manejo adecuado a la documentación de las decisiones, la cual es la información más importante en el proceso de desarrollo de un proyecto. Es así como Rationale se convierte en una herramienta con gran proyección en el campo de la administración de la fundamentación, gestión de proyectos y administración del conocimiento, al servir de apoyo a la comunidad académica e industrial.

La aplicación permite mediante el debate, llegar a la solución de un problema determinado que se tenga, mediante el registro de alternativas para ese problema y que cumplan con los criterios de solución establecidos para el mismo, todo esto en el marco de la realización de un proyecto que esté ejecutando la organización. Esto permite tener registros históricos de solución de problemas que pueden ser útiles en el futuro como base de conocimiento. El sistema admite varios tipos de usuarios, entre esos se encuentran: Administrador, Director, Participante y consultor. Estos usuarios están claramente diferenciados por las zonas a las que tienen acceso y realización de ciertas tareas. Esta aplicación Web fue desarrollada siguiendo la metodología de desarrollo FDD la cual funciona muy bien en ésta clase de proyectos.

La Aplicación Rationale esta orientada a la Web y desarrollada con herramientas de libre distribución como PHP, Ajax y MySQL con el objetivo de permitir un fácil y más amplio acceso a la misma, además de ofrecer un soporte mucho más extenso que sus similares.

* Proyecto de grado

** Facultad de Fisicomecánicas – Escuela de Ing. De Sistemas e Informática
Director: Fernando Rojas Morales

SUMMARY

**TITLE: DESIGN AND IMPLEMENTATION OF A WEB
APPLICATION THAT ALLOWS TO RESOLVE
PROBLEMS OF A JOINT FORM IN ORGANIZATIONS**

AUTHOR: WALTER SERRANO ZAMBRANO

Key words: Rationale, administration, Web, Projects, management, AJAX, Backbase.

This project was born as an answer to a clear necessity, in the organizations the documentation of decisions is not very well handled, wich is the most important information in the development process of a proyect. That way, Rationale becomes as a tool with great projection in fundamentation administration, project management and knowledge management fields serving for support to the academic and industrial community.

The application allows my beans of debate, get to the solution of a determined problem using the registry of alternatives for that problem and that accomplish with the solution criteria established for it, al that encapsuled in in a project that is being executed in the organization. This facilitates have historic registries of problem solutions that may be useful in the future as a knowledge base. The system allows different types of users, wich are: Administrator, Director, participant and consultor. This users are clearly differenciaded by the zones wich they have access and making of certain tasks. This web application was Developed following the FDD development methodology wich works perfect for this kind of web projects.

The Rationale Application is Web oriented and was developed using free to use tools like PHP, Ajax, MySQL with the objective of allow an easier and wider acces to it. It also offers a much extended support than similar.

* Grade Project

** Physicomechanics Faculty – Systems and Informatic Engineering School
Director: Fernando Rojas Morales

INTRODUCCION

La solución de problemas es una actividad determinante del éxito en el desarrollo de proyectos de cualquier índole. Al aumentar la complejidad de un proyecto, tanto en el campo de número de problemas como en el del factor tiempo, es posible que la importancia del camino entre el problema y la solución sea algo importante, ¿Qué pasaría si las características iniciales del problema cambian? ¿Siguen siendo las soluciones antes tomadas las óptimas? ¿Se debería continuar con solución anterior? ¿Qué factores determinaron la toma de una solución?

Proyectos de gran envergadura son sinónimo de grupos de trabajo, éste es un factor que trae consigo otras preguntas que no se podrían responder centrándose solo en la solución de los problemas, ¿Cómo aprovechar el recurso humano de los grupos de trabajo? ¿Cómo compartir los datos relevantes con respecto a un problema? ¿Quién tiene experiencia con la resolución de problemas parecidos? ¿Quién diseñó la anterior solución? ¿Quién tomó la decisión de implementar una forma de abordar un problema?

Registrar los eventos transcurridos para la realización de los proyectos es necesario para responder estas preguntas, se podría decir que el valor agregado de estos datos es algo igual de importante que la solución del problema en si mismo, pero, ¿Cómo realizar la fundamentación de proyectos?

En el mejor de los casos, en las reuniones para la toma de decisiones se maneja un documento con el orden del día, al finalizar a éste se le anexan las conclusiones, decisiones, compromisos, etc., y se genera un acta con los datos de la reunión, ¿Es ésta una forma óptima de manejar éstos datos? ¿Qué pasa si no existe quórum? ¿Cuan fácil es el acceso a los datos del acta? ¿Qué tan flexible podemos ser con este método? ¿Es ésta una forma fiel para llevar el control sobre un proyecto? Por supuesto, estando en el siglo XXI existen tecnologías que podrían brindarnos otras soluciones a éstas preguntas, la Web ha demostrado ser un excelente método para integrar, divulgar y acceder a la información, su uso cada vez se vuelve mas estándar y familiar, y la computación que brinda una gran flexibilidad en el manejo de los datos, lo cual permite centrarnos en lo realmente importante, la información que estos datos nos proporcionan y sus repercusiones en nuestras metas.

Ya existen herramientas que solucionan parcialmente estos inconvenientes, y son por muchas personas actualmente utilizadas, los foros, e-mail, documentos electrónicos, etc. Existe una amplia necesidad de aplicaciones especializadas en la solución de estos menesteres, ésta es precisamente la herramienta tecnológica que se plantea en este libro.

1.PRESENTACION

Con el objetivo de brindar una guía al lector de este documento y a las personas que quieran estudiar el desarrollo de este proyecto se presenta a continuación la estructura del mismo. El documento se elaboró para que sea comprensible a cualquier tipo de lector sin importar su grado de preparación en el tema, por lo cual, muestra de forma clara y concisa el proceso que se llevó a cabo en el desarrollo del proyecto dejando ver la fundamentación teórica utilizada como referencia para la implantación del sistema y el cumplimiento de los objetivos iniciales.

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

En el capítulo 1 se explican los objetivos y la justificación del proyecto.

En el capítulo 2 se exponen los antecedentes al proyecto.

En el capítulo 3 se muestra el estado del arte de los sistemas colaborativos.

En el capítulo 4 se ubica al lector con un amplio marco teórico que comprende la teoría fundamental de los sistemas colaborativos así como tópicos referentes a la ingeniería de sistemas.

En el capítulo 5 se explica la metodología de diseño y desarrollo utilizada.

En el capítulo 6 se manifiesta el porqué se utilizó cada una de las tecnologías que componen al sistema.

En el capítulo 7 se despliegan los diagramas UML y WebUML que se usaron para diseñar el sistema.

En el capítulo 8 se exhiben las capturas de pantalla correspondientes a la interfase de usuario con una breve explicación de su funcionalidad.

1.1 OBJETIVOS

1.1.1 OBJETIVO GENERAL

Facilitar labores de comunicación, coordinación, documentación y aprendizaje de una comunidad en un proceso de resolución de problemas por medio de una aplicación Web basándose en el marco de trabajo administración de la fundamentación.

1.1.2 OBJETIVOS ESPECIFICOS

- Implementar un módulo que permita crear proyectos especificando las fases de los mismos, modificarlos y eliminarlos. Incluye una Vista en forma de árbol de los problemas asociados al proyecto para ilustrar su estructura por fases
- Desarrollar un módulo para la gestión de problemas que incluya:
 - Crear, modificar y eliminar problemas junto con sus criterios, alternativas y argumentos incluyendo la capacidad de subir documentos para respaldar aun mejor la alternativa planteada.
 - Crear, modificar y eliminar comentarios de los usuarios en las alternativas planteadas para facilitar el entorno de discusión necesario con el fin de seleccionar la solución al problema, estos comentarios incluyen su posición frente a la alternativa planteada (A favor, Neutral, En contra).
 - Crear, modificar y eliminar términos en el glosario.
- Realizar un módulo de Administración del sistema que permita:
 - Gestionar los derechos de administración y dirección de los usuarios y si está habilitado o no para operar el sistema.
 - Realizar limpieza de mantenimiento en proyectos, problemas, alternativas, términos del glosario e incluso comentarios de los usuarios.
- Crear un módulo de búsqueda de texto en los aportes hechos por los usuarios con diferentes filtros.

- Generar un módulo de estadísticas que muestre el número de proyectos, usuarios, problemas, alternativas y comentarios que hay en el sistema, Proyecto más activo, problema más activo, usuario más activo, y otros datos en general que proporcionen información adicional de soporte.
- Desarrollar un servicio de envío de correo electrónico automático a los usuarios cada vez que alguien realice un aporte y el envío manual de correo electrónico desde el sistema a los usuarios.

1.2 JUSTIFICACIÓN

La documentación de las decisiones es la información más importante en el proceso de desarrollo de un proyecto que se realiza dentro de una organización. Sin embargo, ésta información generalmente se pierde, debido a factores tales como la falta de comunicación entre el equipo de trabajo, la inexistencia de un procedimiento que permita el registro histórico de los problemas y sus soluciones, para su posterior consulta en caso de una situación similar. Para evitar esto, se invierten recursos tales como tiempo y dinero para capturar la información, de tal manera que sea fácil revisar las decisiones tomadas con anterioridad. La cantidad de recursos que se va a invertir en ésta labor, depende del tipo de proyecto. Si éste es complejo, la información se tiende a revisar y a actualizar varias veces en un largo periodo de tiempo.

En particular, un cliente puede requerir que se documenten todas las decisiones tomadas durante la realización de un proyecto, con el fin de obtener un soporte para el futuro. Por ejemplo, cuando los requerimientos cambian y se le añade funcionalidad, permite a los participantes tener un respaldo de cuáles decisiones necesitan ser revisadas y cuales ya han sido evaluadas. Así, cuando se añaden nuevas personas a un proyecto, pueden familiarizarse rápidamente con las decisiones que se tomaron, simplemente ingresando al sistema de documentación de decisiones. Desafortunadamente, ésta documentación es también la información más difícil de integrar y actualizar por parte de los participantes de un proyecto.

Es por ello, que surge la necesidad de desarrollar un sistema que permita registrar los problemas, las alternativas propuestas y finalmente la solución tomada, todo esto dentro de un proceso de discusión basado en criterios previamente especificados, mediante una aplicación Web robusta, fácil de usar, transparente y que proporcione un mecanismo de comunicación entre los participantes.

2. ANTECEDENTES

La captura de la fundamentación fue propuesta originalmente por Kunz y Rittel desde entonces se han diseñado y evaluado diferentes modelos en el contexto de la ingeniería del software y otras disciplinas de la ingeniería; se comparará a grosso modo tres de ellas: IBIS (sigla en ingles de sistemas de información basado en problemas) DRL (siglas en ingles de lenguaje de representación de decisiones), y QOC (sigla en ingles de preguntas opciones y criterios)

2.1 Sistema de Información Basado en Problemas

El ibis incluye un modelo de problemas y un método de diseño para el tratamiento de problemas mal estructurados u horrorosos. Un problema horroroso se define como un problema que no puede resolverse en forma algorítmica sino que tiene que resolverse mediante discusión y debate.

El modelo de problemas IBIS tiene tres nodos (problemas, posiciones y argumentos) relacionados por varios tipos de vínculos (apoya, se opone a, reemplaza, responde a, generaliza, cuestiona y sugiere) cada problema describe una necesidad de diseño bajo consideración. Los desarrolladores proponen soluciones al problema creando nodos posición. Mientras se generan las alternativas los desarrolladores argumentan acerca de su valor con nodos argumento. Los argumentos pueden apoyar una posición u oponerse.

IBIS fue soportado por una herramienta de hipertexto y se uso para capturar la fundamentación durante reuniones frente a frente. Proporcionó las bases para la mayoría de los modelos de problemas subsiguientes incluyendo DRL y QOC

2.2 Lenguaje de Representación de Decisiones (drl)

El lenguaje de representación de decisiones DRL trata de capturar la fundamentación de las decisiones de un diseño. La fundamentación de una decisión es definir como la

representación de los elementos cualitativos de la toma de decisiones incluyendo las alternativas que se han considerado, su evaluación, los argumentos que condujeron a esas evaluaciones los criterios usados en esas evaluaciones. El DRL esta apoyado por SYBIL una herramienta que permite que el usuario lleve cuenta de las dependencias entre los elementos de la fundamentación cuando revisa las evaluaciones.

Las principales desventajas del DRL son su complejidad y el esfuerzo empleado en la estructuración de la fundamentación capturada.

2.3 Preguntas Opciones y Criterios (qoc)

Preguntas, opciones y criterios es otro aumento al sistema IBIS. Por otro lado las estructuras QOC se construyen como un acto de reflexión sobre el estado actual de diseño. Esta separación conceptual de las fases de construcción y argumentación del proceso de diseño enfatiza la elaboración y estructuración sistemática de la fundamentación, en vez de capturarlas como un efecto secundario de la deliberación. La fundamentación desde la perspectiva del QOC es una descripción del espacio de diseño explorado por los desarrolladores.

3. ESTADO DEL ARTE

Sistemas de Administración de Fundamentación

3.1 Acollab

www.atutor.ca/acollab Versión 1.2

Es un Ambiente de trabajo colaborativo completamente accesible, de código abierto, multigrupo, basado en Web. Es ideal para grupos que trabajan a distancia elaborando documentación, participando en investigación ó escribiendo artículos y es perfecto para educadores en línea que deseen añadir actividades de aprendizaje grupal a sus cursos. Tiene un panel de administración muy completo donde se pueden gestionar los usuarios, permisos, las noticias, foros, y muchas cosas más. Tiene también la posibilidad de uso multilenguaje e incluye integración con *Atutor* de forma nativa.

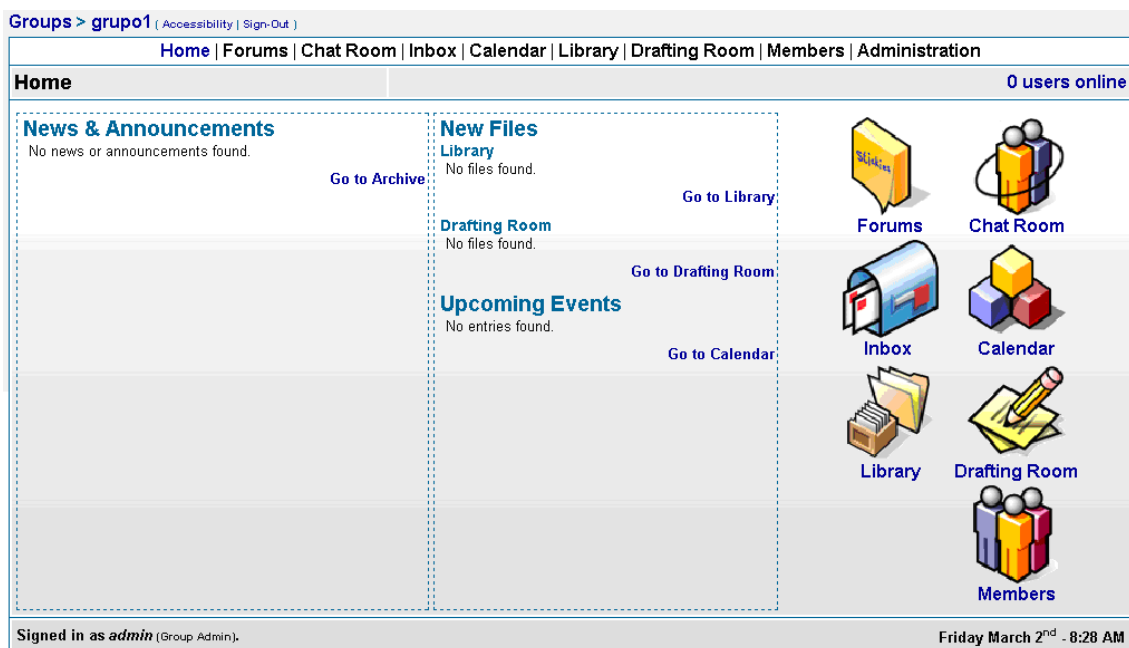


Figura 1. Pantalla Acollab

3.2 dotProject

www.dotproject.net Version 2.0.4

dotProject es un framework (marco de trabajo) realizado en PHP que incluye módulos para compañías, proyectos, tareas – con Diagramas de Gantt – foros, manejo de archivos, calendarios, contactos, sistema de ayuda por turnos, soporte multilenguaje, permisos de usuarios por modulos y diversos estilos gráficos.

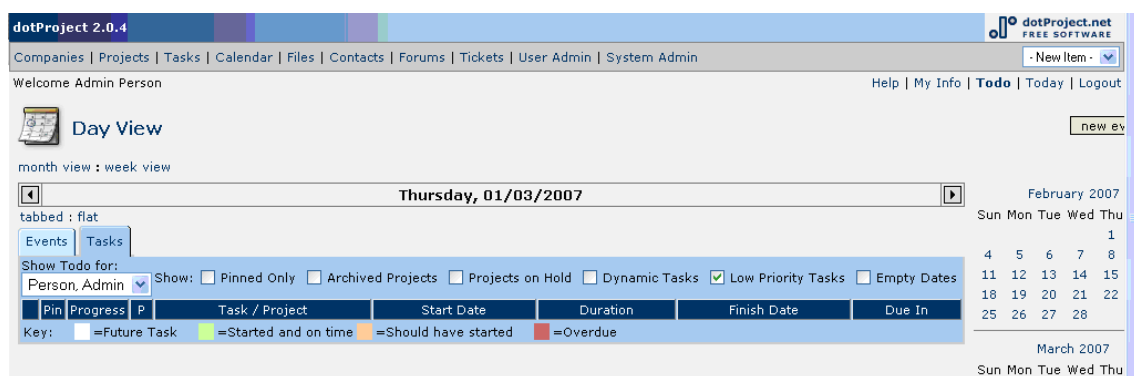


Figura 2. Pantalla dotProject

3.3 Streber

www.streber-pm.org Versión 0.0792

Con Streber, personas freelance y pequeños grupos de trabajo pueden configurar fácilmente proyectos y mantener un registro de las tareas, problemas, intentos de solución, etc. Los derechos de usuario por proyecto pueden ser ajustados para proveer a los clientes una vista limitada del estado actual del proyecto. El sistema es muy completo, posee una interfaz gráfica muy agradable organizada por secciones mediante Tabs, muestra sólo la información relevante para el usuario, tiene un poderoso sistema de búsqueda y función para exportar tablas a formato CSV, haciendo de éste, un sistema con alta portabilidad. De los sistemas de administración de fundamentación, Streber es tal vez el mejor en la actualidad.

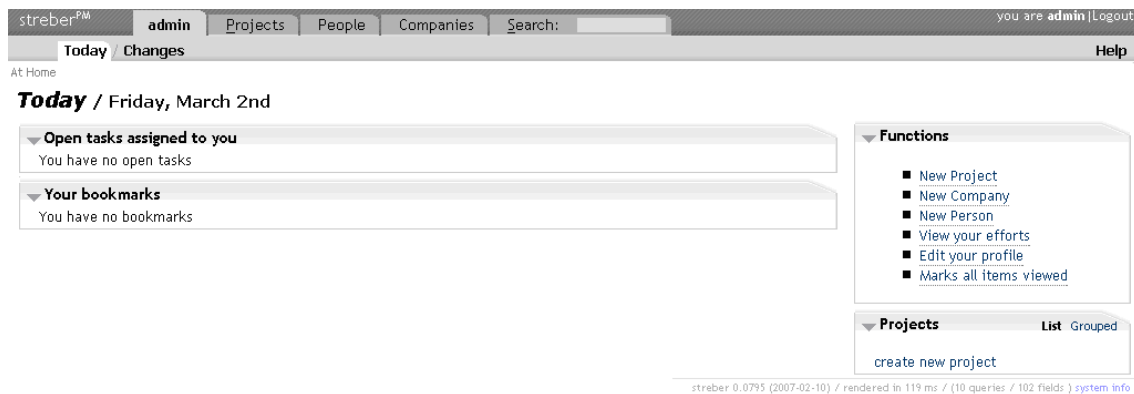


Figura 3. Pantalla Streber

Estos sistemas, desarrollados por comunidades son en general, versátiles, con excelente diseño tanto del sistema, como de la interfase, incluyen herramientas útiles para la elaboración de proyectos, tales como asignación de tareas, equipos de trabajo, hilos de discusión, etc.

Haciendo un análisis exhaustivo de estos sistemas, se encontró que se enfocaron a la administración y gestión de proyectos, algo bastante común en muchas aplicaciones Web e incluso de escritorio, como Microsoft Project. Es por eso que el sistema que se pretende realizar se orienta principalmente a la teoría de administración de la fundamentación, que toma parte notable del desarrollo de proyectos, y así poder ofrecer un complemento a cualquier aplicación de gestión de proyectos.

4. MARCO TEORICO

4.1 Introducción

Las organizaciones actualmente giran en torno a la manera como se administra la información, su gestión; para poder ser realmente competitivas, deben estar preparadas para contrarrestar retos e incrementar su productividad.

La información obtenida diariamente dentro de la organización es muy importante, por lo cual los recursos tecnológicos deben estar orientados a dicha información de manera que ayuden a los gerentes o administradores a tomar decisiones estratégicas y oportunas.

Las personas encargadas de tomar decisiones dentro de la organización reciben y analizan la información proveniente de diferentes medios, la intercambian con otros grupos y la procesan con diferentes herramientas basadas en computador.

Los sistemas de soporte a toma de decisiones (SSD) han originado un lugar muy importante dentro de las organizaciones, las cuales buscan apoyo en un sistema como estos, que permita guiarlos y los conduzca a tomar una decisión óptima y fundamentada, mejor que la obtenida a través de la experiencia. Los SSD se han creado para proyectar y predecir los resultados de las decisiones antes que sean tomadas.

Para la realización de un SSD se debe contar con muchas investigaciones y administraciones eficientes ya que es necesario contar con una estructura principal para justificar que se van a tomar decisiones con la ayuda de las tecnologías de información.

Las etapas que se deben tener en cuenta, en el momento de desarrollar un SSD se relacionan con la identificación de los problemas, la formulación de las políticas y las estrategias.

Mediante el uso de indicadores en las diferentes etapas, se contribuye a sintetizar y analizar información técnica, tomar medidas y acciones, a fijar objetivos y metas de desarrollo, comunicar sobre las tendencias, con el fin de inducir a un desarrollo sostenible.

Los SSD han disminuido las barreras tecnológicas y han contribuido en la toma de decisiones relevantes a administradores y directivos. Las herramientas de soporte administrativo analizan y sintetizan la información y la utilizan en la toma de decisiones, de tal manera que la transforma en conocimiento que puede ser compartido a través de las herramientas de administración.

Los sistemas de soporte a la toma de decisiones (SSD) permiten la recolección, y clasificación de información relevante para el soporte de los análisis realizados por los diferentes miembros de las organizaciones, independientemente de la actividad que esta realice.

El principal objetivo o la finalidad de los (SSD) es mejorar la calidad, rapidez y efectividad de las decisiones tomadas por parte de los usuarios de estos sistemas.

Los principales elementos de un SSD son:

- ✓ El sistema SSD en sí, incluyendo la plataforma tecnológica que lo aloja.
- ✓ La información y datos que el sistema almacena.
- ✓ Los usuarios que pueden ser directivos, gerentes y funcionarios

4.2 Sistema

Según la IEEE Standard Dictionary, sistema es un todo integrado, aunque compuesto de estructuras diversas y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos pueden variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por una sola de las partes individuales. La complejidad de la combinación está implícita."

Aspectos fundamentales del concepto Sistema:

- ✓ La existencia de elementos diversos e interconectados.
- ✓ El carácter de unidad global del conjunto.
- ✓ La existencia de objetivos asociados al mismo.
- ✓ La integración del conjunto en un entorno.

4.3 Ingeniería de Sistemas

Ingeniería de Sistemas es la aplicación de esfuerzos científicos y de ingeniería para:

1. Transformar una necesidad de operación en una descripción de parámetros de rendimiento del sistema y una configuración del sistema a través del uso de un proceso iterativo de definición, síntesis, análisis, diseño, prueba y evaluación.
2. Integrar parámetros técnicos relacionados para asegurar la compatibilidad de todos los interfaces de programa y funcionales de manera que optimice la definición y diseño del sistema total.
3. Integrar factores de fiabilidad, sostenibilidad, seguridad, supervivencia, humanos y otros en el esfuerzo de ingeniería total a fin de cumplir los objetivos de coste, planificación y rendimiento técnico.

4.4 Sistemas Colaborativos

Una aplicación colaborativa se puede definir como un sistema computacional que asiste a un grupo de personas dedicadas a una tarea común, cada una de ellas trabajando en su propio computador, pero compartiendo datos y programas a través de una interfaz multiusuario.

El principal objetivo de las aplicaciones colaborativas consiste en proveer a los usuarios de herramientas que les permitan coordinar sus actividades de trabajo. Así como hay diversidad de definiciones de los sistemas colaborativos, también existen muchos dominios de aplicaciones colaborativas. Por ejemplo hay aplicaciones que facilitan la interacción a distancia entre miembros de un equipo de trabajo. Otras aplicaciones apoyan la comunicación de personas reunidas físicamente. Otro tipo de aplicaciones ayudan a distintas personas en la confección de un mismo documento electrónico, etc. A pesar de los distintos tipos de aplicaciones que hay, todas tienen un factor común que consiste en hacer más fácil el proceso de compartir información entre un grupo de personas que realizan una tarea común.

La construcción de una aplicación colaborativa es una tarea compleja ya que involucra diversas áreas de trabajo tales como sistemas distribuidos, comunicaciones, interfaces humano-computador, inteligencia artificial, bases de datos.

Los Sistemas Colaborativos tienen como finalidad en este sentido aumentar la productividad de toma de decisiones y mejorar la calidad de sus resultados, a través de la estructuración y captura de razonamientos y argumentaciones.

Es en este contexto que se da la realización del presente proyecto, cuya principal contribución radica en satisfacer las necesidades antes descritas. El trabajo consiste en analizar, diseñar e implementar un sistema colaborativo Web que permita mejorar la calidad de las decisiones tomadas dentro de las organizaciones.

En el sistema propuesto se definen los problemas no estructurados como problemas que no se pueden resolver en forma algorítmica sino que deben ser abordados o resueltos mediante el debate y la discusión de un grupo de personas interesadas.

La solución a estos problemas varía de acuerdo a la organización; entre los más comunes se encuentran:

- ✓ Esquema piramidal de las estructuras organizacionales en el cual el responsable de las decisiones es una sola persona.
- ✓ Esquema grupal en el cual las decisiones se toman por un grupo de personas mediante reuniones, discusiones, enfrentamientos cara a cara, etc.

En este proyecto los problemas no estructurados se resolverán por medio de la fundamentación de la información la cual explicamos detalladamente a continuación.

4.5 Fundamentación de la Información del SSD

En el sistema de soporte a la toma de decisiones (SSD) es indispensable la fundamentación de la información para esto se deben seguir los siguientes pasos:

1. Definición de la cuestión **Problema**: Los problemas representan dificultades que no tienen una solución correcta única y que no pueden resolverse de manera algorítmica, por lo general se resuelven mediante discusiones y negociaciones.

2. Exploración del espacio de solución **Alternativa**: Son soluciones posibles que pueden resolver el problema que se esta considerando.
3. Evaluación del espacio de solución **Criterios y Argumentos**: Los criterios son cualidades deseables que debe satisfacer la solución seleccionada. Los argumentos son opiniones expresadas por personas que están de acuerdo o en desacuerdo con una propuesta o una valoración.
4. Descomposición del espacio solución **Resolución o Decisiones**: es la resolución de un problema que representa la alternativa seleccionada de acuerdo al criterio que se uso para la evaluación y justificación de la selección. **[1]**

[1] Tomado de BRUEGGE Y DUTOIT, ingeniería del software orientado a objetos .

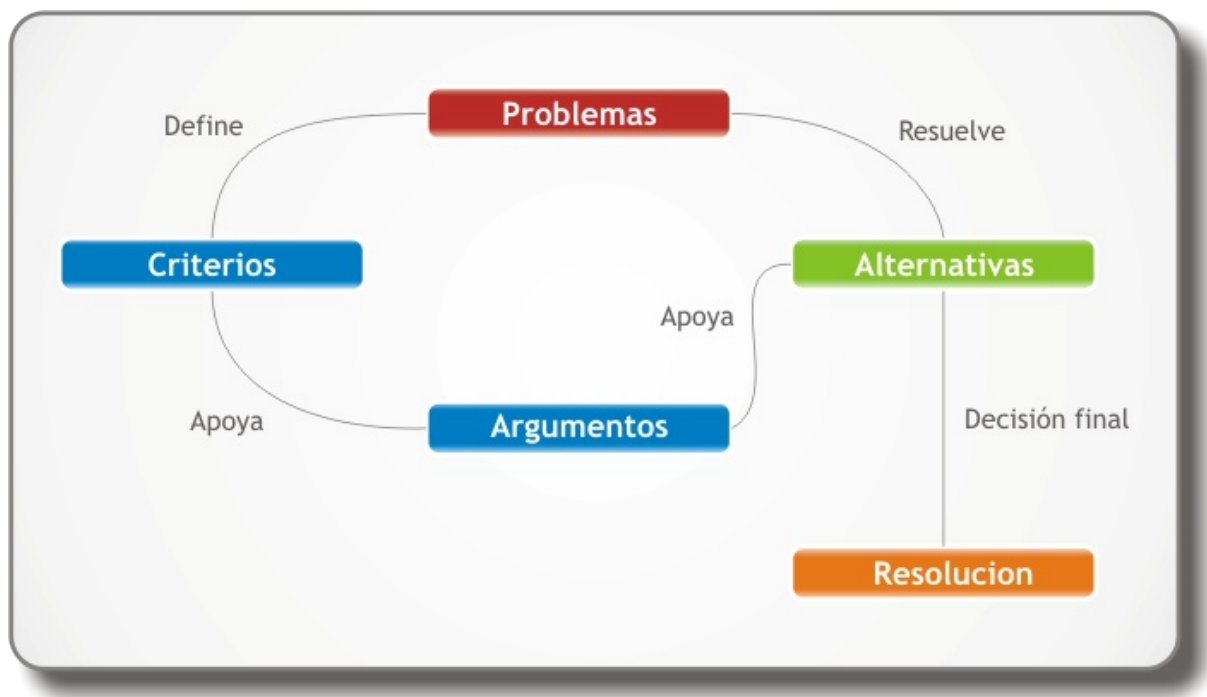


Figura 4. Administración de la fundamentación

En el gráfico se muestra el esquema principal en el que se apoya la fundamentación de la información; el cual consta de cuatro elementos básicos, como son los **problemas, criterios, argumentos y alternativas**, y la relación que existe entre cada uno de estos componentes.

Para resolver los problemas los usuarios deben tener en cuenta los criterios previamente definidos; las alternativas son las soluciones propuestas a estos problemas, por cada miembro del grupo. Los argumentos son las razones en que se basan las alternativas y respaldan a una alternativa o a un criterio, o se oponen ellos. Una vez se vence el plazo del problema se escoge la alternativa que brinda la mejor solución al problema, llamada **resolución**, de esta forma se toma la decisión final y se guarda un registro de esta.

4.6 Componentes

Los componentes de que consta son: hardware, software, recursos humanos y procedimientos. Al unir los componentes de un SSD se pueden organizar diferentes actividades que proporcionen temas relacionados con el proceso de la toma de decisiones.

4.6.1 Hardware

Para poder trabajar con un SSD se necesita contar con los requerimientos mínimos de hardware para este tipo de sistemas:

1. Un dispositivo de entrada/salida, mediante el cual sea posible darle datos de entrada y producir una salida.
2. Un procesador, para realizar los procesos necesarios y generar resultados útiles a los tomadores de decisiones.
3. Una línea de comunicación entre el dispositivo de entrada/salida y el procesador, para permitir la comunicación interactiva entre los miembros del grupo.
4. Una pantalla o monitores individuales, para ver las aportaciones que hace cada miembro del grupo o para analizar resultados.

SSD más sofisticados pueden tener terminales o computadoras personales para cada persona, varios procesadores centrales, equipos de comunicación a larga distancia y pantallas grandes.

4.6.2 Software

Se requiere del software adecuado para realizar el proceso de toma de decisiones en grupo. Los componentes del software de un SSD son:

1. Una base de datos que cuente con información relacionada con la decisión que debe tomarse y que permita la consulta y búsqueda de temas específicos
2. Una base de modelos, de la cual se puedan elegir diferentes alternativas para tomar una decisión.
3. Programas de aplicaciones específicos para que el grupo los use como procesadores de palabras, graficadores, hojas de cálculo o paquetes estadísticos.
4. Una interfase flexible y fácil de usar, que permita al ejecutivo interactuar de la manera adecuada con el sistema sin requerir de mucha asesoría o capacitación.

4.6.3 Recursos humanos

Este componente incluye a las personas que participan en el proceso de toma de decisiones, (usuarios) del SSD. Un administrador, quien se encarga del control de los usuarios, es la persona que conoce el sistema, sabe cómo funciona y cómo operarlo. Un director de grupo, asignado por el administrador quien conoce el potencial del problema a resolver y es la persona que tiene más conocimientos acerca del mismo. El director es en últimas el encargado de tomar la decisión final. Para tener una mejor comprensión de los componentes del sistema y su funcionamiento, observemos el siguiente gráfico.

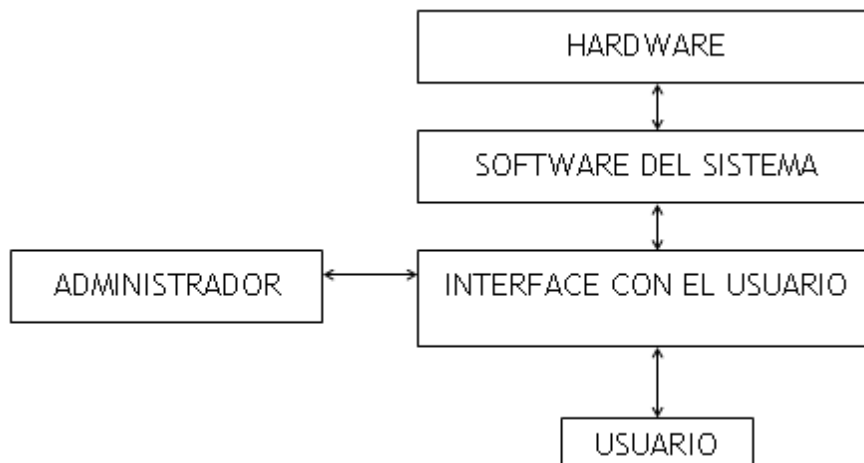


Figura 5. Componentes del Sistema

4.7 Beneficios

En el siguiente gráfico se puede apreciar mejor los beneficios de los sistemas de información para la toma de decisiones.

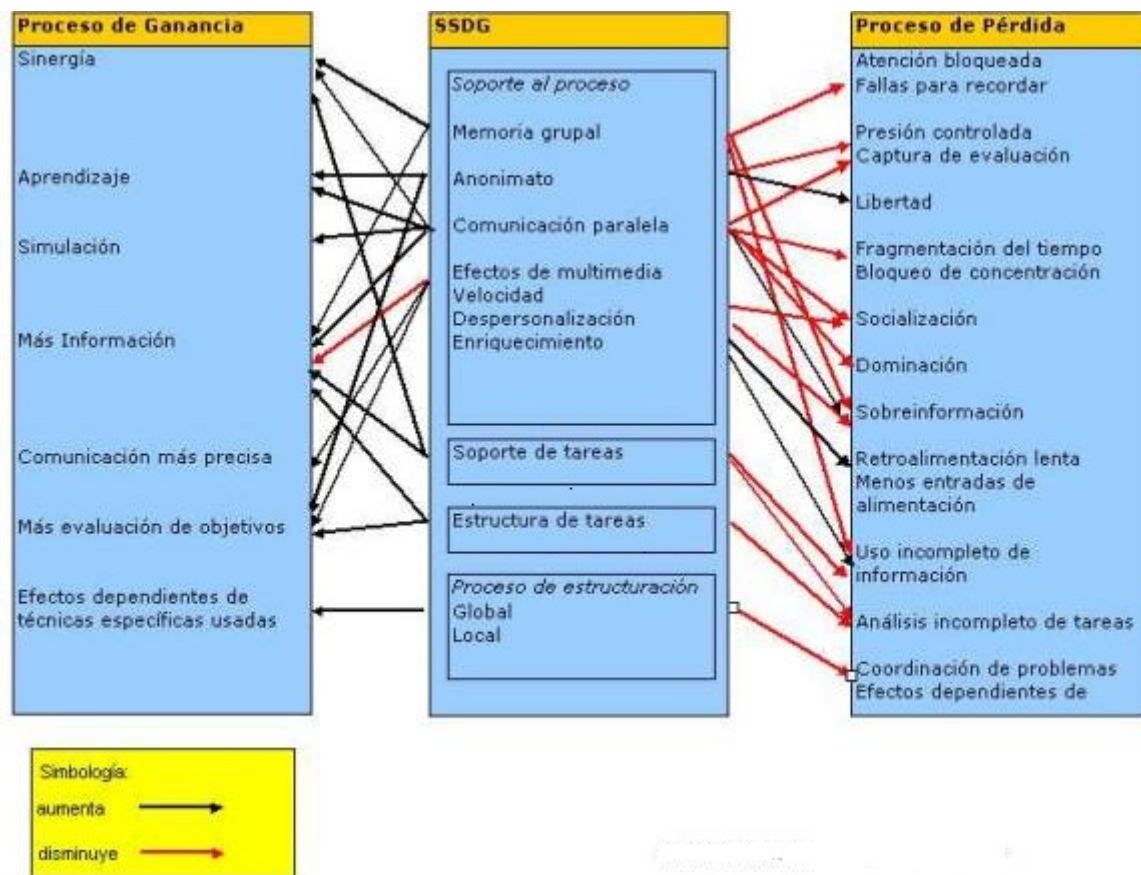


Figura 6. Beneficios del Sistema

4.8 Tipos de Usuario del Sistema

Dentro de las organizaciones existen tres niveles, el estratégico, el administrativo y el operacional.

Los tipos de problemas que ayuda a solucionar un SSD son complejos y semiestructurados y están diseñados para ayudar en el nivel estratégico y administrativo. A nivel operacional este sistema no tiene incidencia directa ya que los operarios realizan tareas repetitivas y de baja complejidad.

4.9 DISEÑO DE LA INTERFASE DE USUARIO

Es el elemento fundamental de la aplicación, la interfase es el puente de comunicación que permite al usuario interactuar con la aplicación, para realizar las tareas y buscar la información que necesita, a través del menú de navegación, enlaces y formularios. Debe ser fácil de usar y aprender, cómoda, usando metáforas de diseño con elementos de diseño

conocidos por todos. Los procesos de usuario como son los formularios de registro, publicaciones o buscadores de información deben ser fáciles de entender y completar.

4.9.1 Diseño gráfico

El Diseño gráfico no es crear formas, sino comunicaciones, que mediante un método específico (diseño) construye mensajes (comunicación) con medios visuales (grafismos). El diseñador debe ser el intérprete del mensaje y para representar la información utiliza diferentes elementos gráficos:

- Elementos gráficos simples: puntos y líneas (libres, rectas, quebradas curvas, etc.)
- Elementos geométricos, con contorno o sin él: polígonos, círculos, elipses, óvalos.
- Fuentes: letras de diferentes formas y estructura, utilizadas para presentar mensajes textuales.
- Gráficos: logotipos, iconos; ilustraciones; fotografías y videos.

Está muy asociada al ser humano, que escribe y dibuja casi siempre con líneas curvas. Son usadas para suavizar la dureza de una forma rectangular en uno o más de sus lados (como los botones).

Estos elementos básicos se combinan unos con otros en un grafismo, utilizando algunos conceptos de diseño gráfico, la unión de todos ellos en la composición gráfica, es una comunicación individual y completa, de complejos elementos humanos asociados con el lenguaje, la experiencia, la edad, el aprendizaje, la educación y la memoria.

4.9.2 Diagramación

La diagramación se define como la organización armoniosa de elementos gráficos (el texto y la imagen), con el objetivo de comunicar visualmente. Diagramar es distribuir, organizar los elementos del mensaje (imagen y texto) en el espacio bidimensional, mediante criterios de jerarquización (importancia) buscando la funcionalidad del mensaje (fácil lectura), con una apariencia estética agradable (aplicación adecuada de tipografías y colores).

La diagramación determina la ubicación, los tamaños, proporciones de los espacios para textos, imágenes, animaciones, así como las áreas blancas, los márgenes para que todos los elementos formen un conjunto agradable, fácil de leer y lo más importante de entender.

Todo esto se logra utilizando las bases de la diagramación: la tipografía, el color y la composición.

El modelo más utilizado esta basado en tres columnas de elementos: barra de navegación, texto y opciones secundarias respectivamente.

4.9.3 Elementos de diagramación

- **Imágenes:** fotografías, diagramas, cuadros estadísticos; viñetas y logotipos.
- **Textos:** títulos, leyendas, texto.
- **Recursos gráficos:** líneas, tramas, signos señaléticos, fondos, recuadros para texto.

4.9.4 Tipografía

Al seleccionar la tipografía el diseñador debe tener en cuenta el significado de las formas de los tipos, tiene la capacidad de darle otros atributos como tamaño, color, linealidad, transparencia... También se les asigna conceptos que le dan un valor connotativo como la pesadez, dinamismo, simpleza, elegancia y fragilidad.

La forma de la letra debe estar asociada con el tema del diseño; los tipos serif son ilegibles, festivos, clásicos, dinámicos; mientras los tipos sans serif son más legibles, serios, conservadores, toscos y estáticos. Luego de determinar la fuente para los textos, se debe escoger el alineado de estos: a la izquierda es recomendado para textos largos, crea una letra y un espacio entre palabras uniforme, el lector es capaz de localizar fácilmente cada nueva línea, es más legible. La alineación a la derecha va en contra del lector, porque es difícil encontrar la nueva línea; puede ser usado para un texto poco extenso; al centro y el justificado (alineado a la derecha e izquierda).

Los tipos de fuente más utilizados en la web son Arial y Verdana, porque visualmente son más claras.

Para facilitar la lectura en la pantalla se debe tener en cuenta el ancho de línea, generalmente superior a los 350 píxeles, utilizando tipografías de tamaño 10, 11 y 12. El contenido se lee mejor con fondo claro y texto oscuro, el más recomendable es el fondo

blanco (#FFFFFF). Según Jakob Nielsen el espacio para el contenido en la pantalla, debería ser por lo menos el 50% o el 80% del espacio de la página.

En dreamweaver las hojas de estilo (CSS), definen el estilo de la tipografía y los colores del sitio. Al definir todo el estilo en un documento, se puede hacer actualizaciones y mejoras en el aspecto del sitio, en muy corto tiempo.

4.9.5 Color

Las formas en que se expresa el color son:

- **Color denotativo:** el icónico expresa la función de identificar (un árbol, la naranja...), el saturado (utilizar el color puro, brillante) y el fantasioso que es una modificación del color natural para generar fantasías visuales (un tomate color azul); son la representación del realismo.
- **Color connotativo:** psicológico, simbólico y estético; expresan las sensaciones.
- **Color esquemático:** el emblemático (escudos, banderas...), el señalético esta asociado a símbolos gráficos, dan identidad a una corporación (deportes, comercio...) y el convencional (son de carácter decorativo o funcional); expresan la funcionalidad y la codificación.

4.9.5 Contraste

El contraste es una herramienta para producir y controlar diferentes efectos visuales y poder transmitir diversos significados. Permite atraer la atención del usuario, mediante la utilización de diferentes contrastes (de color, tono, textura, tamaño...) para dar mayor significado a la comunicación.

- **Contraste de tono:** El contraste entre tonos cálidos y fríos, un contraste claro / oscuro. El elemento de mayor peso es el que tiene más intensidad tonal o el más oscuro.

- **Contraste de colores:** el contraste de colores cálidos – fríos. Las gamas frías verdes-azules tienen un carácter regresivo, las gamas cálidas rojos-amarillos, tienen carácter expansivo. Se puede darle mayor dimensión o proximidad a determinado elemento.

4.9.6 Composición

Las técnicas de composición permiten armonizar la función y la forma, adecuar distintos elementos gráficos dentro de un espacio visual para transmitir un mensaje claro a los usuarios.

El diseño se ordena en una estructura rígida; en la composición los elementos situados a la derecha del área tienen mayor peso visual y dan la sensación de proyección y avance. Los ubicados a la izquierda dan la sensación de ligereza visual, mientras nos acercamos al margen izquierdo. La parte superior posee mayor ligereza visual, el peso de los elementos es mínimo, al estar equilibrados con la parte inferior donde los elementos tendrán mayor peso.

4.9.7 Jerarquización

Permite la organización de los elementos visuales, se le asigna valores a cada uno, para organizarlos en varios niveles. Siempre hay un elemento líder para destacar por tamaño, color, posición, orientación, opacidad, y los demás detrás de él en orden jerárquico.

- **Jerarquía visual**

Es importante tener en cuenta el comportamiento del usuario en el barrido visual de la página, se puede brindarle elementos para jerarquizar la información por el tamaño de los botones, utilizar el contraste de color y el uso de efectos tipográficos, así como efectos de relieve o profundidad para resaltar los elementos.

En toda composición gráfica se debe crear una jerarquía visual para que los elementos más importantes se muestren acentuados, para llamar la atención del usuario y vea la información contenida en la composición de forma organizada y lógica. La distribución de

los elementos de información y navegación, se colocan en la zona superior que tiene mayor jerarquía visual.

Generalmente se aprecia una composición gráfica así: primero el conjunto completo de formas y colores, miramos figuras y fondo; segundo se analiza las partes individuales, comenzando por los gráficos y finalmente con los elementos textuales, que son los que se deben leer palabra por palabra.

La lectura de los documentos se realiza de izquierda a derecha y desde la parte superior a la inferior. Esto sirve para organizar el contenido del sitio Web.

- **Jerarquía visual cromática**

Otro sistema para establecer una jerarquía en los contenidos es usando los colores. Se distingue una zona de atracción y los otros elementos aparecerán difuminados poco relevantes.

En primer orden colores primarios muy saturados: atraen rápidamente, se usan para pequeñas áreas porque recargan demasiado la vista; en segundo orden colores menos saturados, secundarios y terciarios: los que se encuentran en la naturaleza son los más aceptados por el usuario y en tercer orden colorear muy sutilmente: para no atraer demasiado la mirada.

4.9.8 Proporción

Es la relación armónica de las dimensiones entre los diversos elementos que constituyen la composición y entre cada uno de ellos.

4.9.9 Sistemas de proporción

Para lograr combinaciones en las proporciones se pueden integrar sistemas matemáticos y geométricos. Uno de ellos es la sección áurea o también llamada “divina proporción” siendo una proporción que produce armonías perfectas.

También pueden aplicarse otros sistemas como la serie de fibonacci y la regla de los tercios como formas básicas para diseñar.

4.9.10 Formas de proporción

- Los elementos más grandes, altos y anchos tienen mayor carga visual, que los pequeños, crean zonas de atracción más intensas.
- Los colores puros más saturados tienen más peso; los colores secundarios más neutros tienen más o menos peso y los colores terciarios poco saturados tienen poco peso.
- La agrupación por bloques significativos, no usar muchos porque se perderá la proporcionalidad planteada.
- Las formas angulares y alargadas, alargan el campo de visión creando zonas dominantes.
- Las formas redondas crean armonía, suavidad y perfección, las formas simples se recuerdan con mayor facilidad.

4.9.11 Ritmo

Es la estructuración de los componentes de cada mensaje que se presenta, y de sus subcomponentes; para lograr la unidad. En la composición gráfica hay que organizar los elementos independientes y en relación con otros: los textos y sus cambios de tamaño, posición, color; las imágenes; los signos y sus funciones, para lograr un mensaje visual efectivo y funcional.

Algunas técnicas visuales que se puede utilizar para organizar los elementos son: equilibrio, inestabilidad; simetría, asimetría; unidad, fragmentación; transparencia, opacidad; coherencia, variación; plana, profunda; secuencia, aleatoria y continuidad.

4.9.12 Contragrafismo

Es el equilibrio entre el espacio ocupado por los objetos y el espacio libre, las áreas blancas o la sensación visual que provoca el vacío. Los espacios en blanco son una gran ayuda para los usuarios para entender la distribución de los objetos en una página.

4.9.13 Formatos

Se debe tener en cuenta la distribución del espacio dependiendo del formato que vamos a utilizar, el formato debe tener una armonía. Las márgenes definen la caja de diagramación, donde van a ir los textos.

4.9.14 Tamaños de pantalla

- **TAMAÑO ABSOLUTO:** es el tamaño real de la ventana del monitor, medido en pulgadas.
- **RESOLUCIÓN O TAMAÑO RELATIVO:** viene determinada por el número de píxeles que se muestran en la ventana del monitor. Depende de la tarjeta gráfica.

El tamaño absoluto y la resolución deben estar en concordancia para una visualización correcta, estos valores son aceptables para pantallas de 14" y 15" la resolución máxima apreciable es [800 x 600] px, siendo esta la más usada en el diseño de las páginas web; para una de 17" la resolución máxima apreciable es de [800 x 600] px ó [1024 x 768] px y para una de 21" a partir de [1024 x 768] px.

La página debe estar centrada o justificada a la izquierda; debe tener un tamaño fijo o que se adapte al tamaño del navegador. En el caso de aplicaciones Web donde se utilizan formularios, es mejor usar páginas con tamaños fijos, ya que en monitores con resolución de pantalla superior a 1024 la distancia entre la información y los botones es muy grande y se pierde el contexto (botones situados en el margen derecho y el campo de texto en el margen izquierdo).

4.9.15 Botones

Los botones se deben diseñar según la forma y la función para la que fue hecho, para evitar errores y ayudar en la navegación a los usuarios, deben tener la apariencia de volumen para hacer más claro e intuitivo que han de ser apretados. El área donde acaba uno y donde empieza el siguiente debe estar claramente definida.

Los botones o áreas sensibles deben llevar un nombre descriptivo, para que le proporcionen al usuario la información necesaria para conocer su función. Se deben evitar elementos invisibles de navegación, como menús desplegables o indicaciones ocultas, para evitar al usuario sufrir sobrecarga de información.

Además se debe indicar dónde está el usuario y a dónde puede ir de forma clara, lo mejor es tener dos colores, para identificar el elemento activo de la barra de navegación y para identificar los elementos apagados. Para el elemento activo se utilizan los colores cálidos (rojo, naranja) y saturados, los hará ver más cercanos al usuario, frente a los colores fríos (azul y verde) y poco saturados para los elementos apagados.

Para los enlaces se utiliza un color X (azul es la norma), un tono más brillante para el evento *over* o cuando se pulsa y un tono más apagado para el enlace visitado.

4.9.16 Diseño centrado en el usuario (DCU)

Este diseño involucra al usuario en el proceso de diseño. Los diseñadores deben entender al usuario, el entorno en el que desarrolla el trabajo y las tareas que realiza. Además debe mirar el mantenimiento del sistema, asistencia al usuario y la documentación.

4.9.17 Principios del DCU

- El usuario deber ser el que inicie las acciones y controle las tareas.
- El sistema deber ser lo más interactivo posible.
- La accesibilidad de la información y de las opciones van a reducir la carga mental de trabajo del usuario.
- Las metáforas familiares proporcionan una interfaz intuitiva.
- Se asocia un significado con un objeto mejor que con un comando.

- Posibilitar la recuperación de los errores, permitir la reversibilidad y recuperar las acciones.
- La retroalimentación apropiada del sistema, dar respuesta a las acciones del usuario en poco tiempo.
- El estado del sistema (esperando entrada, comprobando, transfiriendo datos...) debería estar siempre disponible para el usuario.
- Proporcionar un entorno agradable para que el usuario entienda la información presentada.
- La interfaz debe ser simple, fácil de aprender y usar.
- Controlar la cantidad de información.

4.9.18 Arquitectura de la información

La información que se dará a los usuarios y el sistema de organización de ésta, se determinan de acuerdo a las necesidades.

4.9.19 Accesibilidad

La accesibilidad de un sitio Web significa que la gente puede percibir, entender, navegar, y trabajar mutuamente con el sitio.

La fabricación de un sitio Web accesible puede ser simple o compleja, depende de muchos factores como el tipo de contenido, el tamaño, la complejidad del sitio, las herramientas y el ambiente de desarrollo.

4.9.20 Usabilidad

La definición de usabilidad **[2]** según la norma ISO 9241-11 (1998) “medida en la cual un producto puede ser usado por usuarios específicos para lograr objetivos específicos con eficacia, eficiencia y satisfacción, en un contexto de uso”. La usabilidad también esta asociada al grado de aceptación de un producto y a la capacidad de una herramienta software para ser comprendida, aprendida, usada y atractiva para el usuario.

[2]Tomado de: http://www.alzado.org/articulo.php?id_art=355

La usabilidad es una característica de la ergonomía, ayuda a que la búsqueda se haga de una forma sencilla analizando el comportamiento humano, y los pasos necesarios para ejecutarla de una forma eficaz. En el diseño de un sistema, hay que tener en cuenta la presentación de la información, la funcionalidad de la aplicación y la arquitectura del software; también se deben considerar los principios de diseño centrado en el usuario y las teorías de fundamentos del diseño.

4.9.21 Estándares internacionales

La Internacional Standard Organization (ISO) [3], es una organización que define los estándares para procedimientos, procesos, requerimientos y atributos de productos y servicios. Algunos estándares que tratan sobre el aspecto ergonómico de sistemas informativos especialmente la usabilidad y el diseño centrado en el usuario. La European Usability Support Centres (EUSC) clasifica los estándares internacionales relacionados con el diseño centrado en el usuario (describen el perfil del usuario, sus tareas y contexto de uso, también evalúan la usabilidad del producto desde el desempeño y satisfacción del usuario) en dos grupos:

- **Estándares internacionales orientados a proceso:** especifican requerimientos para el diseño de procedimientos y procesos.
- **Estándares internacionales orientados a producto:** especifican requerimientos para el diseño y desarrollo de interfaces de usuario.

4.9.22 Métodos de usabilidad

Los métodos de usabilidad son diferentes técnicas que se usan para evaluar la usabilidad de una aplicación. Los métodos más relevantes son:

4.9.23 Evaluación heurística

Según el creador de la técnica Jacob Nielsen, es “el nombre genérico de un grupo de métodos basados en evaluadores expertos que examinan aspectos relacionados con la usabilidad de una interfaz de usuario”. Este método consiste en una evaluación teórica de los hechos observados, es más eficiente para proponer posibles soluciones.

[3] Tomado de: <http://www.iso.ch/iso/en/ISOOnline.frontpage>

4.9.24 Test de usabilidad

Es una medida concreta y objetiva de una herramienta o sistema, tomada a partir de usuarios verdaderos con tareas reales. Este método toma medidas empíricas que muestran donde están los problemas.

Los test de usabilidad o test de los aspectos ergonómicos, son un conjunto de técnicas y métodos usados con el propósito de garantizar un buen diseño centrado en el usuario, definición de Rubin (1994).

El planteamiento y diseño del test de usabilidad garantiza la calidad del producto final. El test esta relacionado con la organización del contenido, la estructura y la operación de la aplicación y la interacción hombre – ordenador. Según los estudios de Nielsen (1993) y Rubin (1998), el test debe ser aplicado a un grupo de usuarios (mínimo 4, máximo 10) seleccionados, dentro de su entorno de trabajo. Se debe diseñar el plan del test, guardar toda la información obtenida durante la prueba; para luego analizarla e interpretarla. Se darán las conclusiones, para recomendar posibles cambios y así poder presentar y publicar los resultados finales.

4.9.25 Tipos de test

Rubin (1994) sugiere cuatro tipos de test asociados a diferentes fases del desarrollo de un producto: exploratorio, evaluación de operaciones y aspectos del producto; validación y comparación.

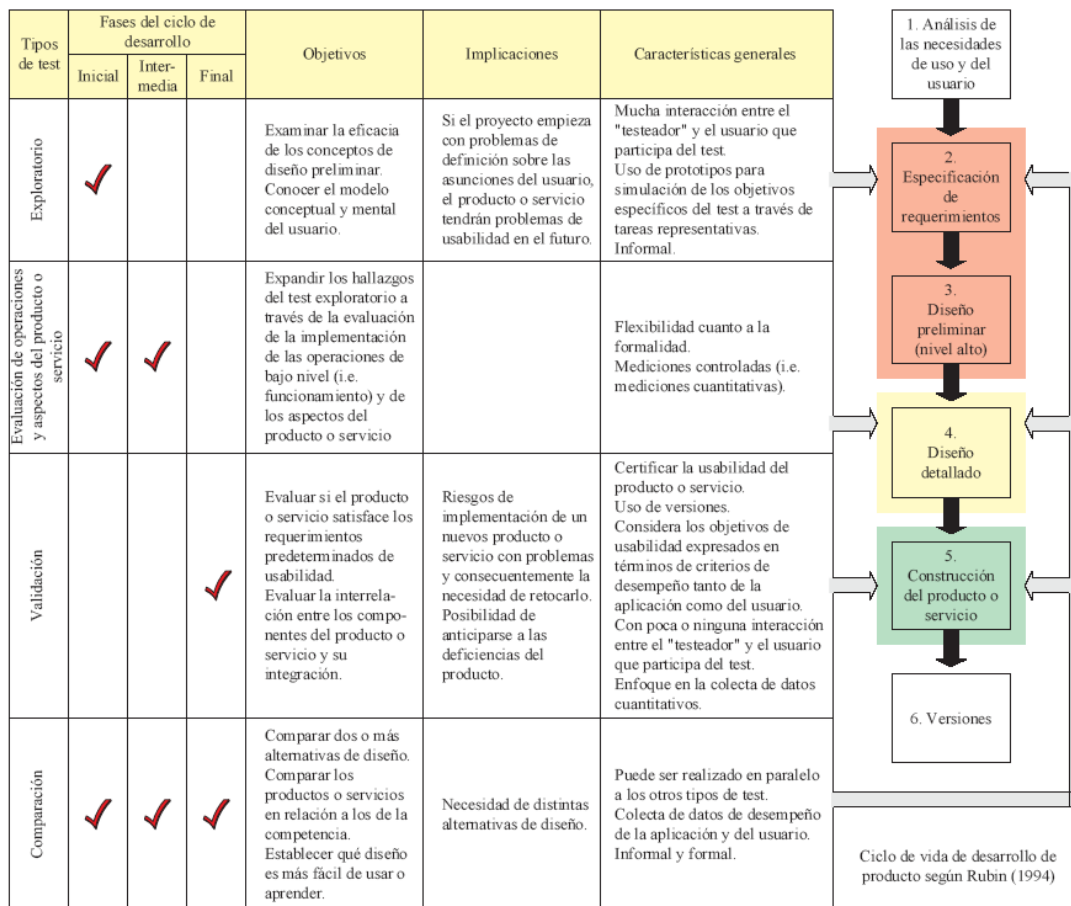


Figura 7. Interrelación entre los tipos de test y las fases de desarrollo de un producto

4.9.26 Principios de la usabilidad

Utilidad, es la capacidad que tiene una herramienta para ayudar a cumplir tareas específicas.

Facilidad de uso, está en relación directa con la eficiencia o efectividad, medida como velocidad o cantidad de posibles errores. Una herramienta fácil de usar permitirá al usuario efectuar más operaciones por unidad de tiempo o hacerlas en menor tiempo y disminuir posibles errores, ya que ninguna herramienta es perfecta.

Intuitiva, para que la realización de las tareas se haga de forma continua, lógica y sencilla.

Rápida, permitiendo al usuario la visualización de todos los elementos, localizar fácilmente la información que busca y realizar en pocos pasos las tareas.

Eficiencia, una vez aprendido debe ser fácil manejarlo, para elevar el nivel de productividad. La Ley de Fitts indica que el tiempo para alcanzar un objetivo con el mouse esta en función de la distancia y el tamaño del objetivo. A menor distancia y mayor tamaño más facilidad para usar un mecanismo de interacción.

Flexibilidad, variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información, las diferentes vías para realizar las tareas. El usuario sólo espera poder introducir información, seleccionar opciones y dar clic en los botones.

Libre de errores o informar al usuario que tipo de error se ha producido y porqué, para que se recupere con rapidez.

Facilidad de aprendizaje, es una medida del tiempo requerido para trabajar con eficiencia al usar la herramienta, para que el usuario reconozca el sistema de navegación y operativo de la herramienta.

Apreciación, es una medida de las percepciones, opiniones, sentimientos y actitudes generadas en el usuario por la herramienta. Un usuario al que no le gusta una interfaz puede generar más errores o tardar más en aprenderla.

Agradable y satisfacción, interfaz diseñada con colores adecuados para proporcionar un entorno de trabajo visualmente relajado y a su vez atractivo.

4.10 Bases de Datos

4.10.1 Base de datos relacionales

Los sistemas relacionales son importantes porque ofrecen muchos tipos de procesos de datos, como: simplicidad y generalidad, facilidad de uso para el usuario final, períodos cortos de aprendizaje y las consultas de información se especifican de forma sencilla.

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base

de datos. La información puede ser recuperada o almacenada mediante consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

En este modelo toda la información se representa a través de arreglos bidimensionales o tablas. Las tablas son un medio de representar la información de una forma más compacta y es posible acceder a la información contenida en dos o más tablas.

Las bases de datos relacionales están constituidas por una o más tablas que contienen la información ordenada de una forma organizada. Cumplen las siguientes leyes básicas:

- Generalmente, contendrán muchas tablas.
- Una tabla sólo contiene un número fijo de campos.
- El nombre de los campos de una tabla es distinto.
- Cada registro de la tabla es único.
- El orden de los registros y de los campos no está determinados.
- Para cada campo existe un conjunto de valores posible.

4.10.2 Diseño de las bases de datos relacionales

El primer paso para crear una base de datos, es planificar el tipo de información que se quiere almacenar en la misma, teniendo en cuenta dos aspectos: la información disponible y la información que se necesita.

La planificación de la estructura de la base de datos, en particular de las tablas, es vital para la gestión efectiva de la misma. El diseño de la estructura de una tabla consiste en una

descripción de cada uno de los campos que componen el registro y los valores o datos que contendrá cada uno de esos campos.

Los campos son los distintos tipos de datos que componen la tabla, por ejemplo: nombre, apellido, domicilio. La definición de un campo requiere: el nombre del campo, el tipo de campo, el ancho del campo, etc.

Los registros constituyen la información que va contenida en los campos de la tabla, por ejemplo: el nombre del paciente, el apellido del paciente y la dirección de este. Generalmente los diferentes tipos de campos que se pueden almacenar son los siguientes: Texto (caracteres), Numérico (números), Fecha / Hora, Lógico (informaciones lógicas si/no, verdadero/falso, etc.), imágenes.

El principal aspecto a tener en cuenta durante el diseño de una tabla es determinar claramente los campos necesarios, definirlos en forma adecuada con un nombre especificando su tipo y su longitud.

5. METODOLOGÍA

FDD (*Feature Driven Development*) – Desarrollo por Rasgos [4]

Es una metodología de desarrollo rápido, especialmente buena cuando se aplica a proyectos Web en los cuales prima el concepto de **modularidad**, siendo el caso exacto del proyecto a realizar. El elemento esencial de FDD son los rasgos. Un rasgo es un pequeño trozo de funcionalidad, un resultado tangible el cual puede ser realizado en 2 semanas o menos. Un rasgo sigue la siguiente estructura:

<Acción> el | la <resultado> <por | de | a | desde | para> un (una) <objeto>

Ejemplo: Calcular el interés para la cuenta del banco.

Conjuntos de Rasgos

En FDD, los rasgos son agrupados en conjuntos. Un conjunto de rasgos se entiende como una pieza de trabajo potencialmente útil para el cliente y el cual podría definirse como un componente. Es un grupo de características de negocio relacionadas.



Figura 8. Metodología

[4] Tomado de <http://www.featuredrivendevelopment.com/>

El Proceso de desarrollo se divide en 5 etapas:

5.1 Etapa 1 – Desarrollar un Modelo General

Este modelo debe ser más ancho que profundo. En otras palabras, el objetivo es construir un modelo de análisis de la función de negocio para todo el requerimiento sin ser demasiado específico. No hay necesidad de desarrollar diagramas de secuencia o largas listas de atributos para ello.

5.2 Etapa 2 – Hacer una lista de Rasgos

La segunda etapa es realizar la lista de rasgos de una combinación de los documentos de requerimientos y el modelo de Objeto. El modelo de Objeto actúa como una fuente para nombrar y ayudar a resolver conflictos de nomenclatura. Es común que en un documento de requerimientos se llame de diferentes maneras cosas similares o incluso idénticas. Si la lista de rasgos es desarrollada correctamente, cada rasgo guiará hacia la **<acción>** como el nombre del método, el **<resultado>** como el valor de retorno y el **<objeto>** como la clase en la cual el método está ubicado.

5.3 Etapa 3 - Desarrollo de un Plan Detallado

La etapa de “plan por rasgo” involucra agrupar los rasgos en conjuntos de rasgos apropiados y temas. En ésta etapa también puede realizarse una estimación del tamaño y tiempo de desarrollo de cada rasgo dirigido a las fases de construcción. Un plan detallado desarrollado apropiadamente entrega estimados increíblemente precisos para la construcción.

5.4 Etapa 4 – Diseño por Rasgo

Los rasgos son agrupados en conjuntos manejables por pequeños equipos de trabajo de 2 o 3 programadores. El conjunto de rasgos debe ser agrupado mediante un diagrama de actividades. Ése diagrama deberá ser revisado por cada miembro del equipo de trabajo encargado de ese rasgo y comparado contra los requerimientos, Las líneas guías para la arquitectura y desarrollo y una lista de chequeo de diseño. Una vez el rasgo apruebe toda la revisión pasa a la etapa de desarrollo.

5.5 Etapa 5 – Desarrollo por Rasgo

El código para cada rasgo es escrito por los miembros del equipo encargado, quienes poseen las clases de negocio afectadas por la funcionalidad del rasgo. Una vez completado, el código es sometido a una revisión. Cuando las pruebas son completadas y el código pasa la revisión, es etiquetado para la inclusión en el sistema.

6. MODELO DE APLICACIÓN

6.1 Aplicaciones Web

Una aplicación Web [5] es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet, utilizando un cliente que popularmente suele ser un navegador Web (browser).

Las aplicaciones Web son el futuro para los sistemas compartidos mientras que las aplicaciones tradicionales (aplicaciones de escritorio) ya llegaron a una madurez, las aplicaciones Web empiezan a crecer en tecnología y funcionalidad.

Este nuevo paradigma trae consigo nuevas metodología en el ciclo de vida del software, las diferencias en las aplicaciones Web conllevan a ventajas como:

1. Mayor potencial de mercadeo: Las aplicaciones Web son independientes de la plataforma (Windows, Mac, Linux) y corren sobre la red, lo cual aumenta el nicho de mercado y la facilidad para llegar a este.
2. Reducción de riesgo: Las aplicaciones Web concentra la lógica y la información del negocio en un solo sitio (el servidor Web), lo cual reduce los puntos de riesgo y facilita su mantenimiento.
3. Mayor cercanía con los clientes: En las aplicaciones Web todos los usuarios tienen acceso a la aplicación final y son al mismo tiempo probadores de mi sistema, ayudando a su retroalimentación continua.
4. Información en tiempo real para la toma de decisiones: En las aplicaciones Web Los usuarios manejan la misma información independiente de la Terminal.

Estos beneficios se ven reflejados prácticas como:

- No necesitan de instalación en la Terminal final.
 - Fácil acceso y usabilidad.
 - Mejoramiento Continuo gracias a su facilidad de actualización.
 - La actualización de las partes, independiente del sistema total.
 - Facilidad para compartir lógica y datos.
-

[5]Tomado de O' Reilly Radar, Web 2.0, Principles and Best Practices

Otra gran sorpresa de las aplicaciones Web, son las tecnologías que de ella han nacido, tecnologías novedosas como los servicios Web (colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones), Web semántica (añadir meta datos semánticos a la World Wide Web, describiendo el contenido, el significado y la relación de los datos, para su mejor clasificación); Son tecnologías que empiezan a mostrar sus beneficios y que nos están llenando de sorpresas.

Rich Internet Application (RIA)

Rich Internet Applications **[6]** (Aplicaciones Ricas de Internet) es un nuevo tipo de aplicación con más ventajas que las tradicionales aplicaciones Web. Esta surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales.

Normalmente en las aplicaciones Web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces, a recargar la misma página con un mínimo cambio.

Otra de las desventajas de las tradicionales aplicaciones Web es la poca capacidad multimedia que posee. Para ver un vídeo se tiene que usar un programa externo para su reproducción. Los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una Base de Datos o de otros ficheros externos.

Las capacidades multimedia son totales gracias a que estos entornos tienen reproductores internos y no hace falta ningún reproductor del Sistema Operativo del usuario.

[6] Tomado de DKTek Software Corporation, Web-based Applications vs. Traditional Software

6.2 Tecnologías de Servidor

Un servidor es el programa encargado de resolver las peticiones que los clientes realizan en la red, existen varias tecnologías de servidor, entre las cuales se destacan:

6.2.1 ColdFusion

Solución implementada por macromedia, utiliza una programación por medio de tags (como el html), es muy rápido, tiene soporte para las plataformas windows, linux, mac y Solaris, depende solo del soporte dado por su casa matriz adobe, dado que es de carácter privativo, lo cual lo tiene un poco atrasado en funcionalidades, y en la difusión para su aprendizaje. Descartamos ColdFusion para la realización de éste proyecto debido a su carácter privativo, alto costo y la dificultad para integrarlo a las otras tecnologías seleccionadas.

6.2.2 Jsp

Solución implementada por Sun Microsystems, utiliza el lenguaje java y una extensa investigación en estándares del software para brindarnos un perfecto paradigma de objetos y un diseño de 3 capas correctamente definido, como es de imaginarse de la casa Sun y su lema para java, “codifica una vez, corre donde sea”, Jsp tiene soporte para todas las plataformas populares, desde el principio a tenido una amplia variedad de librerías funcionales gracias al carácter open-source de los proyectos realizados en su padre java, esto se encuentra en aumento, desde que liberaron el código de la mismísima plataforma, estas características lo han convertido en los preferidos de las corporaciones, pero lo han alejado de los usuarios comunes, debido a que la implementación de toda esa teoría del software hace que la curva de aprendizaje de esta tecnología sea casi una cultura de vida. Descartamos Jsp para la realización de nuestro proyecto debido a que la curva de aprendizaje es muy elevada y extendería el proceso de desarrollo.

6.2.3 Asp

Solución implementada por Microsoft, nacido del VisualBasic, tiene una excelente interoperabilidad entre los productos de Microsoft, como los son ODBC, VBscript, además de un gran soporte dado por su casa matriz, su IDE (Internas de desarrollo) y características adicionales, facilitan el uso, y tiene una curva de aprendizaje que permite una escalabilidad moderada, cabe destacar que asta ahora se esta trabajando en el soporte para otras plataformas diferentes a Windows, pero las demás tecnologías le tienen ventaja en este

campo. Descartamos Asp para la realización de nuestro proyecto debido a su carácter privativo, costo y poca portabilidad.

6.2.4 Perl

El padre de todos los software de servidores, a diferencia de lo que muchos creen en este caso la madurez es un punto fuerte, y para iniciar es una buena plataforma, tiene apoyo de grupos opensource, pero se considera que su hijo mas cercano, PHP, tiene mas ventajas que el.

6.2.5 PHP

PHP como todo software opensource nació en un garaje, de los brazos de perl y con un código basado en C, este producto opensource, de la mano de Linux, tiene implementaciones en las plataformas mas conocidas, el soporte de millones de usuarios en la Web, con infinidad de librerías funcionales, y miles de empresas que desarrollan y corren sus aplicaciones en el, con una curva de aprendizaje a la medida de la necesidad del usuario, e implementaciones en cuanta tecnología haya tocado el opensource, php es un servidor muy adaptable y su vida comercial esta apoyada en su gran difusión en la Web actual.

6.2.6 Por qué se utiliza PHP [7]

La respuesta rápida sería decir que todos los lenguajes indicados tienen ventajas y defectos, y en definitiva la elección será siempre subjetiva. El mejor lenguaje será aquel que mejor se encaje a las preferencias de cada usuario, y sirva mejor al fin que se persigue.

No obstante hay algunas características que pueden servir para decidirse por PHP. En primer lugar, PHP es un lenguaje relativamente nuevo, diseñado desde cero con el fin único de diseñar aplicaciones Web. Esto quiere decir que las tareas más habituales en el desarrollo de estas aplicaciones, pueden hacerse con PHP de forma fácil, rápida y efectiva. Otros lenguajes, como ASP (VBS), Perl o Java serán sin duda más completos y potentes, pero no fueron diseñados con este enfoque especializado.

En segundo lugar PHP resulta fácil de aprender para recién llegados al mundo de la programación. Es fácil dar los primeros pasos y ver los resultados rápidamente. Perl es un completo lenguaje de programación, difícil para el novato.

En tercer lugar, PHP es un lenguaje multiplataforma, y no propietario. Un script PHP normal puede ejecutarse sin cambiar ni una sola línea de código en cualquier servidor que interprete PHP, es decir, en servidores Windows, Linux etc. Por otra parte ASP, es un lenguaje propietario de Microsoft y solo puede ejecutarse en servidores Microsoft. Perl si puede ejecutarse bajo Windows y Unix, pero antes hay que realizar labores de adaptación de cada script al concreto sistema operativo.

PHP esta en el entorno open source. Esto quiere decir que en su desarrollo hay un proceso de colaboración que hace que se tengan inmediatamente disponibles, de forma gratuita, una enorme cantidad de recursos: el lenguaje en si, el servidor para ejecutarlo, manuales y tutoriales.

PHP combina excelentemente con otras herramientas, como son el servidor Apache y la base de datos Mysql (o Postgres), todas ellas gratuitas. ASP necesita, para algunas funcionalidades (algunas realmente básicas) acudir a programas y extensiones comerciales, de terceras empresas. Pagando, por supuesto.

PHP no solo es bonito y barato. También es bueno. Lo que hace lo hace de forma efectiva y rápida. La opinión más común es que es sensiblemente más rápido que ASP. Como desventaja sobre el ASP tiene una peor integración con el paquete Office de Microsoft, pero esto es totalmente lógico, puesto que Microsoft en su empeño de mantener su monopolio, no da las especificaciones del mismo e incluso crea artificialmente problemas entre sus propias versiones para forzar al usuario a comprar las nuevas versiones de su suite.

[7] Tomado de <http://training.gbdirect.co.uk/courses/php/>

6.3 Comparativa de Gestores de Bases de Datos

6.3.1 PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características interesantes, entre las más relevantes para el proyecto a desarrollar incluyen manejo de objetos, Almacenamiento y búsqueda de XML de forma nativa, además de ofrecer Tamaños ilimitados de la base de datos y un rendimiento muy elevado en las consultas. Permite instalar el sistema de administración en entorno Web ó aplicación de escritorio. Es libre y con miles de fuentes de documentación, aunque no es muy fácil de manejar si no se es un usuario experto.

6.3.2 Oracle

Es el sistema de Bases de datos de tipo propietario con las funciones más avanzadas, está enormemente adelantado a otros sistemas como MySQL y PostgreSQL debido a que además de incluir todas las funciones que manejan éstos, incluyendo el manejo nativo de XML con más de 1 año de ventaja, añade todo un sistema de “Advisors” (Consejeros) que son Monitores del sistema que permiten una administración sencilla en medio de un complejo sistema. Al igual que PostgreSQL, hace uso del modelo Objeto-Relacional para almacenar los datos, un punto fuerte a favor. Tiene en contra que se debe tener licencia para usarlo, la cual es muy costosa.

6.3.3 MySQL

MySQL es un sistema de gestión de base de datos, multiplataforma, multihilo y multiusuario con más de seis millones de instalaciones, lo cual lo hace el más popular. En éstos momentos no contempla manejo de XML de forma nativa, sin embargo se planea implementar éstas funciones en una próxima versión. Tiene un límite máximo de 50 millones de registros, lo cual es más que suficiente para una base de datos pequeña o mediana. Su rendimiento es muy bueno, aunque no comparable al de PostgreSQL u Oracle. En cuanto a facilidad de uso, MySQL tiene una enorme ventaja con respecto a PostgreSQL, está pensado para que una persona con conocimientos muy básicos de bases de datos pueda manejarlo.

6.3.4 Porque se escogió MySQL

Se escogió MySQL como sistema de gestión de bases de datos para el proyecto, por su gran facilidad de uso, tiene todas las capacidades necesarias para la ejecución del proyecto, y por supuesto, la gran documentación existente para ésta herramienta. El problema principal con PostgreSQL es su panel de administración el cual no es sencillo de manejar y en vez de facilitar, dificulta muchas de las tareas más sencillas. Oracle tiene en su contra el enorme costo de licencia de uso, totalmente inaceptable.

6.4 Comparativa de Tecnologías de Desarrollo Web para aplicaciones Interactivas

6.4.1 AJAX (Asynchronous JavaScript And XML) [8]

Es una tecnología que permite crear aplicaciones Web con una elevada usabilidad y rendimiento debido al uso que le da al objeto **XMLHttpRequest**, creado por Microsoft en 1998, pero que había tenido muy poca participación hasta 2003, cuando Google introduce su sistema de correo Gmail basado en ésta técnica y que funciona intercambiando datos asincrónicamente entre el servidor Web y el navegador del cliente.

A Favor:

- Basada en HTML, en realidad en XHTML, por lo que debería ser fácilmente estandarizable.
- Accesible. La integración de lectores de pantalla, las posibilidades de modificar el tamaño del texto por el usuario, etc.
- Integración con CSS. Es, por tanto, bastante sencillo separar datos de presentación, por lo que las aplicaciones son más escalables.
- Ligereza. Se supone que una página con sólo HTML es más ligera que un front-end en flash, aunque en realidad el Javascript necesario para funcionar no es tan ligero, por lo que finalmente se depende de mecanismos de caché en servidor.
- Javascript. Lenguaje interpretado en el navegador, sencillo, orientado a objetos (según a quién se le pregunte), especialmente orientado a manipular el DOM de las páginas.
- Motores de búsqueda. Al basar los interfaces en HTML, son fácilmente indexados por los motores de búsqueda. Siempre y cuando su contenido no sea producto de una petición de datos a servidor, claro está.

En contra:

- El usuario puede haber deshabilitado el Javascript en el navegador.
- Se depende de un objeto propiedad de Microsoft para las comunicaciones asíncronas. Lo malo de esto no es que ese objeto sea propiedad de Microsoft o de cualquier otro, sino que es propiedad de alguien.
- Por mucho que se intente, el resultado gráfico final no es comparable al de una Web en Flash.

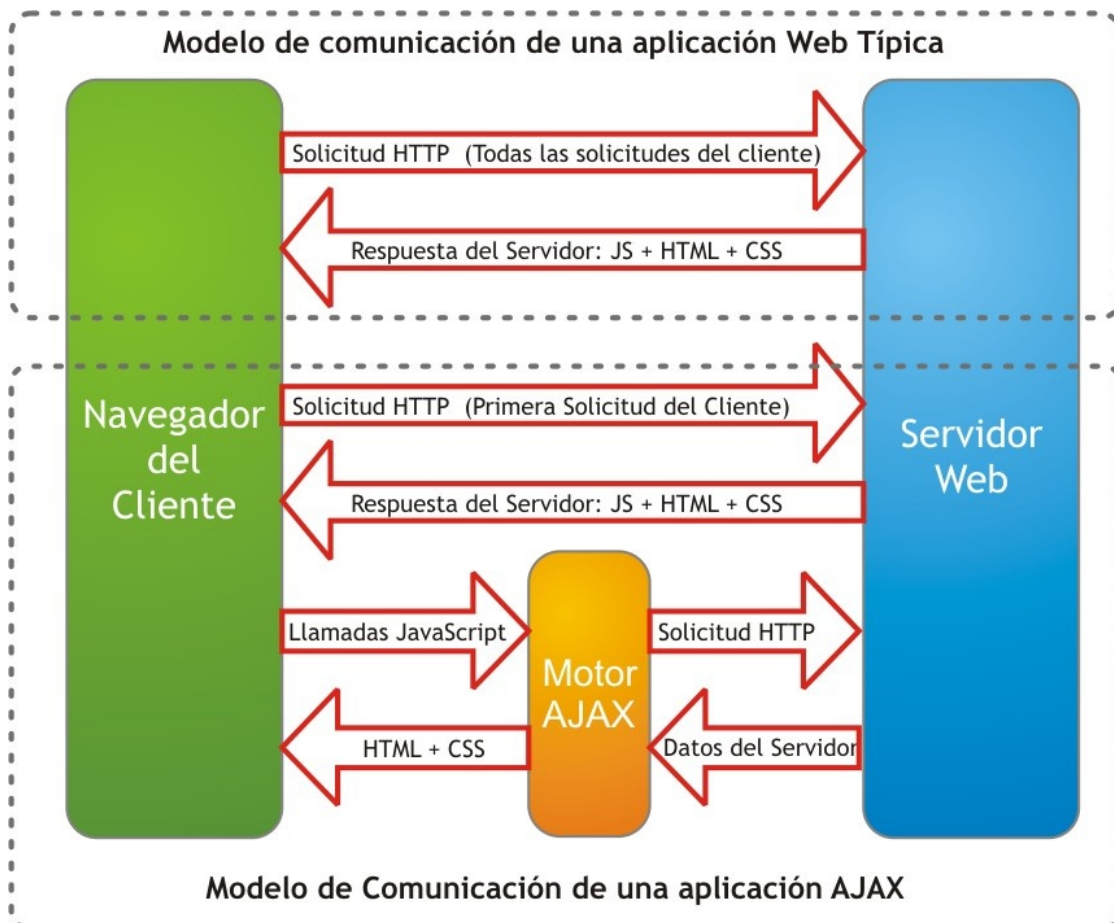


Figura 9. Modelo de Comunicación

[8] Tomado de <http://mosaic.uoc.edu/articulos/ctardaguila0706.html>

6.4.2 Adobe Flash [9]

Consiste en un programa de máquina virtual utilizado para ejecutar los archivos .swf generados con la Suite Flash de Adobe, el cual puede contener información de gráficos vectoriales e imágenes raster, sonido, código de programa, flujo de vídeo y audio bidireccional.

A Favor:

- Player casi universal. Prácticamente todos los ordenadores del mundo lo tienen instalado, aunque eso no implica que el usuario de un ordenador en concreto tenga instalada la última versión del mismo, ni que no lo haya deshabilitado utilizando alguna extensión del navegador.
- Calidad visual. Sobre todo con el avance en el manejo de los textos en las últimas versiones del player de Flash.
- Animación, multimedia. La única forma solvente de incluir animaciones, contenido interactivo, vídeo, audio.
- Lenguaje de programación de alto nivel orientado a objetos, aunque también sea interpretado en el player.

En contra:

- Se depende de un plug-in propietario.
- Se tiende a abusar de las animaciones.
- Integración con el botón de volver del navegador no es nada sencilla.
- Poco accesible, a pesar de que las últimas versiones de Flash suponen una mejora sensible en este aspecto.
- Está inevitablemente unido a dos cosas: el botón de "Skip intro" y los banners invasivos.

[9] Tomado de <http://www.macromedia.com/software/flash/about/>

6.4.3 XML (*eXtensible Markup Language*)

XML proviene de un lenguaje inventado por IBM en los años setenta, llamado [GML](#) (*Generalized Markup Language*), que surgió por la necesidad que tenía la empresa de almacenar grandes cantidades de información. Este lenguaje gustó a la ISO, por lo que en 1986 trabajaron para normalizarlo, creando SGML (*Standard Generalized Markup Language*), capaz de adaptarse a un gran abanico de problemas. A partir de él se han creado otros sistemas para almacenar información.

En el año 1989 Tim Berners Lee creó la web, y junto con ella el lenguaje HTML. Este lenguaje se definió en el marco de SGML y fue de lejos la aplicación más conocida de este estándar. Los navegadores web sin embargo siempre han puesto pocas exigencias al código HTML que interpretan y así las páginas web son caóticas y no cumplen con la sintaxis. Estas páginas web dependen fuertemente de una forma específica de lidiar con los errores y las ambigüedades, lo que hace a las páginas más frágiles y a los navegadores más complejos.

Otra limitación de SGML es que cada documento pertenece a un vocabulario fijo, establecido por el DTD. No se pueden combinar elementos de diferentes vocabularios. Asimismo es imposible para un intérprete (por ejemplo un navegador) analizar el documento sin tener conocimiento de su gramática (del DTD). Por ejemplo, el navegador sabe que antes de una etiqueta **<div>** debe haberse cerrado cualquier **<p>** previamente abierto. Los navegadores resolvieron esto incluyendo lógica ad hoc para el HTML, en vez de incluir un analizador genérico. Ambas opciones de todos modos son muy complejas para los navegadores. Se buscó entonces definir un subconjunto del SGML que permita:

- Mezclar elementos de diferentes lenguajes. Es decir que los lenguajes sean extensibles.
- La creación de analizadores simples, sin ninguna lógica especial para cada lenguaje.
- Empezar de cero y hacer hincapié en que no se acepte nunca un documento con errores de sintaxis.

Para hacer esto XML deja de lado muchas características de SGML que estaban pensadas para facilitar la escritura manual de documentos. XML en cambio está orientado a hacer las cosas más sencillas para los programas automáticos que necesiten interpretar el documento.

VENTAJAS

- Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan bugs y se acelera el desarrollo de la aplicación.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

ESTRUCTURA

La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible. Que la información sea estructurada quiere decir que se compone de partes bien definidas, y que esas partes se componen a su vez de otras partes. Entonces se tiene un árbol de pedazos de información. Ejemplos son un tema musical, que se compone de compases, que están formados a su vez con notas. Estas partes se llaman *elementos*, y se las señala mediante etiquetas.

Una etiqueta consiste en una marca hecha en el documento, que señala una porción de este como un elemento, un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma **<nombre>**, donde *nombre* es el nombre del elemento que se está señalando.

DOCUMENTOS XML BIEN FORMADOS

- Los documentos denominados como "bien formados" (del inglés *well formed*) son aquellos que cumplen con todas las definiciones básicas de formato y pueden, por lo tanto, ser analizados correctamente por cualquier analizador sintáctico (*parser*) que cumpla con la norma. Se separa esto del concepto de validez que se explica más adelante.
- Los **documentos han de seguir una estructura** estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.
- Los documentos XML sólo permiten un elemento raíz del que todos los demás sean parte, es decir, solo puede tener un elemento inicial.
- Los valores atributos en XML siempre deben estar encerrados entre comillas simples o dobles.
- El XML es sensible a mayúsculas y minúsculas. Existe un conjunto de caracteres llamados espacios en blanco (espacios, tabuladores, retornos de carro, saltos de línea) que los procesadores XML tratan de forma diferente en el marcado XML.
- Es necesario asignar nombres a las estructuras, tipos de elementos, entidades, elementos particulares, etc. En XML los nombres tienen alguna característica en común.
- Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador XML espera entender. El resto del documento entre marcas son los datos entendibles por las personas.

Partes de un documento XML

Un documento XML está formado por el prólogo y por el cuerpo del documento.

Prólogo

Aunque no es obligatorio, los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento y otras cosas.

El prólogo contiene:

- una declaración XML. Es la sentencia que declara al documento como un documento XML.
- una declaración de tipo de documento. Enlaza el documento con su DTD, o el DTD puede estar incluido en la propia declaración o ambas cosas al mismo tiempo.
- uno o más comentarios e instrucciones de procesamiento.

Cuerpo

A diferencia del prólogo, el cuerpo no es opcional en un documento XML, el cuerpo debe contener un y solo un elemento raíz, característica indispensable también para que el documento esté bien formado.

Elementos

Los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos.

Atributos

Los elementos pueden tener atributos, que son una manera de incorporar características o propiedades a los elementos de un documento. Deben de ir entre comillas.

Entidades predefinidas

Entidades para representar caracteres especiales para que no sean interpretados como marcado en el procesador XML.

Secciones CDATA

Es una construcción en XML para especificar datos utilizando cualquier carácter sin que se interprete como marcado XML solo se utiliza en los atributos no confundir con 2(#PCDATA) que es para los elementos.

Comentarios

Comentarios a modo informativo para el programador que han de ser ignorados por el procesador.

Los comentarios en XML tienen el siguiente formato:

`<!-- Esto es un comentario --->`

`<!-- Otro comentario -->`

6.4.4 CSS



Figura 10. Hoja de Estilos CSS

Las **hojas de estilo en cascada** (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la *estructura* de un documento de su *presentación*.

Por ejemplo, el elemento de HTML <H1> indica que un bloque de texto es un encabezamiento y que es más importante que un bloque etiquetado como <H2>. Versiones más antiguas de HTML permitían atributos extra dentro de la etiqueta abierta para darle formato (como el color o el tamaño de fuente). No obstante, cada etiqueta <H1> debía disponer de la información si se deseaba un diseño consistente para una página, y además, una persona que lea esa página con un [navegador](#) pierde totalmente el control sobre la visualización del texto.

Cuando se utiliza CSS, la etiqueta <H1> no debería proporcionar información sobre como va a ser visualizado, solamente marca la estructura del documento. La información de estilo separada en una hoja de estilo, especifica cómo se ha de mostrar <H1> : color, fuente, alineación del texto, tamaño, y otras características no visuales como definir el volumen de un sintetizador de voz (véase Sintetización del habla), por ejemplo.

La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo “style”.

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.

Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.

Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser “leída” por un sintetizador de voz.

El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

Hay varias versiones: CSS1 y CSS2, con CSS3 en desarrollo por el World Wide Web Consortium (W3C). Los navegadores modernos implementan CSS1 bastante bien, aunque existen pequeñas diferencias de implementación según marcas y versiones de los navegadores. CSS2, sin embargo, está solo parcialmente implementado en los más recientes.

Diagramado de página en CSS

Antes de que estuviera disponible CSS, la única forma de componer espacialmente una página era el uso de tablas. Aunque es una técnica cómoda y versátil, se está usando un elemento con una semántica particular, que es la de expresar información tabular, solamente por su efecto de presentación. La introducción de CSS ha permitido en muchos casos reemplazar el uso de tablas. Sin embargo CSS todavía no permite la versatilidad que ofrecían las tablas, lograr un diagramado de una página compleja suele ser una tarea difícil en CSS y las diferencias entre navegadores dificultan aún más la tarea. Se espera que futuros desarrollos en CSS3 resuelvan esta deficiencia y hagan de CSS un lenguaje más apto para describir la estructura espacial de una página.

6.4.5 JavaScript

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

1JavaScript permite realizar efectos sobre las páginas Web para presentar así el contenido de las mismas dinámicamente o ejecutar instrucciones en respuesta a los eventos del usuario.

6.4.6 Porqué se decide usar Backbone (Motor AJAX)

Se escogió Backbone debido a que es una gran mezcla de tecnologías (JavaScript, XML y CSS). En el caso de AJAX, existe un producto con librerías listas para usar, llamado Backbone [10] y que tiene una edición gratuita para uso no comercial, que incluye una gran

cantidad de controles preprogramados, que se manejan mediante sencillas líneas de código en XML. El objetivo de Backbase es permitirle a los desarrolladores Web crear Aplicaciones Ricas en Internet (RIA en inglés). En una RIA el procesamiento es hecho del lado del cliente, permitiéndole construir dinámicos e interactivos sitios Web que se comportan más como un escritorio que una aplicación Web tradicional. Para Crear una RIA se necesita una solución que pueda tratar con las limitaciones del navegador y las limitaciones del lenguaje HTML (HyperText Markup Language) y lo hace aprovechando las características del XML, CSS y JavaScript. El renderizado de los datos (presentación) es generado del lado del cliente con una actualización parcial, es decir, sólo lo necesario y no una carga de página completa. También Posee plugins para usar en varios marcos de trabajo como .net, Adobe Dreamweaver y Eclipse.

- *Backbase Presentation Client (BPC)* – Cliente de Presentación de Backbase.

El BPC es un robusto motor AJAX de lado del cliente que despliega una RIA. El BPC interpreta el BXML (Backbase eXtensible Markup Language, en inglés) y lo transforma en la correspondiente estructura DOM (Document Object Model – Modelo de objeto del documento).

- *Backbase eXtensible Markup Language – BXML* Es el Lenguaje declarativo basado en etiquetas de Backbase que extiende el XML. Las etiquetas y atributos BXML son usados para describir componentes de la interfaz de usuario, estructuras de datos, comportamientos, manejo de eventos y ejecución de comandos. Se pueden incluso crear controles, eventos, comportamientos y atributos propios.

[10] Tomado de <http://www.backbase.com>

Cómo funciona Backbase

La siguiente ilustración provee un vistazo de las capas en el software Backbase y sus subcomponentes:

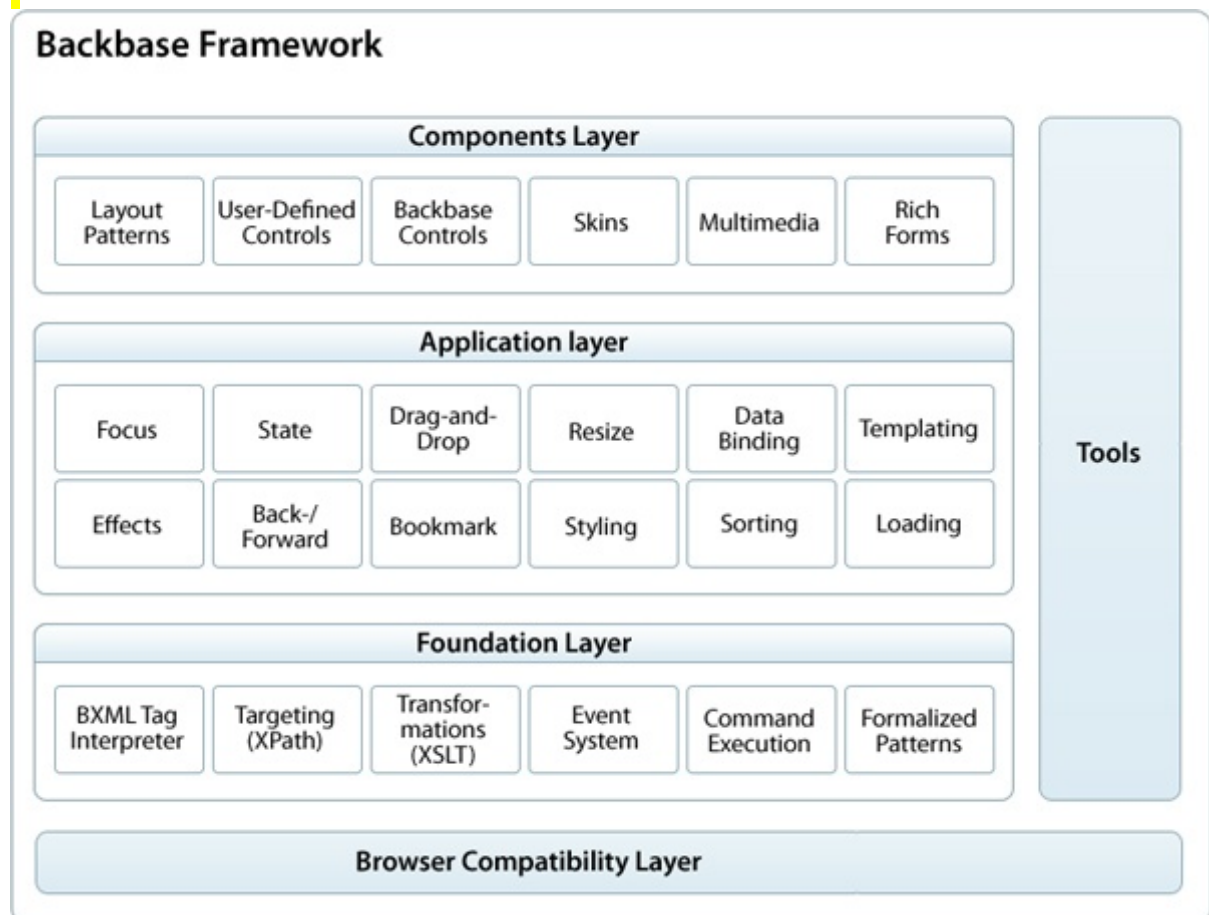


Figura 11. El Framework de Backbase

7. DIAGRAMAS UML

Desde el inicio del proyecto se planteó usar UML como herramienta de modelado del sistema computarizado, dejando abierta la posibilidad de escoger qué diagramas se utilizarían de acuerdo a las necesidades del proyecto.

Para el desarrollo del proyecto, tomando como base que es un sistema Web se escogieron los siguientes diagramas:

- ✓ Diagrama de Contexto
- ✓ Diagrama de casos de uso.
- ✓ Diagrama de Clases
- ✓ Diagrama Entidad - Relación
- ✓ Diagrama de Actividades
- ✓ Diagrama de Navegación.

7.1 Diagrama de casos de uso

El principal motivo por el cual se escogió el diagrama de casos de uso es que son reconocidos y son usados para la gran mayoría de los proyectos. Algunas de sus características importantes son:

- ✓ Permite capturar fácilmente los requisitos funcionales del sistema.
- ✓ Muestra de manera sencilla todo el funcionamiento del sistema facilitando el análisis y el diseño.
- ✓ Permiten delimitar el sistema de su entorno.

Estos diagramas se componen de Actores y Casos de uso. Los actores son las personas que interactúan directamente con el sistema; y los casos de uso son los procesos que estas personas ejecutan dentro del sistema. Los actores y los casos de uso se representan de la siguiente manera.



Figura 12. Actor y casos de uso

Para conectar los actores con los casos de uso, se usan conectores de la siguiente manera.



Figura 13. Conexión entre Actores y casos de uso

Para conectar dos casos de uso se utilizan conectores semejantes a los de actor y casos de uso con una diferencia que es el uso de una etiqueta que define la relación que pueden ser de dos tipos: ver figura 14.

- ✓ <<Extend>> se utiliza para representar que después de un caso de uso se puede acceder o no acceder hacia el otro caso de uso.
- ✓ <<Include>> se utiliza para representar la obligatoriedad de acceder al siguiente caso de uso.

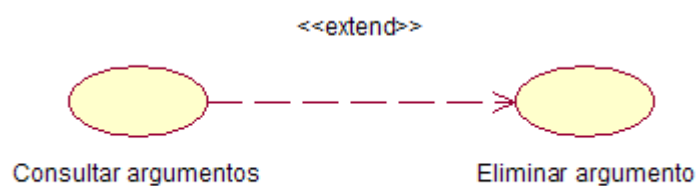


Figura 14. Conexión entre casos de uso

7.1.1 Diagrama de Contexto



Figura 15. Diagrama de Contexto

7.1.2 Diagrama de Casos de Uso - Nivel General

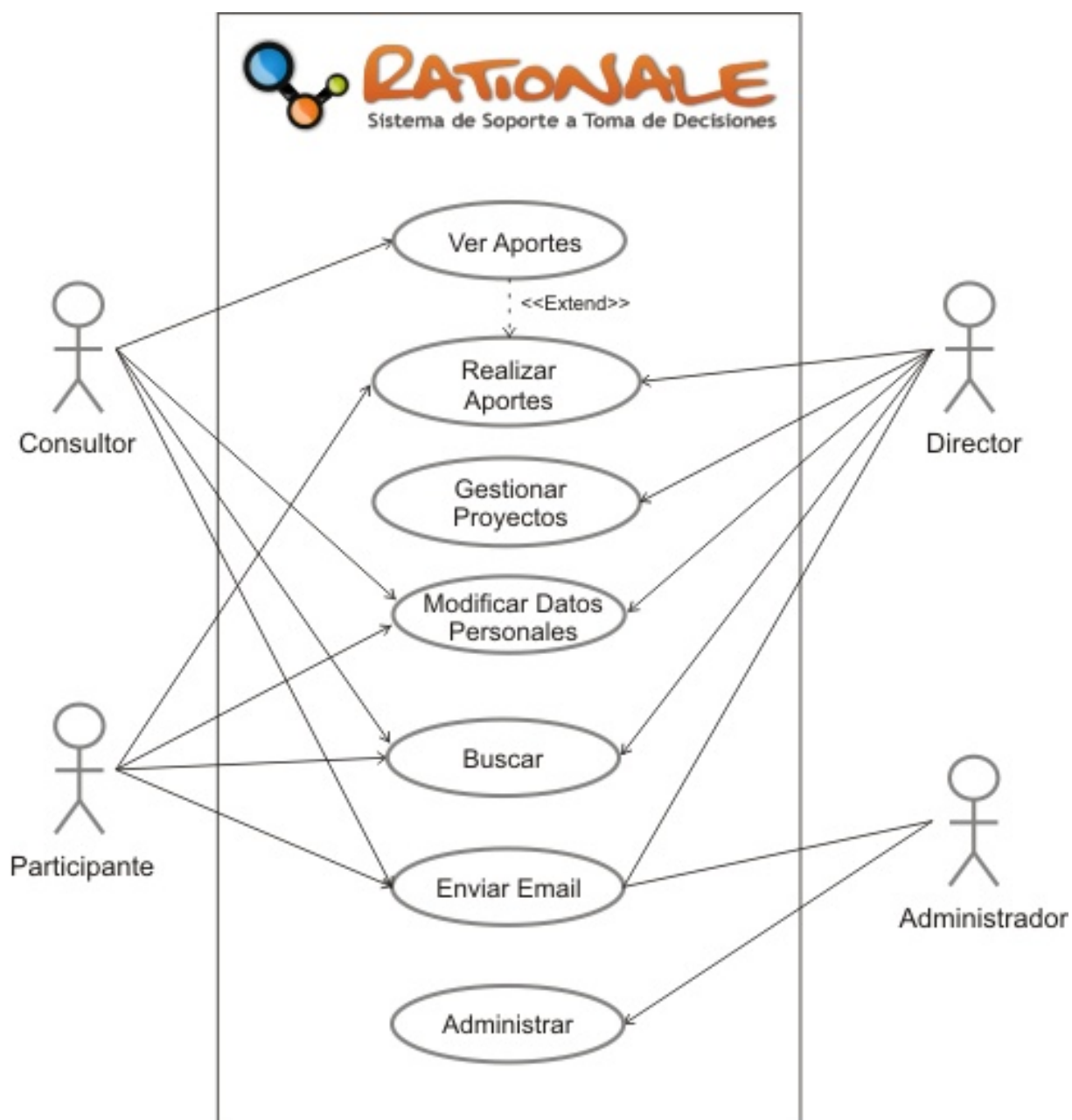


Figura 16. Diagrama Casos de uso – Nivel general

7.1.3 Especificación de Casos de Uso de Nivel General:

Caso de uso	Gestionar Proyectos
Tipo	Básico
Propósito	Conjunto de Acciones que realiza el Director para crear, modificar y eliminar proyectos, así como los problemas que contienen.
Precondiciones	Ingresar al Sistema con permisos de Director
Flujo Principal	A través de una interfaz el Director puede Ver y Crear Proyectos
Subflujos	Editar/Eliminar Proyecto, Ver Problema, Editar problema, Crear Problema, Ver Glosario.
Excepciones	El Proyecto creado ya Existe, campos vacíos en el formulario de creación de proyecto.

Caso de uso	Realizar Aportes
Tipo	Básico
Propósito	Conjunto de Acciones que realiza el Participante o Director para crear o modificar aportes tales como Criterios, alternativas con sus argumentos o términos en el glosario.
Precondiciones	Ingresar al Sistema con permisos de Participante o Director
Flujo Principal	A través de una interfaz el Participante o Director puede Crear, editar y eliminar alternativas, comentarios dentro de las alternativas.
Subflujos	Ninguno
Excepciones	La alternativa creada ya existe, campos vacíos en los formularios de nueva alternativa, editar alternativa.

Caso de uso	Modificar Datos Personales
Tipo	Básico
Propósito	Conjunto de Acciones que realiza cualquier usuario del sistema para modificar su contraseña o sus datos personales.
Precondiciones	Ingresar al Sistema con cualquier nivel de permiso
Flujo Principal	A través de una interfaz el usuario (cualquiera) puede modificar cada uno de sus datos personales que hacen parte del perfil del usuario.
Subflujos	Ninguno
Excepciones	Contraseña vacía, campo de e-mail vacío

Caso de uso	Buscar
Tipo	Básico
Propósito	Permite al usuario realizar búsquedas de aportes, ya sean proyectos, problemas, alternativas, criterios, argumentos, resoluciones y términos del glosario.
Precondiciones	Ingresar al Sistema con cualquier nivel de permiso
Flujo Principal	A través de un formulario el usuario puede escribir las palabras claves de su búsqueda, así como también seleccionar filtros para la misma.
Subflujos	Puede pasar a ver el aporte una vez realizada la búsqueda.
Excepciones	Campo de búsqueda vacío

Caso de uso	Administrar
Tipo	Básico
Propósito	Conjunto de acciones básicas de gestión y configuración del sistema
Precondiciones	Ingresar al Sistema con permiso de administrador
Flujo Principal	A través de una interfaz el usuario (cualquiera) puede modificar cada uno de sus datos personales que hacen parte del perfil del usuario.
Subflujos	Gestión de usuarios, gestión de aportes
Excepciones	Campos de usuario y contraseña de base de datos vacíos

Caso de uso	Enviar E-mail
Tipo	Básico
Propósito	Permite al usuario enviar un mensaje al correo electrónico de los destinatarios seleccionados.
Precondiciones	Ingresar al Sistema
Flujo Principal	A través de una interfaz el usuario (cualquiera) puede enviar un correo electrónico a los demás usuarios.
Subflujos	Ninguno
Excepciones	Campos de Asunto, Mensaje vacíos y/o selección nula de destinatarios.

7.1.4 Caso de Uso: Gestionar Proyectos

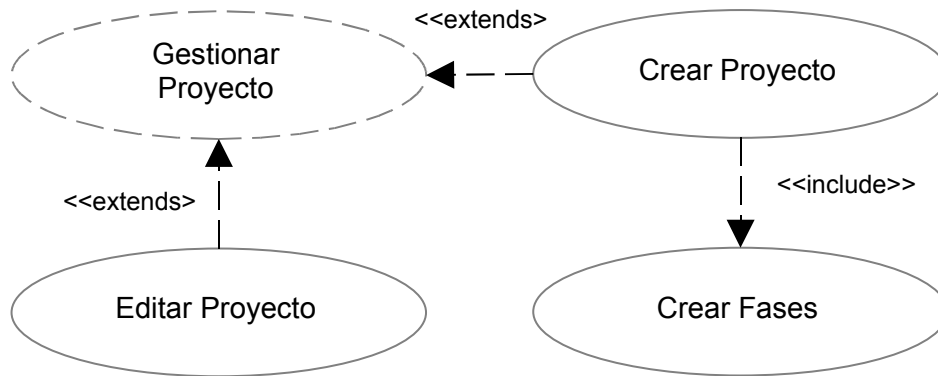


Figura 17. Caso de uso: Gestionar Proyectos

7.1.5 Caso de Uso: Ver Aportes

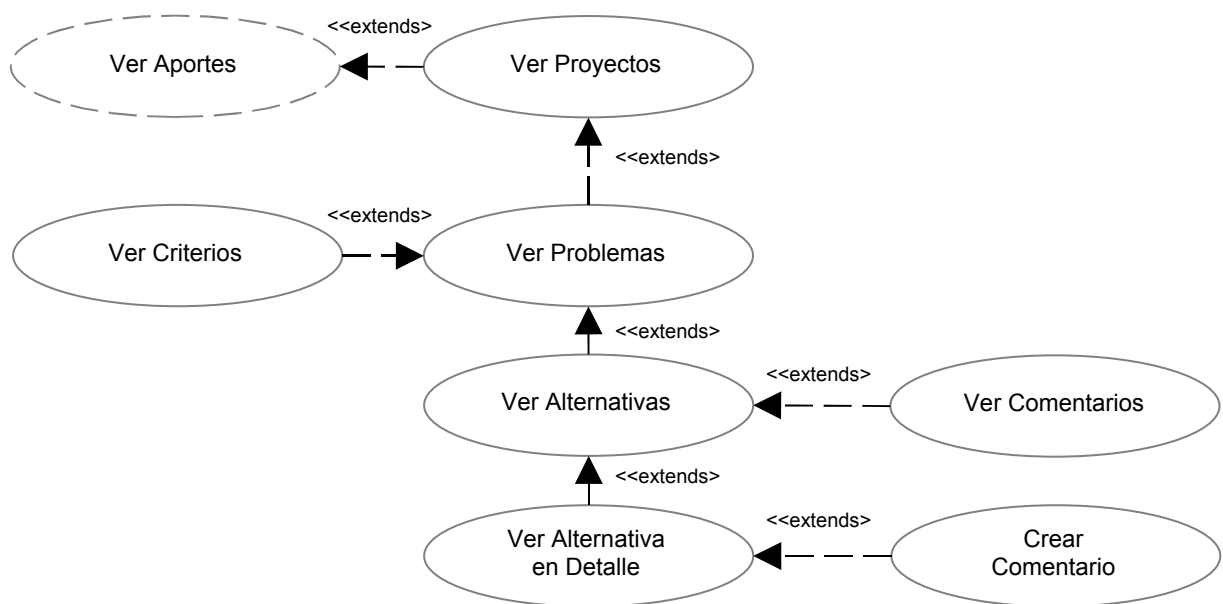


Figura 18. Caso de uso: Ver Aportes

7.1.6 Caso de Uso: Administrar

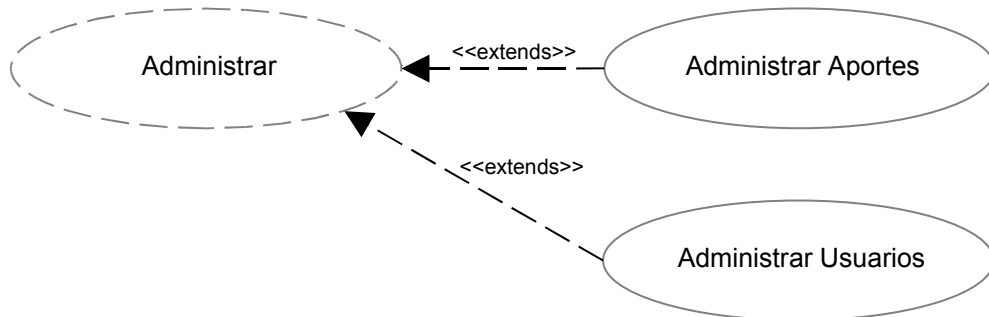


Figura 19. Caso de uso: Administrar

7.1.7 Caso de Uso: Gestionar Aportes

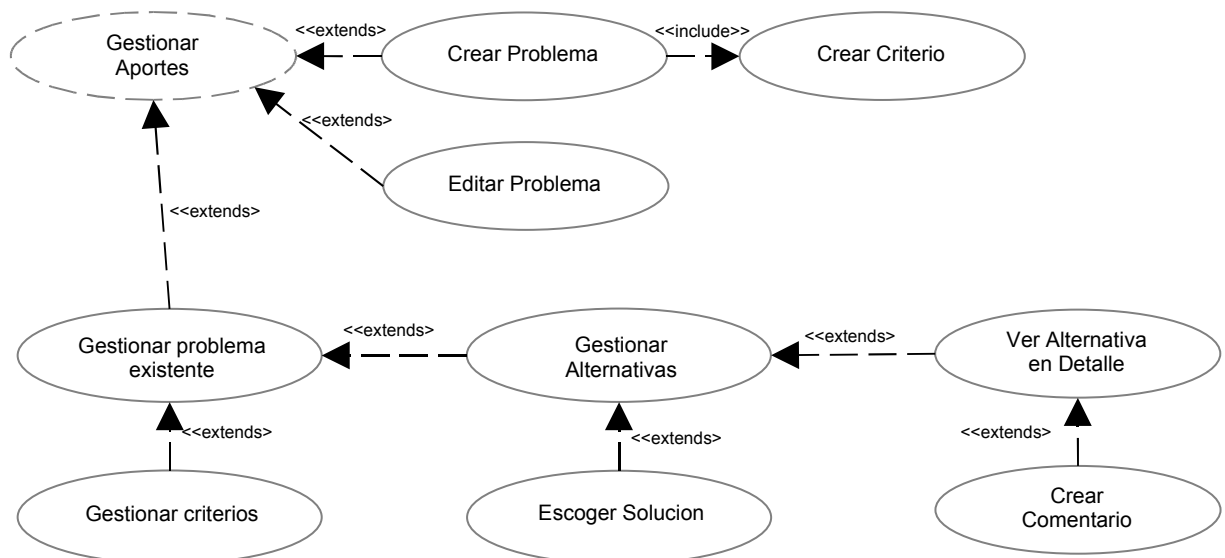


Figura 20. Caso de uso: Gestionar Aportes

7.2 DIAGRAMA DE CLASES: Definición

Para realizar el diagrama de clases, se escribe la descripción del sistema a realizar y se subrayan los “Sustantivos” o lo que pueda representar una clase. Luego se obtiene una lista de clases candidatas, se depuran y finalmente se obtiene el listado de clases del sistema. Posteriormente se logra el diagrama de clases mostrando las relaciones entre ellas.

- **Descripción del problema Subrayado:**

Rationale es un sistema que permite resolver problemas de forma conjunta en las organizaciones. El sistema debe funcionar en un entorno Web agradable.

La pantalla inicial del sistema es el formulario de ingreso al sistema donde el usuario debe ubicar su nombre de usuario y contraseña o bien registrarse en el sistema como nuevo usuario. Al ingresar el formulario, éste es validado y se permite el acceso al menú principal del sistema con las siguientes opciones:

Inicio: Se muestra la información pertinente al usuario, tal como los proyectos en los cuales esta trabajando, las fases que lo componen, los problemas que se han generado en éstas, así también como los últimos aportes registrados en el sistema. Si el usuario decide ver en detalle un proyecto, se mostrará el título y una descripción del mismo, así como una vista en forma de árbol que muestra las fases, los problemas y sus alternativas en orden jerárquico.

El usuario (cualquiera) puede decidir ver en detalle un problema en donde se mostrará el título, la descripción del mismo, los criterios de solución, las alternativas y argumentos planteados y, si existe, una resolución. Tanto en alternativas como en argumentos los usuarios (Director y Participante) pueden dejar comentarios para generar una discusión y eventualmente llegar a la. En ésta vista se incluye también un glosario de términos del proyecto como asistencia adicional.

El usuario (cualquiera) también puede acceder desde inicio a su perfil, el cual puede modificar a su gusto, en éste formulario el usuario puede cambiar su contraseña, su e-mail e incluso una fotografía personal, entre otros datos. El usuario puede escoger si desea recibir notificaciones de cambios que ocurran en el sistema mediante correo electrónico.

E-mail: Módulo para facilitar la comunicación entre los usuarios mediante el envío de mensajes al correo electrónico de los mismos.

Búsqueda: Es un módulo que permite a cualquier usuario del sistema encontrar aportes que concuerden con su criterio de búsqueda. Consiste en un formulario con una casilla de texto en la cual el usuario especifica los términos de búsqueda y unas cajas de chequeo para determinar los filtros que deben aplicarse.

Administración: Módulo que permite únicamente a los administradores configurar parámetros básicos del sistema tal como ver, modificar y eliminar usuarios registrados en el sistema, así como también gestionar los aportes de todos los proyectos existentes.

Estadísticas: Módulo exclusivamente informativo, muestra la cantidad de usuarios registrados, proyectos y problemas más activos, número de aportes, entre otros. A éste módulo puede acceder cualquier usuario registrado.

Cerrar Sesión: Termina la sesión del usuario y lo lleva nuevamente a la pantalla de ingreso del sistema.

- **LISTADO DE CLASES CANDIDATAS:**

Proyecto

Fase

Organización

Aporte

Problema

Vista en forma de Árbol

Criterio

Alternativa

Argumento

Resolución

Comentario

Discusión

Usuario

Glosario

Administrador

- **LISTADO DE CLASES DEL SISTEMA:**

Proyecto

Fase

Problema

Criterio

Alternativa

Comentario

Usuario

Glosario

7.2.1 Diagrama de Clases

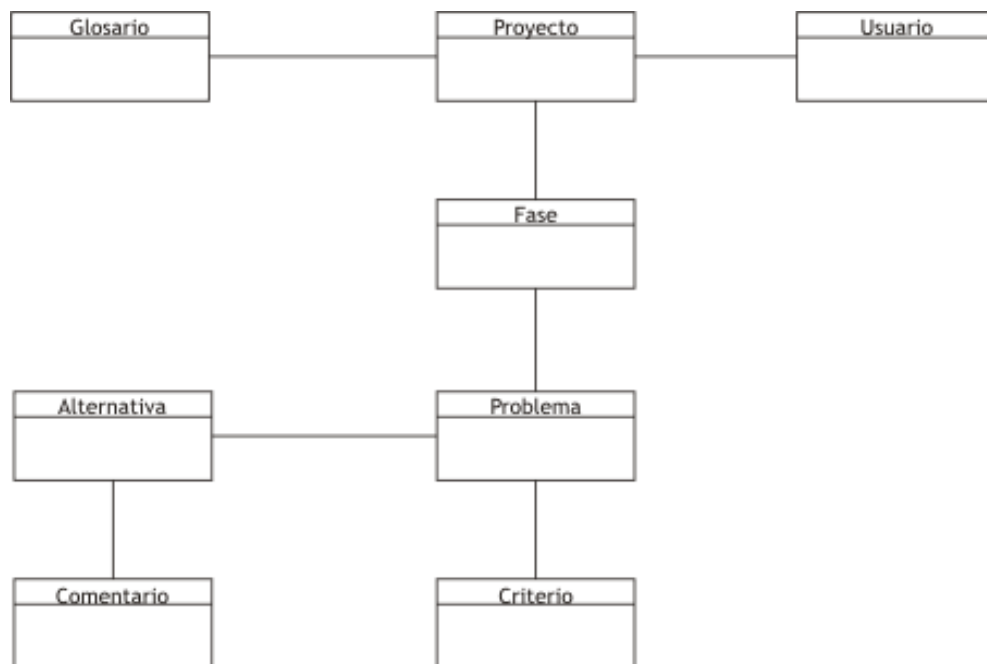


Figura 21. Diagrama de Clases

7.2.2 Diccionario de Clases

Proyectos: Hace referencia a un proyecto que este ejecutando la organización y sobre el cual se va a realizar la documentación de las decisiones. Agrupa problemas dentro de sus fases y usuarios.

Fases: Sección de un proyecto que agrupa problemas, ayuda a estructurar la información.

Problemas: Es un asunto que no ha podido resolverse en la realización del proyecto y que se va a abordar mediante discusiones entre los usuarios para solucionarlo.

Criterios: Los criterios corresponden a cualidades deseables que la solución seleccionada debería satisfacer. Estas cualidades tienen una ponderación, es decir, un grado de importancia para el problema que se intenta resolver.

Alternativas: Son las opciones a elegir como solución de un problema, éstas son dadas por los usuarios y deben ser sometidas a discusión para lograrlo. Es el director del proyecto

quien finalmente decide cuál de esas alternativas se convierte en Resolución. Deben estar soportadas por un argumento que explique porque es la mejor solución al problema.

Comentarios: Son las opiniones que dejan los usuarios de las alternativas o argumentos que dejan sus compañeros de proyecto.

Usuarios: Representa quienes alimentan el sistema con sus aportes en los problemas, los cuales cambian de rol (Director, Participante, Consultor) según el proyecto.

Glosario: Diccionario de términos poco conocidos ó que poseen algún grado de ambigüedad para la realización de un proyecto.

7.3 Diagrama Entidad – Relación: Definición

Formalmente, los diagramas E-R son un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que describen la información que trata un sistema de información y el software que lo automatiza. Los elementos de dicho lenguaje se describen a continuación, por orden de importancia.

Entidades

Una entidad es cualquier "objeto" discreto sobre el que se tiene información. Se representa mediante un rectángulo o "caja" etiquetada en su interior mediante un nombre. Ejemplos de entidades habituales en los sistemas de información son: factura, persona, empleado, etc.

Cada ejemplar de una entidad se denomina instancia. Por ejemplo, "carlos ch y doris ar" pueden ser dos instancias distintas de la entidad "persona". Las instancias no se representan en el diagrama. No obstante, se pueden documentar aparte porque son útiles para inicializar la base de datos resultante. Por ejemplo, los departamentos existentes de una empresa pueden ser relevantes como datos iniciales de la entidad "departamento".

Relaciones

Una relación describe cierta interdependencia (de cualquier tipo) entre entidades. Se representa mediante un rombo etiquetado en su interior mediante un verbo. Además, dicho rombo debe unirse mediante líneas con las entidades que relaciona (es decir, los rectángulos). Una relación no tiene sentido sin las entidades que relaciona. Por ejemplo: una persona (entidad) trabaja (relación) para un departamento (entidad).

Atributos

Los atributos son propiedades relevantes propias de una entidad y/o relación. Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es el identificador de la entidad se suele subrayar dicha etiqueta.

Por motivos de legibilidad, los atributos no suelen representarse en un diagrama entidad-relación, sino que se describen textualmente en otros documentos adjuntos.

Los atributos describen información útil sobre las entidades. En particular, los atributos identificadores son aquellos que permiten diferenciar a una instancia de la entidad de otra distinta.

7.3.1 Diagrama Entidad Relación

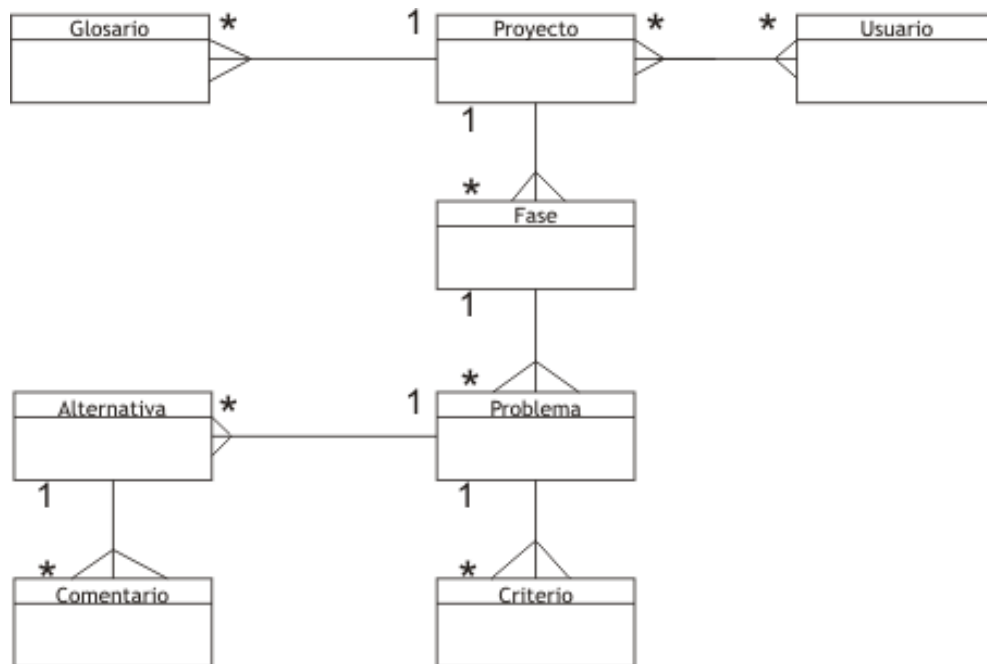


Figura 22. Diagrama Entidad-Relación

7.4 Diccionario de Datos: Definición

Es un catálogo en el que se almacenen las descripciones de los datos y contiene información que describe los datos de la base de datos (metadatos). Normalmente, un diccionario de datos almacena: Nombre, tipo y tamaño de los datos, Nombre de las relaciones entre los datos.

7.4.1 Diccionario de Datos

Alternativas

Campo	Tipo	Descripción
<u>IdAlternativa</u>	int(11)	Identificador de la Alternativa
Alternativa	varchar(64)	Descripción de la alternativa
Argumento	varchar(255)	Explicación de porqué ésta alternativa es la mejor opción
PostInicio	datetime	Fecha y hora de creación de la alternativa
IdUsuario	varchar(16)	Identificador del usuario que creo la alternativa
IdProblema	int(11)	Identificador del problema que se va a resolver con la alternativa
Estado	varchar(16)	Establece si es una solución posible, si fue rechazada o si es la solución definitiva

Comentarios

Campo	Tipo	Descripción
<u>IdComentario</u>	int(11)	Identificador del Comentario
IdAlternativa	int(11)	Identificador de la alternativa a la cual pertenece
IdUsuario	varchar(16)	Identificador del usuario que creo el comentario
PostInicio	datetime	Fecha y hora en la que se realizo el comentario
Posicion	varchar(16)	Valoración que se le da a la alternativa según el punto de vista del usuario
comentario	varchar(255)	Argumento con el cual Soporta la Posición tomada.

Criterios

Campo	Tipo	Descripción
<u>IdCriterio</u>	int(11)	Identificador del Criterio
Titulo	varchar(64)	Titulo del Criterio
Descripcion	varchar(255)	Descripción del Criterio
Rating	tinyint(4)	Nivel de Relevancia del Criterio para el problema
IdProblema	int(11)	Identificador del problema al cual pertenece

Fases

Campo	Tipo	Descripción
<u>IdFase</u>	int(11)	Identificador de la Fase
<i>IdProyecto</i>	int(11)	Identificador del proyecto al cual pertenece la Fase
Nombre	varchar(32)	Nombre de la Fase
Orden	int(11)	Orden en el proyecto

Glosario

Campo	Tipo	Descripción
<u>IdTermino</u>	int(11)	Identificador del Termino del glosario
Termino	varchar(32)	Palabra a definir
Significado	varchar(255)	Definición de la palabra bajo el contexto del proyecto
<i>IdProyecto</i>	int(11)	Identificador del proyecto al cual pertenece el termino
<i>IdUsuario</i>	varchar(16)	Identificador del usuario que creo el termino

Problemas

Campo	Tipo	Descripción
<u>IdProblema</u>	int(11)	Identificador del Problema
Estado	varchar(16)	Establece el estado del problema (Activo o Inactivo)
Titulo	varchar(128)	Titulo del Problema
Descripcion	varchar(256)	Descripción del Problema
PostInicio	datetime	Fecha y hora de creación del problema en el sistema
PostActualizacion	datetime	Fecha y hora del ultimo cambio registrado en el problema
<i>IdProyecto</i>	int(11)	Identificador del proyecto al cual pertenece el problema
<i>IdUsuario</i>	varchar(16)	Identificador del usuario que creo el problema
<i>IdFase</i>	int(11)	Identificador de la Fase a la cual pertenece el problema
limite	date	Fecha limite para realizar aportes para el problema

Proyectos

Campo	Tipo	Descripción
<u>IdProyecto</u>	int(11)	Identificador del Proyecto
Titulo	varchar(128)	Titulo del Proyecto
Descripcion	varchar(256)	Descripción del Proyecto
PostInicio	datetime	Fecha y hora de Creación del proyecto
PostActualizacion	datetime	Fecha y hora de ultima actualización del proyecto
Estado	varchar(16)	Establece el Estado del proyecto (Activo o Inactivo)
area	varchar(64)	Especifica en que Área se desarrolla el proyecto (software, construcción, etc.)

Rol_proyecto

Campo	Tipo	Descripción
<u>IdRol</u>	bigint(20)	Identificador del Rol
<i>IdProyecto</i>	int(11)	Identificador del Proyecto
<i>IdUsuario</i>	varchar(16)	Identificador del Usuario
Rol	varchar(16)	Relación entre el proyecto y el usuario (Consultor, Participante, Director)

Usuarios

Campo	Tipo	Descripción
IdUsuario	varchar(16)	Es el nombre e Identificador del usuario
Password	varchar(16)	La contraseña para ingresar al sistema
Email	varchar(64)	Dirección de e-mail donde se enviarán los mensajes
Estado	varchar(16)	Establece si el usuario esta Activo o no, para efectos de validar el ingreso.
PostLogin	datetime	Almacena la Fecha y hora del ultimo ingreso al sistema
PostRegistro	datetime	Almacena la Fecha y hora del registro del usuario
Nombres	text	Nombres reales del usuario
Apellidos	text	Apellidos reales del usuario
Nacimiento	date	Fecha de Nacimiento del usuario
Sexo	varchar(1)	Define el sexo del usuario (M o F)
Admin	tinyint(4)	Si es 1, El usuario tiene derechos de Administrador.
Dirigir	tinyint(4)	Si es 1, El usuario tiene permisos para Crear un Proyecto y dirigirlo.
Notificar	tinyint(4)	Nivel de notificación del usuario. 1 No notificar, 2 Notificar solo cambios importantes, 3 Notificar todos los cambios en el sistema

7.4 Diagrama de Actividades: Definición

Los diagramas de actividades UML capturan y comunican los detalles de la navegación en la interfase de usuario, la funcionalidad y explica tres estereotipos: acción, presentación y conector.

El software moderno está diseñado para interactuar en algún momento con el usuario. Los desarrolladores de software crearon las interfaces graficas de usuario (GUI) para facilitar la comunicación entre los sistemas computacionales y los usuarios. Estas interfaces están normalmente basadas en algún tipo de ventana con elementos de presentación, que representan el estado actual del sistema, los elementos de control para alterar ese estado actual del sistema y elementos de navegación para cambiar de una sección del sistema a otra.

En la construcción de Diagramas de Actividad, es de gran ayuda usar iconos con colores para indicar actividades específicas y diferenciar visualmente varios pasos en un flujo. Esto es particularmente importante para iconos de conectores que enlazan a escenarios de casos de uso separados.

Acción es el primer elemento del diagrama. Este icono representa actividades realizadas por el sistema o el Actor. Como es el icono más común, típicamente tiene un color neutral o ninguno.

Presentación Éste estereotipo indica que hay una conversación entre el actor y el sistema. Representa una categoría especial de las actividades de acción y es usada para abstraer detalles de la interfase de usuario.

Conector Representa las conexiones hacia flujos diagramados en otro sitio. El uso de Diagramas de Actividad casi siempre se enfoca a la creación de grandes y complejos modelos, así que es útil indicar flujos alternos de navegación. Los iconos conectores para este estereotipo pueden ser usados para enlazar automáticamente a otro diagrama.

7.4.1 Diagrama de Actividades

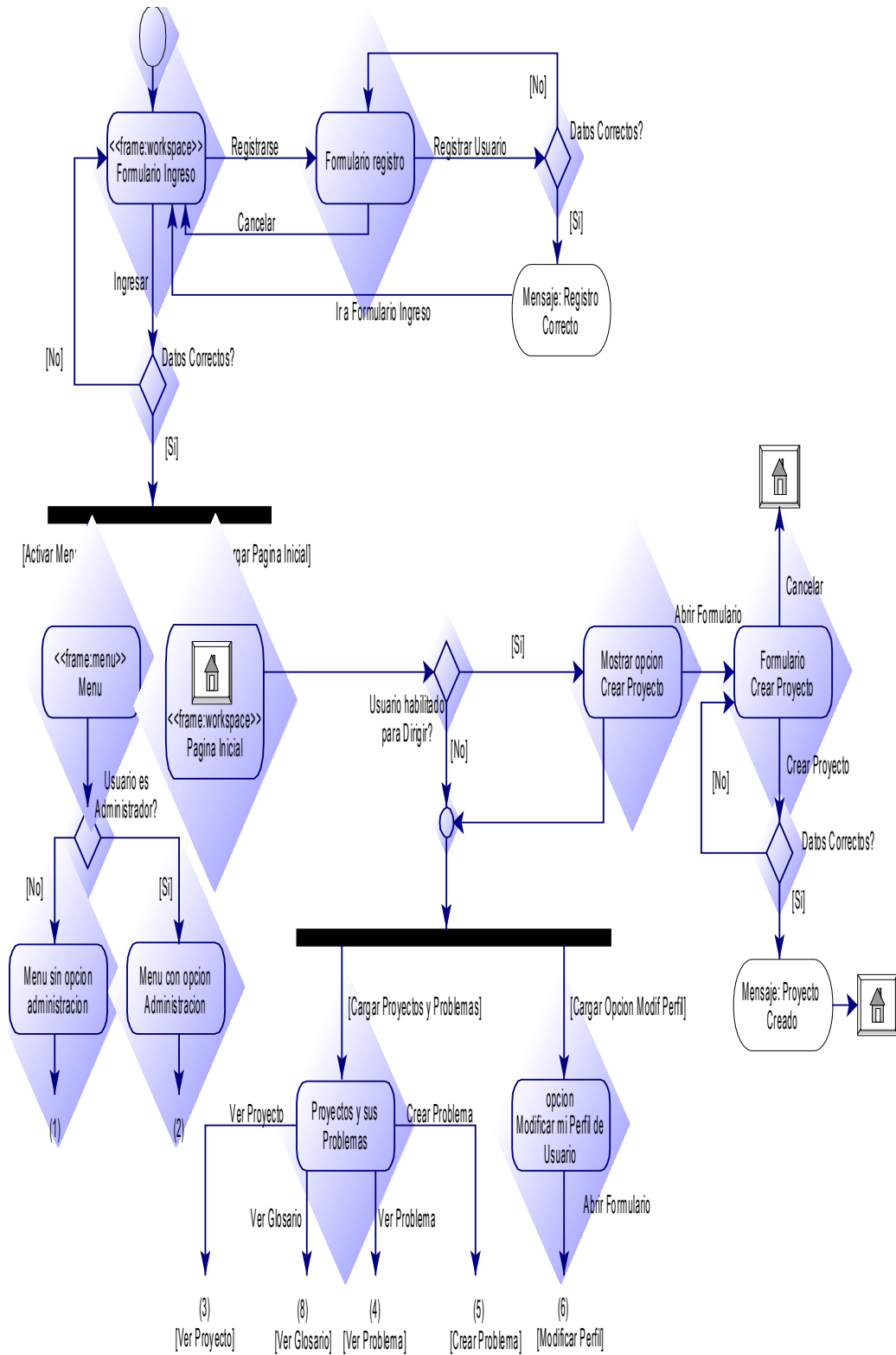


Figura 23. Diagrama de Actividades: inicio

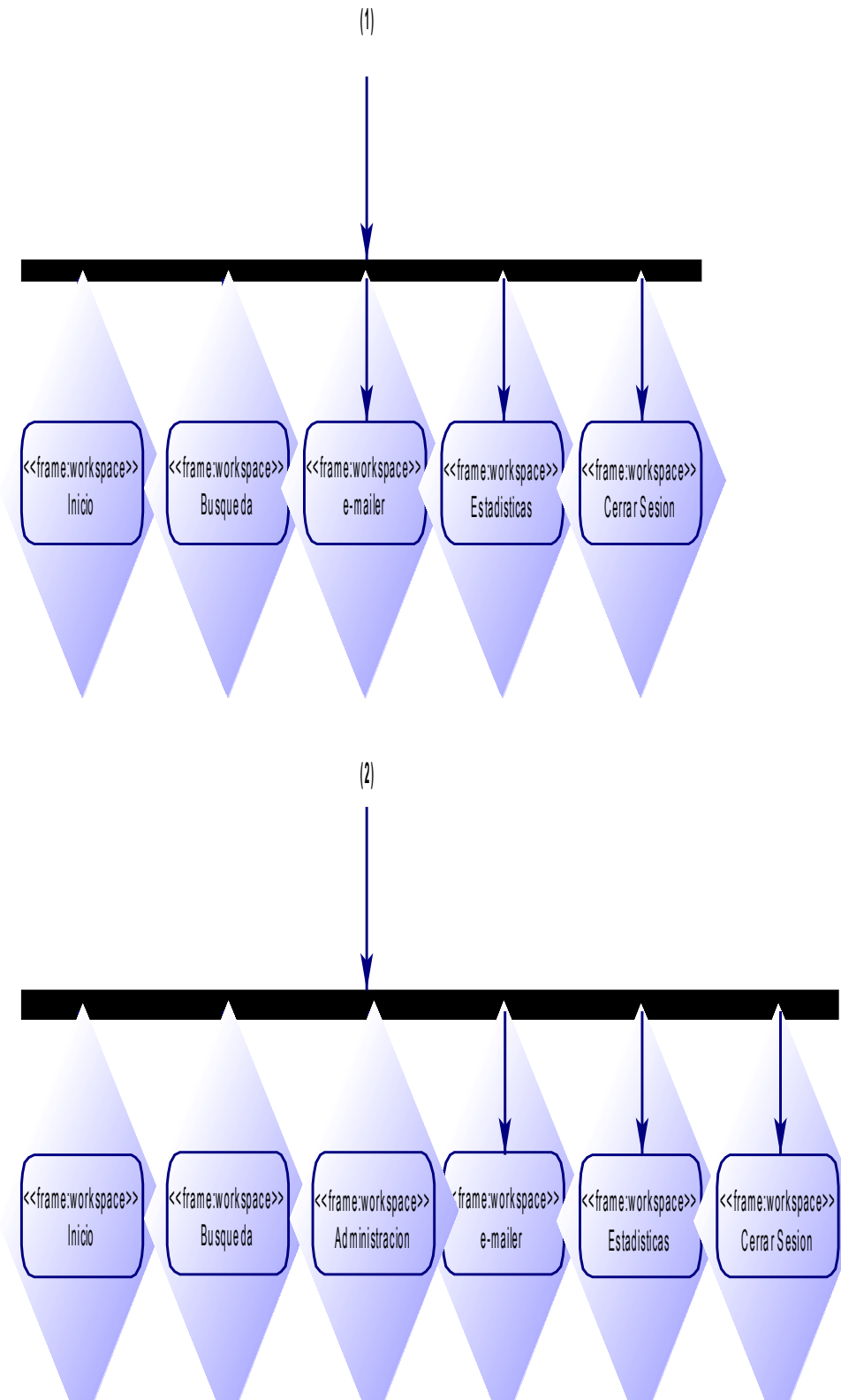


Figura 24. Diagrama de Actividades: Menu Principal

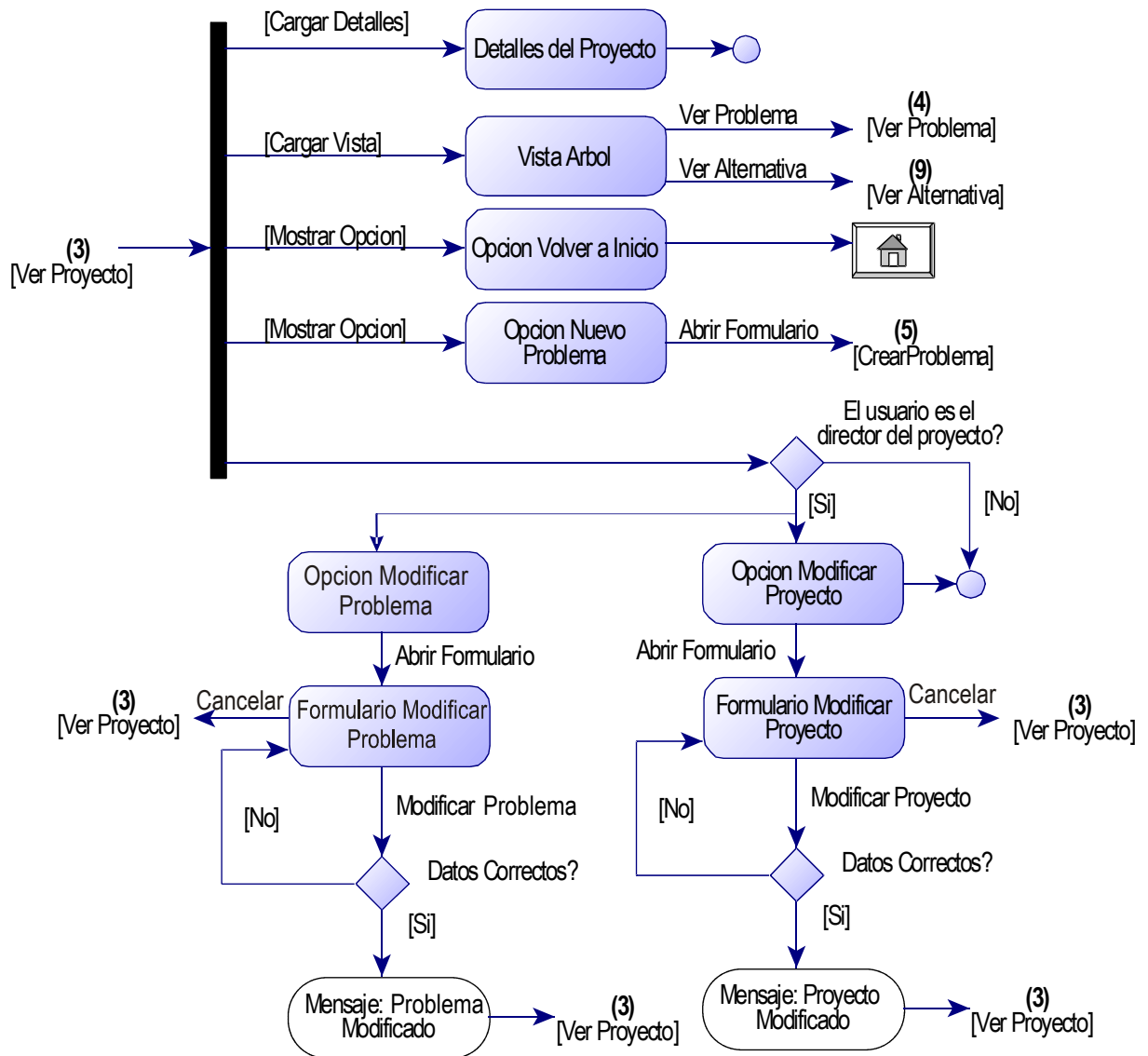


Figura 25. Diagrama de Actividades: Ver Proyecto

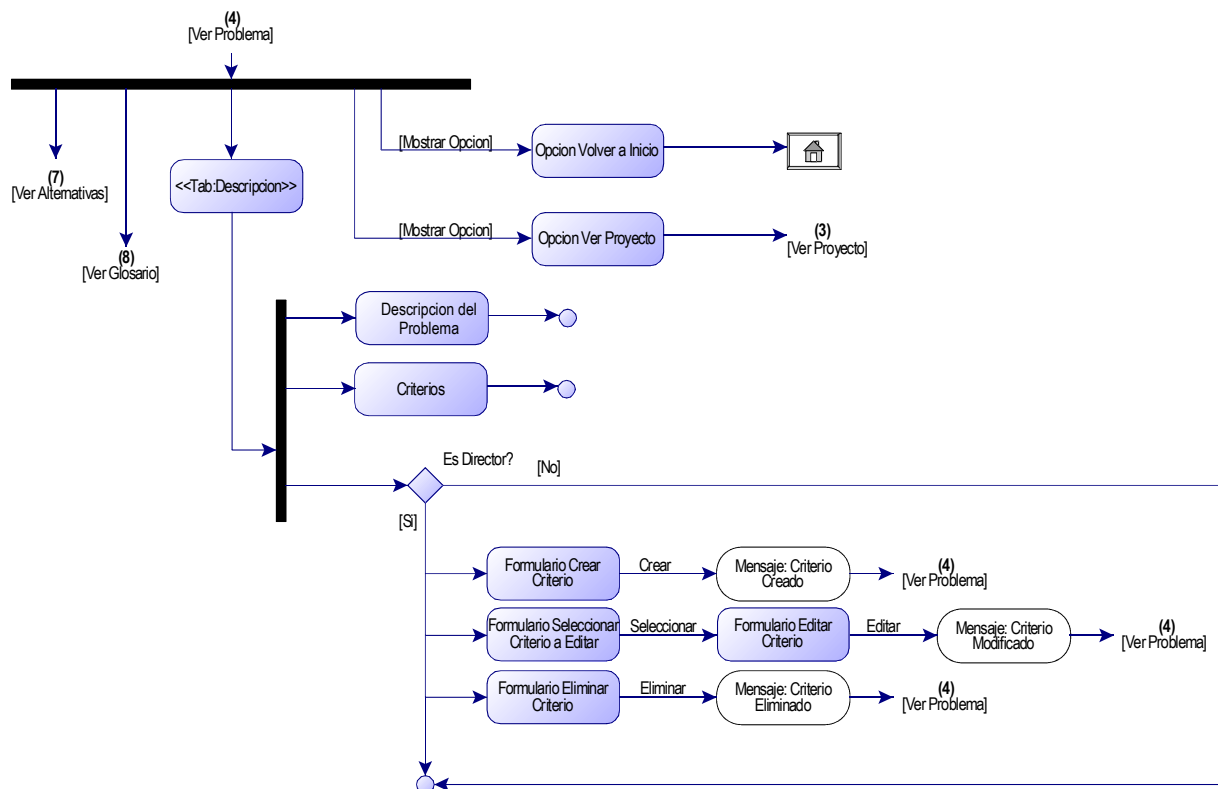


Figura 26. Diagrama de Actividades: Ver Problema

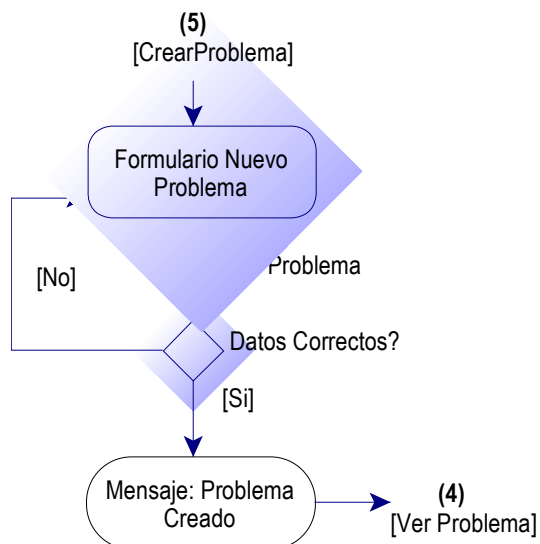


Figura 27. Diagrama de Actividades: Crear Problema

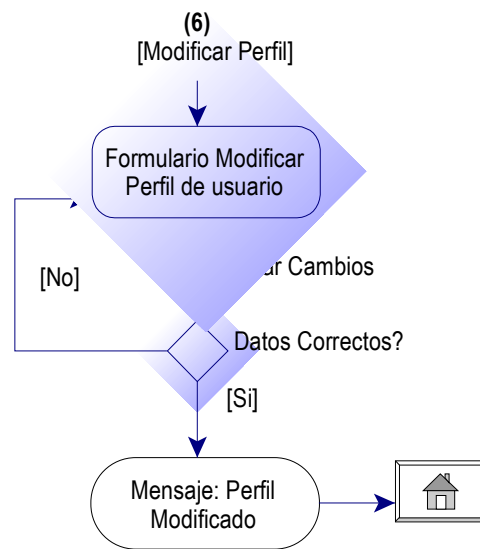


Figura 28. Diagrama de Actividades: Modificar Perfil

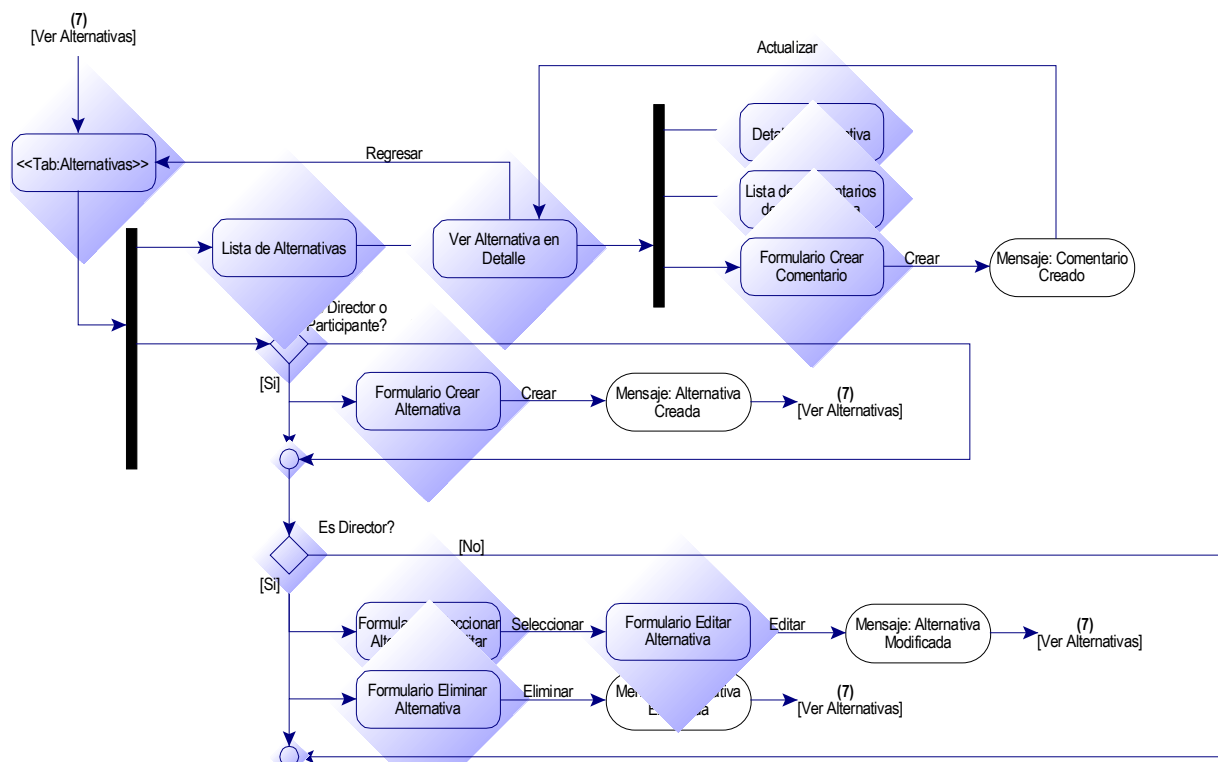


Figura 29. Diagrama de Actividades: Ver Alternativas

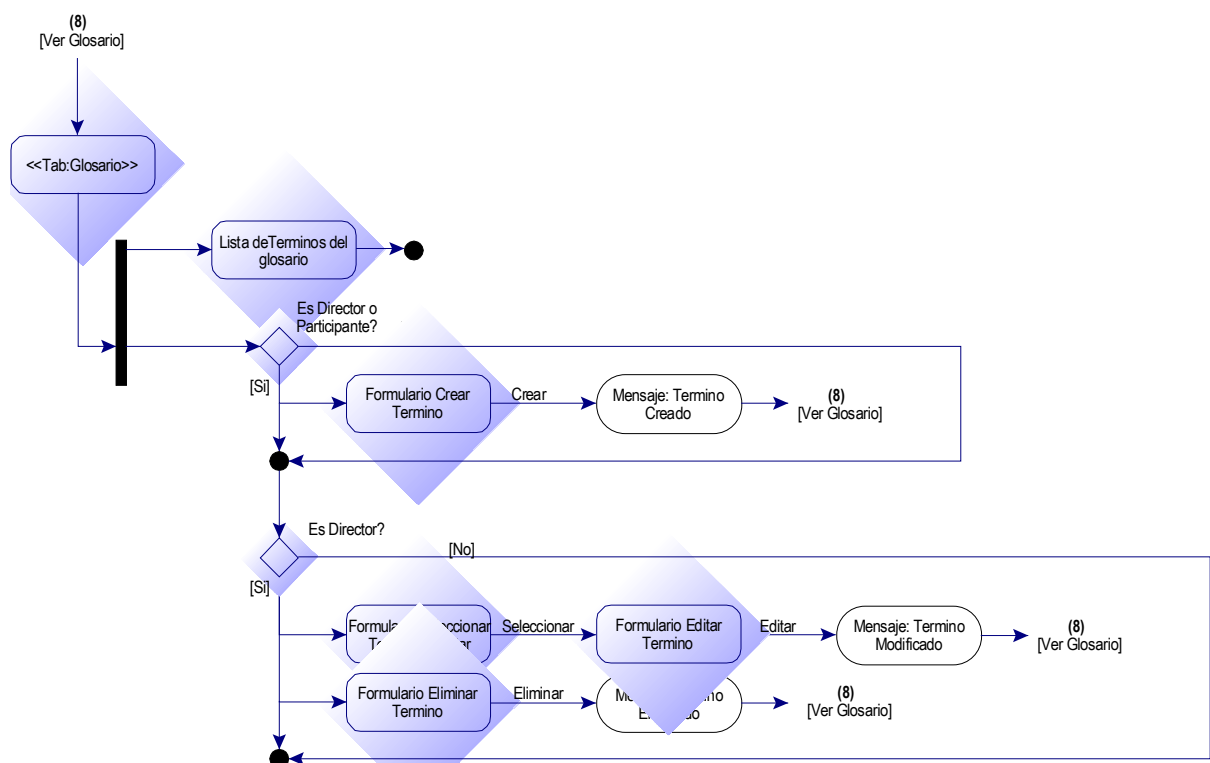


Figura 30. Diagrama de Actividades: Ver Glosario

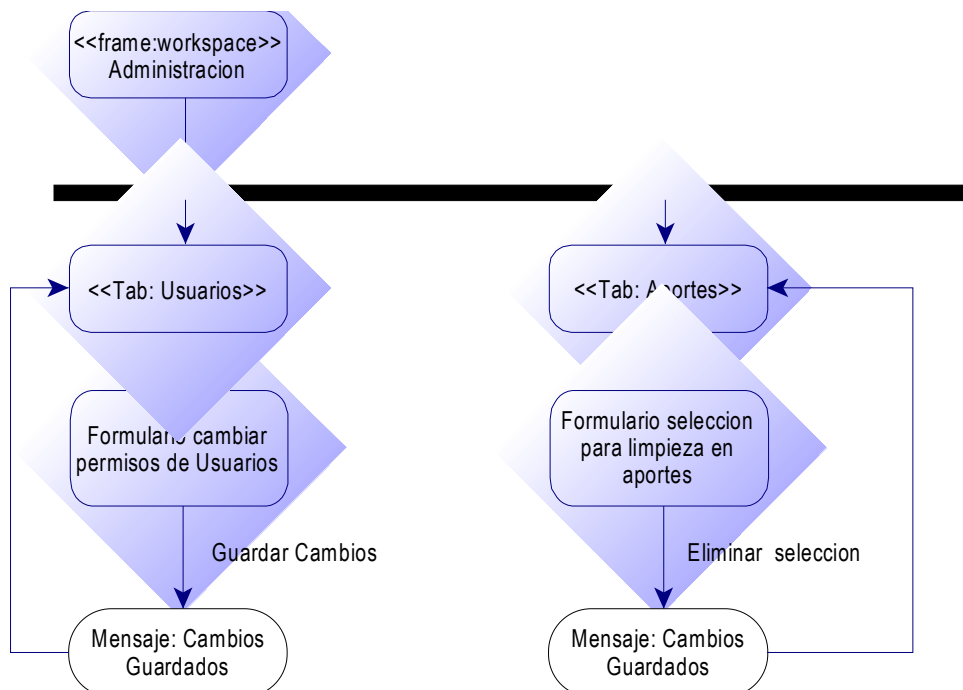


Figura 31. Diagrama de Actividades: Administración

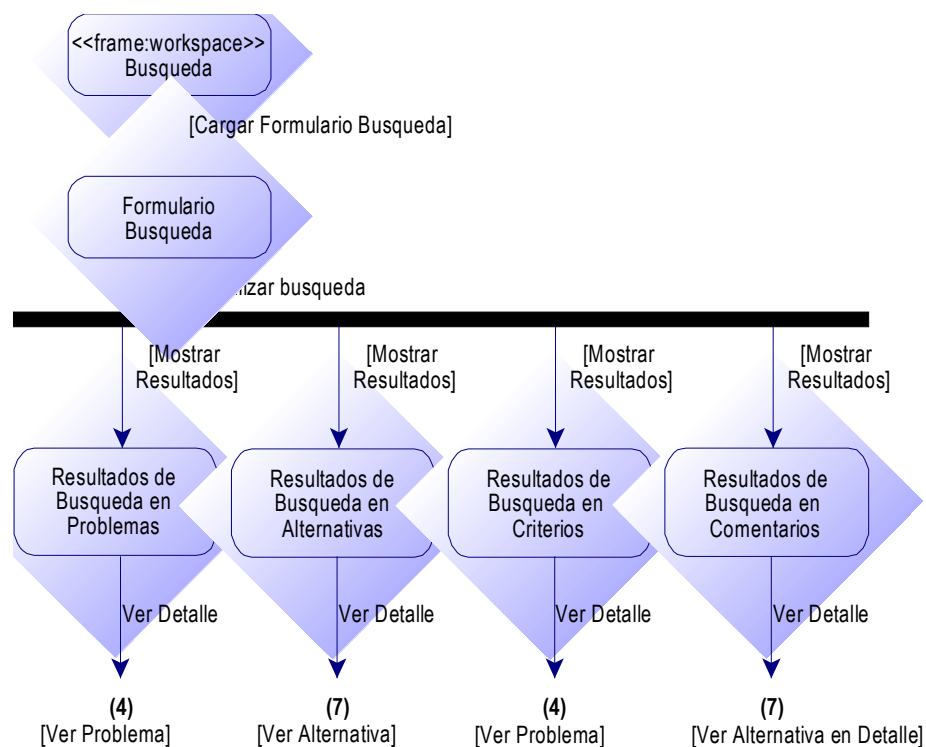


Figura 32. Diagrama de Actividades: Búsqueda

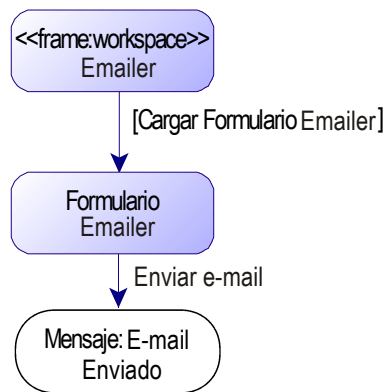


Figura 33. Diagrama de Actividades: E-mailer

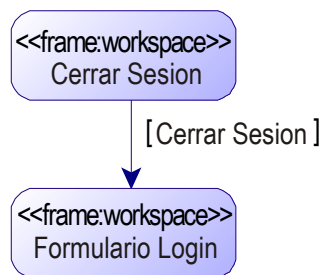


Figura 34. Diagrama de Actividades: Cerrar Sesión

7.5 Diagrama de Navegación: Definición

Es un tipo de Diagrama Web que es una forma de representación más sólida a partir del diagrama de actividades. En éste diagrama se muestran las páginas tal y como fueron nombradas, así como el flujo de navegación que hará el usuario a través del sistema. En el se pueden ver detalles como *frames*, *tabs* o enlaces que existan en cada una de las páginas. Se muestra también las tablas de la base de datos que se utilizan en cada sección y el script en PHP correspondiente para su acceso.

7.5.1 Diagrama de Navegación

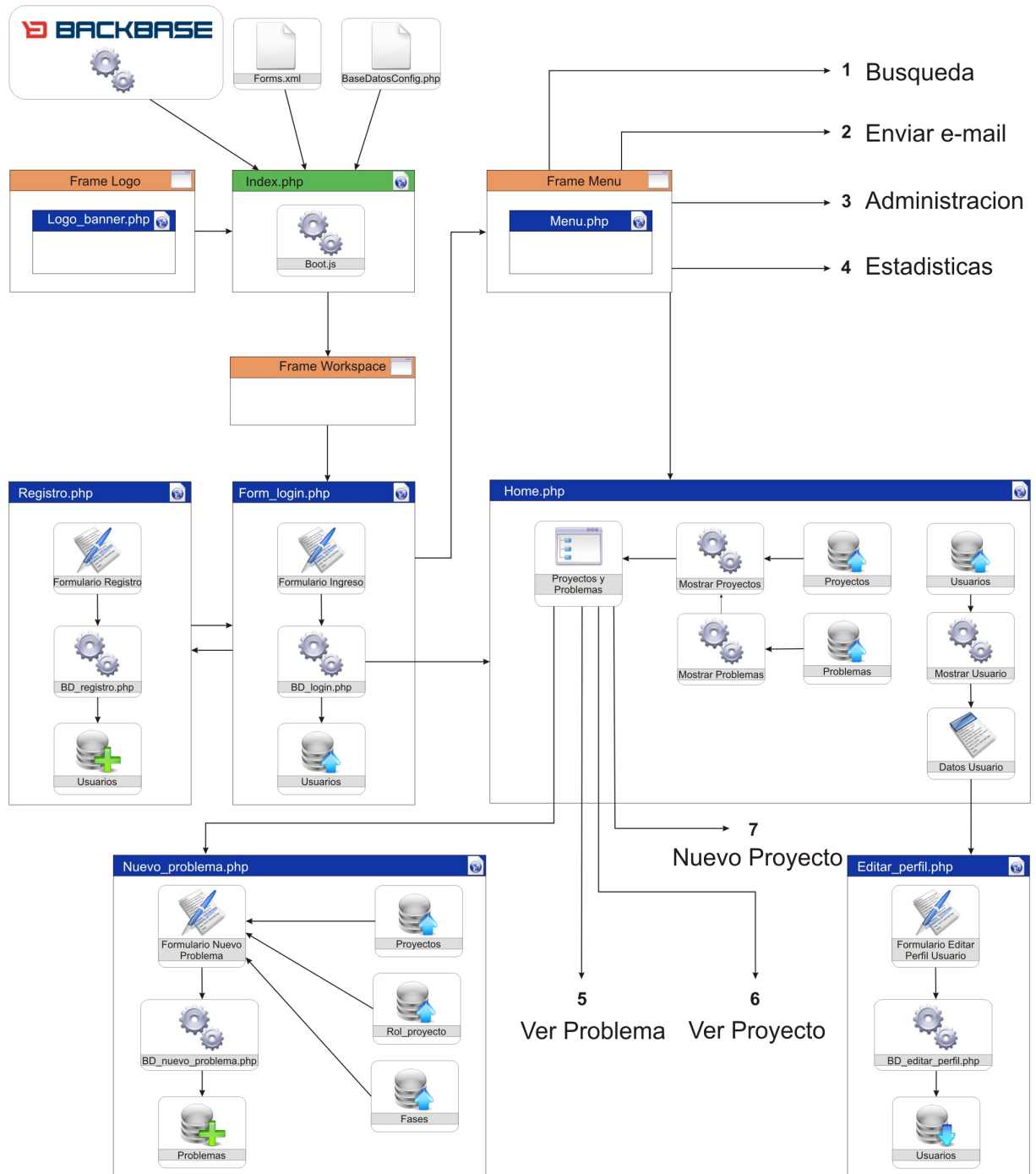


Figura 35. Diagrama de Navegación – Inicio

1. Búsqueda

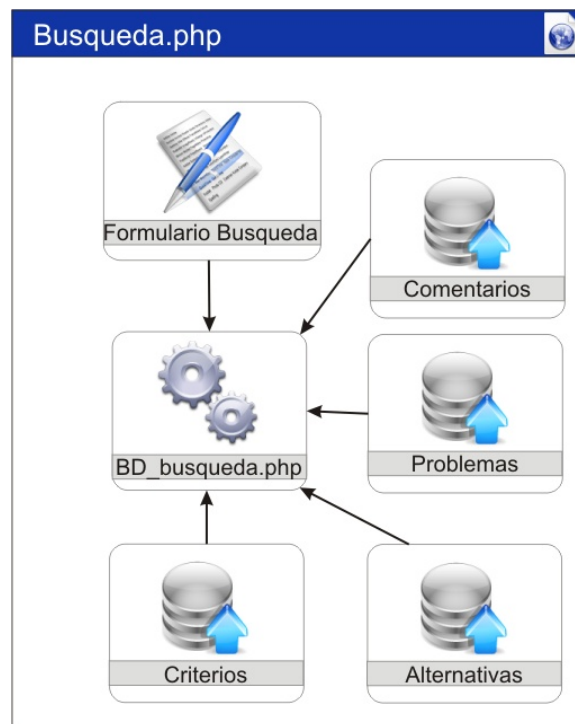


Figura 36. Diagrama de Navegación - Búsqueda

2. Enviar E-mail

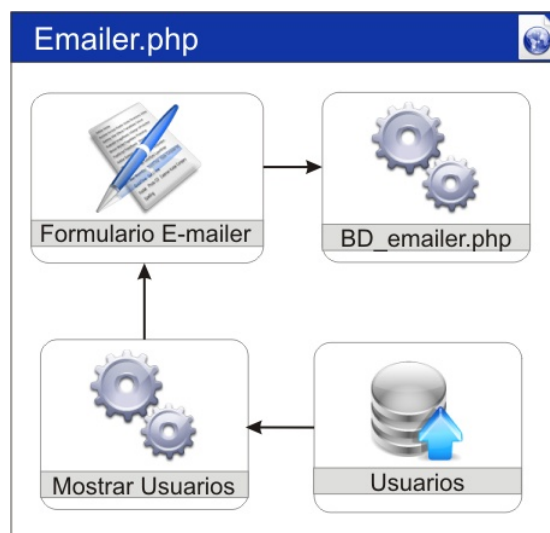


Figura 37. Diagrama de Navegación – Enviar E-mail

3. Administración

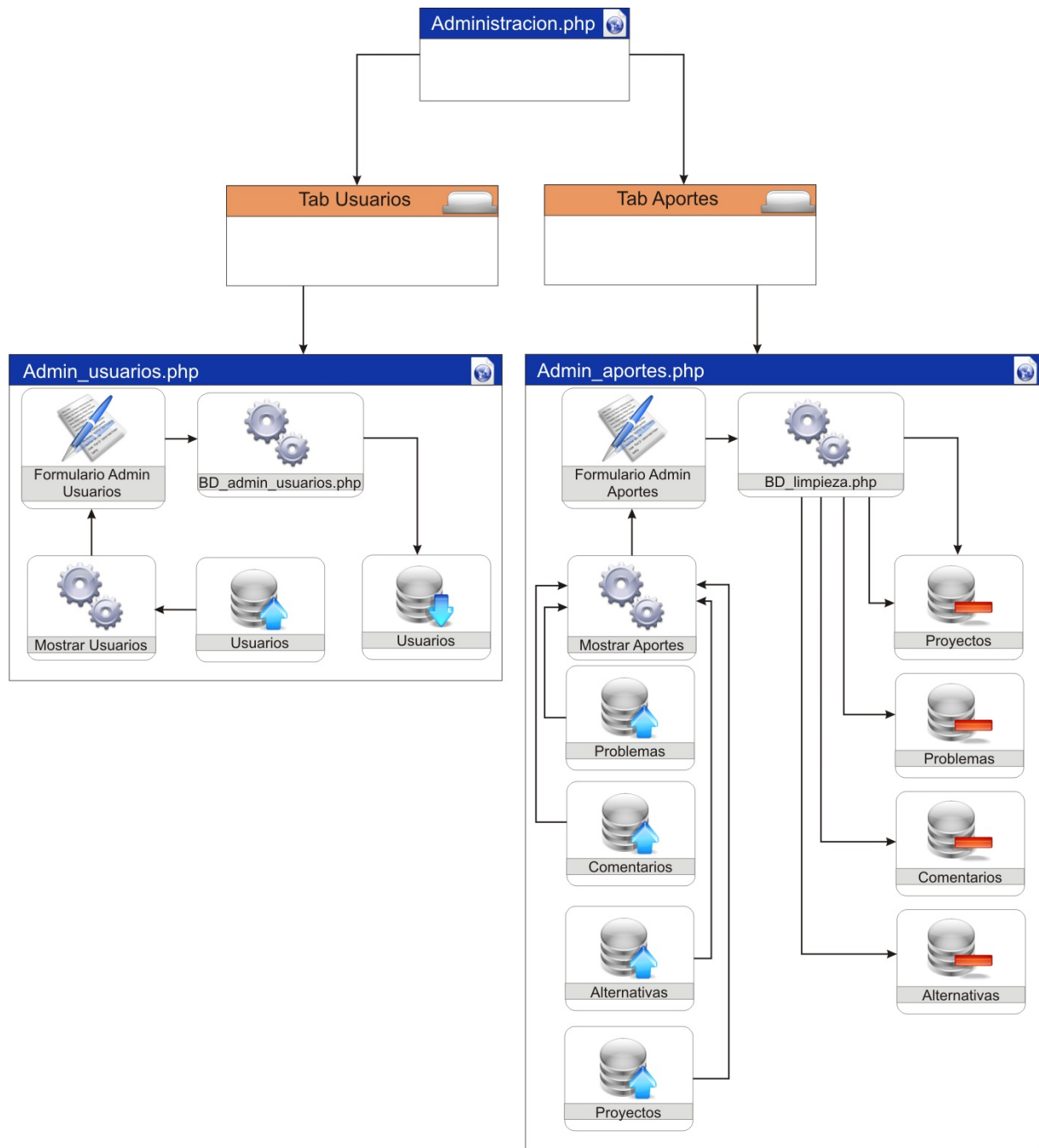


Figura 38 .Diagrama de Navegación - Administración

4. Estadísticas

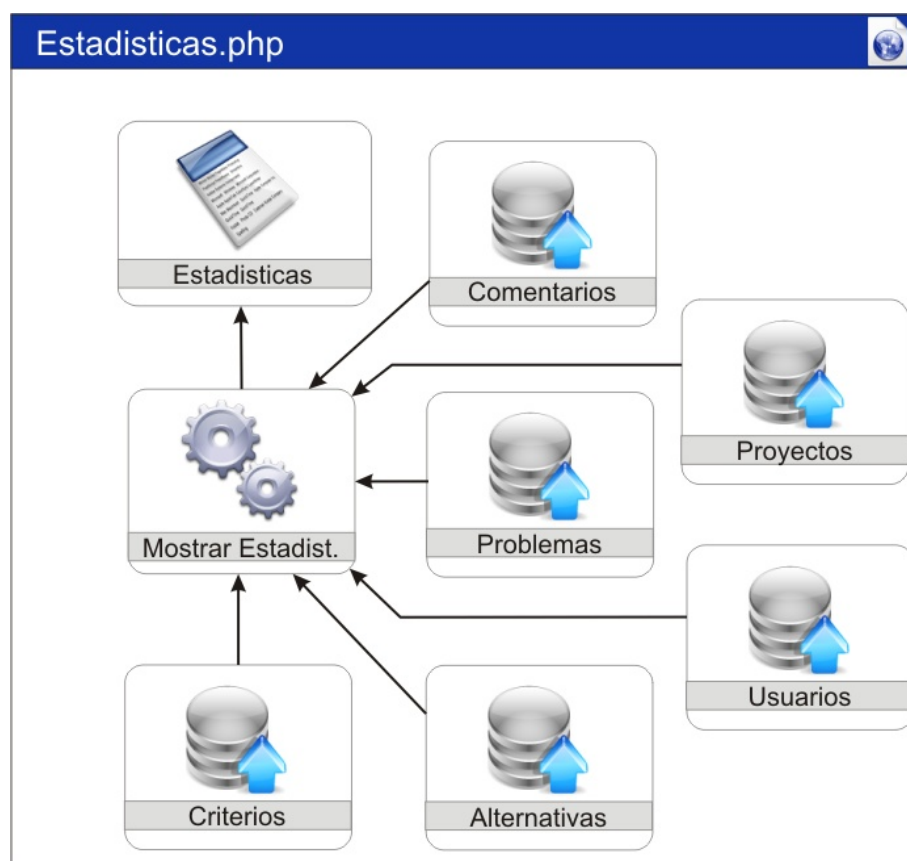


Figura 39. Diagrama de Navegación - Estadísticas

5. Ver Problema

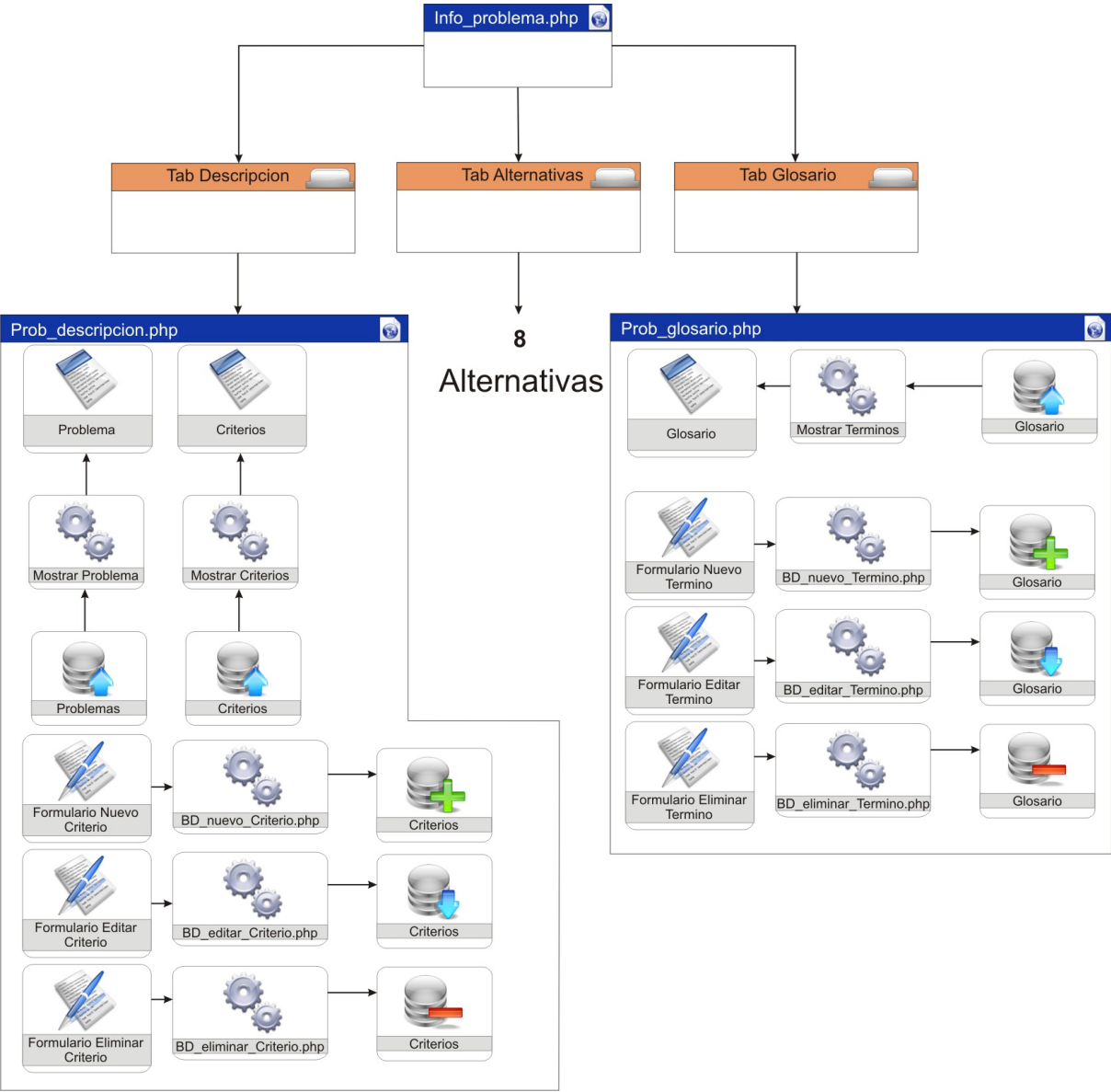


Figura 40. Diagrama de Navegación – Ver Problema

6. Ver Proyecto

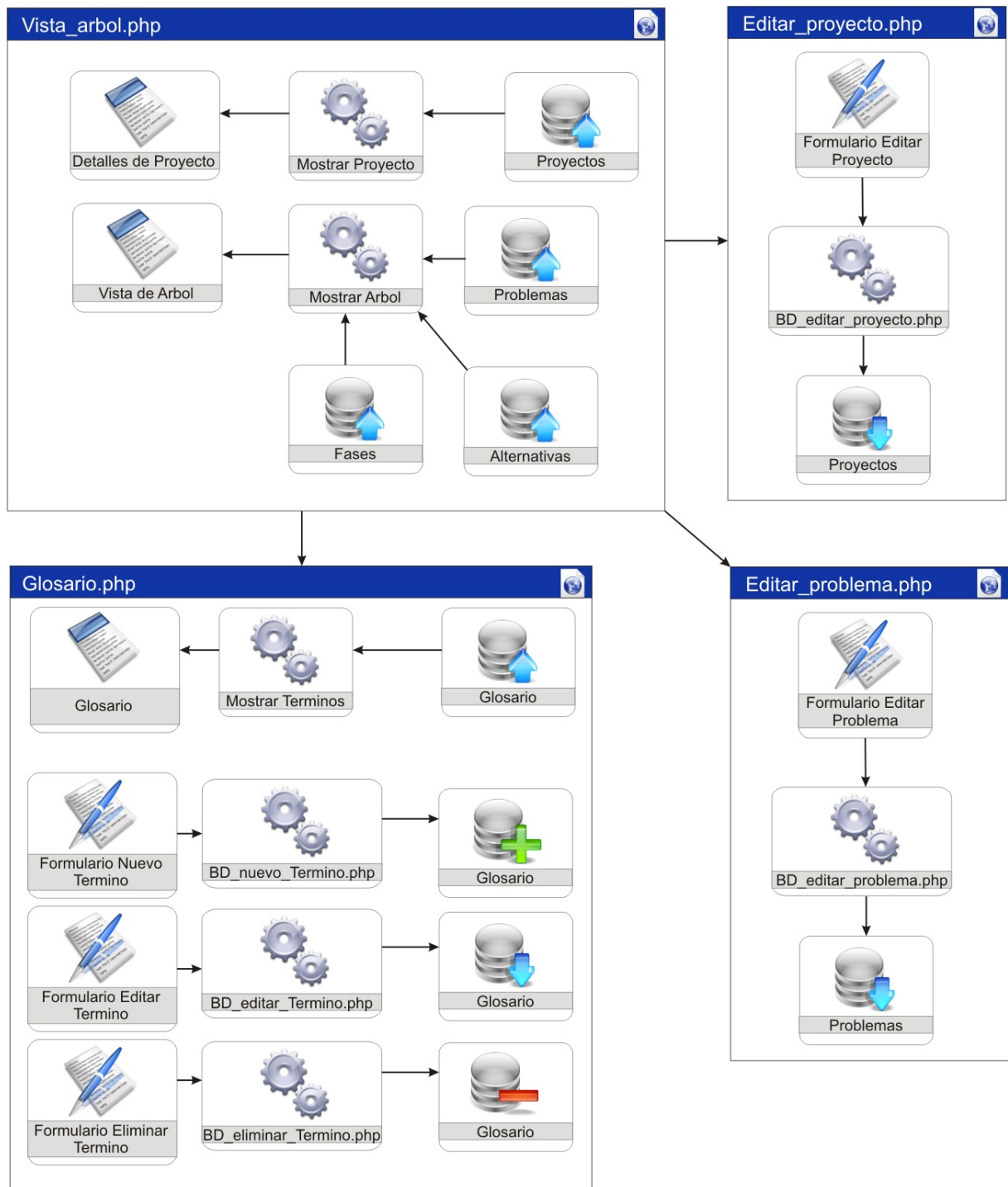


Figura 41. Diagrama de Navegación – Ver Proyecto

7. Nuevo Proyecto

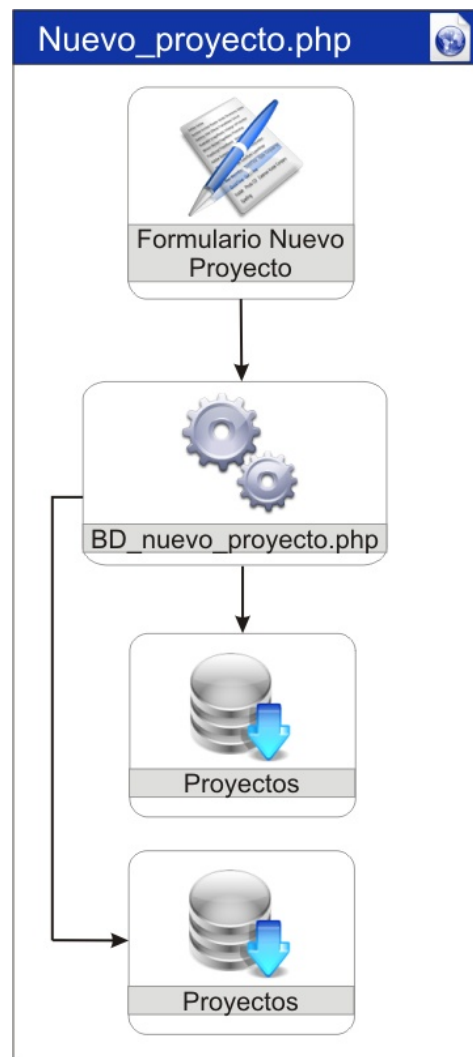


Figura 42. Diagrama de Navegación – Nuevo Proyecto

8. Alternativas

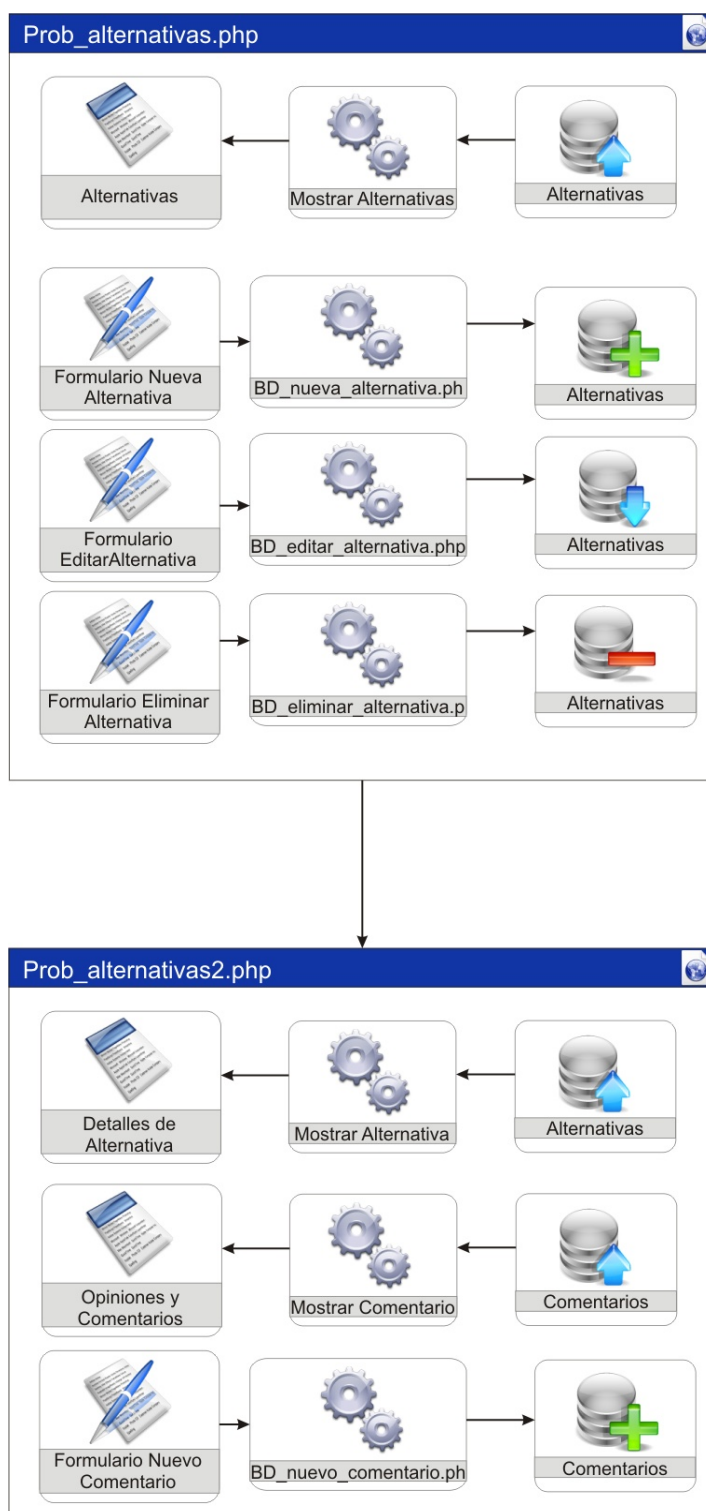


Figura 43. Diagrama de Navegación – Alternativas

8. INTERFACES DEL SISTEMA

Las interfaces de usuario están optimizadas para una resolución de 1024x768 o 1280x1024 para ofrecer mayor campo visual y gran espacio para información. Así mismo se tiene en cuenta el manejo de colores, formas e iconos para familiarizar rápidamente al usuario.

Gracias a Backbase (el motor AJAX), todos los formularios validarán plenamente los datos ingresados por el usuario e informándolo inmediatamente si debe cambiar algo y no le permitirá continuar hasta que lo haga. El sistema también soporta caracteres especiales tales como tildes, ñ, y otros de uso habitual en el español.

Rationale cada vez que va a cargar algo nuevo, informa oportunamente al usuario el estado del sistema con un cuadro que aparece superpuesto en la mitad de la pantalla mientras realiza la operación.

8.1 Pantalla de Bienvenida

Se compone de 3 Zonas o “frames”, el frame superior que contiene un banner con el logo del sistema, el menú principal a la izquierda y el contenido en el frame central.



Figura 44. Pantalla de Bienvenida

8.2 Ingreso al Sistema

Consiste de un formulario en el frame de contenido para el ingreso del usuario si éste ya se ha registrado; si aún no lo ha hecho, puede hacerlo a través del enlace *Regístrate Aquí*. El menú principal permanece desactivado hasta que ingrese correctamente.



The screenshot shows a login interface within a light blue frame. At the top left is a small icon of a person and the word 'Ingresar'. Below this, the text 'Identificate para ingresar al sistema:' is centered. A white box contains the login fields: 'Usuario:' with the text 'admin' and 'Password:' with five black dots. An 'Aceptar' button is at the bottom of the box. Below the box, the text 'Si no estas Registrado, [Regístrate Aquí](#)' is displayed.

Figura 45. Formulario de Ingreso al Sistema

8.3 Registro de Nuevo usuario

Consiste en un formulario de registro en donde se solicitan algunos datos personales del usuario a registrar, tales como un nombre de usuario y contraseña, dirección de correo electrónico, sus nombres y apellidos, fecha de nacimiento, sexo, y nivel de notificación automática, éste campo especifica dado que hayan cambios en el sistema (creación, modificación o eliminación de proyectos, problemas, alternativas, etc.) éste envía un e-mail de manera automática a la dirección que el usuario proporcionó. Cuando el usuario de clic finalmente en el botón Aceptar, si pasa la validación automática, el usuario será registrado en el sistema y se mostrará un mensaje de confirmación con un enlace a la página de ingreso.



Registrarse

Llena el siguiente formulario para registrarte en el sistema:
TODOS los campos son requeridos.

Nombre de Usuario:	<input type="text" value="usuario"/>	(16 Letras Max.)
Contraseña:	<input type="password" value="••••••"/>	(16 Letras Max.)
E-mail:	<input type="text" value="usuario@hotmail.com"/>	(64 Letras Max.)
Nick:	<input type="text" value="El Pibe"/>	(16 Letras Max.)
Nombres:	<input type="text" value="Carlos"/>	(128 Letras Max.)
Apellidos:	<input type="text" value="Valderrama"/>	(128 Letras Max.)
Fecha de Nacimiento:	<input type="text" value="16"/> <input type="text" value="Febrero"/> <input type="text" value="1974"/>	
Sexo:	<input type="text" value="Masculino"/>	
Nivel de Notificación:	<input type="text" value="No Notificarme cambios"/>	

Figura 46. Formulario de Registro


8.4 Menú Principal

Una vez ingrese el usuario correctamente al sistema, se activará el menú principal desde el cual se puede ir a cualquier sección. Si el usuario es administrador, se mostrará también el enlace de *Administración*.




Figura 47. Menú Principal

8.5 Pantalla Inicio




Inicio


Bienvenido: **admin**
Ultima visita: 2008-01-13 11:27:01

[Editar Mi Perfil](#)



Fecha y Hora del Servidor: 13/01/2008 11:27

Mis Proyectos:


[Nuevo Proyecto](#)





[Proyecto A](#)
2008-01-13 08:22:31

[Nuevo Problema](#)[Ver Glosario](#)

Titulo del Problema	Estado	Ultima Actividad
Problema 1	Activo	2008-01-13 08:25:44
Problema 2	Activo	2008-01-13 08:27:02
Problema 3	Activo	2008-01-13 08:34:16




[Proyecto B](#)
2008-01-13 08:38:43

[Nuevo Problema](#)[Ver Glosario](#)

Titulo del Problema	Estado	Ultima Actividad
problema 4	Activo	2008-01-13 11:22:19
Problema 3	Activo	2008-01-13 11:21:47

☒ Abrir Tablero de Mensajes



Tablero de Mensajes Rapidos

Autor:

Decir:

11:29:52 **admin:** aunque deberia hacerse algo mejor

11:30:8 **admin:** mira, a mi forma de ver, creo que el sistema debe ser lo suficientemente rapido

Figura 48. Pantalla Inicio

La primera pagina que se carga después del ingreso, es la de inicio, que contiene información del usuario, los proyectos actuales del sistema y los problemas planteados dentro de los mismos, el usuario puede ejecutar diversas acciones como editar su perfil personal, ver un proyecto en detalle, su respectivo glosario, o ver en detalle algún problema.

En la página de Inicio se incluye adicionalmente un tablero de mensajes rápidos en el cual el usuario puede dejar mensajes a sus compañeros.

8.6 Editar Perfil

Consiste de un formulario que permite modificar los datos personales que ingreso el usuario al momento del registro en el sistema. Incluye cambiar su contraseña, correo electrónico, nivel de notificación automática de cambios en el sistema, entre otros datos personales.



 Editar Perfil de: **admin**

Nick:	<input type="text" value="Walter"/>	(16 Letras Max.)
Contraseña:	<input type="password" value="•••••"/>	(16 Letras Max.)
E-mail:	<input type="text" value="agentwalter@gmail.co"/>	(64 Letras Max.)
Nombres:	<input type="text" value="Walter"/>	(128 Letras Max.)
Apellidos:	<input type="text" value="Serrano"/>	(128 Letras Max.)
Fecha de Nacimiento:	<input type="text" value="1983-06-24"/>	
Nivel de Notificación:	<input type="text" value="No Notificarme cambios"/> ▼	
Sexo:	<input type="text" value="Masculino"/> ▼	

Figura 49. Formulario Editar Perfil de Usuario

8.7 Nuevo Proyecto

En éste formulario se completan los datos del proyecto que se va a crear, junto con sus fases e incluso se puede seleccionar inmediatamente cuáles usuarios existentes tendrán derechos de participante en el mismo. Los usuarios que no se asignen serán tomados automáticamente como *consultores*.



Crear Nuevo Proyecto ?

Título:	<input type="text" value="Proyecto A"/>	(128 Letras Max.)
Descripción:	<input type="text" value="Descripcion del Proyecto A"/>	(255 Letras Max.)
Area de Aplicación:	<input type="text" value="Ingenieria de Sistemas"/>	(64 Letras Max.)
Estado Inicial:	<input type="checkbox"/> Activo <input checked="" type="checkbox"/>	

Fases del Proyecto: ?

Nombre de la Fase	
1.	<input type="text" value="Análisis de Requerimientos"/>
2.	<input type="text" value="Diseño"/>
3.	<input type="text" value="Desarrollo"/>
4.	<input type="text" value="Implementación"/>
5.	<input checked="" type="text" value="Pruebas"/>
6.	<input type="text"/>

Escoge cuales usuarios estaran autorizados para realizar aportes en el proyecto:

IdUsuario	Nick	Habilitar como Participante
usuario	El Pibe	<input checked="" type="checkbox"/>
as	asda	<input type="checkbox"/>
aqweqwe1231	asdasd	<input type="checkbox"/>
asdasd	asdqwe	<input type="checkbox"/>
bnm	sfsdfs	<input type="checkbox"/>

Cancelar

Crear Proyecto

Figura 50. Formulario Crear Nuevo Proyecto

8.8 Ver un proyecto en Detalle

Muestra información detallada del proyecto incluyendo una vista en forma de Árbol jerárquico entre las fases, problemas y las alternativas que hay dentro de cada problema. Pueden ejecutarse varias acciones desde aquí, como Crear un problema o ver el glosario. Si se es el Director del proyecto se mostrarán las opciones de editar el proyecto y los problemas que lo componen.



Ver Proyecto



☒ Ver Detalles

Título:	Proyecto A
Descripción:	Descripción del proyecto A
Fases:	Fase 1 Fase 2 Fase 3 Fase 4 Fase 5
Inicio:	2008-01-13 08:22:31
Última Actualización:	2008-01-13 11:22:48
Estado:	Activo



Vista Arbol


 [Editar Proyecto](#)  [Nuevo Problema](#)  [Glosario](#)

Título	Acción	Tipo
<input type="checkbox"/> Fase 1	----	Fase
<input type="checkbox"/> Problema 1	Editar	Problema
<input type="checkbox"/> Problema 2	Editar	Problema
<input checked="" type="checkbox"/> Fase 2	----	Fase
<input type="checkbox"/> Fase 3	----	Fase
<input type="checkbox"/> Problema 3	Editar	Problema
<input checked="" type="checkbox"/> Fase 4	----	Fase
<input checked="" type="checkbox"/> Fase 5	----	Fase

Figura 51. Pantalla Ver Proyecto

8.9 Editar Proyecto

Es un formulario similar al de Crear proyecto, solo que éste modificará el proyecto existente en vez de crear uno. Posee los mismos campos de detalles del proyecto, fases y usuarios que serán asignados como participantes.



Editar Proyecto

Título:	<input type="text" value="Proyecto A"/>	(128 Letras Max.)
Descripcion:	<input type="text" value="Descripcion del proyecto A"/>	(255 Letras Max.)
Area de Aplicacion:	<input type="text" value="Area de aplicacion del proyecto A"/>	(64 Letras Max.)
Estado:	<input checked="" type="checkbox"/> Activo	

Fases del Proyecto:

Nombre de la Fase	
1.	<input type="text" value="Fase 1"/>
2.	<input type="text" value="Fase 2"/>
3.	<input type="text" value="Fase 3"/>
4.	<input type="text" value="Fase 4"/>
5.	<input type="text" value="Fase 5"/>


Escoge cuales usuarios estaran autorizados para realizar aportes en el proyecto:

IdUsuario	Nick	Habilitar como Participante
usuario	El Pibe	<input type="checkbox"/>
as	asda	<input type="checkbox"/>
aqweqwe1231	asdasd	<input type="checkbox"/>
asdasd	asdqwe	<input type="checkbox"/>
bnm	sfsdfsfs	<input type="checkbox"/>



Figura 52. Formulario Editar Proyecto

8.10 Ver Glosario

Muestra los términos del glosario asociados al proyecto, el cual se irá enriqueciendo a medida que éste se desarrolla. Si el usuario es participante o director, puede crear nuevos términos, y solo si se es director del mismo se pueden modificar o incluso eliminar términos.



Ver Glosario



Terminos asociados al Proyecto: **Proyecto A**

UML	Lenguaje de modelado unificado, permite a los diseñadores de software realizar el proceso de manera sencilla y standard que todos puedan entender
AJAX	Asynchronous javascript y XML, mejora notablemente la usabilidad del sistema mediante una interfaz avanzada

☒ Nuevo Termino +

Agregar nuevo Termino:


Termino	<input type="text"/>
Significado	<input type="text"/>

Enviar


Figura 53. Ver Glosario


8.11 Nuevo Problema

Es un formulario con campos para introducir la información relativa a un problema que ocurra en el proyecto. En el se pueden especificar datos básicos como el titulo, descripción, la fase del proyecto a la que pertenece, el estado inicial, especificar el limite de tiempo, y los criterios y su valoración.



Nuevo Problema ?


[Proyecto A](#)

Título:	<input type="text" value="Problema 1"/>	(128 Letras Max.)
Descripción:	<input type="text" value="Descripción del problema 1"/>	(510 Letras Max.)
Fase:	<input type="text" value="Fase 1"/>	
Estado Inicial:	<input checked="" type="checkbox"/> Activo	
Limitar el problema:	<input type="radio"/> No limitar <input checked="" type="radio"/> <input type="text" value="2008-01-24"/> 	

Criterios del problema: ?







	Título	Descripción	Importancia
1.	<input type="text" value="Criterio 1"/>	<input type="text" value="Descripción del criterio 1"/>	4 
2.	<input type="text" value="Criterio 2"/>	<input type="text" value="Descripción del criterio 2"/>	4 
3.	<input type="text"/>	<input type="text"/>	1 
4.	<input type="text"/>	<input type="text"/>	1 
5.	<input type="text"/>	<input type="text"/>	1 

Figura 54. Formulario Nuevo Problema

8.12 Editar Problema

Formulario similar al de crear Problema, en el que se pueden modificar todos los campos del mismo.



Editar Problema



Título:	<input type="text" value="Problema 1"/>	(128 Letras Max.)
Descripción:	<input type="text" value="Descripción del problema 1"/>	(255 Letras Max.)
Estado:	<input checked="" type="checkbox"/> Activo	
Fase:	<input type="text" value="Fase 1"/>	
Limitar el problema:	<input checked="" type="radio"/> No limitar <input type="radio"/> <input type="text" value="2008-01-30"/> 	

Figura 55. Formulario Editar Problema



8.13 Ver Problema

En esta interfase se muestran datos detallados del problema que se selecciono, el tiempo que queda para realizar aportes en el mismo, se exponen los criterios con su valoración. Si se es Director del proyecto, tiene acceso al panel de gestión de Criterios, que permite crear, editar o eliminarlos. Desde aquí se puede ir a las Alternativas propuestas al problema o incluso al glosario.



Ver Problema

Tiempo para Cerrar el Problema: 16 días
 5% (2008-01-30)



Descripción Alternativas Glosario


☒ Ver Detalles del Problema


Título	Problema 1	
Descripción	Descripción del problema 1	
Fase	Fase 1	
Creado por	Walter	
Inicio	2008-01-13 08:25:44	
Estado	Activo	

☒ Ver Criterios

Criterio 1	Descripción del criterio 1	★★★★★
Criterio 3	Descripción del criterio 3	★★★★★
Criterio 2	Descripción del criterio 2	★★★★★

Gestionar Criterios:

 Nuevo Criterio

 Editar Criterio



 Eliminar Criterio

Figura 56. Pantalla Ver Problema

8.14 Ver Alternativas


Muestra todas las alternativas de solución que tiene el problema en cuestión, quien la realizó, en qué momento y si es considerada como posible solución, si fue rechazada como solución o si es finalmente la solución al problema. Si se es participante ó director del proyecto se muestra la opción de crear una nueva alternativa y si se es Director, se mostrará adicionalmente Editar o eliminar Alternativa. El usuario puede decidir ver una alternativa en detalle, y así conocer su argumento y posibles comentarios y valoraciones que hayan hecho otros usuarios acerca de la alternativa.




Ver Problema

Tiempo para Cerrar el Problema: 16 días

5% (2008-01-30)






Descripcion

Alternativas

Glosario

Alternativa	Autor	Realizada	Estado
Alternativa 1...	admin	2008-01-14 11:25:23	Posible
Alternativa 2...	admin	2008-01-14 11:25:55	Posible
Alternativa 3...	admin	2008-01-14 11:26:21	Posible



Gestionar Alternativas:

☒ Nueva Alternativa 

Alternativa:

Argumento:

Enviar

 Editar Alternativa 




 Eliminar Alternativa 

Figura 57. Ver Alternativas

- 113 -

8.15 Ver Alternativa en Detalle

Si el usuario decide ver en detalle una alternativa, se mostrará información básica de la misma, su argumento base, los comentarios que han realizado los usuarios de la alternativa y la opción de realizar un nuevo comentario. Si se es Director, se mostrará la opción de cambiar el estado de una alternativa de posible a rechazada o solución final. Si se selecciona una alternativa como solución final, el problema quedará automáticamente cerrado, y no permitirá más aportes a menos que el director decida reabrirlo.



Ver Problema

Tiempo para Cerrar el Problema: 16 días
 5% (2008-01-30)

Descripcion

Alternativas


Glosario

[← Volver a Alternativas](#)

☒ Ver Detalles Alternativa

Alternativa	Alternativa 1
Argumento	Argumento de porque la alternativa 1 es una buena opcion para solucionar el problema
Publicada Por	admin
Fecha - Hora	2008-01-14 11:25:23
Estado	<div>Posible</div> <div>Asignar</div>

☒ Ver Comentarios (1)

Detalles	admin - 2008-01-14 11:30:56 -  A Favor
Comentario	Argumento de porque le parece bien esta alternativa







Figura 58. Ver Alternativa en Detalle

8.16 Ver Glosario


Se puede acceder al glosario del proyecto desde cualquier problema del mismo sin salir de él, tiene también las mismas opciones que el glosario de la página principal.

**Ver Problema**

Tiempo para Cerrar el Problema: 16 días
 5% (2008-01-30)




Descripcion Alternativas **Glosario**

**Ver Glosario**

(Terminos asociados a este proyecto)


UML	Lenguaje de modelado unificado, permite a los diseñadores de software realizar el proceso de manera sencilla y standard que todos puedan entender
AJAX	Asynchronous javascript y XML, mejora notablemente la usabilidad del sistema mediante una interfaz avanzada

☒ Nuevo Termino 

Agregar nuevo Termino:

Termino	<input type="text"/>
Significado	<input type="text"/>

Enviar

 Editar Termino



 Eliminar termino

Figura 59. Ver Glosario

8.17 Enviar E-mail

Se puede acceder a ésta opción desde el menú principal y permite enviar un correo a varios destinatarios desde el sistema. Es un sistema de correo sencillo pero bastante útil para facilitar la comunicación. Incluye los campos Asunto, Mensaje, selección de destinatarios y el botón de Enviar E-mail.

 **Enviar Email**

Autor:	admin
Asunto:	<input type="text"/>
Mensaje:	<input type="text"/>

Seleccione El(Los) Destinatario(s):

Seleccionar	IdUsuario	Nick	Administrador	Director	e-mail
<input type="checkbox"/>	admin	Walter	Si	No	agent.walter@gmail.com
<input type="checkbox"/>	usuario	El Pibe	No	Si	usuario@hotmail.com
<input type="checkbox"/>	as	asda	No	No	asdasd
<input type="checkbox"/>	aqweqwe1231	asdasd	No	No	asdad@sakjsdlas.com
<input type="checkbox"/>	asdasd	asdqwe	No	No	lkajsdjklasd@aasdla.com
<input type="checkbox"/>	bnm	sfsdfs	No	No	ajdsasd@dedede.com

Figura 60. Enviar E-mail

8.18 Búsqueda

La opción de búsqueda se puede acceder en el menú principal y se puede buscar en problemas, alternativas, criterios e incluso comentarios que hayan hecho los usuarios acerca de las alternativas. Al mostrar los resultados permite ir directamente al sitio donde está y verlo en detalle.



Administracion

Usuarios

Aportes

Los Usuarios con derechos de administracion no aparecen en esta lista por motivos de seguridad.

IdUsuario	Nick	Nombres	Apellidos	Administrador	Dirigir	Habilitar
usuario	El Pibe	Carlos	Valderrama	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
as	asda	dasd	asda	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
aqweqwe1231	asdasd	sdf	sdfsdf	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
asdasd	asdqwe	asdasd	zxzxczxc	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
bnm	sfsdfs	ertetr	dfgdfg	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Guardar Cambios

Figura 62. Administración de usuarios

8.20 Administración Aportes

Es la sección más delicada del sistema porque permite realizar limpieza de mantenimiento en los aportes, los datos que se seleccionen serán eliminados de arriba hacia abajo en orden jerárquico, es decir, si se decide eliminar un problema, se eliminarán también las los criterios y alternativas propuestas para el mismo y los comentarios que se hicieron para cada una de ésas alternativas. Si se elimina un proyecto se eliminan las fases que lo componen, los roles de usuario asociados al proyecto, los problemas, criterios, alternativas y comentarios que estaban dentro de él. Esta construido en forma de Árbol para facilitar la comprensión del usuario de lo que se está haciendo.

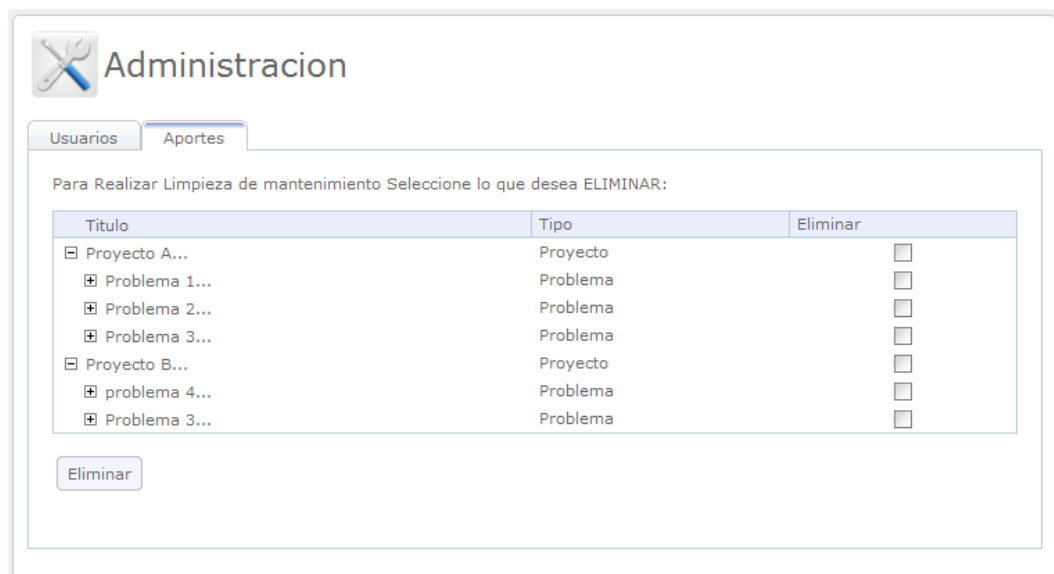


Figura 63. Administración Aportes

8.21 Estadísticas

Se puede acceder a las estadísticas desde el menú principal y permite ver datos estadísticos descriptivos del sistema, como totales, promedios y actividad de proyectos, problemas, alternativas, etc.

Estadísticas

Totales:

Total Proyectos	2
Total Problemas	5
Total Criterios	22
Total Alternativas	3
Total Comentarios	1
Total Usuarios	6

Promedios:

Usuarios/Proyecto	3
Problemas/Proyecto	2.5
Criterios/Problema	4.4
Alternativas/Problema	0.6
Comentarios/Alternativa	0.333333333333

Actividad:

Problemas Resueltos	0
Problemas por Resolver	5

Figura 64. Estadísticas

CONCLUSIONES

- Se creó un software con tecnología Web que facilita resolver y documentar de manera óptima problemas en las organizaciones, permitiendo así a un mejor soporte para el futuro.
- Se diseñó una base de datos relacional que garantiza la integridad, confiabilidad y no redundancia de datos, basados en un desarrollo que tuvo en cuenta la correcta validación de la información antes de ser almacenada.
- El diseño de la interfaz gráfica cumple con las condiciones deseadas que permite una interacción fácil y amigable con la aplicación, pues es el producto de un buen diseño de interfase.
- Dentro de las funcionalidades de la aplicación se puede observar el enfoque a nivel de comunicación entre los usuarios, un factor importante en los requisitos del sistema aportando a la sinergia de la organización.

BIBLIOGRAFÍA

- Bernd Bruegge, Allen H. Dutoit. Ingeniería de software orientado a objetos. Pearson Education, 2002.
- GOMEZ F, Luis Carlos. Planeación de Proyectos, Un enfoque para Ingeniería de Sistemas e Informática. Ediciones UIS, Bucaramanga, 2001.
- HANSEN W, Gary, HANSEN V, James. Diseño y Administración de Bases de Datos. Segunda Edición. Editorial Prentice Hall. España 1997.
- INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACIÓN. Documentación: presentación de tesis, trabajos de grado y otros trabajos de investigación. Colombia. ICONTEC. 2000.
- JOHNSON L, James. Bases de Datos: modelos, lenguajes, diseño. Oxford University Press. México. 2000.
- KENDALL, Kenneth, KENDALL, Julie. Análisis y diseño de Sistemas. Tercera edición. Prentice Hall. México, 1997.
- MCCONNELL, Steve. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. McGraw Hill. España, 1997.
- PARRA P, Leonel. Manual para la elaboración de trabajos de grado para estudiantes para del departamento de sistemas. Ediciones UIS, Bucaramanga 1994.
- PRESSMAN, Roger. Ingeniería del software. Un enfoque práctico. Cuarta edición. McGraw Hill. España, 1998.
- SENN, James. Análisis y diseño de sistemas de información. Segunda edición. McGraw Hill, 1992.