



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DISEÑO Y DESARROLLO DE UN SISTEMA
DE INFORMACIÓN WEB PARA EL
SEGUIMIENTO DE PROYECTOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERA EN COMPUTACIÓN

P R E S E N T A

ROSA ETNA CERVANTES CAMACHO

DIRECTOR DE TESIS

DR. ALBERTO JAIME PAREDES

CODIRECTOR DE TESIS

M.I. JORGE VALERIANO ASSEM



MÉXICO, D.F.

AGOSTO DE 2006



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE

1. INTRODUCCIÓN.....	7
1.1. ANTECEDENTES.....	7
1.2. OBJETIVOS.....	8
1.3. ALCANCES.....	8
2. INFORMACIÓN REQUERIDA PARA DESARROLLAR EL SISTEMA.....	11
2.1. MÉTODO.....	11
2.2. HIPÓTESIS.....	12
2.3. CONTEXTO.....	13
2.4. PLAN DE ACCIÓN.....	13
3. MARCO TEÓRICO.....	15
3.1. INGENIERÍA DE SOFTWARE.....	15
3.2. BASES DE DATOS.....	17
3.3. SERVIDORES.....	20
3.4. LENGUAJES DE PROGRAMACIÓN.....	20
3.5. METODOLOGÍAS DE DISEÑO.....	22
3.5.1. UML.....	22
3.5.2. Diseño de datos.....	23
3.6. HERRAMIENTAS DE SOFTWARE.....	23
3.7. GESTIÓN DE PROYECTOS.....	25
4. DESARROLLO.....	27
4.1. ANÁLISIS.....	27
4.1.1. Diagrama de contexto.....	27
4.1.2. Casos de uso.....	28
4.1.3. Documento de definición.....	35
4.1.4. Determinación de los recursos disponibles.....	39
4.1.5. Selección de tecnologías.....	40
4.2. DISEÑO DEL SISTEMA.....	42
4.3. ARQUITECTURA DEL SISTEMA GEPE.....	43
4.4. DIAGRAMA GENERAL DE LA PÁGINA WEB.....	44
4.5. DISEÑO DE DATOS.....	46
4.6. DESARROLLO DEL SISTEMA.....	48
4.6.1. Base de Datos.....	49
4.6.2. Servidor Web.....	50
4.6.3. Programas generados.....	50
5. RESULTADOS.....	53
5.1. DESCRIPCIÓN DEL SISTEMA DE GESTIÓN DE PROYECTOS (GEPE).....	53
5.2. ANÁLISIS ECONÓMICO.....	59
6. CONCLUSIONES.....	61
7. BIBLIOGRAFÍA Y REFERENCIAS.....	63
GESTIÓN DE PROYECTOS.....	63
INGENIERÍA DE SOFTWARE.....	63
LENGUAJES DE PROGRAMACIÓN.....	63

GLOSARIO	65
ANEXOS	67
ANEXO 1. LISTA DE ACRÓNIMOS	67

Figuras

FIG. 1.	ARQUITECTURA LÓGICA GENERAL DE UNA BASE DE DATOS.....	17
FIG. 2.	DIAGRAMA DE CONTEXTO DEL SISTEMA GEPE.....	27
FIG. 3.	CASO DE USO UC-GEPE1.....	29
FIG. 4.	CASO DE USO UC-GEPE2.....	30
FIG. 5.	CASO DE USO UC-GEPE3.....	30
FIG. 6.	CASO DE USO UC-GEPE4.....	31
FIG. 7.	ARQUITECTURA FÍSICA DEL SISTEMA GEPE.....	43
FIG. 8.	DIAGRAMA GENERAL DE FUNCIONES DE LA PÁGINA WEB.....	44
FIG. 9.	DIAGRAMA DE OPERACIÓN GENERAL DEL SISTEMA GEPE.....	45
FIG. 10.	DIAGRAMA DE COMPONENTES DEL SISTEMA.	46
FIG. 11.	VISTA DE LAS TABLAS DE USUARIOS.	48
FIG. 12.	EJEMPLO DE LA PROGRAMACIÓN EN DBASE 2K.	52
FIG. 13.	PÁGINA DE BIENVENIDA DEL SISTEMA GEPE.	53
FIG. 14.	AUTENTICACIÓN DE USUARIO DEL SISTEMA GEPE.....	53
FIG. 15.	MENÚ DE FUNCIONES DEPENDIENTES DEL USUARIO DEL SISTEMA GEPE.....	54
FIG. 16.	INTERFAZ DE DEFINICIÓN, MODIFICACIÓN Y CANCELACIÓN DE PROYECTOS.	55
FIG. 17.	ACUSE DE RECIBO DE LA BAJA DE 3 PROYECTOS.	55
FIG. 18.	INTERFAZ DE DEFINICIÓN, MODIFICACIÓN Y BAJA DE TAREAS.	56
FIG. 19.	INTERFAZ DE DEFINICIÓN, MODIFICACIÓN Y BAJA DE RECURSOS.....	57
FIG. 20.	CAMBIO DE PERMISOS DE ACCESO A UN PROYECTO.....	58
FIG. 21.	VISUALIZACIÓN GENERAL DE TODOS LOS PROYECTOS.....	58
FIG. 22.	DETALLE DE UN PROYECTO EN OPERACIÓN.	59

Tablas

TABLA 1.	METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	16
TABLA 2.	REGLAS DE NORMALIZACIÓN DE BASE DE DATOS.....	18
TABLA 3.	LISTADO DE LOS PROGRAMAS UTILIZADOS EN EL DISEÑO DEL SISTEMA.....	24
TABLA 4.	LISTADO DE LOS PROGRAMAS UTILIZADOS EN EL DESARROLLO DEL SISTEMA.....	24
TABLA 5.	LISTADO DE LOS PROGRAMAS UTILIZADOS PARA LA IMPLEMENTACIÓN	25
TABLA 6.	RESUMEN DE LOS CASOS DE USO DEFINIDOS PARA EL SISTEMA GEPE.....	28
TABLA 7.	A-GEPE1: ACTOR ADMINISTRADOR DENTRO DE GEPE.	31
TABLA 8.	A-GEPE2: ACTOR INTEGRANTE DENTRO DE GEPE.	32
TABLA 9.	A-GEPE3: ACTOR SUPERVISOR DENTRO DE GEPE.....	32
TABLA 10.	GLOSARIO DE ELEMENTOS DE OPERACIÓN DEL ANÁLISIS POR CASOS DE USO.	33
TABLA 11.	MODELO JERÁRQUICO DEL SISTEMA GEPE.	34
TABLA 12.	SERVIDORES WEB Y SUS PRECIOS.....	40
TABLA 13.	MANEJADORES DE BASES DE DATOS Y SUS PRECIOS.	41
TABLA 14.	TECNOLOGÍAS DEL LADO DEL SERVIDOR PARA GENERAR PÁGINAS DINÁMICAS.	41
TABLA 15.	TABLAS DE METADATOS.....	46
TABLA 16.	TABLAS DE DATOS.....	47
TABLA 17.	SCRIPTS DE GENERACIÓN DE TABLAS PARA LA BASE DE DATOS.	50
TABLA 18.	LISTADO DE LOS PROGRAMAS GENERADOS.....	51

1. Introducción

1.1. Antecedentes

En la División de Ingenierías Civil y Geomática de la Facultad de Ingeniería, UNAM, se realizan proyectos diversos, que van desde las acciones propias de administración e implementación de mejoras, proyectos académicos departamentales y de cada profesor, y servicios a terceros, proyectos y peritajes a gobiernos y empresas.

Los resultados de los proyectos han provocado el incremento de solicitudes, a lo cual se le suma la evolución propia del personal y de la División, generando la necesidad de tener un seguimiento y control de los proyectos más eficiente.

Una manera para atender esta necesidad es aprovechar las nuevas tecnologías para generar un camino alternativo para el manejo, informe y consulta de los avances de los proyectos.

La Gestión de Proyectos propone técnicas y lineamientos para mejorar los resultados de la actividad organizada de grupos de trabajo. Estas metodologías varían en función del ámbito en el que fueron desarrolladas, y aunque algunos lo consideran toda una disciplina¹, de manera general especifican alguna manera de realizar las siguientes tareas:

- Hacer una selección adecuada de los objetivos del proyecto y de los recursos que se han de emplear.
- Definir la tarea que ha de realizarse.
- Especificar el plan de trabajo, el cual implica determinar los recursos humanos y materiales, las tareas que han de realizarse y los productos que han de obtenerse de una manera calendarizada.
- Especificar las técnicas y herramientas que han de utilizarse para dar el seguimiento de las obras.
- Establecer las metas o productos que determinan una adecuada finalización del proyecto² o niveles de operación objetivo.

A pesar de que las organizaciones realizan su actividad a través del trabajo en equipo, la utilización de técnicas para el desarrollo y seguimiento de los proyectos depende directamente del tipo de actividad. Existen áreas de la actividad humana tal como la Ingeniería Civil o la Ingeniería de Software que no podrían haber alcanzado el desarrollo que tienen sin haber generado sus propias técnicas de gestión de proyectos. Sin embargo es relativamente reciente la adopción de estas técnicas en otras áreas como la atención a usuarios. Ante la necesidad económica de ser cada vez más productivos, los esfuerzos por generar mejores técnicas de gestión de proyectos apoyadas en infraestructura de cómputo, se ven reflejados en la importancia, generación e implementación de nuevos estándares³ tales como BS15000⁴ o también llamada ISO 20000 fue el primer estándar de

¹ [Dixon, 2000], 14

² [Ministerio de Administraciones Públicas, 2005], 27

³ [Stephenson, 2004], 2

⁴ [Estándar Británico BS15000]

gestión de servicios de tecnologías de información que sirvió como base para estándares posteriores y su estándar derivado ITIL⁵ – Information Technology Infrastructure Library, que es la metodología más utilizada a nivel mundial de gestión de servicios de tecnologías de información.

El área de gestión de proyectos es grande e implica una diversa gama de actividades que pueden utilizar muy diversas herramientas cuya complejidad, costo y utilidad van en función directa del tamaño de los proyectos que se quieren administrar. Entre los programas de cómputo para tales fines existe una gran variedad de herramientas que van desde las pequeñas como el Microsoft Project, medianas como Métrica3, hasta las empresariales como CA Unicenter Service Plus Service Desk, Niku Clarity.

El presente trabajo se ha enfocado en el desarrollo de una herramienta que sirva para una parte específica de la gestión de proyectos: el seguimiento de tareas.

1.2. Objetivos

El objetivo del sistema de información WEB para el seguimiento de proyectos es ser una página WEB para el seguimiento de las tareas que componen a un proyecto, y de esta manera ser una herramienta de apoyo para la gestión de los proyectos.

En este sistema, que se ha denominado GePe, se muestra la estructura del proyecto, sus avances y recursos asignados. Esta herramienta de apoyo permite el acceso a través de claves de usuario clasificadas por grupos, y la información disponible en el sistema es contextual a ellos.

En el sistema los elementos del equipo son informados acerca de sus tareas asignadas y se les permite informar a través del sistema sus avances y entregar sus informes electrónicos. Los gerentes ven el estado actualizado del proyecto, los detalles de los informes y documentación entregada, y finalmente observan y retroalimentan a sus colaboradores.

Este es un sistema para la División de Ingenierías Civil y Geomática, Facultad de Ingeniería, UNAM.

1.3. Alcances

El sistema permite el seguimiento de proyectos, es utilizable a través de un sitio WEB donde se pueden consultar de manera global y en detalle el estado de los proyectos y sus recursos asignados; donde los usuarios clasificados por tipo pueden entrar y entregar avances que permiten dar seguimiento a los proyectos.

El sistema tiene arquitectura de tres capas y es adecuado para ponerse en funcionamiento de manera inmediata. Es una herramienta con un nivel aceptable de utilidad.

⁵ [Estándar ITIL]

El sistema tiene la funcionalidad siguiente: adecuado para Intranet e Internet, el acceso al sistema está restringido a usuarios autorizados, maneja múltiples proyectos con diferentes grupos de trabajo.

También permite visualizar la estructura de los proyectos, avances registrados al momento y es repositorio de la información, documentos y elementos electrónicos generados en cada proyecto.

Desarrollo del Documento de Tesis

El contenido temático de este documento de tesis es el siguiente. En el capítulo 2 se exponen los elementos de información que se necesitaron para definir la manera en que sería abordado el problema y la estrategia para generar la solución.

Es en el capítulo 3, donde son expuestas brevemente las áreas de conocimiento que fueron investigadas y las conclusiones a las que se llegó en cada una de ellas.

El desarrollo de la tesis es mostrado en el capítulo 4; la manera en que fueron aplicados los métodos y técnicas seleccionados para la generación de la solución del problema planteado, en este caso, el sistema. En éste se muestra cómo se hizo el análisis para la generación del sistema aplicando la metodología UML, la manera en que se hizo el diseño del sistema y los documentos formales producto del diseño y, finalmente, se muestran uno a uno los elementos del desarrollo de este sistema.

Los resultados de la tesis son mostrados en el capítulo 5, donde además se hace un análisis económico del producto terminado. Las conclusiones se muestran en el capítulo 6.

Finalmente se muestra la Bibliografía, el Glosario y la Lista de Acrónimos.

2. Información requerida para desarrollar el sistema

La actividad que consiste en dar el seguimiento de proyectos ejecutados por grupos de trabajo, requiere la recolección de informes, determinación del avance en las tareas específicas, permitir el acceso a información de definición del proyecto. Estas y otras tareas conforman el ejercicio de Informe y Supervisión que permite realizar acciones correctivas y de control de esfuerzo que pueden ayudar a una adecuada ejecución y finalización del proyecto.

Los sistemas de cómputo y las nuevas tecnologías disponibles ponen a disposición herramientas que permiten la automatización, interacción asíncrona, recolección de datos y explotación múltiple de la información, brindando nuevas capacidades, no siempre en sustitución de las anteriores sino sumándose a ellas, para la realización del trabajo.

Es por esto que un sistema de cómputo con interfaz WEB, puede ser el medio para dar el seguimiento de proyectos de manera automatizada y asíncrona, que facilite la disposición de información que las personas necesitan para la toma de decisiones y acciones.

Un sistema de esta naturaleza tiene su utilidad en grupos de trabajo que usen y tengan acceso a medios de cómputo y de telecomunicaciones, y cuyas necesidades de información y retroalimentación requieran el uso de otra herramienta, con el fin de reducir los tiempos de entrega de informes de avance, incrementar los niveles de interacción sin requerir presencia física, registro de avances en el tiempo auditable por terceros y tener un control del estado del proyecto actualizado en cualquier momento.

Como primeros requerimientos del sistema estuvieron el que tuviera un repositorio donde se almacenaran los datos, metadatos y documentos del sistema, con interfaz WEB para permitir el uso sin requerir instalación y a través de diferentes plataformas. Que el acceso a él requiriera de autenticación y que la información mostrada dependiera del perfil del usuario, de manera que las operaciones posibles fueran las autorizadas y adecuadas a él. Finalmente que el sistema permitiera la visualización de los avances de proyecto, de los documentos e informes y de información relevante relacionada.

2.1. Método

Para el análisis del problema se utilizarán los casos de uso de la metodología UML. La selección tecnológica de implementación tendrá en cuenta los recursos disponibles para su desarrollo y uso. El diseño del sistema se hará utilizando la metodología UML ya que es el estándar mundial en especificación de software. Las pruebas y las métricas de software se harán mediante pruebas directas con el usuario.

El repositorio de información puede ser implementado a través de un sistema de archivos o una base de datos. Las bases de datos proporcionan mejores características de seguridad y respaldo que los sistemas de archivos. Sin embargo, se requiere de un sistema de archivos que sirva como antesala a los documentos que se reciben para hacer las verificaciones de seguridad antes de integrarlos al sistema.

Es por esto que se optará por un esquema híbrido donde la base de datos mantenga la información del proyecto y sus metadatos y un sistema de archivos para hacer la recepción de documentos.

Respecto de la selección de la base de datos, tenemos que las más conocidas en el mercado son IBM con DB2 e Informix (35.7% del mercado) Oracle (32.6%), MS SQL (18.1%) y otras como las bases de datos para las plataformas Linux y CA Ingres. Entre las de tipo "open source" la dominante es MySQL, que domina más del 63% de los sitios de Internet. También existen otras bases de código abierto como PostgreSQL, que tienen un mejor soporte para compañías que lo implementan y mejor documentación, aunque su uso no es tan extendido y es utilizable bajo licencia BSD, por la cual tampoco hay que hacer pagos monetarios por ella.

Se utilizará MySQL porque es una base de datos relacional, por su estabilidad ampliamente comprobada y porque no aumentará los costos del sistema, además de que es administrable mediante unos procedimientos calendarizados desde el sistema operativo y dado que en la DICYG no se puede disponer de un administrador de base de datos, es una selección adecuada a los recursos.

La interfaz WEB del sistema se puede hacer mediante plataformas de tecnológicas como HTML, XML y Applet de Java. Estas tecnologías pueden ser implementadas con diversos ambientes de desarrollo. Se empleará HTML implementado a través de dBase 2K, debido a que su arquitectura de generación dinámica de páginas a través de programas ejecutables en el servidor es muy poderosa y segura, no permitiendo accesos directos a la base de datos, incrementando estructuralmente los niveles de seguridad.

2.2. Hipótesis

La hipótesis de la que parte este trabajo es la siguiente: es posible generar una herramienta de software con bajos costos de desarrollo e implementación que sirva para dar seguimiento a las tareas que componen a un proyecto.

Se piensa que esta herramienta de software debe tener la capacidad de trabajarse en red, de manera que múltiples usuarios hagan uso simultáneo de ella. Si el conjunto de datos de configuración (metadatos) y el conjunto de transacciones y estados (datos) se almacenara en una base de datos, quedaría establecido el elemento estructural que permitiera el acceso multiusuario.

Una interfaz web aportaría los beneficios de no requerir instalación en cada máquina de usuario y permitiría a través de la configuración y recursos adecuados el uso por Intranet o por Internet.

El control de acceso y la seguridad de los datos podrían ser implementados a través de la especificación de claves de acceso, que complementados con una clasificación de los usuarios, darían la especificidad de las funciones y registro de los accesos que permitan el control de operación.

Las tareas que el sistema ofrecería serían acceder a la información del proyecto y permitir el registro de los avances en la realización de las tareas que lo componen. De esta manera se podría tener disponible el estado del proyecto conformando una fuente de información actualizada que convertiría al sistema en una herramienta que ayude a dar seguimiento al avance del proyecto.

2.3. Contexto

En la División de Ingenierías Civil y Geomática (DICYG) se llevan a cabo gran cantidad de proyectos que varían en sus características, objetivos, tiempos y grupos de trabajo. Dado que los elementos de cada equipo de trabajo tienen también otras actividades, de manera tal que para ver nuevos avances y resoluciones en determinado proyecto, se requiere de establecer reuniones lo que naturalmente implica la compaginación de agendas.

En la DICYG existen elementos de infraestructura informática como servidores, equipo de cómputo y su red local. Además de esto el personal de la División ya realiza algunos trámites administrativos a través de aplicaciones web.

Es por esto que se vislumbra la utilidad de una herramienta web que sea un medio de comunicación asíncrona entre los elementos de un determinado grupo de trabajo de cada proyecto, permitiendo que tanto el jefe de cada proyecto como sus participantes tengan un medio adicional de informar e informarse, en lo general y en detalle, del estado de avance de las tareas de dichas actividades específicas.

Es deseable que el desarrollo y puesta en operación de esta herramienta no implique costos adicionales para la Institución, sino que tenga características tales que se aprovechen los recursos disponibles.

2.4. Plan de Acción

Para realizar el diseño y desarrollo de un sistema de información web para el seguimiento de proyectos ha de realizarse lo siguiente:

1. Análisis.
2. Determinación de recursos disponibles.
3. Diseño del sistema.
4. Desarrollo del sistema.
5. Pruebas.
6. Liberación.

3. Marco Teórico

En este capítulo se describen brevemente las áreas de conocimiento principales que serán utilizadas en el desarrollo de estos trabajos.

3.1. Ingeniería de Software

La Ingeniería de Software es el área de conocimiento que establece las teorías y las técnicas para el desarrollo de sistemas de cómputo, utilizando tecnologías y prácticas de áreas de conocimiento como las ciencias de la computación, la administración de proyectos, la ingeniería y las áreas de conocimiento específicas del ámbito donde será utilizado el sistema.

Para el proceso de desarrollo de software existen diferentes modelos que describen de manera diferente las actividades o tareas que han de realizarse para generar sistemas de cómputo.

La norma ISO 12207 es un estándar que establece el ciclo de vida de las aplicaciones de cómputo mediante procesos y actividades al adquirir y configurar los servicios de un sistema. Para usar este estándar se requiere seleccionar los procesos e hitos aplicables al proyecto y a las dimensiones del sistema. El estándar establecido por la IEEE/EIA para la Ingeniería de Software es el 610.12. También establece un conjunto de actividades para el desarrollo de sistemas computacionales.

Existen otros modelos de desarrollo entre los que están el Desarrollo Rápido de Aplicaciones, Desarrollo en Incrementos e Iterativo, Modelo en espiral, Modelo de desarrollo de software de código abierto, etc.

Dentro de todas las técnicas que la Ingeniería de Software propone, la arquitectura clásica propone las siguientes actividades:

- Análisis de Requerimientos:** Análisis donde se extraen las características funcionales que va a tener el software a desarrollar en función de lo que los usuarios finales necesitan.
- Diseño:** La manera en que será construido el sistema y se establece tanto su arquitectura como sus componentes y la interacción entre ellos.
- Codificación:** Se programan y construyen los elementos que conformarán al sistema.
- Pruebas:** Verificación de funcionamiento del sistema.
- Documentación:** Generación de los documentos descriptivos de la estructura y funcionamiento del sistema con los objetivos de ayudar al proceso de aprendizaje en el uso y dar mantenimiento futuro a él.

En la actualidad existen diferentes modelos de desarrollo de software, cada uno de ellos tiene diferentes enfoques para analizar, plantear y crear el software. Algunas de las más conocidas se describen en la siguiente tabla.

Tabla 1. Metodologías de desarrollo de software.

Metodología	Descripción
Diagramas de Flujo.	Se describe la solución de un problema a través de una secuencia de tareas y puntos de decisión, de manera que quedan establecidos los caminos de acción. Se utilizan diagramas con rectángulos, rombos y flechas para obtener la representación gráfica o diagrama de la lógica y la secuencia de acciones que realizará el programa.
Programación de arriba hacia abajo.	Este modelo de desarrollo propone hacer primero una descripción general del software e irlo detallando subsecuentemente, refinando el diseño y añadiendo detalle cada vez.
Programación Estructurada	En este modelo se construyen procedimientos y/o funciones que se interrelacionan de una manera funcional para lograr el objetivo del sistema. También se utilizan las estructuras de datos. Se evita el uso de la instrucción goto, haciendo que el flujo del programa esté determinado por su estructura.
Método de Diseño y Análisis de Sistemas Estructurado.	Esta metodología provee una estructura para las actividades del desarrollo de software: coleccionar, almacenar, transformar y distribuir información, de manera tal que se hagan los desarrollos del software económicos y ajustados a las necesidades. La estructura consiste en las siguientes etapas: Análisis de los sistemas actuales, Análisis de requerimientos, Especificación de Requerimientos, de Datos, de la Lógica del sistema, Diseño físico,
Programación Orientada a Objetos.	Esta metodología propone ver a los programas como colecciones de unidades individuales llamadas objetos que interactúan entre ellos. Entonces los objetos deberán ser capaces de enviar y recibir mensajes, y procesar datos, lo que determina un comportamiento específico en ellos. De esta manera se encapsula el detalle operativo, desde el punto de vista de otros objetos y cada objeto es responsable de cumplir su papel.
Rational Unified Process.	Es una metodología de desarrollo de software iterativa y en incrementos, que es adaptable al desarrollo donde se vaya a emplear. Sus principios son desarrollar el software de manera iterativa, administrar los requerimientos, usar los componentes basándose en una arquitectura, hacer modelos visuales del software, comprobar la calidad del software, y controlar los cambios que se hagan a éste.
Máquinas de Estado Finitas.	Es una metodología de especificación de software usado para describir el comportamiento de un sistema de control usando nombres asignados para las entradas (propiedades de control) y las salidas (acciones). Esta es principalmente utilizada en máquinas de control complejas, aplicaciones de instrumentación y de telecomunicaciones.

Modelo de Procesos para la Industria de Software.	Es una norma voluntaria mexicana (tipo NMX) reciente (2005). Es un modelo de procesos para la industria de software nacional, que fomenta la estandarización de su operación a través de la incorporación de las mejores prácticas en gestión e ingeniería de software. La adopción del modelo permite elevar la capacidad de las organizaciones que desarrollan o mantienen software para ofrecer servicios con calidad y alcanzar niveles internacionales de competitividad.
Integración de Modelos de Madurez de Capacidades.	Es una propuesta de mejoramiento de procesos que provee a las organizaciones los elementos esenciales para tener procesos efectivos. Ayuda a integrar las funciones organizacionales, establecer los objetivos y prioridades del mejoramiento de los procesos y dar un punto de referencia para evaluar los procesos actuales.

Se eligió desarrollar bajo la metodología UML debido a su enfoque en el análisis del problema y que permite especificar el sistema; en el sentido de documentar y comunicar, tiene un gran valor utilizar la notación de uso más extendido en el mercado.

3.2. Bases de datos

Las bases de datos son colecciones ordenadas de datos. Dentro del ambiente computacional, están formadas por registros almacenados de una manera sistemática en un sistema de cómputo. De esta forma, un programa de computadora puede consultarlos, las más de las veces a través de la ejecución de preguntas estructuradas de consulta a la base de datos. Los datos son almacenados típicamente como un conjunto organizado de elementos de datos, dicha organización tiene la finalidad de mejorar las capacidades de respuesta en las consultas.

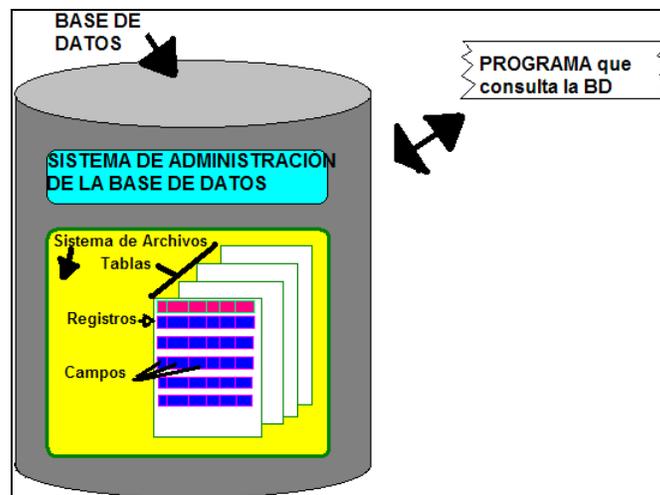


Fig. 1. Arquitectura lógica general de una base de datos.

La arquitectura lógica de las bases de datos se muestra en la figura 1, y está constituida por los siguientes elementos: Sistema de Administración de la Base de Datos, Sistema de Archivos, Tablas, Registros, Campos, Relaciones y Procedimientos Internos.

La base de datos es un conjunto de registros, la descripción y organización de los tipos de información contenidas, así como la manera en que se relacionan constituyen el esquema de la base de datos. Los modelos de base de datos son las diferentes formas de organización de dicho esquema.

Algunos de estos modelos son: El modelo jerárquico, donde los datos están organizados en estructuras arbóreas, las relaciones en ellas son padre-hijo. Las relaciones entre las entidades son del tipo 1-N, uno a muchos. El modelo de red, que es una manera flexible de representar objetos y sus interrelaciones. Cada registro puede tener relaciones N:N de muchos a muchos con registros de otras entidades o tablas. Este modelo es muy útil para bases de datos enormes, abiertas y distribuidas.

El modelo más extendido es el relacional, que consiste en representar la información a través de múltiples tablas con enlaces de correspondencia de integridad de datos entre ellas. Las tablas a su vez estarán conformadas de renglones o registros, y estos de columnas o campos. Es decir, en este modelo se representan todos los datos como relaciones matemáticas k-dimensionales, en donde cada una es un subconjunto del producto cartesiano de k conjuntos, en donde la herramienta de selección de los datos consiste en la evaluación de un predicado lógico, siendo los datos operados a través de proposiciones de cálculo relacional o algebraicas. La consistencia de la representación lógica de la información se consigue a través de la declaración de restricciones en el diseño de la base de datos, también llamado esquema lógico.

También existe un proceso que se ejecuta sobre el esquema lógico llamado normalización, a través del cual se transforma el modelo original a uno de sus equivalentes pero con características específicas deseables. De esta manera se obtiene un modelo donde se mejora la consistencia de los datos a través de la eliminación de la redundancia, organización eficiente de los datos y reducción de la posibilidad de fallos durante la operación. El nivel de normalización del modelo equivalente seleccionado puede medirse a través de las formas normales, que se muestran en la siguiente tabla.

Tabla 2. Reglas de normalización de base de datos.

1FN	Primera forma normal	Establece que el dominio de cada atributo debe estar compuesto de valores simples e indivisibles, y no de conjuntos de ellos. Da flexibilidad en la cantidad de relaciones disminuyendo la complejidad técnica de implementación.
2FN	Segunda forma normal	Establece que un esquema de relación R está en la segunda forma normal si cumple totalmente con la primera forma normal y además cada atributo A no perteneciente a una llave primaria en R, tiene una completa dependencia funcional en su llave primaria. Dicho de otra manera, los atributos en una tabla deben

		<p>dependen del valor de la llave primaria y no del valor de algún otro atributo en ella.</p> <p>Contribuye a que las labores de actualización de información sean procesos sencillos y en un único lugar.</p>
3FN	Tercera forma normal	<p>Establece que una tabla está en la tercera forma normal si cumple con estar en la 2FN y no existen dependencias funcionales no triviales entre los atributos no llave y aquellos atributos que no pertenezcan al superconjunto de atributos candidatos a llave, y si cumple con que todos los atributos que no son llave primaria son mutuamente independientes y sin dependencias transitivas.</p> <p>Evita almacenar información de manera redundante.</p>
4FN	Cuarta forma normal	<p>Establece que una tabla está en la cuarta forma normal si y sólo si no existen dependencias no triviales entre los atributos y ninguna otra cosa que no sea la llave primaria y las dependencias multivaluadas son dependencias funcionales.</p> <p>Sirve para quitar todas las estructuras de datos no deseables, es decir, las dependencias multivaluadas.</p>
5FN	Quinta forma normal	<p>Establece que una tabla está en la quinta forma normal si y sólo si está en la cuarta forma normal y cada dependencia de unión en ella está implicada en alguna llave candidata.</p> <p>Permite que las actualizaciones de atributos que implican dependencias a otros varios elementos puedan hacerse de manera unitaria.</p>

Como se ha visto, las reglas de normalización tienen un objetivo de funcionamiento y/o implementación, sirviendo para ahorrar espacio, disminuir el número de actualizaciones en cadena al actualizar un registro, flexibilizar el crecimiento de los atributos, etc. Sin embargo estas características son deseables sólo si ayudan a cumplir el propósito de los sistemas que usan la base de datos.

Es adecuado normalizar una base de datos cuando es creada para sistemas de procesamiento de transacciones en línea (OLTP), tal como la base de tarjetas de crédito de un banco. No son adecuadas las bases de datos normalizadas para sistemas de procesamiento analítico en línea (OLAP), como una base de datos climatológicos, donde las consultas que se hacen implican el análisis de subconjuntos de datos.

Es por ello que existe el proceso de desnormalización de bases de datos, donde algunas o todas las tablas se transforman en esquemas que no cumplen o cumplen parcialmente con una o varias de las formas normales.

Aún más, en bases de datos específicas y tras las labores de afinación para mejorar el desempeño, aún en los casos de procesamiento de transacciones en línea se desnormalizan tablas específicas para obtener un desempeño adecuado. Existen unas guías generales no estandarizadas llamadas Mejores Prácticas y/o Bases de Conocimiento

para la afinación de desempeño y algunos programas que las usan, para determinar cuáles tablas y qué desnormalizaciones han de realizarse para mejorar el desempeño específico. Estas prácticas incluyen particiones verticales, particiones horizontales, y reversiones de cada una de las formas normales.

Para el sistema se va a utilizar una base de datos relacional, porque a través de la utilización de un servidor de base de datos se tiene acceso múltiple simultáneo y se puede mantener la coherencia entre los datos y tablas.

3.3. Servidores

En el ambiente de sistemas de cómputo se entiende de manera general un servidor como un equipo de cómputo que corre procesos que utilizan otros sistemas de cómputo llamados clientes a través de una red de computadoras.

De hecho este esquema de funcionamiento es llamado arquitectura cliente-servidor. Tiene como ventajas la utilización desde diferentes puntos de las capacidades de cómputo del servidor, permite comunicación de datos a través de aplicaciones y hace las labores de respaldo y de mantenimiento de manera centralizada.

La Arquitectura Cliente-Servidor fue utilizada originalmente por los primeros sistemas de cómputo, y el que se usa actualmente para todos ambientes de mainframes, clusters y supercomputadoras.

Los servidores tienen por lo general características de hardware para mantenerse encendidas 24 horas, mayor capacidad de procesamiento y almacenamiento y sistemas y dispositivos para respaldo de datos. Estas características incluyen fuentes redundantes de datos, conexión a más de una red de computadoras, tecnología de sustitución de partes como discos duros y fuentes de poder sin necesidad de apagarlas, entre otras.

Muchas de los centros de cómputo tienen sus servidores en instalaciones especiales donde se controla el acceso de personal, se provee de energía con sistemas de regulación y/o respaldo, refrigeración ambiental, etc.

Los servidores normalmente son nombrados en función de las tareas que desempeñan como servidores de aplicaciones, servidores de datos, proveedores de servicios, servidores de correo, etc. Esto puede o no implicar que también tengan características de hardware especial que le permitan realizar las funciones objetivo.

3.4. Lenguajes de programación

Existen diferentes tipos y niveles de lenguajes. Van desde los muy apegados a las características arquitectónicas del sistema de cómputo en el que van a funcionar hasta aquellos con los que se producen programas que pueden funcionar en múltiples sistemas, elevando su nivel de abstracción de tareas y especializando las funciones disponibles para objetivos de desempeño específicos.

Existen lenguajes de programación de diferentes niveles. El nivel está determinado por el grado de abstracción entre las funciones del lenguaje y las funciones del microprocesador. Entre más bajo sea el nivel de un lenguaje, más refleja la arquitectura del sistema de cómputo donde se ejecuta. Por ejemplo, el lenguaje de programación C es un lenguaje de procedimientos estructurado mientras que Java es del tipo orientado a objetos.

El lenguaje ensamblador es de bajo nivel, lenguajes como C y Fortran son de mayor nivel, y todos ellos producen programas que sólo pueden ejecutarse en el tipo de computadora para el que fueron compilados, y para correr sobre otra plataforma hay que utilizar otro compilador que corresponda, o definitivamente no pueden utilizarse. Páginas web producidas en lenguajes como Java o HTML sí pueden utilizarse en muy diversos tipos de computadoras, porque cada una tiene de manera nativa, o se le pueden instalar, los elementos de interpretación necesarios para que exactamente el mismo programa se ejecute adecuadamente en cada tipo de computadora.

Dentro del ambiente computacional un lenguaje de programación es aquel que sirve para controlar una computadora. Tiene características como vocabulario, sus reglas sintácticas y semánticas. Su sintaxis define cómo se construye el lenguaje a través de las expresiones regulares definidas con su estructura léxica y su estructura gramatical. Su semántica es la que define las maneras de interpretación de significados.

Con los lenguajes de programación se escriben programas, que serán el conjunto de instrucciones que seguirá la computadora.

La selección del lenguaje de programación se hace en función de los siguientes parámetros:

- Cuál es la tarea que se va a realizar: por ejemplo programar una página web, utilizar una base de datos, hacer un sistema de uso local o distribuido, etc.
- Cuáles características de licenciamiento se desean o se está dispuesto a aceptar: por ejemplo se quiere pagar el derecho de uso anualmente, se quiere una herramienta propietaria que no pague derechos.
- Cuál es la infraestructura sobre la que va a funcionar el producto que se obtendrá, tipo de computadoras, sistemas operativos, restricciones o características de comunicaciones, acceso, seguridad y normas del lugar de implementación.
- Cuáles recursos se tienen para el desarrollo del sistema: por ejemplo se va a contratar el desarrollo o se va a realizar con sus propios recursos. Esto implica tomar en cuenta la cantidad de horas hombre disponibles y conocimientos y dominio de las herramientas.
- De cuánto tiempo se dispone para el desarrollo y la puesta en marcha del proyecto.

El peso de ellos depende de las prioridades y de las restricciones. Si se dispone de un cuerpo reducido fuerza de programación, se seleccionan lenguajes que permitan un desarrollo rápido, a pesar de que esto implique la compra de ambientes de desarrollo, bibliotecas de funciones, herramientas de desarrollo adicionales, etc.

Para el Sistema de Gestión de Proyectos se utilizó el lenguaje de programación dBase para la creación de las páginas web dinámicas. Es un lenguaje orientado a objetos que no

cobra regalías sobre sistemas producidos. Además, dentro de las páginas web se utilizaron los lenguajes HTML y JavaScript para las funciones del lado del cliente y verificación de datos. HTML es un lenguaje basado en marcadores que interpreta el navegador con que se consultan las páginas y JavaScript es un lenguaje básico de tipo guión de instrucciones que interpretan todos los navegadores.

3.5. Metodologías de diseño

Las metodologías de diseño de software son parte del área de conocimiento de Ingeniería de Sistemas. Son producto de la experiencia y de la evolución del conocimiento en el área, así como de la evolución de las computadoras y los sistemas de cómputo.

3.5.1. UML

La metodología de modelado de software más utilizada actualmente se llama Unified Modeling Language (UML) y es producto de la conjunción y mejoramiento de metodologías anteriores como la del modelado de objetos de James Rumbaugh y el modelado de objetos de Gardy Booch y aportaciones de varias otras técnicas. Actualmente varias compañías como IBM y organizaciones académicas, a través del Object Management Group mantienen y desarrollan el proyecto UML, creando nuevas especificaciones y ampliando esta metodología de modelado y se pueden adquirir los documentos originales de especificación (<http://www.omg.org>, <http://www.uml.org>).

Actualmente va en la versión UML 2.0. En esta metodología el software se especifica a través de un conjunto de diagramas de tres tipos: de Estructura, de Comportamiento y de Interacción. La cantidad, tipo y profundidad de diagramas que han de emplearse para especificar un software se basa en la funcionalidad deseada, su complejidad y el tipo de grupo de programación que ha de utilizarse.

Los **diagramas de estructura** son:

- **Diagramas de clases:** En ellos se especifican las clases de objetos que utilizará el sistema, se exponen los objetos que manejarán la información.
- **Diagramas de objetos:** Donde se especifican los objetos instanciados⁶ de las clases y se muestra la relación entre ellos en un punto específico en la operación.
- **Diagrama de estructura compuesta (UML 2.0):** Permite especificar a partir de clases los atributos y las clases a partir de los cuales se definen los atributos, indicando principalmente las asociaciones de agregación de la clase especificada en el diagrama.
- **Diagramas de paquetes:** Aquí se especifican los grupos en los que se puede descomponer un sistema y se muestran las dependencias entre esos grupos.
- **Diagramas de implementación:** Se especifican cómo van a quedar instalados los sistemas dentro de la estructura física de servidores, redes, etc.
- **Diagramas de componentes:** Se especifican los componentes físicos del sistema como archivos, módulos, etc.

Los **diagramas de comportamiento** son:

⁶ Instanciar una clase es el proceso de generar un objeto cuando un programa se ejecuta.

- **Diagramas de actividades:** Es el flujo de control global sobre el negocio, mostrando el flujo del trabajo operacional del sistema modelado.
- **Diagramas de casos de uso:** Da el alcance funcional del sistema respecto de los actores que interactúan con él.
- **Diagramas de máquinas de estados:** donde se especifican a través de máquinas de estados las ocurrencias de los eventos que provocan el cambio de estado de un elemento.

Los **diagramas de interacción** son:

- **Diagramas de secuencia:** Se especifica la secuencia en que los mensajes pasan entre los diferentes objetos en un escenario de operación específico.
- **Diagramas de colaboración (UML 1.x):** Se especifica las interacciones entre los objetos en términos de mensajes secuenciados, mostrando tanto la estructura estática como el comportamiento dinámico del sistema.
- **Diagramas de comunicación y diagrama general de interacción (UML 2.0):** estos dos diagramas reemplazan los diagramas de colaboración. El general de interacción especifica la comunicación entre los objetos principales del sistema a través del pase de los mensajes principales, los diagramas específicos de comunicación hacen lo equivalente de manera más detallada y para escenarios específicos del sistema.

La técnica de desarrollar software a través de un modelado previo en las etapas de análisis y de desarrollo se conoce como Desarrollo dirigido por Modelos. Tiene la finalidad de dirigir, acotar y controlar los esfuerzos de desarrollo, de una manera similar a la función que realizan los planos y proyecto de construcción de un inmueble.

Con los modelos, se puede comunicar las especificaciones de construcción a los grupos de trabajo, estimar costos y esfuerzos y medir el estado del proyecto.

3.5.2. Diseño de datos

El diseño de datos es una metodología utilizada en el diseño de sistemas dirigido por los datos en la que se adaptan los tipos de datos abstractos a la programación orientada a objetos. Esta metodología tiene la ventaja de especificar completamente la estructura física de los datos que utilizarán los objetos, típicamente en una base de datos. Los trabajos más conocidos de esta técnica son los de Wirfs-Brock y Wilkerson.

Típicamente, se obtienen diagramas de entidad relación de la base de datos, diagrama de implementación, diccionario de datos y reglas de nombramiento de los objetos en la base.

3.6. Herramientas de software

Para la elaboración de los modelos del software que se desarrolló se utilizaron los programas listados en la tabla 3.

Tabla 3. Listado de los programas utilizados en el diseño del sistema.

Programa	Compañía	Uso
Case Complete 2.0	Serlio Software http://www.serlio.com	Herramienta de modelado de casos de uso.
DIA 0.95	Fundación de Software Libre http://www.gnome.org/projects/dia	Herramienta para la creación de diagramas UML.
All Fusion ERwin Data Modeler r7	CA http://www.ca.com	Diseño de datos.

Con Case Complete se hicieron los casos de uso, produciendo unos diagramas y unas tablas todos en página web. En él se meten de manera analítica los datos de los casos de uso y el programa genera los diagramas, los informes y el glosario. Con DIA se hacen los diagramas UML, pueden obtenerse en formato de imagen e incorporarse de esta manera a la documentación del sistema. Con ERwin se hace el diagrama entidad-relación de la base y se pueden generar los scripts de creación de la base.

Tabla 4. Listado de los programas utilizados en el desarrollo del sistema.

Programa	Compañía	Uso
dB 2K	dBase Inc. http://www.dbase.com	Lenguaje de programación orientado a objetos. Además se programó en HTML y JavaScript.
MySQL 3.23.37	Free Software foundation http://www.mysql.com	Servidor de Base de Datos relacional.
Apache 1.3	Free Software foundation http://www.apache.org	Servidor web de protocolo HTTP.
Macromedia Dreamweaver MX 6.0	Macromedia http://www.macromedia.com	Diseño y programación de páginas web.
Paint Shop Pro 7.0	Jasc Software http://www.corel.com	Creación de dibujos e imágenes.
UltraEdit 8.	IDM Computer Solutions http://www.ultraedit.com	Edición de archivos de texto.

Para el desarrollo del sistema se utilizaron los programas listados en la tabla4. Con el lenguaje de programación dBase se hicieron los ejecutables que generan las páginas web dinámicas del sistema. Estos son los que se instalan en un directorio de ejecución del servidor web Apache. Con MySQL, además de instalar el servicio de la base de datos, se utilizó el SQL para la generación de la base y sus objetos, corriendo los scripts que produjo el diseño. El Servidor web Apache se instaló para manejar la página durante el desarrollo y pruebas. Con Macromedia Dreamweaver se crearon las páginas estáticas y los modelos que sirvieron para la creación de las páginas dinámicas. El Paint Shop Pro sirvió para dibujar los elementos gráficos de la página. Finalmente el UltraEdit sirvió para las labores de edición y verificación de scripts y páginas web.

Para la implementación del sistema se utilizan los programas listados en la tabla 5.

Tabla 5. Listado de los programas utilizados para la implementación .

Programa	Uso	Ámbito
dB 2K runtime	Contraparte de ejecución requerida para los programas.	Servidor
MySQL 3.23.37	Servidor de Base de Datos relacional.	Servidor
Apache 1.3	Servidor web de protocolo HTTP.	Servidor
Explorador web	Interfaz de acceso al sistema.	Cliente. No requiere instalación específica.

En el servidor hay que instalar 3 programas. Las bibliotecas de ejecución de los programas de dBase (runtime), el Servidor de base de datos MySQL y el Servidor Apache. Estos tres elementos conforman la plataforma requerida para instalar el sistema diseñado y construido en esta tesis.

3.7. Gestión de proyectos

La Gestión de Proyectos es la disciplina que establece los métodos y recomendaciones para ejecutar las tareas de definir y alcanzar objetivos finitos, optimizando la integración y colocando adecuadamente los recursos necesarios para la realización de las tareas que lleven a buen término los objetivos que se han predefinido. De esta manera se genera el proyecto a través de la selección de las actividades que han de realizarse, los recursos que se han de utilizar y estableciendo los objetivos que han de alcanzarse.

Por recursos, se entiende el conjunto de tiempo, dinero, personal, materiales, energía, espacio, suministros, comunicaciones, calidad, riesgos, etc., de que se disponen para la realización del proyecto.

Típicamente para cada proyecto deberá haber un individuo responsable del proyecto cuya función es mantener la productividad y vigilar que los recursos lleguen a tiempo para la realización de las tareas, manteniéndose siempre al tanto del estado del proyecto y solucionando los problemas que surjan para llevar a buen término las actividades según el calendario propuesto.

La teoría de Gestión de Proyectos divide la realización de un proyecto en tres tipos de actividades:

1 Inicio de Proyecto.

- 1.1 Identificar los elementos a desarrollar.
- 1.2 Estimar el esfuerzo.
- 1.3 Planificar.
 - 1.3.1 Estrategia de desarrollo.
 - 1.3.2 Estructura de actividades, tareas y productos.
 - 1.3.3 Calendario de hitos y entregas.
 - 1.3.4 Planificar detalladamente las actividades y recursos necesarios.
 - 1.3.5 Presentar y aceptar la Planificación general del proyecto.

2 Seguimiento y Control del Proyecto.

- 2.1 Asignar detalladamente las tareas.
- 2.2 Comunicar el proyecto al equipo.
- 2.3 Seguimiento de tareas.**
- 2.4 Gestión de incidencias.
- 2.5 Gestión de cambios en los requisitos.
- 2.6 Finalizar tareas.
- 2.7 Actualizar la planificación.
- 2.8 Reuniones de seguimiento.
- 2.9 Aceptación de trabajos elaborados.

3 Finalización.

- 3.1 Cierre de proyecto.

Una vez establecido el proyecto, la tarea más importante es dar el seguimiento de las tareas, en esta actividad la recolección de la información para tomar decisiones y realizar acciones correctivas es crucial para mantener el proyecto en los estándares de tiempo, costo y calidad establecidos en su inicio.

4. Desarrollo

Para el desarrollo de software se utilizan las técnicas de la Ingeniería de Software. En este capítulo se describirán los productos obtenidos por las técnicas seleccionadas, así como las razones de selección y razonamientos principales de análisis, diseño e implementación.

4.1. Análisis

Se realizó un análisis de tarea para encontrar específicamente las funciones que realizará el sistema y que verdaderamente apoyen en el seguimiento de proyectos, que implica la recolección de la información de la realización de las tareas asignadas a los distintos elementos del equipo de trabajo, para la determinación del avance específico y global, tal y como está determinado en el planteamiento del problema, objetivos y alcances en este mismo documento.

El análisis de tarea se hizo a través de la metodología de casos de uso. Para ello se utilizó la herramienta Case Complete 2.0 de Serlio Software.

Se generaron como productos del análisis los siguientes:

- Diagrama de contexto del sistema.
- Casos de uso
 - Cuatro casos de uso
 - Actores
 - Glosario
 - Modelo jerárquico
- Documento de definición del sistema.
- Documento de recursos disponibles.

4.1.1. Diagrama de contexto

El diagrama de contexto nos describe de manera general cómo el sistema va a interactuar con los actores principales y cuáles son las tareas generales que desempeñará. El diagrama de contexto del sistema GePe es el mostrado en la figura 2.



Fig. 2. Diagrama de Contexto del Sistema GePe.

Los actores principales para el seguimiento de proyectos en la DICYG fueron identificados como sigue:

Administrador del proyecto, que es el líder que da el seguimiento y quien tomará las acciones correctivas en caso de que el proyecto no vaya bien, además de ser el responsable del cumplimiento. Las tareas que ha de realizar son principalmente la definición de las tareas a realizar y la asignación de los recursos para cumplirla, además de supervisar las acciones conforme se vayan dando.

Integrante del proyecto, que es el elemento del equipo de trabajo al cual le es asignada la realización de una, parte de una o más de una tarea. Las acciones principales que ejecutará sobre el sistema serán el informarse de sus actividades asignadas, los tiempos y términos de ejecución y actualizar el estado y avance de ellas.

Supervisor del proyecto, que es el rol de supervisión global de uno o más proyectos. Las tareas que ha de realizar son el informarse del estado de avance de los proyectos.

4.1.2. Casos de uso

Para obtener los documentos de especificación formal mostrados en el diagrama contexto, se seleccionó realizar un análisis de casos de uso. De esta manera se obtiene un conjunto de documentos que especifican la funcionalidad requerida al sistema.

Se obtuvieron un conjunto de documentos .htm, que típicamente se pondrían en una Intranet para análisis, revisión y supervisión conjunta.

Se definieron cuatro casos de uso porque fueron considerados los casos de utilización más relevantes, estos se muestran en la tabla 6. UC-GePe1: Caso en que el actor Administrador define dentro del Sistema GePe el proyecto al que se dará seguimiento, UC-GePe2: Caso en el que un actor administrador consulta el estado de avance de un proyecto, UC-GePe3: Caso en el que un actor Integrante del proyecto consulta las tareas que tiene asignadas y UC-GePe4: Caso en el que un actor Integrante del proyecto actualiza el estado de avance de sus tareas asignadas.

Tabla 6. Resumen de los casos de uso definidos para el Sistema GePe.

Use Case Brief Descriptions		
(Sorted by ID)		
Miércoles, 22 de Marzo de 2006, 18:13:19		
ID	Use Case	Description
UC-GePe1	Definir el proyecto	Establecer en el sistema las características del proyecto y las tareas que lo conforman.
UC-GePe2	Consultar el estado del proyecto	Ver cuál es el avance en la realización de las tareas.
UC-GePe3	Consultar las tareas que ha de realizar.	informarse de cuáles son las tareas que debe de realizar un Integrante
UC-GePe4	Informar del avance en la realización de las tareas.	Ingresar en el sistema el estado de las tareas que tiene a su cargo.

La definición de cada uno se muestra a continuación. Cada caso de uso tiene una identificación única de documento, en este caso de la forma UC-GePe(n), un nombre de identificación del caso. Se define también cuál es la estructura jerárquica de operación al establecer si tiene un caso de uso padre, de manera que si se indica el nombre del sistema (GePe) se está indicando que es una tarea básica o de primer nivel.

El actor primario establece al rol principal involucrado en el caso. Para ninguno de los casos fueron definidos actores de apoyo, por no necesitarse concurrencia explícita dentro del manejo para los fines del software.

En la descripción breve se estableció el objetivo principal de la tarea descrita en cada caso de uso.

Finalmente se describió el flujo de los sucesos dado el escenario de operación exitosa. Esta descripción no implica que en la operación real no haya otras actividades o desviaciones específicas, ni tampoco implica que no estén contempladas en el software acciones correctivas para la realización de las tareas tal y como se espera que sean ejecutadas. La descripción hecha en estos casos de uso tienen como fin establecer los escenarios de operación.

Las siguientes tablas fueron producidas con el software de modelado de casos de uso Case Complete y son el reporte de los elementos definidos.

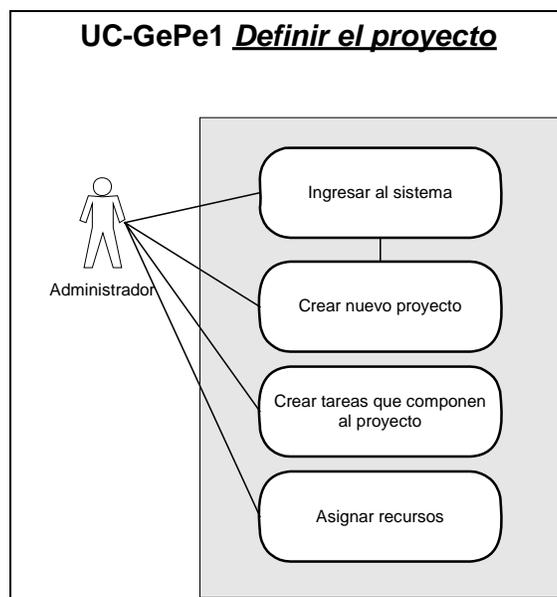


Fig. 3. Caso de uso UC-GePe1.

En la figura 3 se muestra el caso de uso donde el actor Administrador se enfrenta al caso de definir un proyecto dentro del sistema. Para llevarla a cabo requiere hacer las tareas mencionadas en la figura, ingresar al sistema con su nombre de usuario y clave de acceso. El nivel de administrador le permite tener acceso para crear un nuevo proyecto definiendo sus parámetros generales. Después habrá de definir dentro del sistema las tareas que componen al proyecto, estableciendo sus características generales y estableciendo el peso

de aportación relativo que tendrá al avance del proyecto. Finalmente se asignarán los recursos materiales y humanos destinados al proyecto.

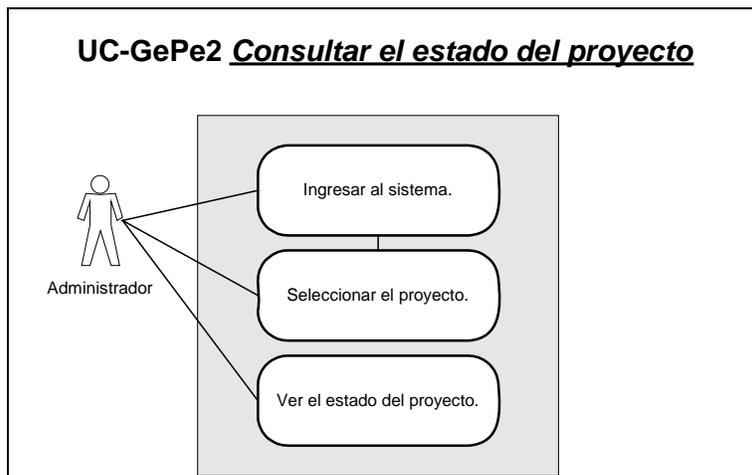


Fig. 4. Caso de uso UC-GePe2.

En la figura 4 se muestra el caso en el que un administrador ha de consultar el estado de avance de un proyecto que ya esté en funcionamiento. Entonces deberá ingresar al sistema, seleccionar el proyecto y revisar su avance. El sistema le informará del porcentaje de avance general del proyecto, el cual es calculado al momento de la consulta a partir del avance de las tareas que lo componen y ponderado según haya sido especificado en el momento de la creación de las tareas.

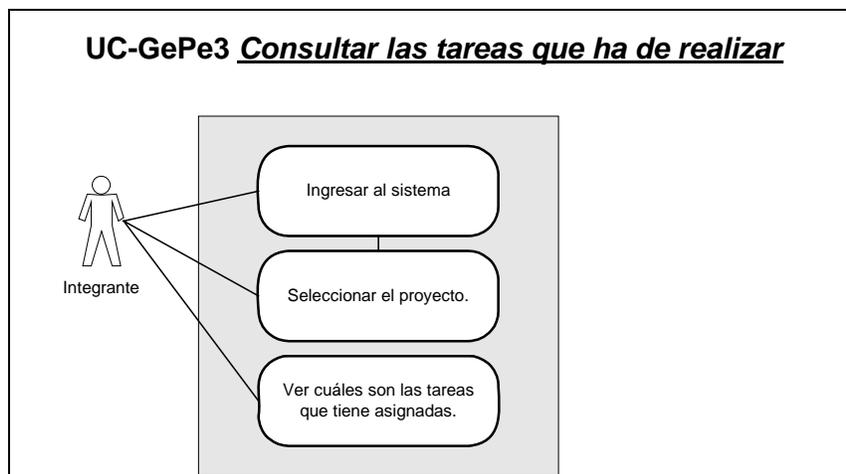


Fig. 5. Caso de uso UC-GePe3.

En la figura 5 se muestra el caso en que un Integrante del equipo de trabajo de un proyecto ha de consultar las tareas asignadas asociadas a un proyecto. Esto requiere que el Administrador haya asignado específicamente una tarea a un grupo o a un Integrante.

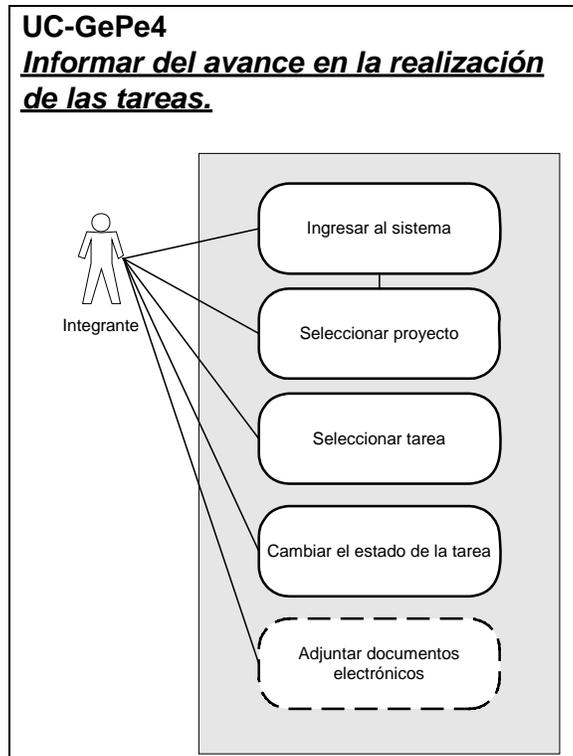


Fig. 6. Caso de uso UC-GePe4.

En la figura 6 se muestra el caso donde un Integrante actualiza el estado de una tarea de un proyecto, para ello ha de ingresar y establecer el contexto. Entonces podrá modificar el estado de avance de la tarea, documentando y respaldando con documentos electrónicos.

Los actores, también requieren de una especificación formal. En este caso todos los actores son representaciones de personas y no de otros sistemas o programas.

Los actores también tienen un código de identificación única, en este caso A-GePe(n), un nombre único con el que siempre son nombrados dentro de otras especificaciones y documentos. Además, se les describe para especificar cuál es la abstracción entendida para el sistema y cuáles son sus objetivos de operación dentro del sistema.

Las siguientes tablas son el producto de la herramienta de modelado Case Complete, que las produce en formato htm para ser incorporadas en la documentación del proyecto.

Tabla 7. A-GePe1: Actor Administrador dentro de GePe.

A-GePe1	Administrador
Parent: GePe	
Description	
Elemento del grupo de trabajo del Sistema quien es el encargado que define al proyecto dentro del sistema para su manejo a través de él Es el responsable directo de la	

arquitectura del proyecto dentro del sistema y requiere del conocimiento de todas las tareas y quién definirá los roles de los Integrantes. Estas estructuras sirven para mantener informado del estado del proyecto a los Supervisor es.

Goals

Definir el proyecto (UC-GePe1)

Consultar el estado del proyecto (UC-GePe2)

En la tabla 7 se muestra la especificación del Administrador. En la descripción se delimitan las funciones que tendrá dentro del sistema. Los objetivos (Goals, en la tabla) definen funciones que habrán que programarse.

Tabla 8. A-GePe2: Actor Integrante dentro de GePe.

A-GePe2	Integrante
Parent: GePe	
Description	
Elemento del grupo de trabajo quien participa en al menos alguna de las tareas y de la cual va a informar el estado.	
Goals	
Consultar las tareas que ha de realizar. (UC-GePe3)	
Informar del avance en la realización de las tareas. (UC-GePe4)	

En la tabla 8 se establece el actor Integrante y las funciones que ha de realizar en el sistema. La especificación de los actores permite una mejor comunicación entre el equipo de modelado de software y el de desarrollo, evitando que sean interpretados de manera diferente los componentes y objetos requeridos para el sistema.

Tabla 9. A-GePe3: Actor Supervisor dentro de GePe.

A-GePe3	Supervisor
Parent: GePe	
Description	
Jefe o directivo de la organización que a través de una clave de acceso puede revisar el estado del proyecto.	
Goals	
Consultar el estado del proyecto (UC-GePe5)	

En la tabla 9 se define al actor Supervisor; su objetivo es obtener información del sistema, al consultar el estado del proyecto.

El glosario es el conjunto de nombres o términos especificados para uso en el modelo. Establece los elementos de uso en el sistema, y que posteriormente quedarán modelados e implementados como objetos o como datos en el sistema. En algunos casos se definen algunos sobrenombres que ayudan a clarificar o expresar el término en algunos ambientes en otros puntos del modelado. El glosario queda definido de la siguiente manera:

Tabla 10. Glosario de elementos de operación del análisis por casos de uso.

Glossary		
Miércoles, 22 de Marzo de 2006, 18:12:42		
Term	Definition	Aliases
Administrador	Encargado de la definición del proyecto, dar el seguimiento y tomar acciones para enfrentar contingencias.	-
clave de acceso	Palabra secreta para cada usuario del sistema con la cual se permite el acceso.	pwd
clave de usuario	Palabra con la que se identifica a un elemento del proyecto y con la cual se permite el acceso y se registran sus movimientos	login
estado	Realidad actual de ejecución de una tarea	-
informar	Actualizar el estado de una tarea	informarse
Integrante	Miembro del equipo de trabajo que realiza alguna tarea.	-
plan	Prospecto de ejecución de una tarea	-
proyecto	Cada elemento objeto de seguimiento, compuesto por las tareas y definido en el GePe.	proyecto
recursos	Bienes o medios que se utilizarán para la realización de una tarea	-
tarea	Elemento unitario que compone a un proyecto , el cual es realizado por al menos un Integrante . Del cual se tiene registrado un plan de acción y un estado de avance	-
Supervisor	Elemento de revisión del estado general de los proyectos.	-

Finalmente, la estructura jerárquica del proyecto muestra las dependencias de existencia y operación requeridas por el sistema. Es relevante sobre todo para sistemas complejos. Para GePe la estructura jerárquica queda modelada como se muestra en la tabla 11.

Tabla 11. Modelo Jerárquico del Sistema GePe.

<p style="text-align: center;">Model Hierarchy</p> <p style="text-align: center;">Miércoles, 22 de Marzo de 2006, 18:12:46</p> <hr/> <p>1. GePe</p> <p><u>Actors:</u></p> <p>A-GePe1 Administrador A-GePe2 Integrante A-GePe3 Supervisor</p> <p><u>Use Cases:</u></p> <p>UC-GePe1 Definir el proyecto UC-GePe2 Consultar el estado del proyecto UC-GePe3 Consultar las tareas que ha de realizar. UC-GePe4 Informar del avance en la realización de las tareas.</p>

El documento donde se concretan de manera formal y completa, todo el análisis realizado es el documento de definición, que forma en sí mismo un documento autocontenido que describe al sistema. Este documento se utiliza para comunicar la manera en que se ve el problema y cómo ha de resolverlo el sistema, tal y como veremos a continuación.

4.1.3. Documento de definición

El documento de definición es la definición formal del sistema en su conjunto de manera conceptual y funcional. En él se establecen los objetivos, alcances y restricciones, problemas que ha de resolver el sistema, usuarios objetivo y otros usuarios del sistema, características objetivo del sistema, la visión general que incluye su propio diagrama de la operación general.

Además se establecen de manera específica las características que tendrá el sistema y los beneficios punto a punto que aportará cada una de ellas.

El documento de definición del sistema GePe es el siguiente:

Documento de Definición
GePe

Sistema de Gestión de Proyectos

Última actualización: 22/03/2006

INTRODUCCIÓN

El sistema de Gestión de Proyectos (GePe), permitirá a través de su interfaz web a un grupo de trabajo hacer el seguimiento, la entrega de informes de avance de las diferentes tareas que componen al proyecto. La información estará definida y respaldada en un servidor con una base de datos en la que se manejarán los diferentes niveles de información.

OBJETIVOS

ÁREA DE OPORTUNIDAD

Dentro de las organizaciones el seguimiento de los proyectos u actividades específicas cotidianas que realizan grupos de trabajo requieren de una mejor comunicación para mantener en contacto a los diferentes elementos del grupo.

Los sistemas de cómputo para el seguimiento de proyectos no son adoptados en las organizaciones principalmente por dos razones. La primera, es la manera en que se está acostumbrado a trabajar en equipo. Los sistemas de cómputo para el seguimiento de proyectos son herramientas adicionales, que permiten la atención de diferentes tipos de asuntos, algo similar al servicio que proporciona un teléfono, que es una herramienta útil para establecer comunicación entre elementos del equipo y que ayuda a solucionar cierto tipo de problemas.

La segunda razón, es por los costos que genera la adopción de una herramienta nueva. Estos costos implican la compra de licencias del software, la capacitación en el uso de los elementos del equipo, la instalación y mantenimiento necesarios para mantener la herramienta funcionando y los elementos de infraestructura necesarios como servidores, computadoras, redes de computadoras, etc.

De esto se vislumbra que, si se puede generar una herramienta de seguimiento de proyectos de bajo costo, con gran facilidad de uso y capacidades de autoadministración, se puede aportar una herramienta de comunicación entre los elementos de los grupos de

trabajo de proyectos de manera tal que se permita obtener el estado del proyecto en cualquier momento durante su ejecución.

PROBLEMA A RESOLVER

Para proyectos en los cuales las tareas que los componen son realizadas por diferentes elementos del equipo, la determinación de cuáles son las tareas específicas que realizará cada uno, y el ejercicio de informar el estado y/o avance de cada una de ellas, puede utilizar un sistema de cómputo en red como herramienta de comunicación, seguimiento, control y almacenamiento de información.

Este seguimiento e interacción entre los elementos de los grupos de trabajo, en ocasiones no puede darse en los niveles de eficiencia y cantidad necesarias por problemas de transportación, coordinación de agendas y el no establecimiento específico de las actividades de proporcionar y recibir información.

De esta manera al tratar de especificar el estado actual de la tarea o del proyecto, se tienen imprecisiones debidas a la falta de información y a la subjetividad de avance.

Las tecnologías actuales de comunicación electrónica y los sistemas de cómputo permiten generar nuevas maneras de interacción, más económicas en tiempo y dinero, que complementan y facilitan los caminos de comunicación entre los elementos del grupo de trabajo, y además permiten contabilizar automáticamente los niveles de avance contra objetivos específicos.

USUARIOS OBJETIVO

Este sistema está pensado para proyectos con equipos de trabajo cuyos elementos hacen uso cotidiano de equipos de cómputo, o que pueden tener acceso y hacer uso de ellos, y en los cuales las tareas que componen al proyecto son realizadas por diferentes elementos del equipo.

CARACTERÍSTICAS OBJETIVO

Las características que se quieren en este sistema de Gestión de Proyectos son:

- Que el sistema permita el manejo de la información relevante para la gestión de proyectos.
- Que permita la interacción ordenada y controlada de los elementos del equipo de trabajo para la comunicación de las tareas a realizar y realizadas y el informe del estado del proyecto.
- Que sea sencillo de utilizar.
- Que sea de bajo costo de implementación y manutención.

OTROS USUARIOS DEL SISTEMA

Elementos de supervisión, como jefes y directivos pueden beneficiarse del sistema al tener acceso en cualquier momento al estado de los proyectos, desde lo general hasta el detalle de las tareas. Este elemento de información da certeza y ayuda a dar el seguimiento que se necesite de manera directa, sin el filtro que siempre supone el informe a través de jefes intermedios, y permite una mejor sensibilización del tipo de actividades que se realizan y el avance en ellas.

Usuarios ocasionales, como los solicitantes de un trámite, proveedores y participantes externos, pueden mantenerse informados del estado de su solicitud o actividad específica a través del sistema. De esta manera se puede dar una atención personalizada y permanente a un bajo costo.

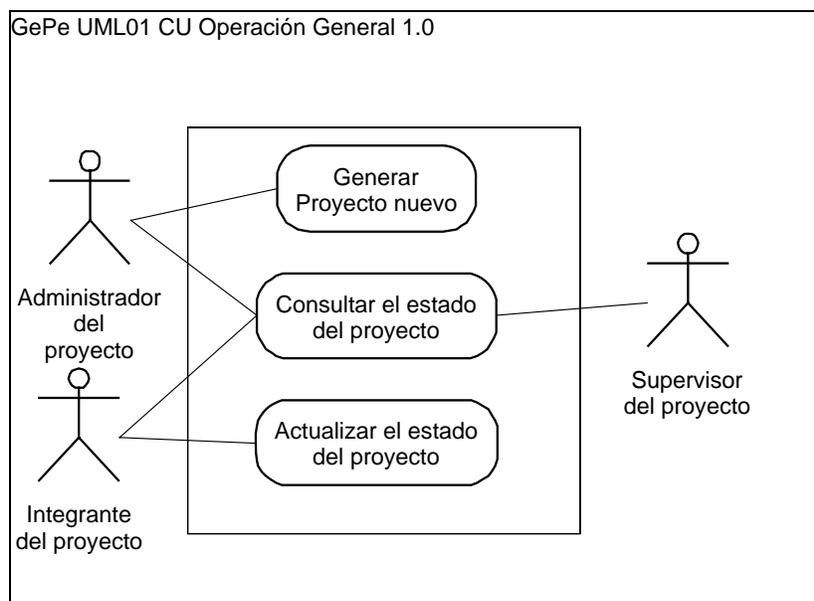
VISIÓN GENERAL DEL SISTEMA GEPE

El GePe se instalará en un servidor y dará servicio a través de la Intranet de la organización o a través de Internet, según sean las necesidades que se quieran cubrir. Luego en el sistema serán definidos los usuarios que participarán en el proyecto, determinando los roles de cada uno y estableciendo sus nombres de ingreso y claves de acceso.

Para cada proyecto, el administrador de éste dará de alta la representación de dicho proyecto y definirá las tareas y subtareas que lo componen, determinando quiénes realizarán unas y otras.

Finalmente, los integrantes del equipo de trabajo entrarán a una página web del proyecto utilizando su clave de acceso para informarse de cuáles tareas requieren realizar e informarán los avances que tienen en cada una, complementando con los documentos específicos requeridos, entregándoselos vía web al sistema.

De esta manera, los administradores y supervisores al ingresar al sistema vía web, podrán ver el estado del proyecto de manera general y de manera particular, teniendo acceso a los documentos complementarios que hayan sido ingresados.



DD Fig 1 Diagrama de caso de uso. Operación general del sistema GePe.

CARACTERÍSTICAS Y BENEFICIOS

Las principales características y beneficios del sistema GePe serán las siguientes:

DD Tabla. 1 Características y beneficios del sistema GePe.

Característica	Beneficios
Interfaz web	<ul style="list-style-type: none"> No requiere instalación del sistema en cada equipo y es multiplataforma. Gran parte del personal que utiliza cotidianamente computadoras sabe cómo utilizar y navegar con interfaces web. Los elementos del equipo pueden acceder a ver o actualizar información desde cualquier computadora.
Servidor web y base de datos.	<ul style="list-style-type: none"> El servidor web dedicado permite la concentración de las funcionalidades, de manera que es el único equipo que requiere mantenimiento de la aplicación y respaldo, bajando los costos que implican la mano de obra especializada en cómputo. La información centralizada facilita las labores de respaldo y mantenimiento de seguridad de la información. El servidor web y la base de datos serán implementadas en herramientas gratuitas que han probado su efectividad y estabilidad, por lo que se bajan los costos por pago de software.
Validación de usuarios	<ul style="list-style-type: none"> Permite el establecimiento de operaciones válidas para cada elemento del equipo en función de su papel, controla el acceso y la información desplegada para cada uno y permite el seguimiento de quién puede hacer cuáles tareas y cuándo las ha hecho.
Auto administración	<ul style="list-style-type: none"> Los mismos elementos de los equipos de trabajo que desempeñan las labores de liderazgo en ellos son quienes pueden ser administradores de sus propios proyectos, de manera tal que parametricen a la herramienta para cada uno de los proyectos que manejan, sin necesidad de que un experto en cómputo realice estas labores.

ALCANCES Y RESTRICCIONES

El sistema de gestión de proyectos GePe que va a ser desarrollado tendrá una visión general en su manejo, con las funcionalidades que han sido ya descritas. Será desarrollado de manera tal que no se requieran licencias de software adicionales para su instalación y operación. El servidor tendrá como plataforma el sistema operativo Windows debido a su amplia utilización.

Responsable del documento: EC

----- Fin del documento de definición del sistema GePe -----

4.1.4. Determinación de los recursos disponibles

Los recursos de que se dispone para la realización de un proyecto determinan la restricción del tamaño y funcionalidad del producto a obtener o la cantidad de recursos que han de obtenerse para la realización del proyecto.

La determinación de los recursos disponibles es uno de los análisis que es importante realizar en la definición de un sistema. Tanto como los demás análisis, es trascendente en el éxito del desarrollo y puesta en marcha, sobre todo porque dirige de manera muy importante la selección de las tecnologías y técnicas que han de utilizarse para la generación del sistema.

Primeramente, se registra la infraestructura de tecnologías de información que se tiene y la que es posible adquirir, cuando se da el caso. Se determina también el perfil del personal que operará al sistema.

El análisis debe realizarse a dos niveles, recursos disponibles para el desarrollo y los disponibles para la implementación.

Los equipos de cómputo de los usuarios son x86 con versiones de Windows 98 y XP. También se dispone de servidores x86 donde podría correr la aplicación.

Con el fin de que la herramienta utilice los recursos que ya dispone el usuario, es conveniente que la aplicación no requiera de la compra de software adicional.

El producto de este análisis es el Documento de Recursos disponibles, que es el siguiente:

DOCUMENTO DE RECURSOS DISPONIBLES

Documento de Recursos Disponibles
GePe

Sistema de Gestión de Proyectos

Última actualización: 22/03/2006

Para Implementación

- Servidor x86 Windows.
- Clientes x86 Windows.
- Navegadores IE4 o FireFox.
- No comprar licencias de software adicional.
- No incluir compra de equipo adicional.
- Perfil de usuarios de uso medio en sistemas de cómputo.

Para Desarrollo

- 640 horas / hombre de programación.

Responsable del documento: EC

En resumen, los productos obtenidos a través del proceso de análisis son los siguientes:

- Un diagrama de contexto.
- Cuatro casos de uso, tres actores, un glosario de elementos de operación.
- Un modelo jerárquico
- Un documento de definición del sistema.
- Un documento de recursos disponibles.

4.1.5. Selección de tecnologías

Para cubrir el objetivo de que sea un sistema que pueda utilizar el personal de la DICYG, es una característica deseable que se acceda a través de una página web, esto tiene como ventajas:

- No requerir instalación especial de software.
- Puede instalarse en la red local de la DICYG o en Internet.
- Cada quién puede acceder desde su propio equipo.

La alternativa a esto es tener un programa que se instale en cada equipo, pero en función de la disponibilidad de personal de apoyo para actividades de mantenimiento de equipo de cómputo, se prefiere que sea web.

Entonces, se requiere de un servidor web que proporcione páginas dinámicas y que utilice una base de datos para contener la información.

Los servidores web disponibles más comunes se muestran en la tabla 12.

Tabla 12. Servidores web y sus precios.

Servidor	Compañía	costo	planta instalada en Internet a julio 2006
Apache HTTP Server	Apache Software Foundation. http://www.apache.org	Gratis (Sólo el programa servidor web)	63.09%
Internet Information Services (IIS)	Microsoft. http://www.microsoft.com	Incluido con el sistema operativo servidor Windows. \$890 US. \$150 US. versión académica. (Sólo el programa servidor web)	29.48%
Sun Java System Web Server	Sun Microsystems http://www.sun.com	\$8,300 US Sun Fire V20z servidor incluido, más \$50 US por empleado por año por el Sun Java Web Infrastructure Suite.	0.37%
Zeus Web Server	Zeus Technology http://www.zeus.com	\$2,600 US (Sólo el programa servidor web)	0.57%

Hay que destacar que a julio del 2006 el 63% de los servidores web del mundo son tipo Apache⁷, y que la tendencia del mercado de los otros tipos va a la baja. Entonces, por conveniencia económica y eficiencia probada, se selecciona el servidor web Apache.

La tecnología que implican las bases de datos para el manejo de información es superior a la alternativa tecnológica anterior de "archivos planos", debido a sus características de estructura, acceso concurrente, seguridad y estabilidad. Ahora, las bases de datos más utilizadas en el mercado se muestran en la tabla 13.

Tabla 13. Manejadores de bases de datos y sus precios.

Manejador de base de datos	Compañía	Precio
DB2	IBM http://www.ibm.com	La UNAM tiene un acuerdo vigente bajo el cual se puede utilizar.
Oracle	Oracle http://www.oracle.com	\$15,000 US por procesador. Edición estándar.
Microsoft SQL Server	Microsoft http://www.microsoft.com	\$475 US EMS SQL Enterprise Bundle for SQL Server
MySQL	MySQL AB http://www.mysql.org	Gratis

Para aplicaciones tipo web, MySQL es utilizada por compañías como Yahoo, UPS, Continental Airlines, Nokia, Lucent, Ericsson, Cisco, UNICEF, United Nations FAO, NASA entre muchos otros⁸. Además tiene las características de base de datos relacional y estabilidad que se requieren para la aplicación que se generará. Además, debido a que utiliza el sistema de archivos del sistema operativo y no su propio modelo como todas las otras, se requiere de muy poca actividad de administración de base de datos, una vez que esté en funcionamiento. Es por ello que se elige MySQL para el sistema.

Para generar páginas web dinámicas existen diferentes tecnologías. Del lado del cliente existen las opciones de JavaScript, Java y VBScript. Java y VBScript requieren de la instalación de un intérprete para que puedan ser ejecutadas, mientras que JavaScript utiliza las capacidades nativas del lenguaje HTML. Es por ello que se elige JavaScript para la ejecución del lado del cliente.

Del lado del servidor se puede elegir una de las interfaces mostradas en la tabla 14.

Tabla 14. Tecnologías del lado del servidor para generar páginas dinámicas.

SSI	Server Side Includes. Adecuado para añadir pequeñas cantidades de información dinámica administrativa a las páginas.
CGI	Common Gateway Interface. Protocolo estándar para enlazar programas externos con el servidor web. En este esquema programas separados del servidor web procesan y preparan el contenido dinámico. Los lenguajes comunes en los que se realizan los programas que CGI correrá como

⁷ http://news.netcraft.com/archives/web_server_survey.html

⁸ <http://www.mysql.com/customers/>

	procesos independientes son C/C++, Pearl, Scripts de sistema operativo, Visual Basic y Fortran.
PHP	PHP: Hypertext Preprocesor. Lenguaje de programación de script compilado parecido a C y a Pearl que cuando se usa del lado del servidor produce las páginas web que el servidor web entregará. Es ampliamente utilizado.
ASP	Active Server Pages. Tipo de soporte de scripts en VBScript que solamente puede ser utilizado con el servidor IIS.
ASP.Net	Nueva versión de ASP de Microsoft que sirve para construir páginas dinámicas y aplicaciones web y servicios web. Forma parte de la plataforma .NET

La desventaja de PHP y ASP estiba en que establecen conexiones directas a las bases de datos, desde el punto de vista del cliente, y entonces hay que establecer mejores controles de seguridad para mantener la seguridad, ya que pueden ser ejecutados códigos maliciosos. Esta desventaja es superada en la nueva tecnología ASP .Net a través de los servicios web. La nueva tecnología .Net tiene muchas ventajas y herramientas de desarrollo que facilitan el desarrollo, mantenimiento y seguridad de las transacciones.

Sin embargo, la tecnología CGI tiene la ventaja de poder utilizarse ejecutables completamente compilados que de manera natural son un freno total a la inspección de código malicioso, si a esto se le agrega comprobación detallada de los parámetros recibidos, los niveles de seguridad aumentan.

La velocidad de desarrollo es también parámetro de decisión, ya que las curvas de aprendizaje pueden afectar las velocidades de entrega del producto.

Es por estas razones que se selecciona el modelo CGI para la generación de contenido dinámico. Los programas que correrán del lado del servidor serán realizados en dBase 2K, que es un lenguaje con manejo completo de objetos que proporciona una velocidad de desarrollo mejor que la de otros lenguajes y se tiene experiencia en ese lenguaje.

Fin del análisis.

4.2. Diseño del sistema

El objetivo del diseño del sistema es obtener el modelo técnico mediante el cual el equipo de codificación ha de generar los programas de éste. En el proceso de diseño se toman decisiones que afectan la implementación, como la selección de las tecnologías sobre las que se implementará.

Primero se tomará en cuenta que el objetivo general del sistema es: Permitir a través de su interfaz web a un grupo de trabajo hacer el seguimiento y la entrega de informes de avance de las diferentes tareas que componen a un proyecto. Luego, se toman en cuenta los documentos generados por el análisis.

El detalle de los lineamientos generales puede verse en el Documento de Definición en la página 35.

4.3. Arquitectura del sistema GePe

La arquitectura de un sistema es la estructura bajo la cual funcionará el sistema, describe su organización fundamental. Incluye los elementos componentes, sus relaciones y los principios que gobernarán su diseño y evolución. (definida en la norma IEEE 610.12 y ANSI/IEEE 1471-2000).

La arquitectura de un sistema puede describirse desde diferentes puntos de vista, funcional o lógica, del código, estructural, concurrencia/procesamiento/hilos, física o respecto de las acciones del usuario.

Se eligió describir la arquitectura física del sistema por la relevancia de implementación que muestra, tal y como se puede ver en la figura 7, donde se puede ver que un equipo que fungirá como servidor, contendrá los elementos de servicio de páginas y la base de datos. A través de la red interna (Intranet) o a través de Internet atenderá las peticiones de los diferentes usuarios, que utilizarán un explorador estándar desde sus equipos de cómputo.

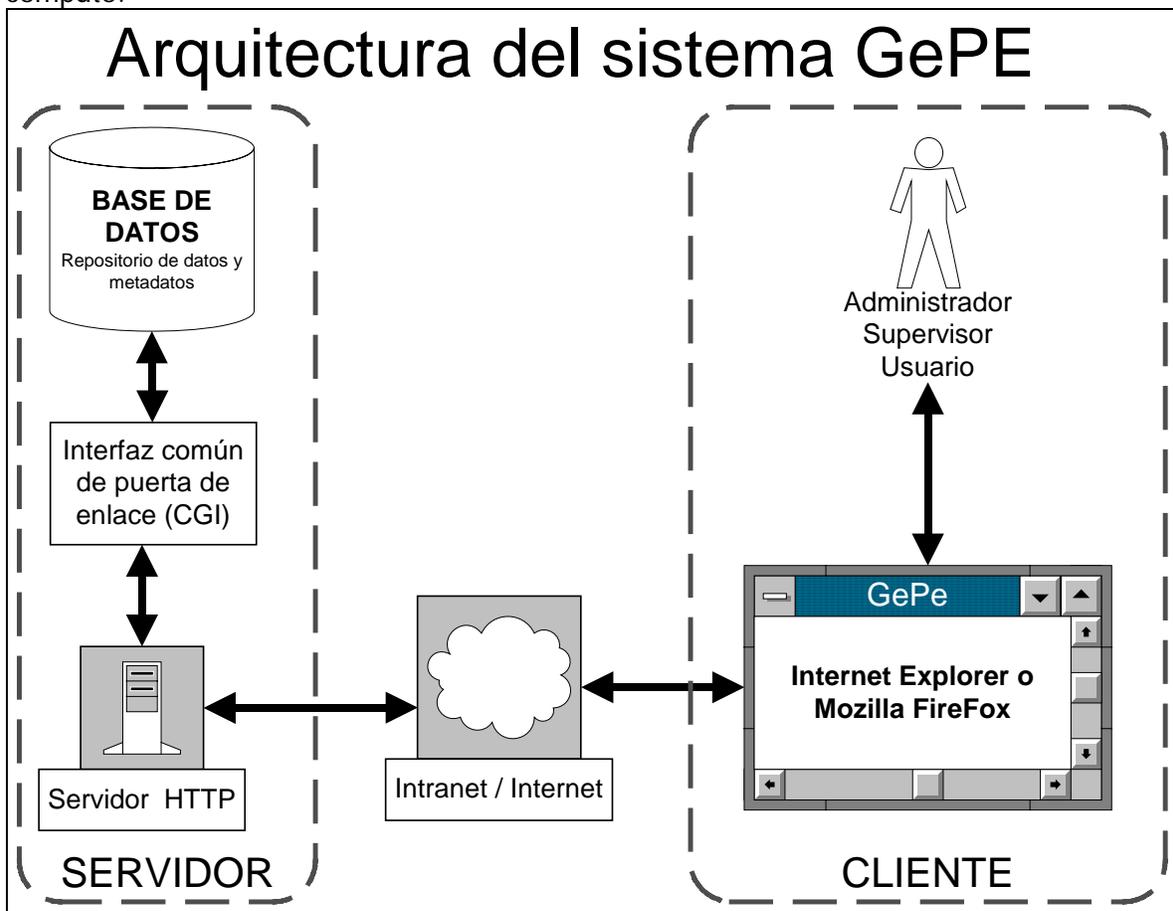


Fig. 7. Arquitectura física del sistema GePe.

4.4. Diagrama General de la Página Web

El diagrama general de la página web del Gestor de Proyectos que se muestra en la figura 8, muestra la estructura funcional desde el punto de vista de codificación. En él queda explícito que las funciones del sistema son accesibles después del proceso de autenticación del sistema.

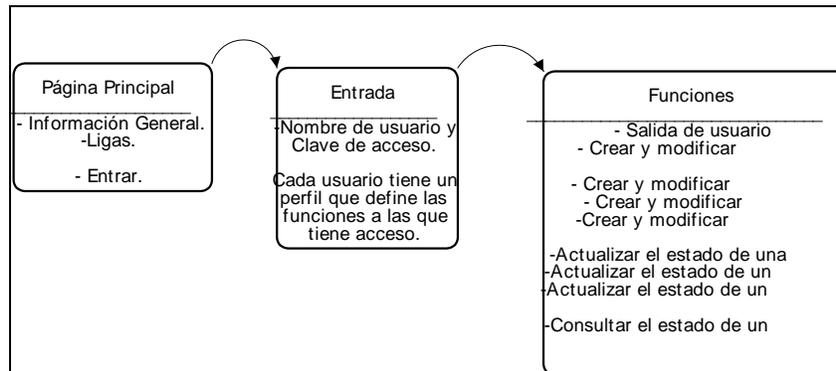


Fig. 8. Diagrama general de funciones de la página web.

El diagrama de comportamiento del sistema muestra a los actores y a las funciones que tendrán en la Operación General, tal y como se muestra en la figura 9, donde se describen a los 3 tipos de usuarios del sistema y a las tareas principales que éste tendrá, que son Generar un proyecto nuevo, Consultar el estado y Actualizar el estado del proyecto.

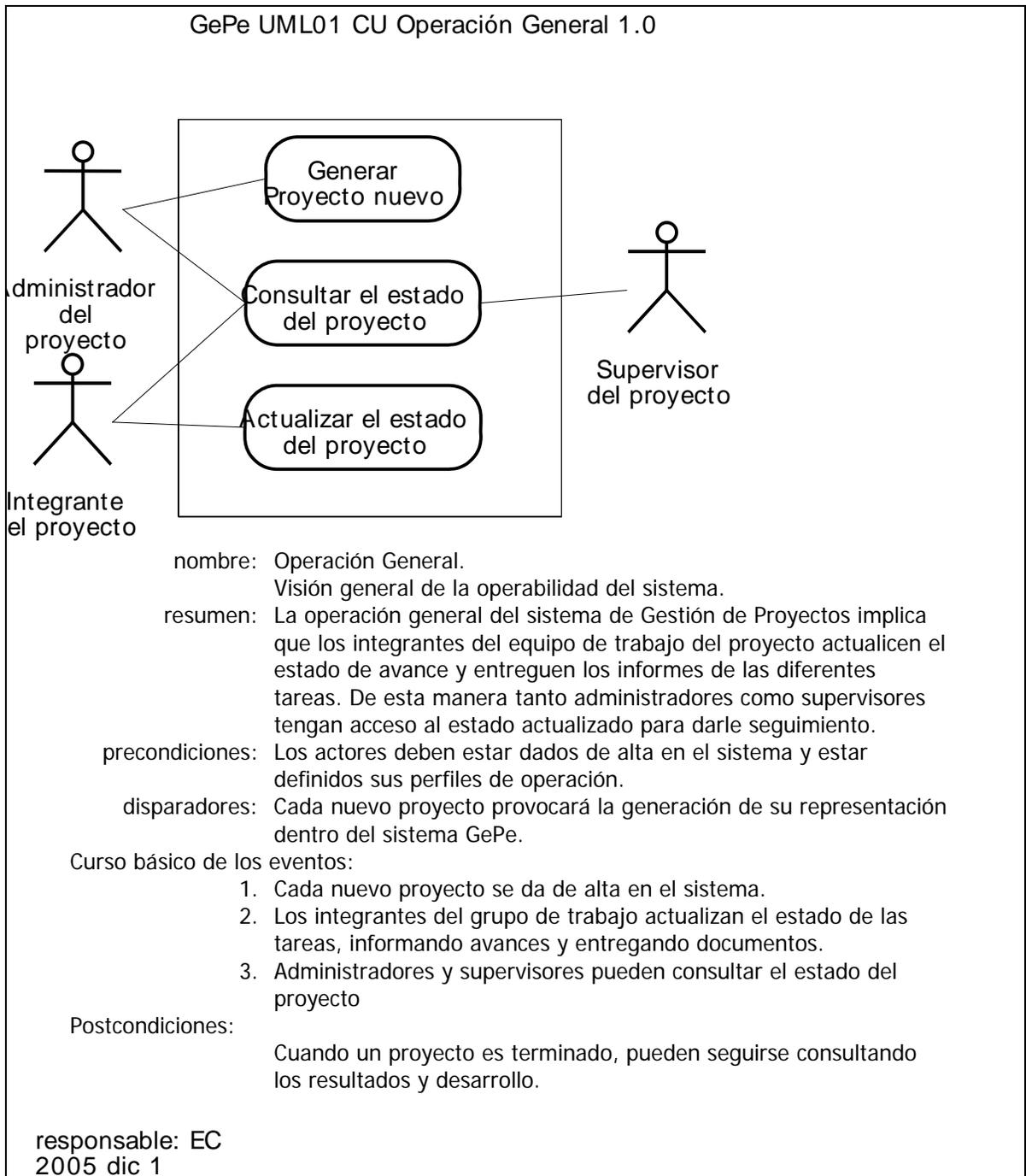


Fig. 9. Diagrama de operación general del sistema GePe.

En el diagrama de componentes del sistema, mostrado en la figura 10, se identifican los módulos de que estará compuesto. La página inicial, el menú de funciones genéricas, y la entrada del usuario serán componentes estáticos; mientras que, los módulos de salida de usuario, menú de funciones adecuadas al usuario, y la operación de cada una de ellas serán definidos como clases que se instanciarán con datos diferentes en función de los metadatos existentes en la base de datos en ese momento.

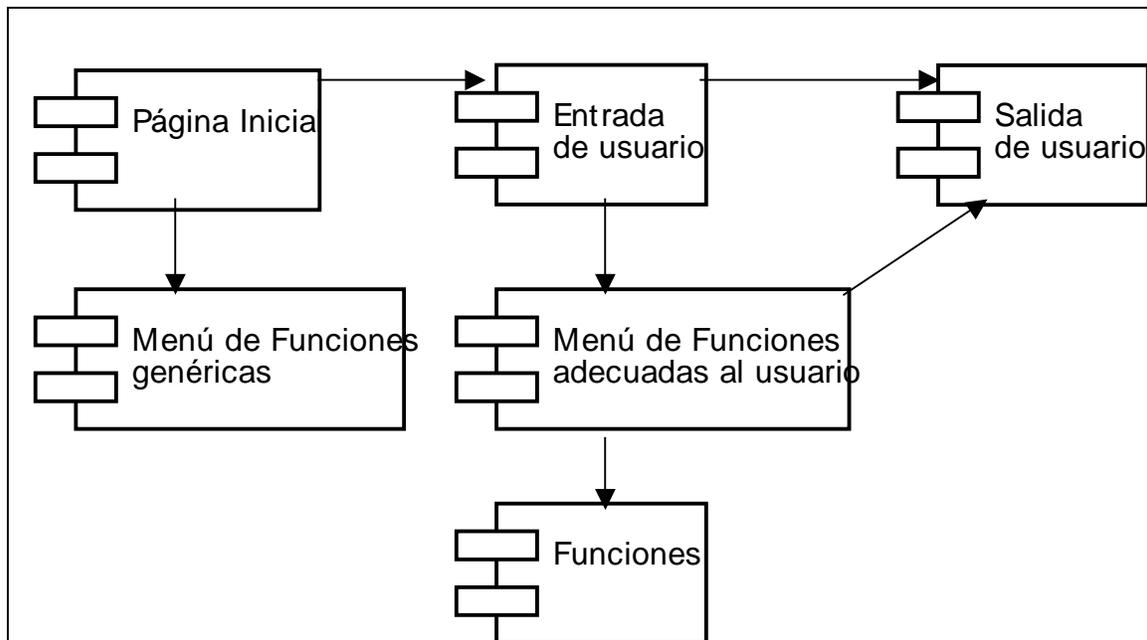


Fig. 10. Diagrama de componentes del sistema.

4.5. Diseño de datos.

En el diseño de datos se definen las características que tendrán las tablas y los campos donde se guardará la información del sistema. Aquí se distinguen 2 tipos de datos, los metadatos del sistema y los datos. El conjunto de información que conformarán los metadatos del sistema serán las características de configuración de operación, y se muestran en la tabla 15.

Tabla 15. Tablas de metadatos.

Contenido	Nombre de la tabla
Catálogo de clases de proyectos	clasesproyectos
Parámetros de las funciones del sistema	funciones
Niveles de las funciones del sistema para control de acceso en función del tipo de usuario.	nivelesfunc
Catálogo para establecer la jerarquía de los proyectos.	nivelesproyectos
Catálogo de tipos de funciones del sistema.	tiposfunciones
Catálogo de tipos de objetos manejados por el sistema.	tiposobjetos
Catálogo de tipos de permisos asignables a los tipos de usuarios.	tipospermisos
Catálogo de tipos de proyectos manejables por el sistema.	tiposproyectos
Catálogo de los tipos de recursos manejables por el sistema.	tiposrecursos
Catálogo de los tipos de tareas manejables por el sistema.	tipostareas
Catálogo de los tipos de usuarios que pueden ser definidos en el sistema.	tiposusuarios
Referencias a los archivos de configuración de estilo globales de las páginas.	webestilos

Los datos del sistema son aquellos que se modifican con la operación del sistema y / o sobre los cuales los usuarios con los permisos adecuados pueden modificar para personalizar el funcionamiento del sistema, tal y como se detalla en la tabla 16.

Tabla 16. Tablas de datos.

Contenido	Nombre de la tabla
Registros de acceso por sesión que indican las operaciones realizadas cada vez por cada usuario.	bitacora
Tabla de registros de verificación de sesión que viajará por internet.	claveasesiones
Catálogo de los estados en los que se puede encontrar un proyecto.	estadosproyectos
Catálogo de los estados en los que se puede encontrar un recurso.	estadosrecursos
Catálogo de los estados en los que se puede encontrar una sesión ofrecida por el sistema a un usuario.	estadosesiones
Catálogo de los estados en los que se puede encontrar una tarea.	estadostareas
Registros que conforman los grupos de los proyectos.	grpproyectos
Registros que conforman los grupos de los recursos.	grprecursores
Registros que conforman los grupos de las tareas.	grptareas
Registros que conforman los grupos de los objetos.	gruposobjetos
Grupos de permisos.	grupospermisos
Grupos de usuarios.	gruposusuarios
Catálogo del estado que define la jerarquía de atención de un proyecto.	jerarquias
Catálogo de lugar asociado a un proyecto.	lugares
Registros de objetos.	objetos
Catálogo de organización asociado a un proyecto.	organizaciones
Catálogo de permisos de acceso a usuarios.	permisos
Tabla de relación entre permisos y funciones.	permisosfunc
Registros de los proyectos.	proyectos
Registros de los recursos.	recursos
Registros de las sesiones de usuario.	sesiones
Catálogo de suborg1 de un proyecto.	suborg1s
Catálogo de suborg2 de un proyecto.	suborg2s
Catálogo de suborg3 de un proyecto.	suborg3s
Registros de las tareas.	tareas
Tabla de relación entre tipos de grupos de usuarios y permisos de acceso de usuarios.	tipogrupopermiso
Tabla de relación entre tipos y grupos de objetos.	tipogrupobjetos
Tabla de relación entre tipos y grupos de usuarios.	tipogrupousuarios
Tabla de relación entre tipos y grupos de proyectos.	tipogrpproyectos
Tabla de relación entre tipos y grupos de recursos.	tipogrprecursores
Catálogo de grupos de tareas.	tipogrptareas
Catálogo de grupos de objetos.	tipogruposobjetos

Catálogo de grupos de permisos.	tipogrupospermisos
Tabla de relación entre tipos de usuarios y permisos de funciones.	tupf
Catálogo de ubicación de proyectos.	ubicaciones
Tabla de relación de permisos de acceso de usuario a proyecto.	usuarioproj
Tabla de relación de permisos de acceso de usuario a recurso.	usuariorecu
Tabla de usuarios.	usuarios
Tabla de relación de permisos de acceso de usuario a tarea.	usuariotare

Una vista del modelo de la base de datos es un subconjunto de tablas. Se utilizan vistas cuando los modelos son grandes o se requiere concentrar en un área o tipo de tablas de la base. La siguiente es una vista que muestra las relaciones entre las tablas para el ámbito de usuarios. En la figura 11 se muestra la de usuarios.

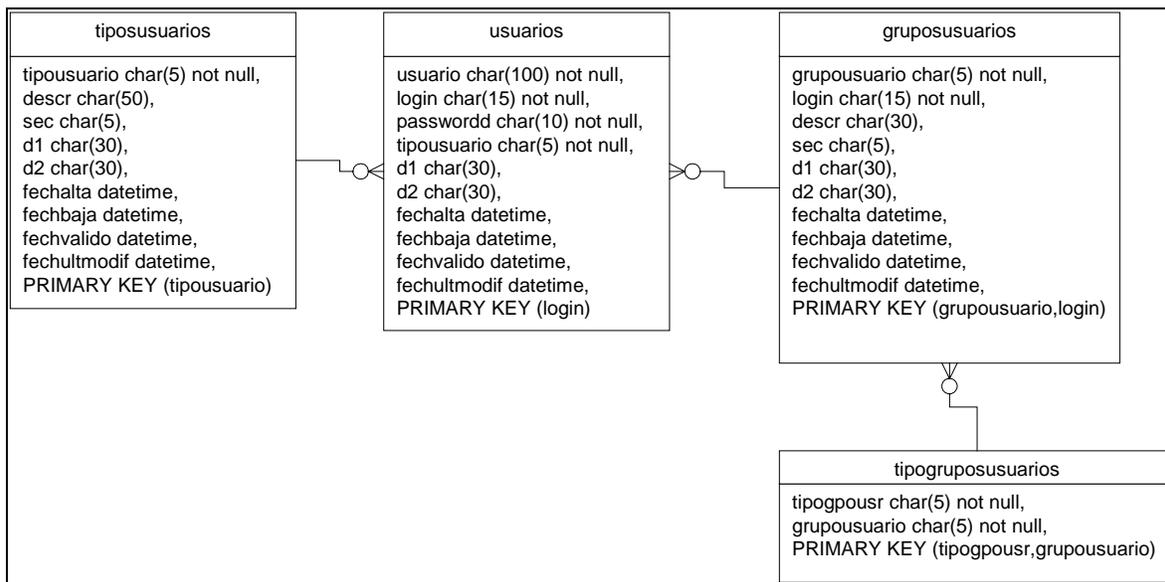


Fig. 11. Vista de las tablas de usuarios.

4.6. Desarrollo del sistema

La etapa de desarrollo del sistema consiste en la programación o codificación de los componentes y procesos de configuración necesarios para la implementación.

Todos los componentes de configuración, creación y código fuente del sistema se incluyen en el disco que acompaña a esta tesis.

4.6.1. Base de Datos

Una vez instalado el manejador de la base de datos en el servidor, se puede generar la base de datos a través de una interfaz de administración o a través de un conjunto de instrucciones en lenguaje SQL que se ejecutan desde línea de comando o a través de la interfaz de comando de línea del manejador.

La interfaz de comando de línea del manejador de la base de datos se accede de la siguiente manera. En una ventana de shell o de comando se cambia al directorio del manejador de la base de datos, por ejemplo:

```
c:\>cd mysql
```

Y después se llama la interfaz:

```
c:\mysql>bin\mysql
```

Entonces se pueden ejecutar los scripts de la siguiente manera:

```
mysql>source creabaseencu.sql
```

Las instrucciones dentro de los scripts instruyen a crear la base de datos, luego la seleccionan para ir creando una a una todas las tablas con sus respectivos campos y llaves, tal y como se muestra a continuación.

```
create database if not exists encu;  
use encu;
```

```
create table encuestas (  
  encuesta char(5) not null,  
  nombre char(30),  
  descr char(250),  
  fechcreac datetime,  
  activa integer(1),  
  valida datetime,  
  ultimamodif datetime,  
  PRIMARY KEY (encuesta));
```

También son metidos los metadatos del sistema, como se muestra a continuación:

```
insert into tipostareas (tipotarea,descr,d1) values  
("ADMIN","Administrativo"," "),  
("ACADE","Academico"," "),  
("LOGIS","Logistico"," "),  
("ALUMN","Estudiantil"," "),  
("CURSO","Curso"," "),  
("CATED","Catedra"," "),  
("PERMA","Permanente"," "),  
("TEMPO","Temporal"," "),  
("EXTRA","Extraordinario"," "),  
("NVOPR","Nuevo proceso en implementacion"," "),  
("SOPOR","Soporte"," ");
```

Los scripts de generación de base de datos que se realizaron para la generación de la base de datos son los mostrados en la siguiente tabla.

Tabla 17. **Scripts de generación de tablas para la base de datos.**

creabaseencu.sql
creatablapermisosfunc.sql
creatablasconf.sql
creatablasfunciones.sql
creatablaspermisosdeacceso.sql
creatablasproyectos.sql
creatablasrecursos.sql
creatablasesiones.sql
creatablastareas.sql
creatablasusuarios.sql
creatablasweb.sql
creatablaTUPF.sql
metecatalogosproyectos.sql
metecatalogosrecursos.sql
metecatalogostareas.sql
metepermisosfunc.sql
nuevasfunciones.sql

4.6.2. Servidor Web

El servidor web debe instalarse de manera que convenga a las condiciones de operación de la red. Las únicas consideraciones especiales es el establecimiento de los alias y los directorios de documentos y ejecución cgi-bin.

En el archivo httpd.conf del Apache deben de incluirse las siguientes líneas:

```
DocumentRoot "C:/Apache Group/Apache/htdocs"  
ScriptAlias /cgi-bin/ "C:/Apache Group/Apache/cgi-bin/"  
Alias /geomatica/ "C:/Geomatica/sitioDICYG/"
```

Para los programas realizados, el servidor web debe de estar en plataforma Windows o utilizar la característica de proxy pass through de un servidor Apache Linux a un servidor Apache Windows donde se utilizará la contraparte proxy pass reverse.

Además debe instalarse el Runtime de dBase para que puedan ejecutarse los programas generados.

En el directorio de documentos "htdocs" del servidor Apache han de colocarse las páginas estáticas del sitio, que incluyen el archivo principal index.html.

En el directorio de ejecutables "cgi-bin" del servidor Apache han de colocarse los programas generados, *.exe.

4.6.3. Programas generados.

Estos programas fueron codificados en lenguaje dBase 2K y forman cada uno un ejecutable para el servidor Apache. Su función principal es recibir los parámetros de una

página web llamada por el usuario y parametrizada por él, operar los cambios al sistema y regresar una página web al servidor Apache para que sea entregada al usuario que hizo la solicitud, presentando en esa página la respuesta a la consulta o el resultado de la operación realizada por el usuario.

Los programas generados se muestran en la tabla 18.

Tabla 18. Listado de los programas generados.

Nombre del programa	Propósito
busca2715.prg	Entrada del usuario y generación de clave de sesión. Registro en bitácora y verificación de permisos y funciones adecuadas al usuario.
busca2717.prg	Regeneración del menú de funciones para usuarios con sesión abierta.
GProAlta.prg	Generación de interfaz para la operación de características de proyectos, altas, bajas y cambios.
GPROALTAP.PRG	Procesamiento de los cambios realizados a los proyectos.
GPROCONS.PRG	Generación de la interfaz de consulta y operación de los proyectos. Altamente recursiva. Genera múltiples tipos de respuesta.
GPreAlta.prg	Generación de interfaz para la operación de características de Recursos, altas, bajas y cambios.
GPreAltaP.prg	Procesamiento de los cambios realizados a los recursos.
GTarAlta.prg	Generación de interfaz para la operación de características de Tareas, altas, bajas y cambios.
GTarAltaP.prg	Procesamiento de los cambios realizados a los Tareas.
GPUsuAlta.prg	Generación de interfaz para la operación de características de usuarios, altas, bajas y cambios.
GPUsuAltaP.prg	Procesamiento de los cambios realizados a los usuarios.
GPUsuEntra.prg	Entrada dinámica al sistema.
GPUSUGRP.PRG	Interfaz múltiple de configuración de catálogos.
GPUsuGrpP.prg	Procesamiento de cambios de la interfaz múltiple de configuración de catálogos.
GPUsuSalir.prg	Cierre de sesión de un usuario.

Fueron programadas más de 40 mil líneas de código.

Un ejemplo de cómo fueron implementadas las clases de las distintas páginas web se muestra en la figura 12.

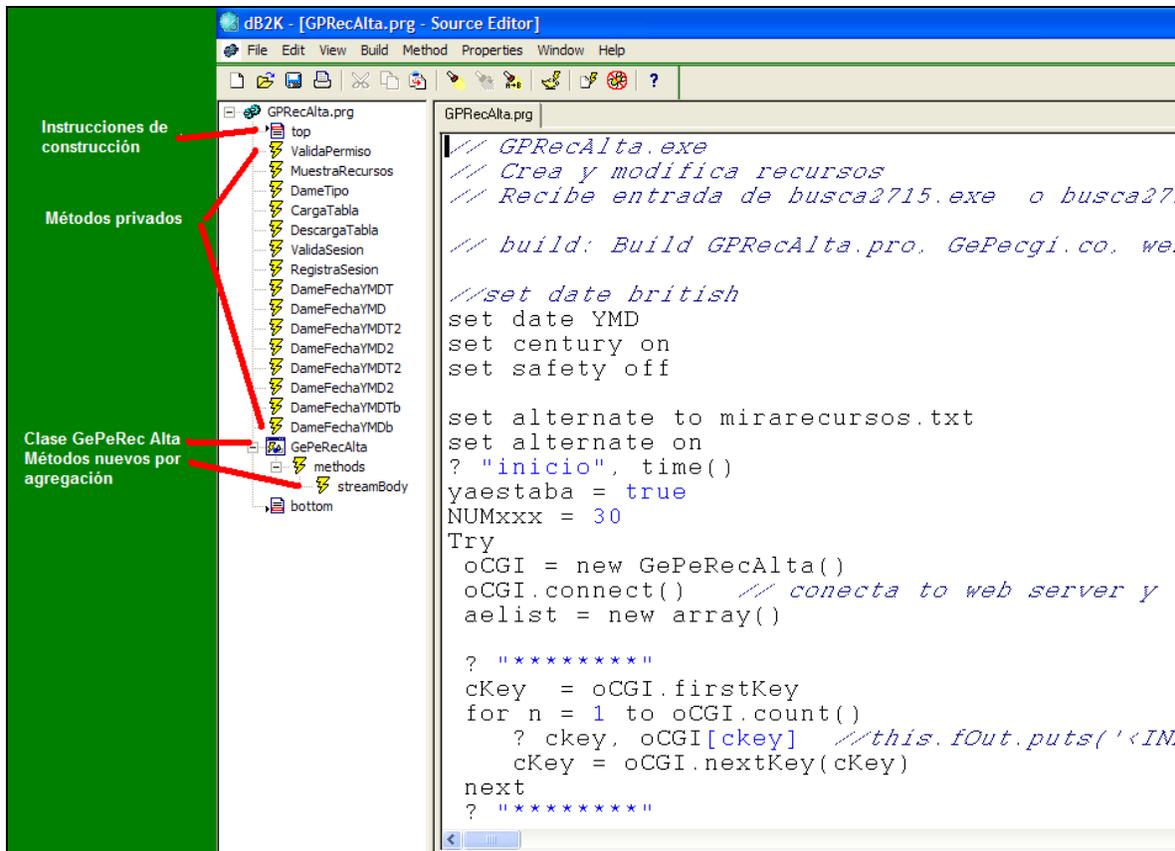


Fig. 12. Ejemplo de la programación en dBase 2k.

Cada tipo de página es una clase hija de la clase GePeCGI, la manera en que se hace la herencia en cada caso, tiene la siguiente forma:

```
Class GePeProAlta of GePeCGISession from "GePeCGI.cc"
```

```
Function streamBody
private i, ubic, a1
set database to
if file("gepeserv.dbf")
a1 = alias()
use gepeserv in select()
select gepeserv
ubic = ltrim(rtrim(ubicacion)).... etc.
```

Todo el código fuente está incluido en el disco que acompaña a este documento.

5. Resultados

5.1. Descripción del Sistema de Gestión de Proyectos (GePe)

Se obtuvo un sistema con interfaz web que tiene elementos estáticos y dinámicos. A través de éste se puede dar el seguimiento de un proyecto. La figura 13 muestra la página inicial.

La parte estática tiene información general y es de acceso público.



Fig. 13. Página de bienvenida del sistema GePe.

El estilo y fondo de la página puede cambiarse en el servidor a través de una hoja de estilos encu.css. De esta manera se pueden actualizar los tipos de letra, tamaños, colores y fondos que ha de utilizar la página. Las páginas dinámicas también usan la hoja de estilo mencionada, de manera que se puede mantener una imagen uniforme y corporativa de la aplicación.

La entrada al sistema se realiza utilizando un nombre de usuario y una clave de acceso como se muestra en la siguiente figura:

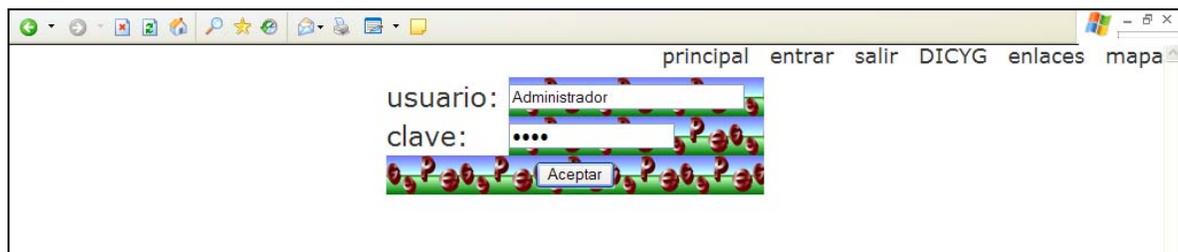


Fig. 14. Autenticación de usuario del sistema GePe.

Entonces se despliega el menú que depende de las funciones autorizadas para cada usuario. Aún que se tengan las mismas funciones, los usuarios están sujetos a

condiciones de ámbito que están relacionadas con los proyectos asignados. Un ejemplo se muestra en la figura 15.

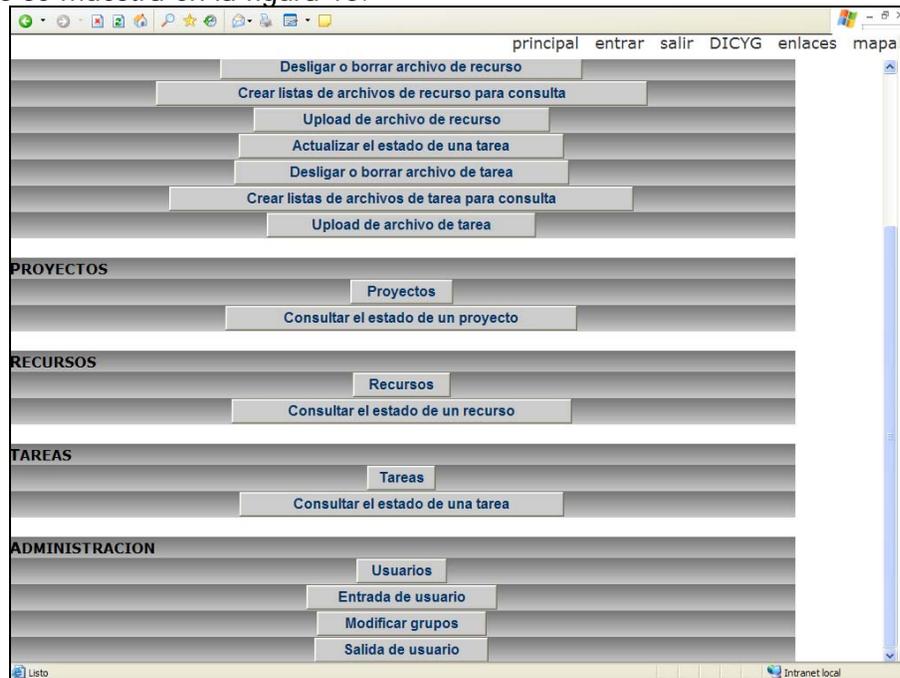


Fig. 15. Menú de funciones dependientes del usuario del sistema GePe.

En la interfaz de definición de proyectos, que se muestra en la figura 16, se definen las características generales de estos. En la misma pantalla se pueden realizar las modificaciones, bajas y altas. La información desplegada depende de las condiciones de ámbito del usuario, de manera que los administradores locales puedan manejar sus propios proyectos sin por ello tener la posibilidad de afectar los de otros usuarios administradores.

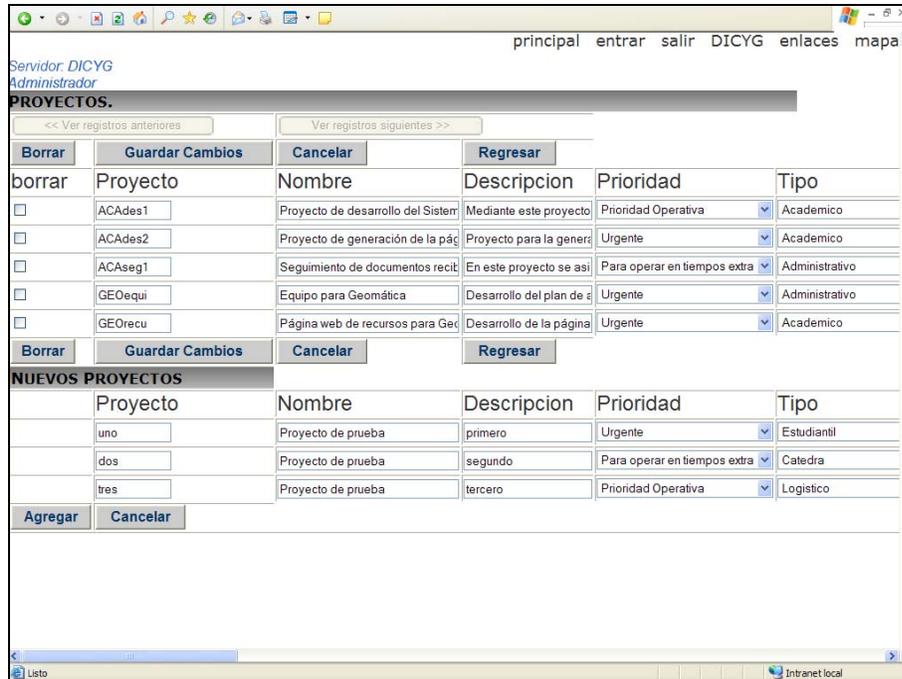


Fig. 16. Interfaz de definición, modificación y cancelación de proyectos.

Todos los elementos, a saber, usuarios, proyectos, tareas y recursos, se definen de manera analítica en el sistema, estableciendo sus características.

Una vez que se realizan las operaciones deseadas, el servidor responde con un acuse de recibo que confirma qué cambios fueron realizados exitosamente, como se muestra en la figura 17.

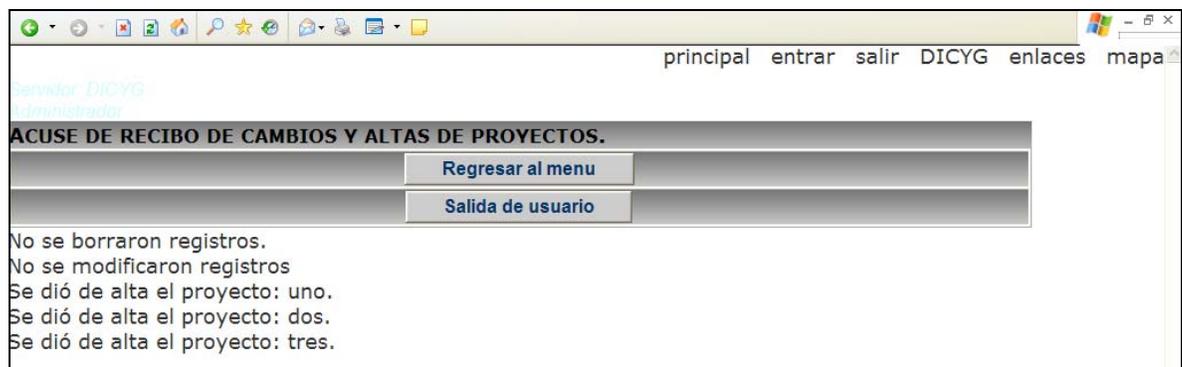


Fig. 17. Acuse de recibo de la baja de 3 proyectos.

Además, el sistema en todo momento despliega el nombre del usuario que está operando y el identificador del servidor al cual está conectado, en la parte superior izquierda de la pantalla.

Dentro del sistema cada proyecto está conformado de un grupo de tareas asociadas, que tiene su propia interfaz de definición. Las tareas deberán definirse después de que el proyecto haya sido creado. La interfaz se muestra en la figura 18.

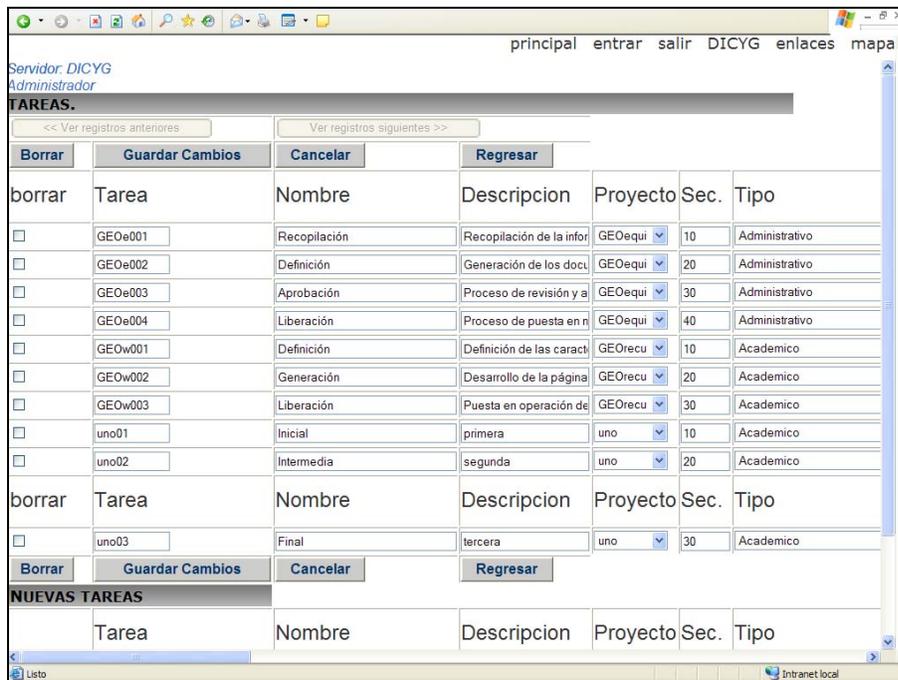


Fig. 18. Interfaz de definición, modificación y baja de tareas.

Las pantallas tienen características basadas en criterios ergonómicos tales como la repetición de los encabezados para listas largas, multiplicidad de botones arriba y abajo de la página para un acceso expedito, separación de funciones a través de elementos altamente visibles y conversión de claves de campos a sus nombres equivalentes definidos por catálogos.

También pueden definirse recursos para los proyectos, esto se hace a través de su propia interfaz, como se muestra en la figura 19.

La agregación de registros en cualquiera de las pantallas está limitada a 3 elementos cada vez. Esto obedece al criterio de evitar pérdidas de grandes cantidades de trabajo del usuario en caso de que las telecomunicaciones con el servidor fallen. Este criterio es cambiante a nivel de programación, estableciendo otro valor a un parámetro global a cada uno de los ejecutables.

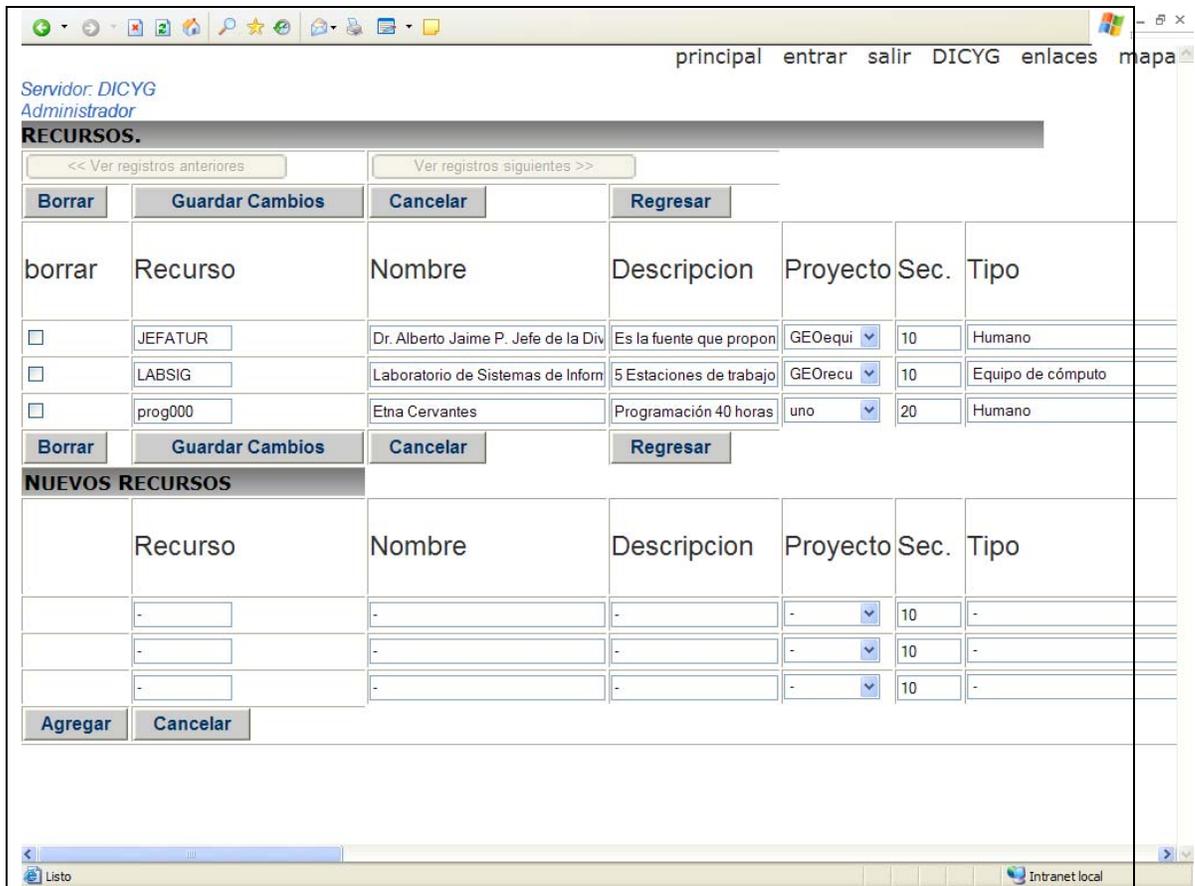


Fig. 19. Interfaz de definición, modificación y baja de recursos.

Cada uno de los elementos definibles en el sistema tiene una gran cantidad de parámetros a configurar. Se ayuda al usuario proponiendo valores por omisión en algunos de ellos que garantizan el funcionamiento estándar de los elementos.

Como herramienta para la modificación del funcionamiento estándar, se permite la configuración de grupos y tablas, por ejemplo cambiar los permisos de acceso a un proyecto.

Al generar un proyecto, se generan dos usuarios, el usuario administrador y el usuario de operación, sin embargo se pueden agregar y definir los usuarios que se necesiten a cada uno de ellos.

Se pueden modificar Grupos de recursos, tareas, usuarios y proyectos. Permisos de proyectos, recursos y tareas, para cada uno de los elementos correspondientes definidos en el sistema. Esta especificación de permisos por elemento da flexibilidad y control. La interfaz se muestra en la figura 20.

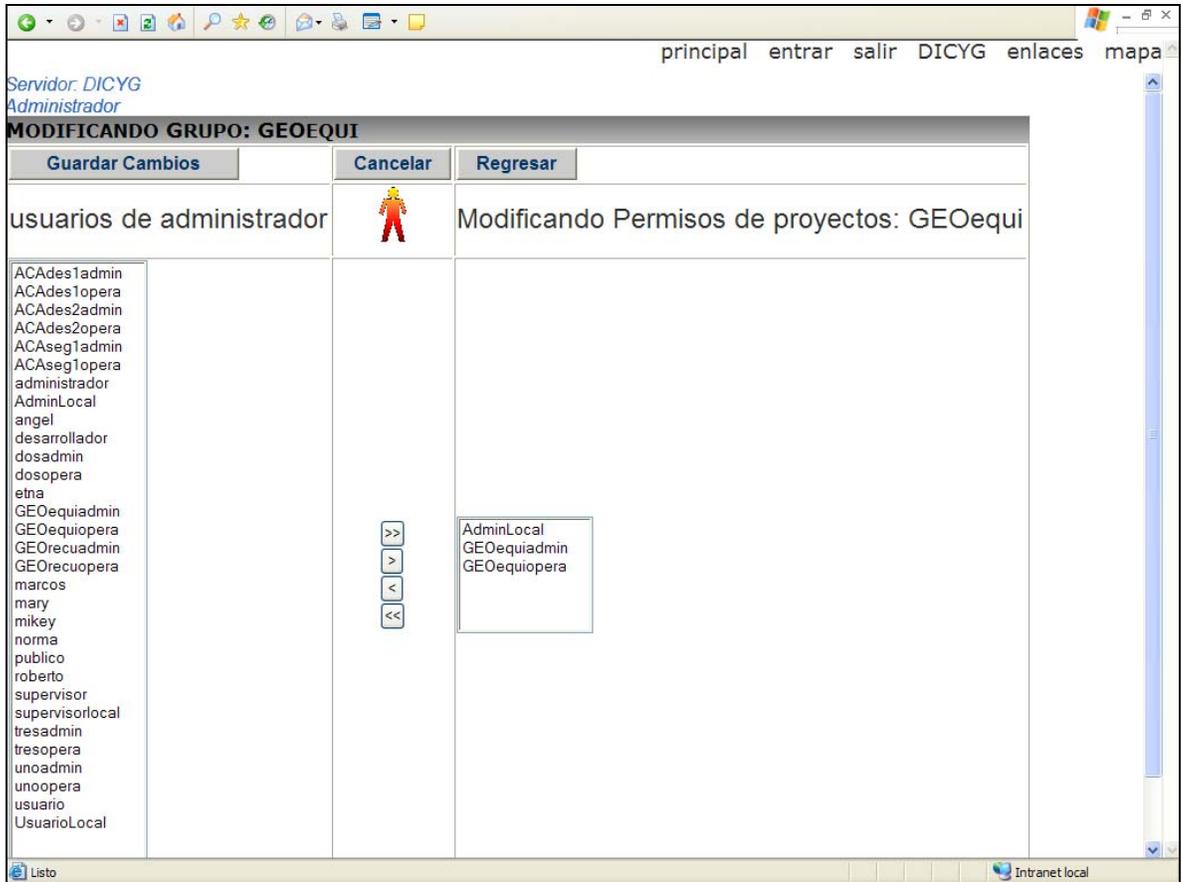


Fig. 20. Cambio de permisos de acceso a un proyecto.

Finalmente, cada usuario hace las actualizaciones y entregas de documentos correspondientes, para que se obtengan los diagramas de visualización o consultas del estado del proyecto.

La visión general de todos los proyectos es como muestra la figura 21.

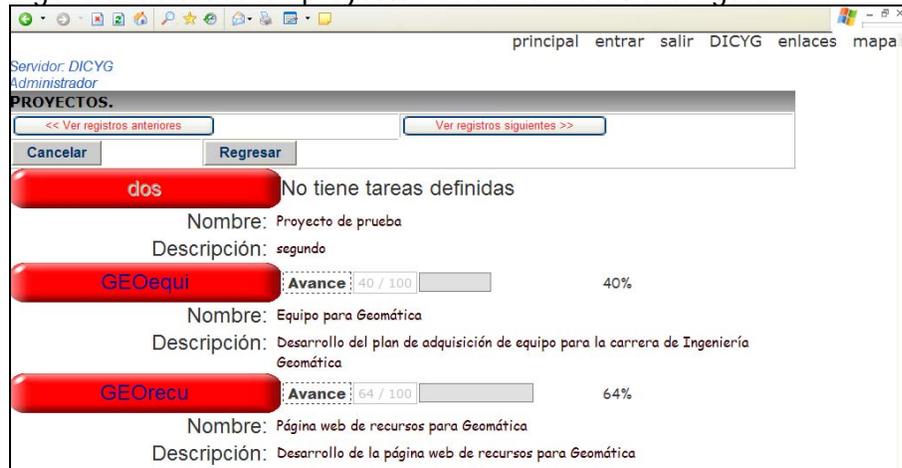


Fig. 21. Visualización general de todos los proyectos.

La barra de avance de los proyectos muestra los datos justo como se encuentran en ese momento registrados en la base. El cálculo del avance se hace al momento en que se solicita la página. Los criterios de cálculo del avance total del proyecto son definidos en cada una de las tareas que lo componen, de manera que se puede dar un peso uniforme a cada una o establecer que alguna de las tareas no contribuya al avance o establecer ponderaciones diferentes a cada una de ellas.

El detalle del proyecto se accede desde esa misma página y se muestra en la figura siguiente.

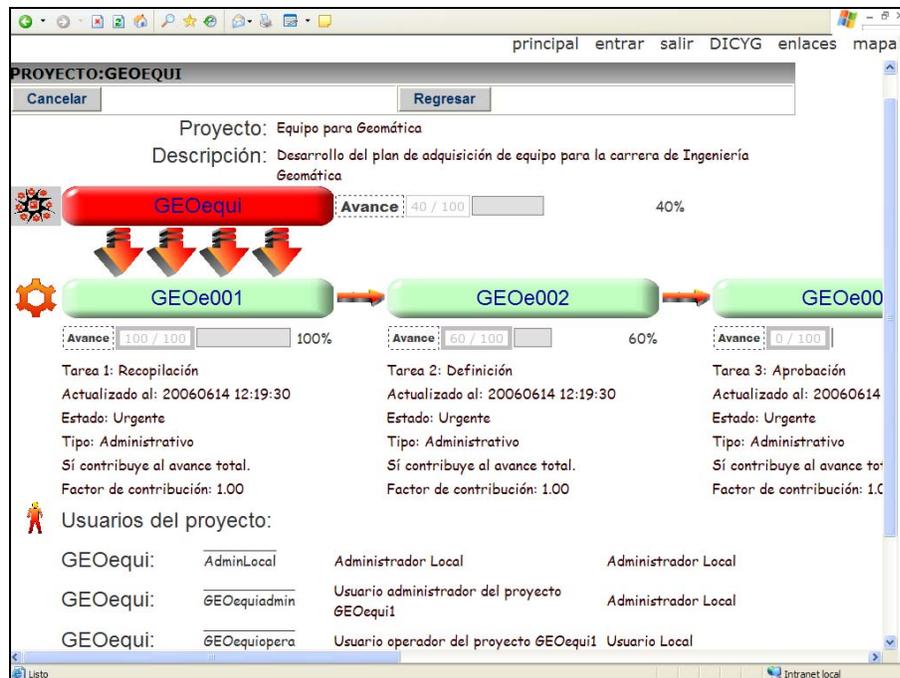


Fig. 22. Detalle de un proyecto en operación.

De esta manera se puede consultar el avance global y el avance por tareas. También se pueden desde ahí mismo consultar los detalles de cada tarea y de los demás elementos del proyecto, que son los usuarios con acceso a él y sus recursos.

5.2. Análisis Económico

Esta sección es una análisis económico del producto terminado. El costo del software desarrollado, si se hubiera hecho en el mercado abierto, se calcula de la siguiente manera:

Análisis del proyecto:	80 horas @ \$200.00/h	\$ 16,000.00
Diseño del sistema:	160 horas @ \$1,000.00/h	\$160,000.00
Desarrollo del sistema:	640 horas @ \$250.00/h	\$160,000.00

De esta manera el costo de este diseño y desarrollo sería de \$336,000.00, en 22 semanas.

Para comparar con las soluciones similares que pueden adquirirse en el mercado, se tiene que limitar la cantidad de usuarios modelados y con acceso al sistema, debido a los esquemas de licenciamiento que manejan las empresas que venden el software.

Supongamos un equipo de trabajo de 20 Integrantes.

MS-Project 2003 \$10,000.00 licencia completa por cada computadora / Integrante del proyecto que se quiera que maneje el sistema.

Costo por 20 usuarios: \$200,000.00

Unicenter Service Plus Service Desk \$30,000.00 licencia anual por usuario concurrente.

Costo por 20 usuarios por un año: \$600,000.00

Niku Clarity \$50,000.00 licencia anual por usuario concurrente.

Costo por 20 usuarios por un año: \$1,000,000.00

6. Conclusiones

Se desarrolló un sistema que permite el seguimiento de proyectos a través de una interfaz web y que puede trabajar con la infraestructura de la DICYG.

Tiene las características de control de usuarios, manejo de diversos niveles de proyectos con sus componentes tareas, recursos y usuarios. Se puede ver de manera global el estado de los proyectos y el detalle del avance y de los documentos relacionados entregados por los diferentes integrantes del equipo de trabajo del proyecto.

Es una herramienta que ayuda a comunicar el estado de los proyectos.

El siguiente paso dentro de la DICYG, es ponerlo en operación diaria y hacerle los ajustes que los usuarios rutinarios y el administrador del servidor recomienden.

Una vez que el sistema entre en operación, podría fortalecerse con otras maneras de visualización de los avances y los proyectos, por ejemplo hacerlo de manera comparativa con grupos y tipos seleccionables. Otra característica que se visualiza útil es la incorporación automatizada de proyectos pasados, tomando los datos históricos que se tengan disponibles. Esto serviría como una base de conocimiento donde se pueden buscar las soluciones del pasado.

Si se hubiera desarrollado el sistema en el mercado abierto su costo hubiera sido de \$336,000. Si se hubiera comprado MS-Project se hubieran invertido \$200,000, o Unicenter Service Plus Service Desk \$600,000 o Niku Clarity \$1,000,000. De donde se ve la ventaja económica para la DICYG de haber desarrollado un sistema de Gestión de Proyectos a través de un trabajo de tesis, porque además el sistema es adecuado a las necesidades e infraestructura de la DICYG y puede manejar todos los usuarios que la División necesite.

7. Bibliografía y Referencias

GESTIÓN DE PROYECTOS

- [MAP05] Ministerio de Administraciones Públicas, Gestión de Proyectos (España: Ministerio de Administraciones Públicas, 2005) 2,13,27.
- [BER05] Scott Berkun, The Art of Project Management, (EUA: O'Reilly, Media Inc., 2005)
- [APM00] Miles Dixon, Project Management. Body of Knowledge. (Reino Unido: Association for Project Management, 4a.ed, 2000), 14.

INGENIERÍA DE SOFTWARE

- [STE04] Mike Stephenson and John Kapman, Delivering Best Practices for Complex IT Environments. (EUA: CA, 2004) 2.
- [ITIL] <http://www.itil.co.uk/>
- [BS15000] <http://www.bs15000.org.uk/>
- [COL03] Colleen Graham and Kevin H. Strange, Gartner study shows worldwide RDBMS market declined in 2002. (EUA, 2003) 2
- [SHO87] Martin L. Shooman, Software Engineering. Design, Reliability and Management, (Singapore, McGraw-Hill, 5a.Ed., 1987),408-437.
- [CHA97] Dennise de Champeaux, Object Oriented development process and metrics, (USA, Prentic-Hall Inc., 1a.Ed., 1997).

LENGUAJES DE PROGRAMACIÓN

- [AXM2K] David Axmark and Michael Widenius, MySQL Reference Manual for version 3.23.37. (Finlandia
- [MyS]
- [MAS01] Mark Maslakowski, Aprendiendo MySQL en 21 días, (México: Pearson Education, 2001)
- Mercados de Software
- [MyS05] MySQL AB, Guide to Cost-effective Database Scale-Out using MySQL. For web an Data Warehouse Applications. (<http://www.mysql.com>, 2005) 5-7.

Glosario

arquitectura cliente-servidor	En Informática, estructura de acceso a servicios de cómputo en la cual, el beneficiario accede a los servicios desde un equipo diferente al equipo que los proporciona, usando una red de computadoras.
arquitectura de tres capas	Tipo de arquitectura cliente-servidor compuesto de tres niveles: nivel de interfaz de usuario, nivel de lógica de proceso funcional y nivel de datos, en la cual cada nivel es desarrollado y mantenido de manera independiente.
Apache	Servidor web de protocolo HTTP de código abierto.
base de conocimiento	En informática se refiere a una base de datos especializada en el conocimiento de un área específica, con herramientas de búsqueda inteligentes.
campo	En bases de datos, cada una de las características o tipos de datos que se guarda de un registro en una tabla.
codificación	En informática, programación.
comunicación asincrónica	Comunicación donde el emisor y el receptor no se encuentran en el canal simultáneamente. Típicamente realizada a través de paquetes de comunicación
DB2	Base de datos relacional de IBM.
desnormalización de una base de datos	Proceso de transformación de un modelo de una base de datos a otro equivalente donde se evita cumplir con alguna de las formas normales, con la finalidad de obtener un desempeño diferente de ella.
estándar	Que sirve como tipo, modelo, norma, patrón o referencia.
forma normal	Reglas del proceso de normalización de una base de datos, que tienen como objetivo dar un funcionamiento u optimización específico a una base de datos. Denotadas como 1FN, 2FN, etc.
Gestión de Proyectos	Disciplina que tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un proyecto.
hardware	En informática, Conjunto de los componentes que integran la parte material de una computadora.
hipertexto	En computación, paradigma de interfaz de usuario para el despliegue de documentos.
HTML	HyperText Marking Language. Lenguaje de presentación de documentos basado en marcadores. Uno de los tipos más utilizados para la
HTTP	Hypertext Transfer Protocol. Protocolo de transferencia de hipertexto.
informe electrónico	Documento en formato electrónico usado a través de las computadoras.
Informix	Base de datos relacional.
Ingres	Base de datos relacional.
interfaz	Elementos de interacción de un sistema con el usuario.
Internet	Red informática mundial, descentralizada, formada por la conexión directa entre computadoras u ordenadores mediante un protocolo especial de comunicación.
Intranet	Red informática local a una organización, formada por la conexión directa entre computadoras u ordenadores mediante un protocolo especial de comunicación.
java	lenguaje de programación multiplataforma
máquina de estados finita	Autómata finito. Modelo de comportamiento compuesto de estados, transiciones y acciones.
mejores prácticas	Idea administrativa que asegura que existe una técnica, método, proceso, actividad, incentivo o recompensa tal que es más efectiva en la entrega de salidas. La idea es que con los adecuados procesos, verificaciones y pruebas, un proyecto puede ser realizado con menores problemas.
metadatos	En computación, datos que describen otros datos. Si un sistema maneja datos para realizar su objetivo, los datos que describen cómo son los pertenecientes al primer conjunto.

método	Modo de decir o hacer con orden.
metodología	Conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.
Microsoft Project	Programa básico para el manejo de proyectos de marca Microsoft.
MS SQL	Base de datos de la marca Microsoft.
MySQL	Base de datos de código abierto.
norma	Regla que se debe seguir o a que se deben ajustar las conductas, tareas, actividades, etc.
normalización de una base de datos	Proceso de transformación de la estructura de una base de datos a un modelo equivalente que se hace utilizando las formas normales para obtener unas características de desempeño especificadas por éstas.
OLAP	On Line Analytical Processing. Tipo de sistema donde se hace procesamiento analítico en línea.
OLTP	Online Transaction Processing. Tipo de sistema donde se hace procesamiento de transacciones en línea
Oracle	Base de datos transaccional de marca Oracle.
programa	En Informática, conjunto unitario de instrucciones que permite a un ordenador realizar funciones diversas, como el tratamiento de textos, el diseño de gráficos, la resolución de problemas matemáticos, el manejo de bancos de datos, etc.
protocolo	En el ambiente de comunicación de computadoras, reglas de intercambio de mensajes entre ellas.
proyecto	Conjunto de escritos, cálculos y dibujos que se hacen para dar idea de cómo ha de ser y lo que ha de costar una obra de arquitectura o de ingeniería.
red de computadoras	Sistema de comunicación entre equipos de cómputo.
registro	En bases de datos, cada uno de los elementos de una tabla.
runtime	Tiempo de ejecución de un programa.
sistema	Ensamble de elementos relacionados que conforman una unidad.
sistema de archivos	Organización y control específico para leer y escribir información en un medio controlado por un sistema de cómputo
software	En Informática, conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.
SQL	Structured Query Language. Lenguaje basado en la construcción de preguntas a través del cual se consulta una base de datos.
tabla	En bases de datos, unidad de almacenamiento de registros.
usuario	En informática, entidad que utiliza un sistema de cómputo.
web	Red informática mundial. Tecnologías para ella.
XML	Extensible Markup Lenguaje. Lenguaje basado en marcadores similar al HTML, donde además pueden definirse tipos de datos, esquemas y hojas de estilos para pasar entre aplicaciones.

Anexos

ANEXO 1. LISTA DE ACRÓNIMOS

Acrónimo	Descripción
ANSI	American National Standards Institute. Organización estadounidense no lucrativa que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas para los Estados Unidos de Norteamérica. http://www.ansi.org
DICYG	División de Ingenierías Civil y Geomática, FI. UNAM. http://www.ingenieria.unam.mx/~dictyg/
DIE	División de Ingeniería Eléctrica, FI. UNAM. http://www.fi-b.unam.mx/
EIA	Electronic Industries Alliance. Asociación de comercio estadounidense para fabricantes de electrónicos. http://www.eia.org
FI	Facultad de Ingeniería, UNAM. http://www.ingenieria.unam.mx
GePe	Gestionador de Proyectos. Nombre del sistema realizado en esta tesis.
IBM	International Business Machine. Compañía que desarrolla y vende hardware, software e implementaciones de tecnologías de la información. http://www.ibm.com
IEEE	Institute of Electrical and Electronic Engineers. Organización Internacional Profesional no lucrativa para el fomento del desarrollo tecnológico relacionada a la electricidad. http://www.ieee.org
ISO	International Organization for Standardization. Cuerpo Internacional dedicado a establecer estándares industriales y comerciales, compuesto de cuerpos nacionales. http://www.iso.org
ITIL	Information Technology Infrastructure Library. Estructura de propuestas de mejores prácticas que pretenden facilitar la liberación de servicios de tecnologías de la información de alta calidad. http://www.itil.co.uk
HTTP	Hypertext Transfer Protocol. Protocolo de comunicación entre computadoras a través de una red de tipo solicitud – respuesta entre clientes y servidor. Es uno de los protocolos más utilizados en la red mundial de computadoras.
UML	Unified Modeling Language. Lenguaje de Modelado Unificado. Conjunto de técnicas para modelar software. http://www.uml.org
UNAM	Universidad Nacional Autónoma de México. http://www.unam.mx