

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

"Sistema para control de tareas o proyectos y medición de
rendimiento basado en resultados"

TESIS DE GRADO

Previo a la obtención del Título de:

INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN EN SISTEMAS DE INFORMACIÓN

Presentada por:

William George Murillo Párraga

GUAYAQUIL – ECUADOR

2008

AGRADECIMIENTO

A todas las personas que
estuvieron conmigo durante
mi carrera Universitaria.
Gracias por sus sabios
consejos y su apoyo
incondicional.

DEDICATORIA

A mis padres

A Dios

TRIBUNAL DE GRADUACIÓN

ING. HOLGER CEVALLOS
SUBDECANO DE LA FIEC
PRESIDENTE DEL
TRIBUNAL

ING. CARMEN VACA
DIRECTORA DE TESIS

ING. MONICA VILLAVICENCIO
VOCAL PRINCIPAL

ING. VERONICA MACÍAS
VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de ésta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL).

William George Murillo Párraga

RESUMEN

El propósito de este proyecto de tesis de grado es presentar una herramienta para administrar tareas dentro de un ambiente laboral, permitiendo una mejor comunicación entre líderes de grupo y colaboradores.

En el capítulo 1 se analiza los conceptos de control, delegación y comunicación, los cuales han sido la base que fundamenta la idea de este proyecto. Estos conceptos se complementan con el de medición, en el cual se analiza la importancia de usar métricas.

En el capítulo 2 se presentan diversas herramientas que se encuentran en el mercado, las cuales se tomó como referencia para compararlas con el proyecto y presentar sus beneficios. También se describen las características de la herramienta.

En el capítulo 3 se define todo lo relacionado al diseño de la aplicación. Se presenta la plataforma tecnológica sobre la cual se trabajó, los estándares usados y se presentan los módulos que tiene la herramienta.

En el capítulo 4 se explica de qué manera se implementó el proyecto, esto incluye la arquitectura usada y cómo interactúan cada uno de los módulos.

El capítulo 5 presenta el tipo de pruebas realizadas en el proyecto, el perfil de los usuarios seleccionados y como se configuró el ambiente de pruebas. Finalmente se muestra la lista de observaciones recogidas de las pruebas que fueron luego aplicadas.

ÍNDICE GENERAL

AGRADECIMIENTO.....	II
DEDICATORIA.....	III
TRIBUNAL DE GRADUACIÓN.....	IV
DECLARACIÓN EXPRESA.....	V
RESUMEN.....	VI
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS.....	XI
INTRODUCCIÓN.....	XIII
1.- ADMINISTRACIÓN Y CONTROL DE TAREAS.....	1
1.1.- El Control: Elemento Esencial en un Negocio.....	1
1.2.- Delegación + Comunicación = Control Adecuado.....	3
1.3.- Medición: Métricas de Cumplimiento.....	6
2.- PROPUESTA DE LA SOLUCIÓN.....	9
2.1.- Oferta Existente en el Mercado.....	10
2.2.- Objetivos de la Solución.....	12
2.3.- Características.....	13

2.4.- Comparación con las Ofertas del Mercado.....	16
2.5.- Beneficios.....	19
3.- ANÁLISIS Y DISEÑO.....	20
3.1.- Plataforma Tecnológica.....	21
3.2.- Estándares Web.....	24
3.2.1.- CSS.....	25
3.2.2.- XHTML.....	25
3.2.3.- AJAX.....	26
3.3.- Roles de Usuarios.....	26
3.4.- Módulo Hoja de Trabajo.....	27
3.5.- Módulo de Mantenimiento.....	31
3.5.1.- Administración de Usuarios.....	31
3.5.2.- Administración de Áreas.....	32
3.5.3.- Permisos por Rol de Usuario.....	32
3.5.4.- Usuarios Por Área.....	33
3.5.5.- Parámetros del Sistema.....	33
3.6.- Módulo de Métricas.....	34
4.- IMPLEMENTACIÓN.....	36

4.1.- Arquitectura.....	37
4.2.- Capa de Acceso a Datos.....	42
4.2.1.- Modelo de Base de Datos.....	42
4.2.2.- Modelo de Acceso a Datos.....	45
4.2.3.- Encriptación de Configuraciones.....	49
4.3.- Capa Lógica de Trabajo.....	50
4.4.- Capa de Presentación.....	52
4.5.- Módulo Hoja de Trabajo.....	56
4.6.- Módulo de Mantenimiento.....	58
4.7.- Módulo de Métricas.....	61
4.8.- Seguridad.....	62
5.- Pruebas del	
Sistema.....	64
5.1.- Pruebas de Funcionalidad.....	65
5.2.- Observaciones Obtenidas de las Pruebas.....	69
CONCLUSIONES Y RECOMENDACIONES.....	71
BIBLIOGRAFÍA.....	73
ANEXOS.....	76

ÍNDICE DE FIGURAS

	Pág.
Figura 3.1	Diagrama básico de interacción del usuario en la herramienta 28
Figura 4.1	Modelo 3 capas 37
Figura 4.2	Distribución de carpetas del proyecto 39
Figura 4.3	Programa para configurar Membership 40
Figura 4.4	Sección Membership en el Web.config 40
Figura 4.5	Control para validar usuarios 41
Figura 4.6	Diagrama de base de datos 43
Figura 4.7	Ejemplo de un adaptador de datos 46
Figura 4.8	Código común para recuperar datos de la base de datos 47
Figura 4.9	Código para recuperar datos de la base de datos en ASP.NET 48
Figura 4.10	Sección ConnectionString en el archivo Web.Config 48
Figura 4.11	Ejemplo de como instanciar un adaptador de datos 50
Figura 4.12	Instrucción para identificar métodos una clase como fuente de datos 51
Figura 4.13	Instrucción para identificar una clase como fuente de datos 53

Figura 4.14	Propiedades de un ObjectDataSource	52
Figura 4.15	Etiqueta que instancia un administrador de Script AJAX	53
Figura 4.16	Etiquetas que permiten ejecución asincrónica	54
Figura 4.17	Carpeta de temas del proyecto	55
Figura 4.18	Sección del Web.Config para definir tema	55
Figura 4.19	Hoja de trabajo con lista de tareas	58
Figura 4.20	Etiquetas que representan un visor de reportes	61
Figura 4.21	Métrica de cumplimiento	62

INTRODUCCIÓN

El mundo de los negocios, desde hace mucho tiempo atrás ha venido aumentando su dinamismo. En igual o mayor proporción la carga laboral aumenta para los empleados. Esto debido a nuevos modelos de negocio que se implementan, pero sobre todo al auge que ha tenido la tendencia de reducción de personal en la mayoría de las empresas. Cuando se reduce personal en una empresa, no se reducen a la par las actividades, estas son redistribuidas entre el personal que se mantiene, lo que aumenta las responsabilidades.

Esta y otras razones han generado que se empiece a buscar maneras de distribuir las tareas y de organizar el tiempo. El mercado brinda herramientas para ayudar a administrar actividades y tareas. Así tenemos, por ejemplo, los clientes de correo que incluyen agendas, calendarios, tareas, etc. Existen a su vez los teléfonos inteligentes (Smartphone por su denominación en inglés) que también ofrecen estas características.

A nivel de aplicativos específicos para la gestión de tareas o proyectos, el más conocido es Microsoft Project, una herramienta que es muy completa, pero esta virtud lleva a que las personas necesiten tener una capacitación previa de cómo usarlo. Para personas que no están muy familiarizadas con el uso de herramientas de oficina o sistemas de información, el manejo de este aplicativo puede llegar a ser engorroso.

Por otra parte, es posible encontrar más opciones en Internet, pero en su mayoría tienen un nivel de complejidad alto porque incluyen una gran cantidad de opciones, otras en cambio simplifican tanto el manejo de las tareas que no brindan información que puede ser valiosa para realizar análisis posteriores.

En este trabajo se presenta el análisis y la implementación de una herramienta para el control de tareas. Esta herramienta tiene una interfaz sencilla y recopila la mayor cantidad de información importante por cada tarea, con la finalidad de usar esta información para generar métricas que permitan medir el rendimiento basado en el cumplimiento.

CAPÍTULO 1

1. ADMINISTRACION Y CONTROL DE TAREAS

En toda organización existen siempre tareas por cumplir, desde algunas críticas para el desenvolvimiento del negocio hasta otras de menor importancia, pero que son parte de las obligaciones del empleado y cuentan al momento de evaluar el rendimiento enfocado en resultados obtenidos.

En este capítulo se analizan brevemente los problemas que se presentan a los empleados al momento de organizar sus tareas, como por ejemplo, los problemas de comunicación entre los encargados de las tareas y los jefes, y explicaremos las bases en que se fundamenta este proyecto: delegación, comunicación, control y medición.

1.1 El Control: Elemento Esencial en un Negocio

El control es una etapa primordial en la administración, pues, aunque una empresa cuente con magníficos planes, una estructura organizacional adecuada y una dirección eficiente, el ejecutivo no podrá verificar cuál es la situación real de la organización si no existe un mecanismo que ayude a que los altos mandos se cercioren e informen si el trabajo realizado está de acuerdo con los objetivos [18].

El concepto de control es muy general y puede ser utilizado en el contexto organizacional para evaluar el desempeño general frente a un plan estratégico [17].

El control se emplea para:

Mejorar la calidad: Las fallas del proceso se detectan y el proceso se corrige para eliminar errores.

Enfrentar el cambio: Este forma parte ineludible del ambiente de cualquier organización.

Producir ciclos más rápidos: Una cosa es reconocer la demanda de los consumidores para un diseño, calidad, o tiempo de entregas mejorados, y otra muy distinta es acelerar los ciclos que implican el desarrollo y la entrega de esos productos y servicios a los clientes. El cliente en un ambiente laboral es el mismo negocio.

Agregar valor: Los tiempos veloces de los ciclos son una manera de obtener ventajas competitivas.

Facilitar la delegación y el trabajo en equipo: La tendencia contemporánea hacia la administración participativa también aumenta la necesidad de delegar autoridad y de fomentar que los empleados trabajen juntos en equipo. Esto no disminuye la responsabilidad última de la gerencia. Por el contrario, cambia la índole del proceso de control. Por tanto, el proceso de control permite que el gerente controle el avance de los empleados, sin entorpecer su creatividad o participación en el trabajo.

1.2 Delegación + Comunicación = Control Adecuado

“No trate de mantener ocupado a su personal puesto que puede terminar siendo experto en mantenerse ocupado usted mismo ... lo que le puede impedir alcanzar Resultados” Eric Gaynor Butterfield, Ph. D. (abd) – Presidente de “The Organization Development Institute International”, Latinoamérica (Board member de The Organization Development Institute-Worldwide) [19].

Esta frase inicial se enfoca en comprender lo difícil que puede llegar a ser el controlar las tareas que un jefe delega a su personal y a su vez, cuán complicado se puede volver el trabajo cuando se empieza a delegar responsabilidades desmesuradamente, llevando a perder el control y perder mucho tiempo en una reorganización y recapitulación en muchos casos de lo realizado.

En lenguaje de negocios, cuando uno desperdicia el tiempo está desperdiciando el dinero. La delegación es importante y se vuelve imprescindible cuando no tenemos tiempo y en algunos casos capacidad para resolver todas las tareas encomendadas, pero antes de delegar tenemos que pensar si la tarea es delegable, si la respuesta es sí, hay que ponerse a pensar cómo delegarla, y hay pasos a seguir [20]:

- Encontrar la persona adecuada
- Explicar claramente los objetivos
- Dar la autoridad necesaria y los medios para realizar la tarea
- Mantenerse en contacto brindando apoyo, sin ahogar
- Demostrar confianza

Queda claro que a pesar de delegar una tarea, la responsabilidad del resultado final, sigue siendo nuestra como líderes de un grupo. Esto nos mantiene pendientes siempre del avance de la tarea.

En lo que respecta a la comunicación, estudios sobre problemas en el trabajo [21] demuestran que cuatro de cada cinco problemas son causados por una mala o deficiente comunicación. La experiencia nos demuestra que muchas personas, incluso excelentes profesionales 'técnicos', personal con una enorme valía en su 'oficio' son pésimos comunicadores. En muchos casos no son conscientes de la importancia de la comunicación en su trabajo. Les interesa el tornillo, el proyecto, el producto, el dato, pero no demasiado las

personas que deben apretar el tornillo, realizar el proyecto, acabar el producto, anotar el dato.

Obviar la importancia de la comunicación puede generar muchos problemas, por tal motivo es importante que se mantenga una comunicación constante, y que, tanto el líder como el colaborador estén debidamente informados y puedan hablar un mismo idioma.

Delegación y comunicación son virtudes esenciales en un líder, toda persona que tenga a su cargo la dirección de un grupo de personas a través de una jefatura y más aún a nivel de gerencia debe tener cualidades de líder. Según estudios realizados en la Universidad Estatal de Ohio [16], el liderazgo se relaciona con dos dimensiones: el estilo autoritario y el estilo participativo.

Eric Gaynor Butterfield hace mención a dos intervenciones de consultoría donde encuentra sustento al hecho de que algunas personas tienen “en sus cabezas” una forma de liderar diferencial, y que la misma está relacionada con la visión que tienen respecto de cómo opera una empresa [16].

En estos trabajos se encontró que aquellos líderes que concebían que la empresa funcionaba principalmente en base a “su estructura organizativa”, lo cual implica privilegiar el organigrama y los manuales de descripción y funciones y manual de procedimientos, tendían a adoptar y hacer uso de un estilo de relacionamiento con el personal de tipo autoritario.

Mientras que aquellos directivos y ejecutivos en la empresa que tenían en sus cabezas que pasan muchas “otras cosas más” en la empresa y que no

se relacionan con los aspectos formales y estructurales de la organización, se orientaban a adoptar un estilo de relacionamiento con el personal de mayor consideración hacia ellos, más participativo.

El estilo de liderazgo que nos podemos encontrar va a depender de varios factores: uno es la formación y experiencia que ha tenido una persona a lo largo de su trayectoria, otro factor es el ambiente laboral y el tipo de relación interpersonal que prime en el lugar donde nos desempeñemos, y otros factores externos.

Este proyecto de tesis intenta ser útil en un ambiente laboral independiente del estilo de liderazgo que se maneje en el mismo, brindando a los líderes la capacidad de delegar y a un grupo de trabajo la capacidad de estar comunicados del avance de las tareas.

1.3 Medición: Métricas de Cumplimiento

Lo que se puede medir, se puede manejar y, por tanto, se puede mejorar.

La necesidad de la medición es algo evidente. Después de todo es lo que nos permite cuantificar y por consiguiente gestionar de forma más efectiva. Pero la realidad puede ser muy diferente. Frecuentemente la medición conlleva una gran controversia y discusión [14]:

1. ¿Cuáles son las métricas apropiadas para el proceso y para el producto?
2. ¿Cómo se deben utilizar los datos que se recopilan?
3. ¿Es bueno usar medidas para comparar gente, procesos o productos?

Estas preguntas y otras tantas docenas de ellas siempre surgen cuando se intenta medir algo que no se ha medido en el pasado.

La medición es muy común en el mundo de la ingeniería. Medimos potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos.

Desgraciadamente la medición se aleja de lo común en el mundo de la ingeniería del software.

Encontramos dificultades en ponernos de acuerdo sobre qué medir y cómo evaluar las medidas.

Hay varias razones para medir un producto.

1. Para indicar la calidad del producto.
2. Para evaluar la productividad de la gente que desarrolla el producto.
3. Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
4. Para establecer una línea de base para la estimación
5. Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Las mediciones del mundo físico pueden englobarse en dos categorías: medidas directas y medidas indirectas.

Medidas Directas. En el proceso de ingeniería se encuentran el costo, y el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el

tamaño de memoria y los defectos observados en un determinado periodo de tiempo.

Medidas Indirectas. Se encuentra la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc.

Controlar y medir son actividades difíciles de realizar manualmente. Si a esto le sumamos que la memoria humana es frágil para recordar gran cantidad de detalles, y que las tareas por lo general tienen algunos detalles inmersos, se hace imprescindible contar con una herramienta que nos ayude cumplir con estas actividades. Una herramienta apropiada nos permitirá conseguir los objetivos de delegación, comunicación, control y medición mencionados en este capítulo.

En el siguiente capítulo se analizarán algunas herramientas que se pueden encontrar en el mercado y se planteará además la propuesta de esta tesis.

CAPÍTULO 2

2. PROPUESTA DE LA SOLUCIÓN

En el mercado existen herramientas comerciales para el seguimiento de proyectos y o tareas, la más conocida es Microsoft Project. A pesar de ser una herramienta comercial no es de fácil uso para el usuario común y se puede agregar a esto que las empresas no la adquieren en gran escala porque deben pagar una licencia por usuario. En este capítulo se analizan las características que podemos encontrar en varias aplicaciones comerciales y se presenta una comparación entre estas aplicaciones y la herramienta propuesta para este proyecto de tesis mostrando las diferencias y beneficios que presenta respecto a los productos ya existentes.

2.1 Oferta Existente en el Mercado

En el mercado podemos encontrar variedad de productos que nos permiten trabajar administrando nuestras tareas y proyectos. Para el análisis de este proyecto hemos escogido 4 productos, primero el más conocido y usado, MS Project 2007; segundo TaskBin; el tercero es KMKey y por último B-Kin Project Monitor.

Cada uno de estos sistemas ha sido seleccionado tras analizar las diferentes opciones encontradas en el mercado, con el fin de tener opciones que acrediten una buena comparación con el proyecto que se sustenta.

- Microsoft Office Project Standard 2007 [10] ofrece herramientas sólidas de administración de proyectos, con el fin de realizar esta tarea con mayor eficacia y eficiencia. Permite mantenerse informado y controlar el trabajo, la programación y las finanzas del proyecto. Se integra con los conocidos programas del sistema Microsoft Office y tiene la opción de elaborar de informes.
- En TaskBin [12], el usuario es siempre parte de un grupo o varios grupos. Estos grupos definen el contexto de las tareas. Una vez que el usuario pasa a formar parte de un grupo, los demás miembros del grupo pueden agregar tareas a su lista personal o editarla. Así que en cierto sentido, todas las tareas son "compartidas". TaskBin no incluye un calendario o planificador, que es muy común en otras herramientas, por ende no hay necesidad de especificar fechas

concretas. Las características principales de TaskBin, mencionadas en el sitio Web de la aplicación [12] son:

- A las tareas se le pueden asignar las siguientes prioridades: Ahora, Hoy, Mañana, En esta semana y Pronto.
 - Todas las tareas emiten notificaciones por correo electrónico.
 - La aplicación es ideal para trabajar con equipos de localización cruzada y tiene un gran soporte para múltiples zonas horarias.
 - El usuario puede ver lo que está sucediendo en su grupo, lo que se ha añadido o actualizado recientemente, lo que se debe hacer hoy, etc.
- KMKey Project [11] es un software de gestión de proyectos donde cualquier empresa de servicios puede disponer de toda la información necesaria para desarrollar su negocio, está especialmente indicado para llevar el control de proyectos de cualquier tipo: desarrollo de proyectos. Sus características principales son: la planificación del proyecto, el manejo de flujos de trabajo, la visualización de un calendario, el control de periodos de ejecución, la comparación de fechas reales frente a fechas previstas, y el análisis de tareas fuera de plazo. También se puede integrar con Microsoft Project para generar el flujo de trabajo.
- B-Kin Project Monitor [13], planifica tiempos, costes, recursos humanos, esfuerzos, trabajos, usando herramientas como diagramas

de Gantt, planificadores rápidos de tareas y recursos, tareas predecesoras, etc.

Permite compartir la información en forma de documentos, foros, informes, listados. Cuenta con más de cincuenta informes ya configurados sobre esfuerzos y costes. Permite configurar y compartir informes personalizados. Está preparado para diferentes tipos de usuario como responsables de programas, jefes de proyecto o de área, responsables de tareas intermedias o personas que realizan los trabajos.

Una vez que se han presentado algunas alternativas existentes en el mercado, procedemos a presentar la propuesta de este proyecto de tesis, empezando por los objetivos que se persiguen, luego explicando las principales características y finalmente comparándola con las herramientas antes mencionadas.

2.2 Objetivos de la Solución

Crear una herramienta Web que a través de un manejo simple permita mantener un control detallado, automatizado y progresivo de tareas o proyectos aprovechando toda esta información para generar a su vez métricas de rendimiento basadas en el cumplimiento. Es importante recalcar que la herramienta no se enfoca en la administración de proyectos, sino en la administración de tareas diarias.

Los objetivos específicos se pueden listar de la siguiente manera:

- Mantener un registro electrónico del avance histórico de las tareas.
- Conocer en tiempo real cómo va el avance de las tareas.
- Generar notificaciones de correo electrónico a las personas involucradas en las tareas.
- Configurar los parámetros para las condiciones de funcionamiento y las reglas de control.
- Medir el rendimiento basado en métricas.
- Uso más sencillo que herramientas comerciales tradicionales.
- Acceso a la herramienta a través del Web sin necesidad de instalar una aplicación en cada máquina

2.3 Características

El proyecto propuesto es una herramienta con características simplificadas en relación a herramientas comerciales para la gestión de proyectos, enfocado en brindar opciones que son realmente necesarias para el usuario.

A continuación se detallan algunas de las principales características:

- *Acceso vía Web.*- La aplicación podrá ser accedida a través de un navegador Web. Se necesitará de un servidor dentro de la empresa para alojar la aplicación y que funcione dentro de la intranet. Esto facilita que no sea necesario instalar una aplicación en cada máquina de cada usuario y que incluso puedan revisar la información o

actualizar los datos desde cualquier computador con acceso a la intranet. Si la empresa lo requiere, incluso se podrá acceder a través de internet o una VPN de la propia empresa.

- *Autenticación con LDAP o ActiveDirectory* .- La forma más común que las empresas manejan para autenticar sus usuarios es a través de servidores LDAP o de ActiveDirectory. Esta aplicación se integrará fácilmente a cualquiera de estas dos opciones de autenticación.
- *Control automático del estado de las tareas*.- La aplicación manejará 4 estados para las tareas: Nuevo para proyectos nuevos, Abierto para proyectos en ejecución y dentro de su tiempo estimado, Retrasado para proyectos en ejecución pero que han pasado de su tiempo estimado inicial y Cerrado para proyectos ya culminados. La asignación del estado a cada proyecto la realizará la aplicación en base a las reglas de funcionamiento.
- *Agente de recordatorios*.- La aplicación complementará su funcionamiento con un agente que todos los días revisará cada una de las tareas para notificar a el responsable por: Puntos nuevos que han pasado más de 7 días sin tener una fecha de apertura, Puntos que no han recibido comentarios en más de una semana, Puntos a los cuales se les acerca la fecha de cierre y otro tipo de alertas según se desee configurar.

- *Gráficos estadísticos.*- Dentro de la aplicación se podrán revisar las estadísticas sobre el estado de cada tarea y el rendimiento de cada área o persona que es evaluada. Estos resultados estadísticos podrán ser llevados a Excel o PDF.
- *Registro de auditoría por cada tarea.*- Cada una de las tareas contará con un registro de auditoría que va a permitir conocer las fechas en que cada tarea cambió de estado, cuando se ha modificado la fecha estimada de culminación de un proyecto, cambio de prioridad, etc.
- *Documentos anexos.*- A cada tarea se podrán anexar documentos para compartirlos con el resto de los usuarios de la aplicación.
- *Manejo de fases.*- Existen tareas que no son tan sencillas para ser manejadas como un simple elemento de nuestra lista de tareas. Proponemos un manejo de fases, pero que cada fase se maneje como una tarea independiente, manteniendo una relación entre las tareas a través de una etiqueta. Esto permite enfocarse en cada una de las tareas, sin tener toda la información que compete al proyecto que muchas veces no es usada, y se la puede visualizar sólo de ser requerido por el usuario.
- *Simplicidad de manejo.*- Hacer las cosas más sencillas para el usuario pasa de ser un objetivo a ser una obligación para cada sistema. Esta aplicación intenta dar los recursos necesarios y útiles a los usuarios y no llenarlos de opciones que generalmente no se usan.

- *Costos menores.*- Para una empresa resulta más económico adquirir una aplicación de estas características que otro tipo de herramientas. Con un solo precio pueden acceder todos los usuarios que la empresa crea necesarios sin tener que comprar una licencia para cada uno

2.4 Comparación con las Ofertas del Mercado

De todos los sistemas que se han puesto en consideración, el único que no tiene una opción de ver los proyectos en Web es MS Project. Claro está que no es imposible, porque existe el complemento llamado MS Project Server, que brinda esta opción, pero esto hace que las empresas incurran en más gastos. Dadas estas condiciones, MS Project funciona más como un aplicativo personal que depende de un archivo. Esto impide que exista un debido control de las tareas por parte de los jefes, ya que deben pedir a sus colaboradores una copia del archivo original, incurriendo en problemas comunes como tener versiones desactualizadas, pérdidas de archivos, etc. Los demás sistemas y el proyecto propuesto funcionan como aplicaciones Web, teniendo un repositorio central de todas las tareas.

TaskBin maneja un concepto más sencillo de tareas, tomando como base el concepto que presenta Microsoft Outlook, que es muy parecido al que se presenta también en los PDAs. Lo que se hace es agregar descripciones sencillas que implican una tarea, se le asigna una categoría y una fecha de cierre estimada. La principal desventaja es que no permite medir el

cumplimiento de las tareas. Se puede decir que es una opción electrónica de manejar los apuntes que usualmente llevamos en papel. El grupo creador de TaskBin indica que ellos no creen en reportes de estado de los proyectos, por lo que su filosofía es que todo sea transparente. Lastimosamente esto no es muy útil a niveles ejecutivos en empresas, donde es imprescindible tener métricas que ayuden a evaluar resultados.

KMKey y B-Kin, son aplicaciones Web que muestran una interfaz muy familiar a la ofrecida por MS Project. Su objetivo está más enfocado a la gestión general de un proyecto, con fases, sub-fases y tareas. Un usuario puede tener múltiples proyectos creados. Tal como en Project puedo manejar los tiempos, los recursos, etc. Es prácticamente una versión Web de MS Project, por ende su manejo llega a tener un nivel de complejidad un poco elevado y abarca muchos temas que a través de nuestro proyecto buscamos evitar.

A continuación mostramos un cuadro en el cual se compara por diferentes criterios, este proyecto de tesis y los demás aplicativos que se han venido analizando.

	Microsoft Project	TaskBin	KMKey	B-Kin	Proyecto de Tesis
Interfaz Web		x	x	x	x
Genera Informes/Métricas	x		x	x	x
Registro de Auditoría de las Tareas					x
Archivos Adjuntos			x	x	x
Autenticación LDAP /AD					x
Agente de Recordatorios					x
Notificaciones por E-mail		x	x	x	x
Gestión de Tiempo / Finanzas	x		x	x	
Roles de Usuario			x	x	x
Ambiente Colaborativo		X	x	x	x
Gestión de Reuniones					x
Prioridades Por Tarea		X	x	x	x
Diagramas(Gantt,etc)	x		x	x	

Como se explicó en la parte introductoria, las empresas tienden a reducir personal, elevando a la par la carga laboral de quienes se mantienen en sus puestos de trabajo. Al aumentar esta carga laboral, las personas tienden a querer hacer las cosas lo más sencillas posibles sin perder el control. Lo que se propone con este proyecto es una herramienta que ayude a las personas a que manejen sus actividades evitando el tener que llenar tanta información en un sistema, o realizando una planificación tan estricta. Este tipo de actividades son más apropiadas para Administradores de Proyectos o

Planificadores, quienes se han preparado para hacer un análisis minucioso de los tiempos, tareas, etc., que forman parte de cualquier proyecto.

De todos los sistemas que se han analizado del mercado, es clara la tendencia de abarcar muchas facetas de un proyecto, ofreciendo a su vez gran cantidad de opciones en el sistema, que a la larga no son utilizadas. Sólo TaskBin mantiene una relación con el concepto que proponemos, pero no permite tener un control adecuado.

2.5 Beneficios

Los beneficios que vamos a obtener de este proyecto son los siguientes:

- Creación de un repositorio común de tareas a cumplir y cumplidas.
- Envío de notificaciones automáticas vía correo electrónico.
- Acceso a través de Internet o la intranet.
- Registro del avance de cada tarea a lo largo del tiempo que se tomó para resolverla.
- Empleados pueden revisar información sobre cómo es medido el nivel de rendimiento.
- Fomentar que las personas estén al tanto de sus tareas asignadas y pendientes del avance de las mismas.
- Ahorro económico en relación a otras aplicaciones comerciales

CAPÍTULO 3

3. ANÁLISIS Y DISEÑO

Generalmente las ideas son difíciles de definir, pero una vez que uno define bien una idea se encuentra con otro muro: ¿Cómo elaboro un buen diseño que plasme correctamente la idea?

En este capítulo se analizará la plataforma tecnológica que será la base para el desarrollo y funcionamiento de la aplicación, y los estándares que se emplearán al momento de elaborar la interfaz de usuarios.

Posteriormente en este mismo capítulo se presentarán los roles de usuario que van a interactuar en la aplicación y los diferentes módulos que van a conformar el proyecto en su totalidad.

3.1 Plataforma Tecnológica

Framework .NET 2.0 [4] es el componente de Microsoft Windows que permite crear y ejecutar la próxima generación de aplicaciones de software y servicios Web. Estos componentes facilitan la integración compartiendo datos e información a través de la red mediante protocolos y estándares independientes de la plataforma, como por ejemplo, XML, SOAP y HTTP.

Framework .NET ofrece lo siguiente:

- Un entorno enormemente productivo y basado en estándares para integrar el código existente con aplicaciones y servicios de próxima generación
- La flexibilidad para afrontar los retos que suponen la implementación y la operación de aplicaciones en toda la empresa.

Framework .NET se compone en dos elementos principales: Common Language Runtime (CLR) y un conjunto unificado de bibliotecas de clases, tales como ASP.NET para aplicaciones Web y Servicios Web, Windows Forms para clientes inteligentes y ADO.NET para el acceso a datos sin rigidez.

Usando el Framework .NET obtenemos las siguientes ventajas:

- Arquitectura flexible.- Está diseñado para usar servicios Web XML como mecanismo de comunicación nativo. Sus avanzadas funciones

permiten convertir aplicaciones actuales en servicios Web sin necesidad de re-escribir o re-compile código.

- Desarrollo rápido.- Es posible integrar varias aplicaciones en distintos lenguajes, sin necesidad de invertir tiempo en formación extra. Visual Studio es uno de los entornos de desarrollos más avanzados del mercado. Utiliza un diseño totalmente basado en componentes y sin estructura que permite a los programadores centrarse en escribir la lógica empresarial.
- Operaciones más eficientes.- ASP.NET 2.0 incorpora funciones avanzadas de compilación y almacenamiento en caché que pueden aumentar notablemente el rendimiento de aplicaciones Web. Gracias a la tecnología de metadatos que se utiliza, la instalación de aplicaciones es tan fácil como copiarlas en un directorio. Administra la memoria, los procesos y los subprocesos con el objeto de evitar las pérdidas de memoria, obteniendo de esta forma aplicaciones más confiables.

ASP.NET [2] es una plataforma para desarrollo Web, una de las más poderosas del mercado, que provee todas las herramientas y servicios necesarios para construir aplicaciones Web empresariales, comerciales o de tipo personal. ASP.NET se nutre y beneficia de todas las características que ofrece el Framework .NET.

Existe una gran cantidad de ventajas que se obtienen de usar ASP.NET para los desarrollos, pero las que justifican el uso de esta tecnología para el proyecto actual son [3]:

- Compatibilidad con herramientas de primer nivel: El marco de trabajo de ASP.NET se complementa con un diseñador y una caja de herramientas muy completos en el entorno integrado de programación (IDE) de Visual Studio. La edición WYSIWYG, los controles de servidor de arrastrar y colocar y la implementación automática son sólo algunas de las características que proporciona esta eficaz herramienta
- Eficacia y flexibilidad: Debido a que ASP.NET se basa en Common Language Runtime, la eficacia y la flexibilidad de toda esa plataforma se encuentra disponible para los programadores de aplicaciones Web. La biblioteca de clases de Framework .NET, la Mensajería y las soluciones de Acceso a datos se encuentran accesibles desde el Web de manera uniforme
- Simplicidad: ASP.NET facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios.

- Seguridad.- Junto con ASP.NET viene incluido un módulo de seguridad que facilita y ahorra gran cantidad de tiempo y esfuerzo en temas relacionados al manejo de la seguridad y validación de usuarios
- Menos líneas de código: Con la tareas comunes que se encapsulan en controles y que pueden ser usados a través de todo el sitio Web, muchas de las cosas que se hacen tradicionalmente ahora se realizan con mucho menos código que en versiones anteriores de ASP u otras herramientas.
- Rendimiento y escalabilidad.- ASP.NET se ha creado para desempeñarse usando un modelo de ejecución compilado para manejar solicitudes de las páginas. Es compatible con procesadores y servidores de 64-bits. Incluye además mejoras en el manejo de la caché.
- ASP.NET AJAX: Junto con ASP.NET 2.0, el uso de AJAX [1] para la creación de sitios Web interactivos es más sencillo. Existe la posibilidad de con solo agregar un control, seguir usando la programación del lado del servidor, o a su vez usar programación del lado del cliente, e incluso combinar ambas.

3.2 Estándares Web

Este proyecto se desarrollará aplicando los últimos conceptos y estándares para aplicaciones y sitios Web, tomando como referencia la denominada

Web 2.0 [6] la cual se caracteriza por el uso de estándares en el diseño del sitio Web o aplicación Web.

Los estándares Web a ser aplicados en este proyecto son:

- CSS
- XHTML
- AJAX

3.2.1 CSS

Las hojas de estilo en cascada (CSS) [8] son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

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

3.2.2 XHTML

XHTML [7], acrónimo inglés de Extensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas Web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una Web semántica, donde la información, y la forma de presentarla estén

claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo (como las hojas de estilo en cascada) y JavaScript su aspecto y diseño en distintos medios

3.2.3 AJAX

AJAX [9], acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo Web para crear aplicaciones ricas o RIA (Rich Internet Applications). Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma

3.3 Roles y Usuarios

El control de acceso de la aplicación se lo hará a través de roles, lo que permite distribuir de manera más práctica las funciones que van a tener los usuarios y facilitará el trabajo de los administradores de la aplicación.

Para el efecto se han determinado los siguientes roles:

- Administrador: Rol que por defecto tiene habilitados todos los accesos a la aplicación.
- Líder: Rol para representar al líder de un área.

- Líder alternativo: Rol para representar a quien suplanta a un líder cuando este no está disponible.
- Consultor: Usuarios en general. Todos los usuarios tienen por defecto asignado este rol.

3.4 Módulo “Hoja de Trabajo”

La hoja de trabajo es la principal herramienta de trabajo para los usuarios. Aquí se crean, editan y eliminan las tareas. La Figura 3.1 muestra un esquema general del funcionamiento de la herramienta y como interactúa un usuario con sus tareas y áreas a las que pertenece.

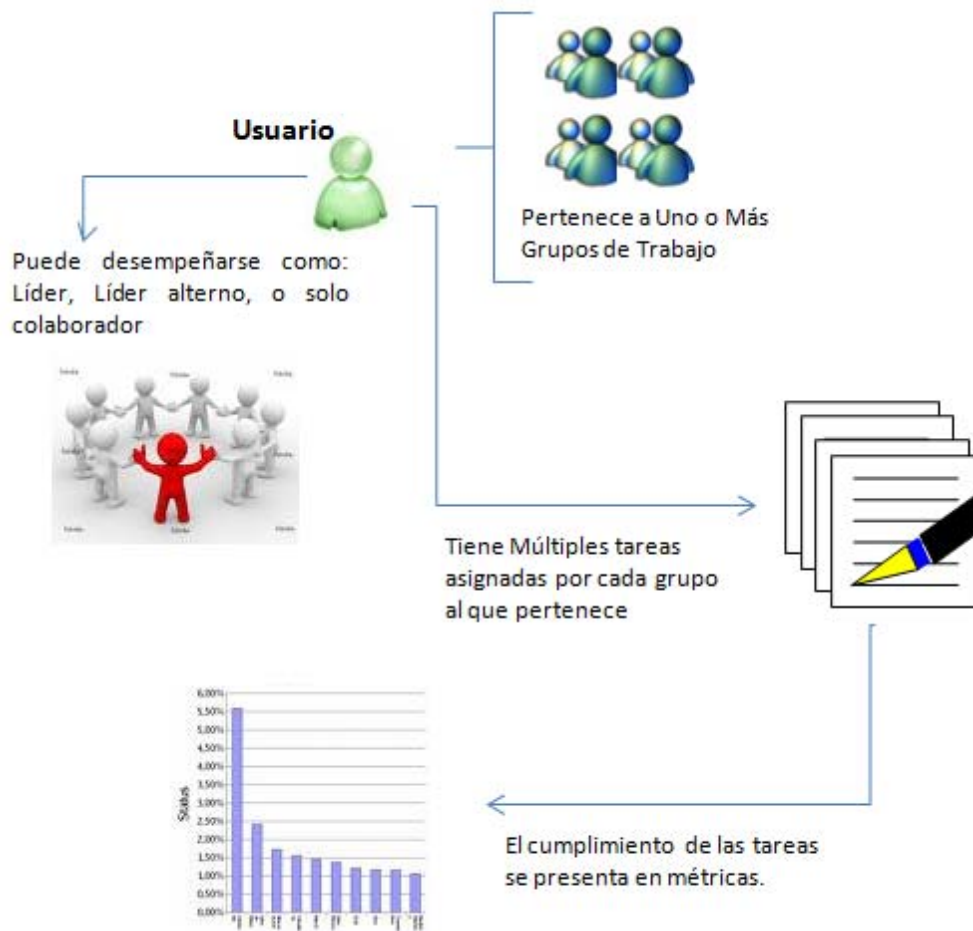


Figura 3.1 Diagrama básico de interacción del usuario en la herramienta.

Las tareas aparecen listadas en una tabla. Esta tabla consta de las siguientes columnas:

- Código: Número que identifica a la tarea. Es generado automáticamente por la aplicación.
- Descripción: El detalle de la tarea.
- Prioridad: Prioridad asignada a la tarea. Las prioridades posibles son: 1, 2 y 3.

- Fecha de Inicio: Fecha de creación de la tarea. Generada automáticamente por el sistema.
- Fecha de cierre: En esta columna se listan todas las fechas de cierre que se han ido proponiendo para la tarea.
- Estado: Estado en el que se encuentra la tarea. Los estados posibles son: Nuevo, Abierto, Retrasado, Cerrado.
- Comentarios: Los comentarios que se han ingresado sobre la tarea. Cada vez que se agrega una nueva fecha de cierre a la tarea se solicita al usuario que ingrese un comentario, para dejar una justificación por la cual se retrasa la misma.
- Menú: Columna con menú de opciones posibles sobre una tarea. Las opciones son: Ver Historial de Tarea, Ver Archivos Adjuntos, Ver Temas, Eliminar Tarea.

La lista de tareas se puede ordenar por:

- Código
- Prioridad
- Estado
- Fecha de creación
- Fecha de cierre

La lista de tareas se puede filtrar por:

- Prioridad
- Estado
- Tema

Cuando un usuario ingresa a la hoja de trabajo y es líder del área que ha seleccionado, en la lista de colaboradores se muestran los nombres de las personas que trabajan en esa área y a quienes él lidera.

Cuando un usuario no es líder del área que está revisando, en la lista de colaboradores se muestran los nombres de las demás personas que colaboran en esa área.

Sobre cada tarea hay acciones específicas que se pueden realizar, estas acciones son las siguientes:

- Crear una tarea
- Ver tareas de otros colaboradores
- Eliminar una tarea
- Cambiar prioridad de una tarea
- Cambiar estado de una tarea
- Editar descripción de una tarea
- Agregar comentarios a una tarea
- Asignar un tema a una tarea
- Crear Tema
- Agregar fecha de cierre a una tarea

- Agregar archivo adjunto a la tarea
- Ver archivos adjuntos de una tarea

3.5 Módulo de Mantenimiento

El mantenimiento de la aplicación comprende la administración de usuarios, administración de áreas, los permisos de acuerdo al rol, la relación entre los usuarios y los grupos de trabajo y la configuración de los parámetros de la aplicación.

3.5.1 Administración de usuarios

La administración de usuarios comprende la creación, actualización y eliminación de usuarios.

Para agregar un usuario se debe especificar el nombre de usuario, con este parámetro la aplicación se conecta al servidor LDAP o Active Directory de la empresa y toma los datos necesarios para registrarlo.

Sobre los usuarios ya creados se puede actualizar la información del correo electrónico, nombre, apellido, e indicar si el usuario tiene el rol de administrador.

Cuando un líder o líder alternativo ingresa a la opción de crear usuarios, en la lista de usuarios activos sólo aparecen aquellos usuarios que están relacionados con las áreas que lidera. Al intentar eliminarlos, lo que se hará es borrar todas las relaciones de ese usuario con las áreas del líder.

3.5.2 Administración de áreas

La administración de áreas comprende la creación, y eliminación de las áreas a través de los cuales se relacionan los usuarios.

Para agregar una nueva área se debe especificar el nombre de la misma. Al eliminar un área se eliminan también todas las tareas relacionadas a la misma.

3.5.3 Permisos por rol de usuario

La opción de permisos por rol, permite definir qué permisos tiene disponible cada usuario según el rol que tiene asignado. Los permisos giran en torno a las tareas, el uso de la hoja de trabajo, la relación de los usuarios con las áreas, más no a la parte de parámetros, que compete a los administradores.

Los permisos que se pueden relacionar a cada rol son las siguientes:

- Crear una tarea
- Ver tareas de otros colaboradores
- Eliminar una tarea
- Cambiar prioridad de una tarea
- Cambiar estado de una tarea
- Editar descripción de una tarea
- Agregar comentarios a una tarea
- Asignar un tema a una tarea
- Crear Tema
- Agregar fecha de cierre a una tarea

- Agregar archivo adjunto a la tarea
- Ver archivos adjuntos de una tarea

3.5.4 Usuarios por área

En esta opción se puede relacionar a los usuarios con las áreas que se han creado. Se especifica si el usuario pertenece a cada área y si cumple las funciones de líder o de líder alternativo.

Cuando un administrador ingresa a esta opción se le muestra la lista de todos los usuarios y de todas las áreas.

Cuando un líder o un líder alternativo ingresa a esta opción, en la lista de usuarios se muestra sólo aquellos usuarios que están relacionados a las áreas que lidera. En la lista de áreas se muestran sólo las áreas que lidera.

Un líder o líder alternativo también puede relacionar a sus áreas a otra persona que se encuentra previamente registrada en el sistema.

3.5.5 Parámetros del sistema

La opción de reglas de trabajo permite configurar los parámetros que controlan ciertas funcionalidades del sistema. Estos parámetros son:

- Tiempo requerido para abrir una tarea (asignar fecha de cierre).
- Días que una tarea puede estar sin comentarios.
- Frecuencia de envío de notificaciones de correo electrónico
- Días límite para enviar mail recordatorio por cierre de tarea

3.6 Módulo de Métricas

El módulo de reportes y métricas permite conocer el rendimiento de los usuarios en el manejo de sus tareas a través de métricas. También existen reportes sobre el uso de la aplicación. Para el efecto se ha definido las siguientes métricas:

- Métrica de cumplimiento: Permite conocer cuantas tareas estimadas a completarse se cerraron realmente. Se compara el total general de tareas con las tareas estimadas a cerrar en el mes, tareas estimadas a cerrar en meses posteriores y las tareas cerradas en el mes. Junto a estos valores se agrega un valor de rendimiento, calculado de la división de tareas cerradas en el mes para el total de tareas a cerrar en el mes. Se puede analizar por mes o por trimestre.
- Métrica de retrasos: Muestra la cantidad de retrasos que tiene cada usuario sobre sus tareas. La idea de esta métrica es conocer cuántas tareas retrasa un usuario y en qué proporción. Se mide número de tareas dentro del tiempo correcto, tareas retrasadas por una fecha y tareas retrasadas más de una vez.

Los reportes son los siguientes y pueden analizarse por mes o por trimestre:

- Reporte de uso del sistema: Muestra las fechas y hora en que los usuarios han ingresado al sistema durante el mes

- Lista de áreas: Muestra la lista de todas las áreas con su respectivo líder.

CAPÍTULO 4

4. IMPLEMENTACIÓN

Luego de haber planteado las ideas que sustentan este proyecto y haber elaborado un diseño que ayude a plasmar estas ideas, se debe pensar cómo implementarlo.

Es importante aplicar conceptos de arquitectura de software para diseñar y elaborar un sistema, esto nos permite evitar problemas de reescritura de código a causa de cambios no esperados dentro de nuestros proyectos, nos brinda a su vez una buena base para que el sistema sea escalable, razones que sirvieron para aplicar en este proyecto el modelo de 3 capas sugerido en el portal oficial de ASP.NET [22], el cual se analizará brevemente en este capítulo.

También se revisará las técnicas aplicadas en el desarrollo de la aplicación, aprovechando las bondades que Microsoft .NET ofrece usando el Visual Studio 2005.

4.1 Arquitectura

Para el acceso a datos se aplica el modelo de 3 capas, tomando como referencia la sugerencia del portal oficial de ASP.NET [22], que se explica a continuación.

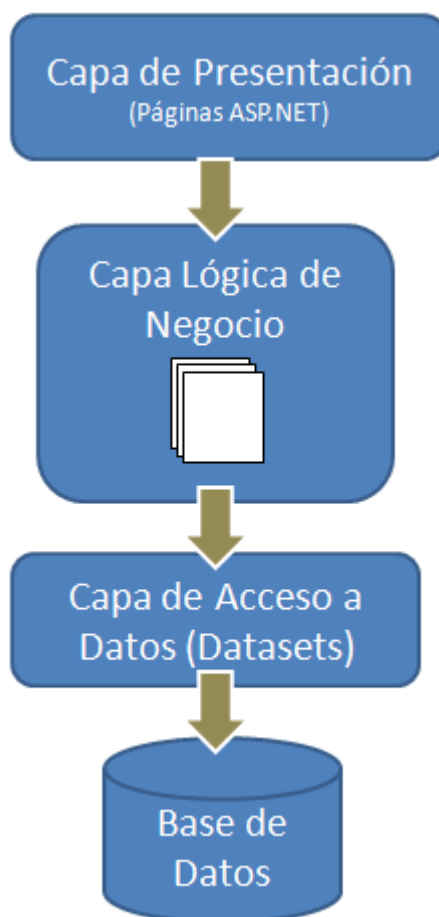


Figura 4.1 Modelo 3 capas

La capa de acceso a datos se compone por archivos de tipo Dataset, los cuales en esencia son esquemas XML que representan un medio de acceso a la base de datos.

La capa lógica o capa de negocios se maneja con clases, para lo cual se ha escogido el lenguaje Visual Basic .NET. Esta capa invoca a la capa de acceso a datos y es la que realiza el procesamiento debido de la información para enviarla a la capa de presentación.

La capa de presentación se compone de formularios Web, los cuales se componen de dos archivos, uno el archivo .aspx que contiene todo el XHTML que genera la información y el otro archivo es el .aspx.vb que contiene el procesamiento que se realiza para la interacción del usuario con la aplicación.

Para una mejor administración de las clases se usan 6 carpetas que representan a los objetos dentro de la aplicación, estas carpetas son:

- Areas: Para el manejo de los grupos de trabajo
- Reportes: Para el manejo de los reportes y las métricas
- Seguridad: Administra toda la seguridad de la aplicación
- Tareas: Para el manejo de las tareas
- Usuarios: Administra a los usuarios de la aplicación
- Util: Maneja información común para toda la aplicación

En cada una de estas carpetas existen 2 subcarpetas. La carpeta BOL que contiene los archivos de la capa lógica de negocio y la carpeta DAL que contiene los archivos de la capa de acceso a datos. La Figura 4.2 muestra cómo están organizadas las carpetas en el proyecto.



Figura 4.2 Distribución de carpetas del proyecto

Control de acceso de usuarios

Para la seguridad de la aplicación y el manejo de usuarios se usa el complemento llamado Membership que provee ASP.NET. Este complemento se encarga de administrar la sesión de los usuarios y la seguridad en el sitio Web. Lo primero que se debe hacer es ejecutar una herramienta llamada aspnet_regsql.exe para crear en la base de datos unas tablas que se usan para este propósito. La Figura 4.3 muestra el programa que se debe ejecutar y la ruta en la que se encuentra.

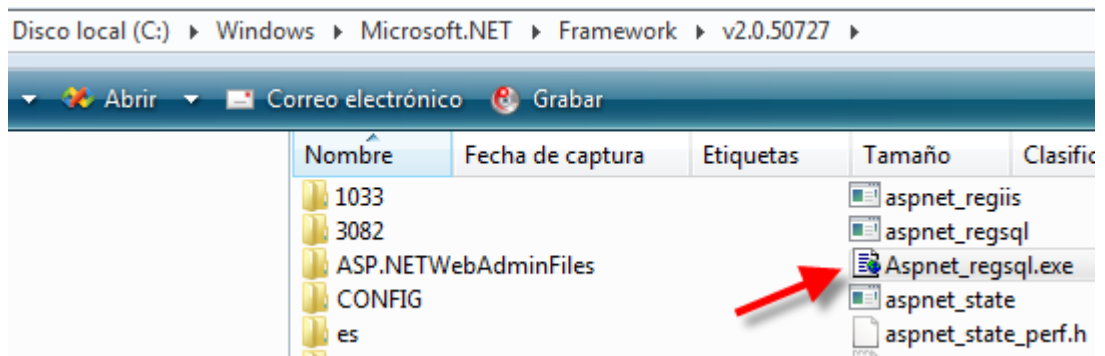


Figura 4.3 Programa para configurar Membership

Este programa requiere el servidor en que está alojada la base de datos y el nombre de la misma, luego crea las tablas que necesita para la validación de usuarios. Las tablas creadas tienen el prefijo aspnet, y se explican en la sección Capa de Acceso a Datos.

Membership incluye también unos controles de usuario para iniciar sesión, crear usuarios, etc. Para que esto esté habilitado, en el archivo Web.Config se define la sección “membership” dentro de la sección “system.web”, tal como se muestra en la Figura 4.4.

```

<system.web>
  <roleManager enabled="true"/>
  <membership>
    <providers>
      <remove name="AspNetSqlMembershipProvider"/>
      <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider" data-path="~/WebResource/AspNetSqlMembershipProvider.dll" data-key="AspNetSqlMembershipProvider" data-value="AspNetSqlMembershipProvider" data-order="1" data-user-provider="AspNetSqlMembershipProvider"/>
    </providers>
  </membership>
  <authentication mode="Forms"/>

```

Figura 4.4 Sección Membership en el Web.config

En la sección membership se define el elemento “providers”, dentro del cual se agrega el nombre que va a tener nuestro administrador de usuarios y se le definen varios parámetros como la cadena de conexión que debe usar, el tipo de proveedor, y otros parámetros específicos de funcionalidad [23].

Esto se complementa con la sección “roleManager” y su atributo “enabled” con valor “true”, lo cual habilita el uso de roles en la aplicación. También se necesita tener la sección “authentication” y su atributo “mode” con el valor “Forms”, esto indica que la autenticación se hace a través de la Web.

Luego para que los usuarios validen su usuario y clave, se usa el control Login, el cual está en la barra de herramientas y genera un control con las cajas de texto y un botón para procesar la autenticación del usuario. La Figura 4.5 muestra el control Login usado en la página de validación de usuarios del proyecto.

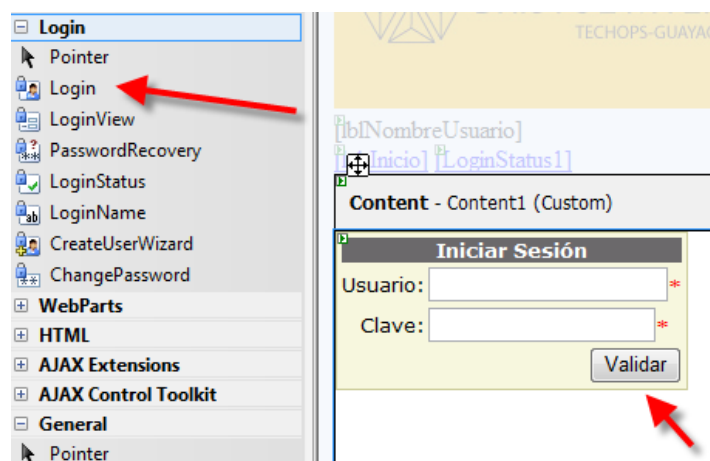


Figura 4.5 Control para validar usuarios

Al dar clic en el botón “Validar”, Membership se encarga de realizar la autenticación del usuario y en caso de ser exitosa direcciona al mismo a la página principal de la aplicación.

Usar Membership nos ahorra tener que crear las tablas para manejar los usuarios en la base de datos, manejar la sesión de los usuarios y controlar los accesos a la aplicación. Incluye también métodos para crear, editar o eliminar usuarios, crear, editar o eliminar roles y asignar roles a los usuarios. Vale recalcar que se debe crear una interfaz para este propósito, pero se aprovechan los métodos antes mencionados. En la sección Mantenimiento de este capítulo se explica la interfaz creada en este proyecto para la administración de usuarios.

4.2 Capa de Acceso a Datos

Para el acceso a datos se usa el motor SQL Server, el cual es el complemento ideal para proyectos con .NET. Se puede usar desde la versión 7.0 en adelante.

4.2.1 Modelo de base de datos

La Figura 4.6 presenta el diagrama de la base de datos de este proyecto.

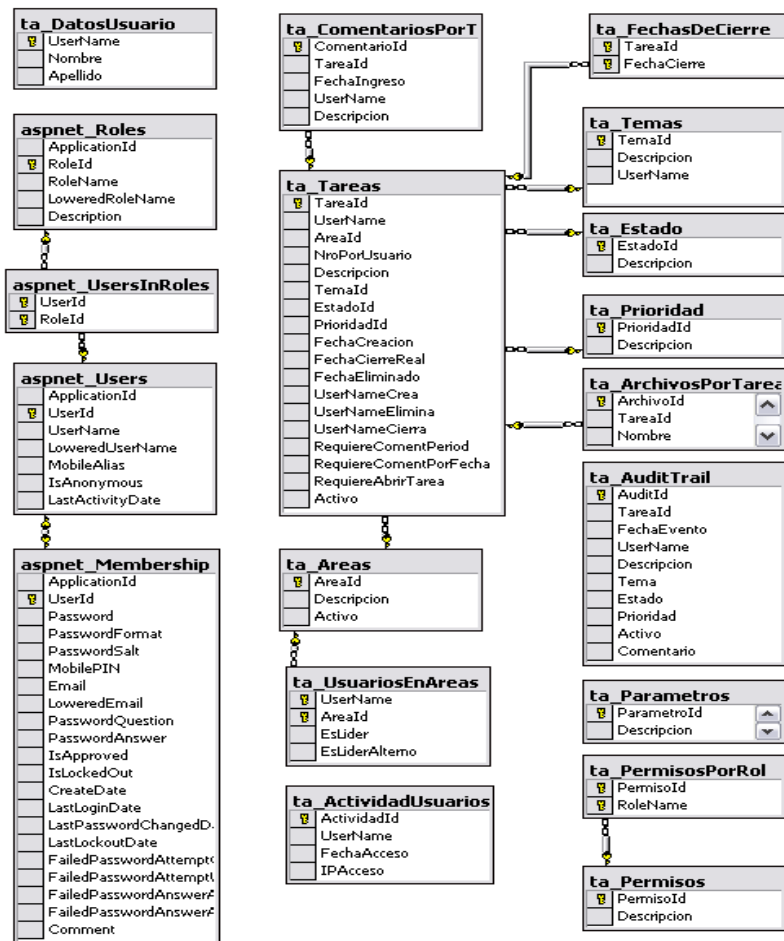


Figura 4.6 Diagrama de base de datos

Las tablas creadas por el componente Membership de ASP.NET para la seguridad se identifican por el prefijo “aspnet_” en su nombre. A continuación una descripción de la información almacenada en cada una de estas tablas:

- aspnet_Roles: Los roles que maneja la aplicación
- aspnet_Users: Los usuarios de la aplicación
- aspnet_Membership: Datos de la cuenta de cada usuario

- aspnet_UsersInRoles: Relación de los usuarios con los roles.

Para complementar la información de los usuarios se ha creado la tabla ta_DatosUsuario, en la cual se graban el nombre y el apellido de cada usuario, información que no es contemplada por el componente Membership.

Las demás tablas guardan la información referente a las tareas. Estas tablas se identifican por el prefijo “ta_” en su nombre. Las tablas son:

- ta_Tareas: Se almacenan las tareas de todos los usuarios
- ta_FechasDeCierre: Fechas de cierre que se han ido agregando a cada tarea
- ta_ComentariosPorTarea: Lista de Comentarios que se han ido agregando a cada tarea.
- ta_Temas: Temas que se han creado por los usuarios y se relacionan a las tareas.
- ta_Estado: Lista de estados posibles de una tarea
- ta_Prioridad: Lista de prioridades posibles de una tarea.
- ta_ArchivosPorTarea: Archivos adjuntos por cada tarea.
- ta_Areas: Lista de grupos de trabajo creados.
- ta_UsuariosEnAreas: Almacena la relación de los usuarios con los grupos de trabajo.
- ta_Permisos: Lista de permisos que se pueden asignar a los usuarios para el manejo de las tareas

- ta_PermisosPorRol: Relación de los permisos asignados a cada rol dentro de la aplicación
- ta_Parametros: Lista de parámetros generales de la aplicación con su respectivo valor, que puede ser modificado por el administrador.
- ta_AuditTrail: Guarda el historial de cada tarea.
- ta_ActividadUsuarios: Guarda el historial de uso de la aplicación por cada usuario.

4.2.2 Modelo de acceso a datos

Para el acceso a datos, comúnmente en las aplicaciones se crea una clase que instancia una conexión a la base de datos y otras clases en las cuales se generan manualmente las sentencias SQL. Luego se realizan procedimientos para pasar los parámetros respectivos, abrir la conexión a la base de datos, ejecutar la sentencia, capturar los datos si es que aplica y luego volver a cerrar la conexión. A esto se le agrega el control que por lo general se realiza para capturar excepciones. Realizar este tipo de procedimientos toma en los casos más pequeños unas 12 líneas de código. En aplicaciones que usan sentencias SQL más complejas se puede tomar de 30 a 50 líneas de código y esto por cada acción que se desee realizar.

Para ahorrar tiempo y líneas de código, en este proyecto se aprovecha la ventaja que provee ASP.NET a través de los esquemas XML que representan un Dataset.

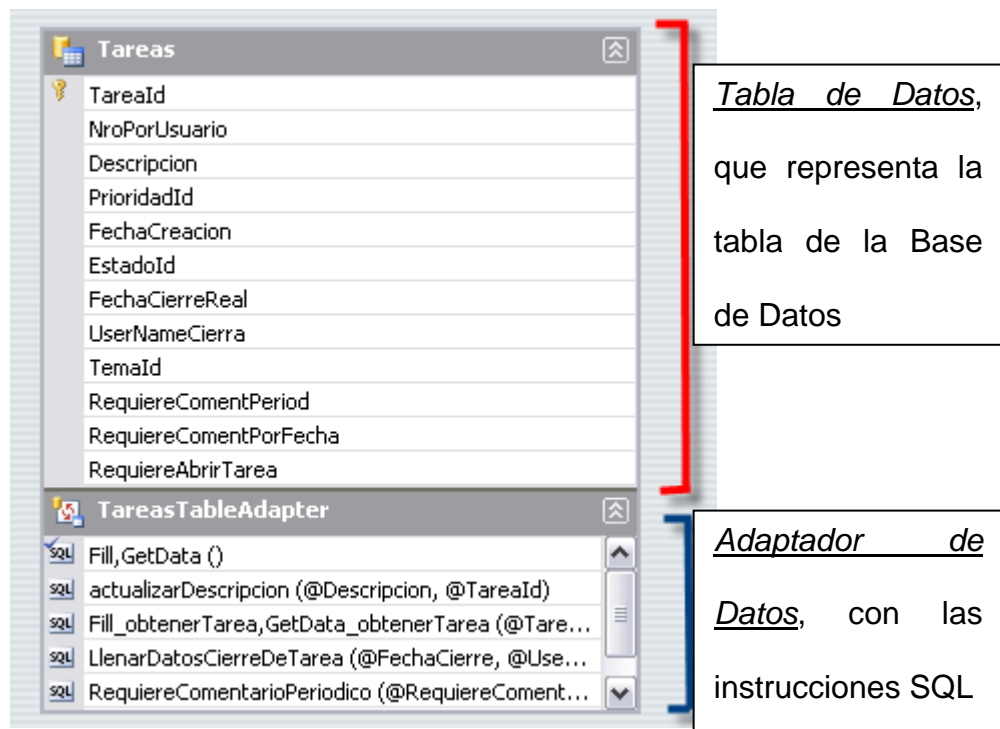


Figura 4.7 Ejemplo de un adaptador de datos

El Dataset físicamente es un archivo en el cual se crean uno o más adaptadores, los cuales generan una representación de la tabla de la base de datos en la aplicación. La representación de la tabla de datos es la que se marca con rojo en la Figura 4.7.

Luego en el adaptador se pueden agregar múltiples sentencias SQL, que se muestran en la parte de color azul de la Figura 4.7. Para crear las sentencias SQL se usa una opción que brinda Visual Studio la cual permite generar todo tipo de sentencias SQL y probarlas antes de grabarlas. Esto nos ahorra gran cantidad de tiempo y líneas de código. A cada sentencia SQL se la crea como un método y los parámetros se pasan como parámetros de cualquier

método. Con esto nos ahorramos instrucciones para instanciar parámetros, abrir y cerrar la conexión, crear la sentencia SQL, etc., lo que generalmente se implementaría como se muestra en la Figura 4.8.

```
Public Function obtenerTarea(ByVal TareaId As Int32) As TareasDataTable
    Dim dc As New SqlCommand
    Dim cn As SqlConnection
    Dim dr As SqlDataReader
    Dim dt As New TareasDataTable

    cn = New SqlConnection(ConfigurationManager.ConnectionStrings("TareasCN").Connect:
    dc.CommandType = CommandType.Text
    dc.Connection = cn
    dc.CommandText = "SELECT t.[TareaId], t.[NroPorUsuario], t.[Descripcion], "
    dc.CommandText &= "t.[PrioridadId], t.[FechaCreacion], t.[EstadoId], "
    dc.CommandText &= "t.[FechaCierreReal], t.[UserNameCierra], t.[TemaId], "
    dc.CommandText &= "t.[RequiereComentPeriod], t.[RequiereComentPorFecha], "
    dc.CommandText &= "t.[RequiereAbrirTarea]"
    dc.CommandText &= "FROM [ta_Tareas] t"
    dc.CommandText &= "WHERE t.Activo = 1      AND t.TareaId = @TareaId"

    dc.Parameters.Add(New SqlParameter("@TareaId", TareaId))

    dc.Connection.Open()
    dr = dc.ExecuteReader()
    If dr.HasRows Then
        'instrucciones para llenar el TareasDataTable
    End If
    dc.Connection.Close()
    Return dt
End Function
```

Figura 4.8 Código común para recuperar datos de la base de datos

Usando la técnica propuesta por ASP.NET, el mismo ejemplo de la Figura 4.8 se reduce a la porción de código de la Figura 4.9, con la cual, para ejecutar la sentencia SQL y que se retornen los datos se llama al método "GetData_obtenerTarea". Este método está definido en el adaptador llamado

TareasTableAdapter (ver Figura 4.7), el cual es parte de la capa de acceso a datos de la clase Tarea.

```
Public Function obtenerTarea(ByVal TareaId As Int32) As TareasDataTable  
    Return Me.Tareas.GetData_obtenerTarea(TareaId)  
End Function
```




Figura 4.9 Código para recuperar datos de la base de datos en ASP.NET

La conexión a la base de datos también se simplifica. Cada adaptador de datos lee la cadena de conexión que se encuentra en el archivo Web.Config de la aplicación y se encarga de administrar por su cuenta cada conexión que se realice a la base.

Para definir la cadena de conexión a la base de datos se agrega la sección “connectionStrings” [24] en el archivo Web.Config, según se muestra en la Figura 4.10.

```
<connectionStrings>  
    <remove name="LocalSqlServer"/>  
    <add name="LocalSqlServer" connectionString="Data Source=  
    <add name="cadenaConexion" connectionString="Data Source=  
</connectionStrings>
```

Figura 4.10 Sección ConnectionString en el archivo Web.Config

4.2.3 Encriptación de configuraciones

Escribir la cadena de conexión en un archivo que no es compilado o encriptado puede resultar riesgoso por diversos motivos, en especial porque la cadena de conexión contiene el nombre de la base de datos, el usuario y clave para conectarse a la misma. Para evitar este problema, el Framework .NET ofrece una herramienta para agregar encriptación en cualquier sección del archivo Web.Config. Esta herramienta está ubicada en la misma ruta en la que se encuentra la herramienta para aplicar Membership, bajo el nombre aspnet_regiis, según se muestra en la Figura 4.3.

Esta herramienta debe ejecutarse bajo línea de comandos usando la siguiente instrucción:

```
aspnet_regiis -pe "connectionStrings" -app  
"/Nombre_Aplicacion_En_IIS" -prov  
"RsaProtectedConfigurationProvider"
```

En la instrucción mostrada arriba, Nombre_Aplicacion_En_IIS representa el nombre que tiene nuestra aplicación cuando la publicamos en el Internet Information Server.

Cuando una página de la aplicación es solicitada a través del navegador de Internet, ASP.NET se encarga de descifrar de manera automática en tiempo de ejecución las secciones que se han encriptado [25].

4.3 Capa Lógica de Trabajo

Como es común en una capa lógica de trabajo, aquí se procesa toda la información que proviene de la base de datos a través de la capa de acceso a datos.

La comunicación con la capa de acceso a datos se realiza creando instancias por cada adaptador según se requiera. Esto se consigue definiendo una propiedad de solo lectura del tipo del adaptador al que hacemos referencia. La Figura 4.11 muestra cómo se instancia el adaptador `TareasTableAdapter` mostrado en la Figura 4.7.

```
Protected ReadOnly Property Tareas() As TareasTableAdapter
    Get
        If _TareasAdapter Is Nothing Then
            _TareasAdapter = New TareasTableAdapter()
        End If
        Return _TareasAdapter
    End Get
End Property
```

Figura 4.11 Ejemplo de cómo instanciar un adaptador de datos

La figura muestra cómo se instancia el adaptador de datos de `TareasTableAdapter` que fue explicado en la sección de la capa de acceso a datos. Con esta instancia podemos acceder a cada una de las sentencias SQL ahí definidas.

Para establecer una comunicación adecuada con la capa de presentación, es importante que en cada método implementado se identifique claramente su

tipo, es decir si es un método que retorna datos, que actualiza, inserta o elimina información. Esto se lo logra ubicando sobre cada método una instrucción como la que se muestra en la Figura 4.12.

```
<System.ComponentModel.DataAnnotations.DataObjectMethodAttribute _  
(System.ComponentModel.DataAnnotations.DataObjectMethodType.Select, True) > _
```

Figura 4.12 Instrucción para identificar métodos una clase como fuente de datos

En esta instrucción se debe especificar el tipo de función, los valores posibles son:

- Select: Para funciones que retornan registros
- Insert: Para funciones que graban registros
- Update: Para funciones que actualizan registros
- Delete: Para funciones que eliminan registros

El último parámetro es un valor de verdadero y falso, con el cual se indica si el método es el método principal de la clase.

Para completar la definición de las funciones, se necesita agregar al momento de declarar la clase una instrucción que indique que la clase provee datos. Estos datos serán luego utilizados por la capa de presentación.

La Figura 4.13 muestra la sentencia que se debe especificar antes de declarar una clase.

```

<System.ComponentModel.DataObject()> _
Public Class Tarea

```

Figura 4.13 Instrucción para identificar una clase como fuente de datos

A continuación las instrucciones dentro de cada método de la clase se generan de acuerdo a la lógica de la aplicación.

4.4 Capa de Presentación

Para presentar a los usuarios la información que viene a través de la capa lógica usamos unos componentes conocidos como ObjectDatasource.

Un ObjectDatasource puede tener referenciados un método para retornar información, uno para actualizar, uno para eliminar y uno para crear. Se le debe especificar la clase de la capa lógica en la cual están los métodos. La Figura 4.14 muestra las propiedades que tiene un ObjectDatasource.

Data	
(Expressions)	
ConflictDetection	OverwriteChanges
ConvertNullToDBNull	False
DataObjectTypeName	
DeleteMethod	
DeleteParameters	(Collection)
FilterExpression	
FilterParameters	(Collection)
InsertMethod	
InsertParameters	(Collection)
OldValuesParameterF	original_{0}
SelectMethod	obtenerListaDeTareas
SelectParameters	(Collection)
SortParameterName	
TypeName	Tarea
UpdateMethod	
UpdateParameters	(Collection)

Figura 4.14 Propiedades de un ObjectDataSource

En la Figura 4.14 se muestra con línea azul el lugar donde se especifican los métodos según el tipo y con línea roja donde se especifica la clase que contiene a los métodos. Toda esta información es asignada automáticamente por el Visual Studio al momento de crear el ObjectDataSource. Esto nos permite ahorrarnos muchas líneas de código al momento de presentar la información al usuario.

Un ObjectDataSource se puede usar como fuente de datos para cualquiera de los controles de ASP.NET.

ASP.NET AJAX

Aplicar AJAX con ASP.NET [1] es una tarea muy sencilla. Uno mantiene la programación normal que realizaría si no usa AJAX. Para que se genere contenido asíncronico lo que se debe hacer es primero agregar en la página un administrador de script, usando la instrucción que se muestra en la Figura 4.15.

```
<asp:ScriptManager ID="ScriptManager1" runat="server" />
```

Figura 4.15 Etiqueta que instancia un administrador de Script AJAX

Luego la sección de la página que deseamos manejar de manera asíncronica la ubicamos dentro de la etiqueta UpdatePanel y su sub-etiqueta ContentTemplate, tal como se muestra en la Figura 4.16.

```
<asp:UpdatePanel ID="UpdatePanel2" runat="server">
  <ContentTemplate>
    <!-- CONTENIDO QUE TRABAJARÁ ASINCRÓNICAMENTE CON AJAX-->
  </ContentTemplate>
</asp:UpdatePanel>
```

Figura 4.16 Etiquetas que permiten ejecución asincrónica

De esta forma, el Framework de ASP.NET se encarga de administrar la acción de manera asincrónica sin que sea necesario que el desarrollador cree funciones en JavaScript como se requeriría comúnmente en otra plataforma.

Diseño con hojas CSS

En lo que respecta al diseño, para el manejo de hojas de estilo se aplica temas. Los temas permiten crear un grupo de archivos CSS y archivos SKIN que van a ser interpretados por todas las páginas sin hacer referencia a cada uno de ellos en cada página. Por lo general, cuando no se usan temas, en cada página hay que hacer una referencia directa a cada archivo CSS de la siguiente manera:

```
<link rel="stylesheet" href="archive_de_estilo.css" type="text/css" />
```

Esto implica que al tener múltiples hojas de estilo hay que repetir esto por cada hoja de estilo y esto en cada una de las páginas que se usen.

Para aplicar los temas en ASP.NET, se crea la carpeta App_Themes y dentro de esta carpeta se crea una subcarpeta que representa al tema. Dentro de la

carpeta del tema se crean todos los archivos CSS que uno desee. La Figura 4.17 muestra la carpeta con el tema usado en el proyecto.

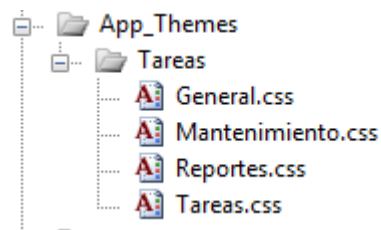


Figura 4.17 Carpeta de temas del proyecto

Para que estos archivos CSS sean interpretados en las páginas, en el archivo Web.Config se hace referencia al tema que se desea usar. Esta referencia se hace en la sección “pages”, según se muestra en la Figura 4.18.

```
<pages theme="Tareas">  
<controls>  
  <add tagPrefix="asp" n  
</controls>  
</pages>  
<!--
```

Figura 4.18 Sección del Web.Config para definir tema

Esto nos ahorra gran cantidad de tiempo y líneas de XHTML. Solo se necesita agregar al atributo “theme” el nombre de la carpeta del tema.

4.5 Módulo Hoja de Trabajo

La hoja de trabajo es la pantalla principal de la aplicación, donde se trabaja en las tareas. Cuando un usuario ingresa a esta sección, el sistema toma el nombre de usuario y carga la siguiente información:

Área: El sistema muestra la lista de las áreas a las cuales pertenece el usuario.

Usuarios: Cuando el usuario que está revisando la hoja de trabajo es líder de área o líder alterno, en la lista de usuarios aparecen además de su nombre los nombres de todos aquellos que pertenecen al área que lidera. En el caso de que se configure la aplicación para que una persona que no es líder de área o líder alterno pueda ver las tareas de las demás personas del área al que pertenece, éstas también aparecerán en la lista.

Dependiendo del usuario seleccionado en la lista de usuarios y el área escogida se muestra la siguiente información:

Temas: Se presenta la lista de temas creados por el usuario.

Tareas: Las lista de tareas del usuario según el área que se está revisando.

Por defecto, cuando ingresa un usuario, se presentan directamente sus tareas para la primer área que aparece en la lista de grupos, los cuales se presentan en orden alfabético.

Las Tareas se presentan en otra página embebida en un control HTML llamado IFRAME. Esta lista se actualiza cada vez que el usuario cambia alguno de los filtros presentes en la pantalla. Desde la hoja de trabajo se

genera la URL que se carga en el IFRAME, esta URL contiene los siguientes parámetros:

- Orden: Instrucción SQL que indica el campo a través del cual se ordenan las tareas
- Filtro: Instrucción SQL que indica los filtros seleccionados por el usuario para presentar las tareas
- Areald: El código del área al que pertenecen las tareas.
- UserName: El nombre de usuario de la persona cuyas tareas se están revisando.
- TipoDeVista: Indica si se presentan las tareas activas, cerradas o todas juntas.

La página embebida llama a la capa lógica de las tareas a través del método `ObtenerListaDeTareas`, el cual necesita se le indique el nombre del usuario dueño de las tareas, el código del área a la cual están relacionadas las tareas y el tipo de vista. Desde la capa lógica se retorna un `DataTable` con la información solicitada, la cual se usa para generar una vista en la que se aplican los filtros y condiciones de ordenamiento seleccionadas por el usuario y transmitidas a través de la URL. La Figura 4.19 muestra la hoja de trabajo con una lista de tareas.

VER TAREAS DE								
GRUPO: Sistemas		USUARIO: William Murillo		Nueva Tarea		Reportes y Métricas		Inicio
FILTROS								
VER TAREAS: Activas			PRIORIDAD: Todas		TEMA: Todos			
#	Cod	Descripción	Prio	Creado	Fecha-Fin	Estado	Comentarios	Menu
1	1	abc	1	17/09/2008 21:20:25		Nuevo	<< Ver Todos >>	
2	2	efg	1	17/09/2008 21:20:33		Nuevo	<< Ver Todos >>	
3	3	xyz	1	17/09/2008 21:20:41	18/09/2008	Abierto	17/09/2008 21:20:55 - murillow: aasdas << Ver Todos >>	

Figura 4.19 Hoja de trabajo con lista de tareas

4.6 Módulo de Mantenimiento

El mantenimiento de la aplicación se compone de 5 opciones:

Mantenimiento de usuarios

El mantenimiento de usuarios se puede implementar usando dos mecanismos diferentes:

- Usando un servidor LDAP o Active Directory de Windows.
- Administrando directamente los usuarios.

Cuando se usa un servidor LDAP o Active Directory de Windows, al crear un usuario solo se requiere ingresar el nombre de usuario y el sistema automáticamente captura el resto de la información a través de la capa lógica de Usuarios.

Si se administra directamente la información de los usuarios, el administrador de la aplicación debe proveer toda la información, la cual se valida y procesa por la capa lógica de usuarios.

Cuando un usuario es líder de grupo tiene la posibilidad de agregar nuevos usuarios para relacionarlos luego a los grupos que lidera y también la posibilidad de quitarle cualquier relación que tenga con ellos.

Para conseguir esto, cuando ingresa un usuario a esta sección, en la capa lógica de usuarios se verifica si el usuario tiene el rol de Administrador del sistema, si no lo tiene se devuelve sólo la lista de usuarios que están relacionados a los grupos que lidera.

Mantenimiento de áreas

Esta opción permite crear, eliminar y editar información de las áreas. Se compone de dos partes, la primera es el panel de creación, en el cual se debe ingresar el nombre del nuevo grupo para poder crearlo.

La segunda parte es la lista de áreas existentes, en la cual se muestra el código del área y su nombre. Junto a cada área también están 2 botones, uno para actualizar la información y otro para eliminar.

Usuarios por área

Esta opción nos permite definir qué usuarios pertenecen a una determinada área y quién es líder o líder alternativo. Esta opción, como todas las de mantenimiento la tiene un usuario con rol de administrador, pero para facilitar

el trabajo de un líder, se permite que este pueda manipular la relación de los usuarios con los grupos de trabajo que lidera.

Cuando un usuario ingresa a esta opción, en la capa lógica de grupos de trabajo se verifica si el usuario tiene el rol de administrador, caso contrario en la lista de áreas se muestran las áreas de las cuales es líder y se habilita la opción de especificar si una persona puede ser o no líder alterno de un área. Para mostrar la lista de usuarios se hace una validación parecida a la que se explicó en la sección de administración de usuarios.

Configuración de parámetros

La sección de configuración de parámetros permite definir parámetros de trabajo, estos son:

- Tiempo requerido para abrir una tarea (asignar fecha de cierre).
- Días que una tarea puede estar sin comentarios.
- Días límite para enviar mail recordatorio por cierre de tarea

Esta opción está habilitada solo para el administrador del sistema.

Permisos por rol

Esta opción permite guardar una relación entre los roles y las diferentes acciones que se pueden realizar en la aplicación, en especial en lo que respecta a las tareas. Cuando un usuario ingresa a la aplicación se cargan todos los permisos que tiene el usuario de acuerdo a los roles que tiene asignados.

4.7 Módulo de Métricas

Para las métricas se usa la tecnología Microsoft Reporting a través de archivos de reporte, los cuales tienen extensión .rdlc.

La relación con los datos se realiza a través de los archivos DataSet creados en la capa de Acceso a Datos.

Para visualizar un reporte en un formulario Web se debe usar el visor de reportes, el cual requiere el nombre del archivo de reporte y una fuente de datos que va a gestionar la conexión con la capa de acceso a datos y pasar la información al reporte para que sea presentada al usuario.

La Figura 4.20 muestra la porción de código del visor de reportes con la métrica de cumplimiento explicada en el capítulo 3.

```
<rsweb:ReportViewer ID="ReportViewer1" Visible="false" runat="server" Font-Names="Ver
>
  <LocalReport ReportPath="Reportes\CumplimientoMes.rdlc">
    <DataSources>
      <rsweb:ReportDataSource DataSourceId="ObjectDataSource1" Name="MetricasDf
    </DataSources>
  </LocalReport>
</rsweb:ReportViewer>
<asp:ObjectDataSource ID="ObjectDataSource1" runat="server" SelectMethod="GetData"
  TypeName="MetricasDSTableAdapters.ta_Metricas_Cumplimiento_MensualTableAdapter">
  <SelectParameters>
    <asp:Parameter DefaultValue="0" Name="AreaId" />
    <asp:Parameter DefaultValue="1" Name="Mes" />
    <asp:Parameter DefaultValue="2099" Name="Anio" />
  </SelectParameters>
</asp:ObjectDataSource>
```

Figura 4.20 Etiquetas que representan un visor de reportes

La Figura 4.21 presenta ya en la aplicación, la métrica de cumplimiento, en el visor de reportes.

Reportes » Métrica de Cumplimiento

AREA: Sistemas RANGO DE ANÁLISIS: Por Mes Por Trimestre Mes: 2do Trimestre

Generar Reporte

1 of 1 100% Find | Next Select a format Export

MÉTRICA DE CUMPLIMIENTO TRIMESTRAL

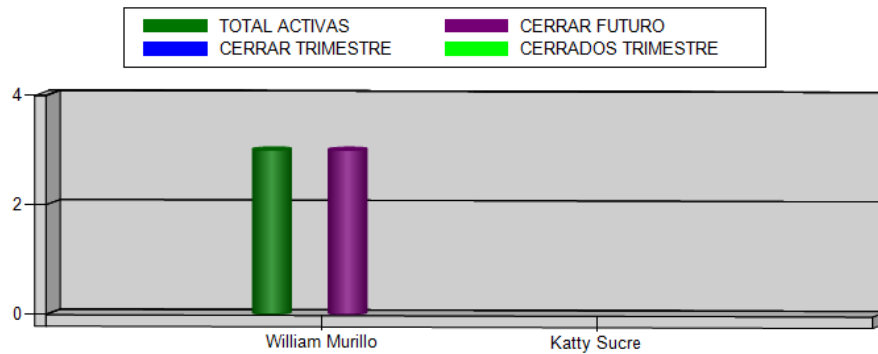


Figura 4.21 Métrica de cumplimiento

4.8 Seguridad

Cada uno de los permisos que se asignan a cada rol, tienen un código alfanumérico. Cuando un usuario ingresa a la Hoja de Trabajo, se leen todos los códigos de los permisos que se le han asignado dependiendo de sus roles y se ejecuta un procedimiento que va preguntando por cada uno de los códigos y de estar asignado se lo habilita. Por defecto todos están deshabilitados.

Autenticación de usuarios

Para la autenticación de usuarios, se invoca al método ValidarUsuario dentro de la clase Seguridad. Este método valida los usuarios, recibiendo como parámetros el nombre de usuario y la clave. Es importante aclarar que el

Membership siempre se usa a pesar de que LDAP esté activo. Membership siempre tiene registrado los usuarios, la diferencia radica en que cuando se activa LDAP, Membership no administra las contraseñas, éstas son validadas directamente con el servidor LDAP o Active Directory.

CAPÍTULO 5

5. PRUEBAS DEL SISTEMA

Una vez que se ha finalizado la implementación del proyecto, es necesario hacer una evaluación que permita conocer otros puntos de vista diferentes a los que se han tenido al momento de crear el proyecto. Para obtener un producto de calidad es conveniente que personas ajenas al desarrollo del mismo puedan revisarlo y emitir sus comentarios, lo que a la postre se transformará en mejoras importantes y valiosas. En este capítulo se explicarán las diversas pruebas que se han efectuado y finalmente se hablará de las observaciones que se han podido recopilar y aplicar.

5.1 Pruebas de Funcionalidad

Para este proyecto se definió hacer pruebas informales con usuarios que tengan experiencia en computación y usuarios enfocados en otras áreas, no relacionadas directamente con la computación. El fin de este tipo de pruebas es obtener retroalimentación de estos usuarios respecto a la funcionalidad y las características del sistema.

Para estas pruebas se definieron 5 casos de prueba, y se usó un usuario por cada caso. Los casos usados son los siguientes:

1. Usuario que es líder de un área y colaborador en otra área.
2. Usuario que es líder en dos áreas.
3. Un usuario que es colaborador de solo un área
4. Usuario que es colaborador en más de un área
5. Usuario líder alternativo de un área y colaborador de otra área.

Previo a empezar las pruebas se explicó a los usuarios de qué se trataba el sistema. Se explicó a demás a qué caso de prueba iban a estar vinculados para que tengan una mejor noción de sus posibilidades.

El sistema se preparó con los siguientes permisos por cada rol:

Consultor:

- Crear una tarea
- Ver historial de una tarea
- Cambiar estado de una tarea
- Editar descripción de una tarea

- Agregar comentarios a una tarea
- Asignar un tema a una tarea
- Agregar fecha de cierre a una tarea
- Agregar archivo adjunto a la tarea
- Ver tareas cerradas
- Ver tareas de otros colaboradores
- Ver archivos adjuntos de una tarea

Líder:

- Crear una tarea
- Eliminar una tarea
- Ver historial de una tarea
- Cambiar prioridad de una tarea
- Cambiar estado de una tarea
- Editar descripción de una tarea
- Agregar comentarios a una tarea
- Asignar un tema a una tarea
- Agregar fecha de cierre a una tarea
- Agregar archivo adjunto a la tarea
- Ver tareas cerradas
- Ver tareas de otros colaboradores
- Ver archivos adjuntos de una tarea
- Crear usuarios

- Actualizar datos de usuario
- Relacionar usuarios con áreas

Líder alterno:

- Crear una tarea
- Eliminar una tarea
- Ver historial de una tarea
- Cambiar prioridad de una tarea
- Cambiar estado de una tarea
- Editar descripción de una tarea
- Agregar comentarios a una tarea
- Agregar fecha de cierre a una tarea
- Agregar archivo adjunto a la tarea
- Ver tareas cerradas
- Ver tareas de otros colaboradores
- Ver archivos adjuntos de una tarea
- Actualizar datos de usuario
- Relacionar usuarios con áreas

Los usuarios que participaron en las pruebas tenían el siguiente perfil:

Usuario 1:

Sexo: Femenino

Edad: 24 años

Estudios: Ing. en Sistemas Multimedia

Usuario 2:

Sexo: Masculino

Edad: 23 años

Estudios: Ing. en Sistemas de Información

Usuario 3:

Sexo: Femenino

Edad: 48 años

Estudios: Ing. Química

Usuario 4:

Sexo: Masculino

Edad: 21 años

Estudios: Ingeniería Comercial

Usuario 5:

Sexo: Femenino

Edad: 24 años

Estudios: Economía

Cada usuario tuvo la libertad de efectuar las pruebas durante 3 días de manera independiente, se brindó soporte cuando existía alguna duda pero nunca hubo influencia directa en el momento de ejecutar las pruebas.

5.2 Observaciones Obtenidas de las Pruebas

Luego de que cada usuario notificara haber culminado sus pruebas se procedió a conversar con cada uno revisando el sistema para obtener la mejor retroalimentación posible.

De la retroalimentación brindada se generaron las siguientes observaciones:

- La etiqueta del tema se mostraba inicialmente debajo de la descripción de la tarea. Se recomendó mostrarla sobre la descripción.
- Permitir agregar archivos adjuntos a las tareas.
- Los temas se podían crear por cada usuario. Se recomendó que el sistema permita que un líder de área cree un tema para todos los usuarios perteneciente a dicha área.
- Ordenar las tareas por fecha de cierre estimada. El sistema permite ordenar las tareas por algunos campos pero no se había considerado la fecha de cierre estimada por el usuario.
- El gráfico de pastel que aparece en la pantalla inicial de la aplicación mostraba unos 0 si no habían tareas de algún estado. Se recomendó no mostrar esos 0, es decir que sólo se muestren los valores de los estados que sí tengan tareas.
- Al crear tarea, permitir escoger usuario y área. El proyecto en un principio mostraba el nombre del área y del usuario sobre el cual se crea la tarea. Se recomendó permitir escoger usuario y área, en

especial por los líderes de grupo que pueden necesitar crear tareas para diversos usuarios.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Para el éxito de esta herramienta, es importante que los usuarios tengan autodisciplina y compromiso. La herramienta sirve como un medio para llevar un control y para generar recordatorios, pero el trabajo depende de los usuarios.
- Esta herramienta es una alternativa de comunicación en un grupo de trabajo, pero quien lidera el grupo debe hacer un seguimiento continuo del avance de las tareas y estar en constante comunicación con sus delegados.
- Usar ASP.NET ha permitido ahorrar más del 50% del tiempo que hubiera tomado desarrollar esta herramienta usando otras tecnologías como Java o PHP, esto gracias a los múltiples controles de usuario de Visual Studio 2005 que simplifican muchas tareas, permitiendo enfocar el tiempo en generar ideas.
- Aplicar conceptos de arquitectura para el desarrollo resultó beneficioso al momento de hacer las modificaciones que resultaron de las observaciones hechas por los usuarios durante las pruebas del sistema.
- La retroalimentación que los usuarios brindaron tras las pruebas del sistema ayudaron a agregar nuevas ideas de valor, la cuales durante la etapa de desarrollo del proyecto no fueron concebidas.

Recomendaciones

- Previo a la utilización de esta herramienta se recomienda establecer una reunión entre el líder del grupo o área y sus colaboradores para definir los lineamientos que regirán en el uso de la misma, especificando cada qué tiempo es necesario se actualicen los comentarios de las tareas, cada qué tiempo se analizarán las métricas, cuánto tiempo pueden tomarse para definir una fecha de cumplimiento de una tarea, entre otros temas.
- Para la instalación de la herramienta, se recomienda crear una interfaz que se ejecute una sola vez para configurar los parámetros del proyecto. Esta opción se suele conocer como de ejecución única. Se debe poder definir el tipo de autenticación de usuarios, la dirección LDAP para la autenticación de usuarios si es que aplica, el nombre del servidor en el cual está alojada la base de datos, el nombre de la base de datos y los parámetros de conexión a la base.
- Una alternativa que puede brindar mucho valor agregado es crear una interfaz WAP para actualizar las tareas desde dispositivos móviles. Aquellas personas que no pasan mucho tiempo en la oficina pero tienen dispositivos móviles con acceso a internet y navegación Web/WAP podrían usar esta opción acceder a la herramienta.

BIBLIOGRAFÍA

- [1] ASP.NET AJAX, <http://www.asp.net/ajax/>
- [2] Guía de ASP.NET, <http://quickstarts.asp.net/QuickStartv20/aspnet/doc/whatsnew.aspx>
- [3] Cambios en ASP.NET 2.0, [http://msdn2.microsoft.com/es-es/library/aa479401\(en-us\).aspx](http://msdn2.microsoft.com/es-es/library/aa479401(en-us).aspx)
- [4] Información general acerca de las características de Windows .NET Framework 2.0, http://www.microsoft.com/spanish/msdn/netframework/framework20_InformacionCaract.msp
- [5] Introducción a Genéricos en .NET, <http://www.15seconds.com/issue/031024.htm>
- [6] Web 2.0, http://en.wikipedia.org/wiki/Web_2
- [7] XHTML, <http://es.wikipedia.org/wiki/XHTML>
- [8] CSS, http://es.wikipedia.org/wiki/Cascading_Style_Sheets
- [9] AJAX, <http://es.wikipedia.org/wiki/AJAX>
- [10] Descripción general de Microsoft Office Project Standard 2007, <http://office.microsoft.com/es-es/project/HA101656383082.aspx>
- [11] KMKey Software de gestión de proyectos, http://www.kmkey.com/productos/kmkey_project
- [12] TaskBin Administrador de Tareas, <http://www.taskbin.com>
- [13] B-kin Project software de gestión de proyectos, www.b-kin.com

- [14]** Métricas, Estimación y Planificación en Proyectos de Software, http://www.willydev.net/Descargas/WillyDEV_PlaneaSoftware.Pdf
- [15]** Desarrollo de Métricas de Rendimiento, <http://www.orau.gov/pbm/documents/overview/uc.html>
- [16]** Fleishman, E. A. & others. Leadership and supervision in industry. Columbus: Ohio State University, Bureau of Educational Research
- [17]** Control Definicion E Importancia, <http://www.mitecnologico.com/Main/ControlDefinicionEImportancia>
- [18]** Control, <http://www.monografias.com/trabajos14/control/control.shtml>
- [19]** Desarrollo organizacional y administración del tiempo, <http://www.gestiopolis.com/canales6/ger/desadtiempo.htm>
- [20]** La Delegación de Tareas como Herramienta de Crecimiento, <http://www.webnova.com.ar/articulo.php?recurso=313>
- [21]** La comunicación en el trabajo, <http://www.gestiopolis.com/canales5/comerciohispano/63.htm>
- [22]** Tutorial de Acceso a Datos, <http://www.asp.net/learn/data-access/>
- [23]** Elemento membership (Esquema de configuración de ASP.NET), [http://msdn.microsoft.com/es-es/library/1b9hw62f\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/1b9hw62f(VS.80).aspx)
- [24]** Elemento connectionStrings (Esquema de configuración de ASP.NET), [http://msdn.microsoft.com/es-es/library/bf7sd233\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/bf7sd233(VS.80).aspx)

[25] Cifrar y descifrar secciones de configuración,

<http://msdn.microsoft.com/es-co/library/zhhddkxy.aspx>

[26] Autenticación de Active Directory desde ASP .NET,

[http://msdn.microsoft.com/es-es/library/ms180890\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/ms180890(VS.80).aspx)

ANEXOS

ANEXO A

INSTRUCTIVO DE USO DE LA HERRAMIENTA PARA LOS USUARIOS

Objetivo

El Objetivo de este manual es explicar como usar la sección Hoja de Trabajo del sistema para gestión de tareas y que el usuario entienda como puede administrar sus actividades a través de la herramienta.

Contenido

Pantalla Inicial

Cuando un usuario ingresa a la herramienta, a su derecha va a encontrar un grafico de tipo pastel, el cual le mostrará en resumen, cuantas tareas tiene por cada uno de los estados posibles de una tarea para el mes en curso. Ver Figura 1.

Para ingresar a la hoja de trabajo, debe dar clic en la imagen de la hoja de trabajo, que se muestra en la Figura 1.

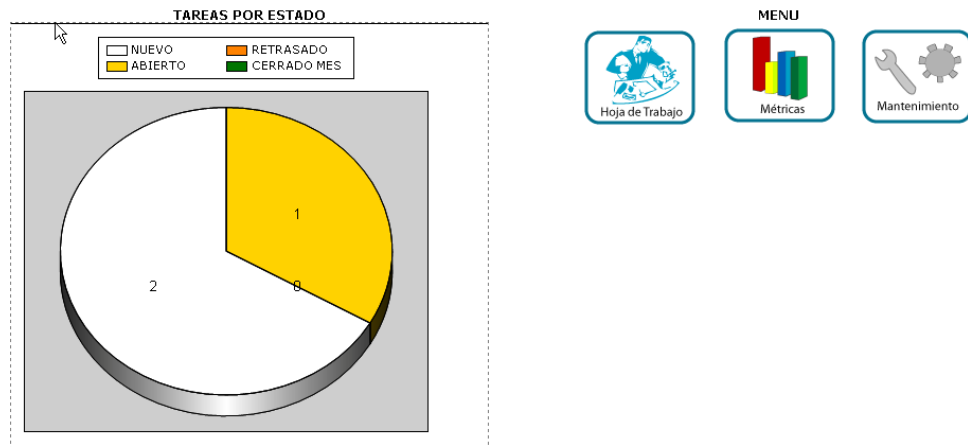


Figura 1. Pantalla inicial de la herramienta

Hoja de Trabajo

Al ingresar a la hoja de trabajo, en la parte superior de la pantalla se muestra la lista de Usuarios y de Áreas. Las tareas se muestran de acuerdo al área y al usuario seleccionado. La Figura 2 muestra la pantalla de la Hoja de Trabajo.

VER TAREAS DE										
AREA:		Equipo Calificación de Personal	USUARIO:		William Murillo	<input type="button" value="Nueva Tarea"/> <input checked="" type="checkbox"/> Reportes y Métricas <input type="button" value="Inicio"/>				
FILTROS										
VER TAREAS:						Activas	PRIORIDAD:	Todas	TEMA:	Todos
#	Cod I	Descripción	Prio	Creado	Fecha Fin	Estado	Comentarios	Menu		
1	1	Necesitamos que se cree una carpeta en el disco I para almacenar la información de los entrenamientos	1	06/11/2008 14:25:32		Nuevo	<< Ver Todos >>			
2	2	Se necesita ver la posibilidad de que a través de cognos se muestre la información del cumplimiento de los entrenamientos por empleado	1	06/11/2008 14:30:16		Nuevo	<< Ver Todos >>			

Figura 2. Pantalla de la Hoja de Trabajo

Luego están los filtros, las tareas se pueden filtrar de acuerdo a su estado, prioridad o tema. Para esto hay tres combos, uno con el nombre VER TAREAS, el cual permite ver las tareas según su estado, las opciones son:

- Activas: Incluye las tareas en estado: Nuevo, Abierto, Retrasado
- Cerradas: Todas las tareas en estado: Cerrado
- Todas: Incluye todas las tareas independiente de su estado

El siguiente combo, con el nombre PRIORIDAD, permite filtrar las tareas según su prioridad. El combo muestra todas las combinaciones posibles entre todas las prioridades.

El último combo, con el nombre TEMA, permite filtrar las tareas por uno de los temas que ha creado el usuario.

La Figura 3 muestra los 3 filtros nombrados en los párrafos superiores.



The image shows a header bar with the word 'FILTROS' in white text on a blue background. Below the header, there are three dropdown menus. The first is labeled 'VER TAREAS:' and has 'Activas' selected. The second is labeled 'PRIORIDAD:' and has 'Todas' selected. The third is labeled 'TEMA:' and has 'Todos' selected. Each dropdown menu has a small downward-pointing arrow on its right side.

Figura 3. Filtros para las tareas en la Hoja de Trabajo

La otra opción que tienen los usuarios es ordenar las tareas. Para esto deben dar clic en la cabecera del campo por la cual deseen ordenar. Al dar clic se mostrará una flecha que indica el orden en que se ha ordenado el campo. Ver Figura 4.



The image shows a table with three columns. The first column is labeled '#', the second is labeled 'Cod ↓', and the third is labeled 'Descripción'. The 'Cod ↓' header has a small downward-pointing arrow next to it, indicating that the table is sorted by this column in descending order.

Figura 4. Flecha que indica ordenamiento del campo Cod.

Los únicos campos por los cuales se puede ordenar las tareas son:

- Código
- Prioridad
- Fecha de Creación
- Fecha de Cierre (Fecha Fin)
- Estado

La lista de tareas se compone de 10 campos (Ver Figura 5), 8 campos que tienen los datos de las tareas, más un primer campo que muestra el número de tarea y un último campo que muestra un menú con varias opciones.

#	Cod	Descripción	Prio	Creado	Fecha Fin	Estado	Comentarios	Menu
1	1	Necesitamos que se cree una carpeta en el disco I para almacenar la información de los entrenamientos	1	06/11/2008 14:25:32		Nuevo	<< Ver Todos >>	

Figura 5. Campos de la lista de tareas.

Existen campos con características especiales, es decir que no solo muestran texto. A continuación se explica cada una de ellas:

El campo de Descripción permite editar su contenido al dar clic en un lápiz que aparece en la esquina superior derecha de la celda. Al dar clic en el lápiz, se abre una ventana en la cual se muestra la descripción de la tarea en cuadro de texto para poder modificarla.

El campo de Prioridad presenta un combo en el que están enlistadas las prioridades posibles para una tarea y aparece seleccionada la que está

asignada para cada tarea. Al seleccionar otra prioridad automáticamente se le asigna esta nueva prioridad a la tarea.

El campo de fecha de cierre se compone de 3 controles, un caja de texto para ingresar la nueva fecha, al dar clic en la caja se activa un calendario para escoger la fecha. El segundo control es un botón, al dar clic a este botón se procede a grabar la nueva fecha de cierre. El tercer y último control es un List Box, en el cual salen listadas las fechas de cierre que se han ido asignando a la tarea. Cuando se agrega una nueva fecha de cierre, el sistema valida que la fecha no sea menor a la fecha actual ni menor a una fecha ya agregada. Además, por cada nueva fecha de cierre se solicita que se ingrese un comentario que justifique el cambio de la fecha.

El campo de Estado presenta un combo en el que están enlistados los posibles estados por cada tarea. El estado de una tarea se maneja automáticamente por el sistema usando las siguientes condiciones:

- Cuando una tarea se crea recibe el estado Nuevo hasta que se le asigne la primera fecha de cierre. Esta celda tiene un fondo blanco cuando la tarea está en este estado.
- Al recibir su primera fecha de cierre, una tarea recibe el estado Abierto. Esta celda tiene un fondo amarillo cuando la tarea está en este estado.
- Cuando se agrega una fecha adicional de cierre o cuando ha pasado su fecha de cierre y no se ha dado por terminada la tarea, se le asigna

el estado de Retrasado. La celda tiene un fondo naranja cuando la tarea está en este estado.

- El único estado que se puede asignar manualmente es el estado Cerrado. Esto se puede hacer cuando la tarea está en estado Abierto o Retrasado. Al asignar el estado Cerrado, la celda toma un fondo color verde y sale de la lista de tareas activas, se mueve a tareas cerradas.

El campo Comentarios presenta los 3 últimos comentarios que ha recibido la tarea, precedido de la fecha y hora en que se ingresó el comentario más el nombre de usuario de quien lo ingresó. Para ver el resto de los comentarios, hay un enlace en la parte inferior izquierda de la celda con el nombre <<Ver Todos>>, al dar clic a este link se abre una nueva ventana en la que aparecen todos los comentarios. Para agregar un nuevo comentario, se debe dar clic en un lápiz que se encuentra en la parte superior derecha de la celda. Al dar clic se abre una nueva ventana en la cual aparece un cuadro de texto para agregar el mensaje y debajo de este cuadro de texto los demás mensajes que se han ingresado. Cuando el sistema requiere que se ingrese un comentario a una tarea, la celda de comentarios aparece con fondo rojo.

El sistema puede requerir un comentario por dos motivos:

- Por una nueva fecha de cierre asignada a la tarea
- Porque ha pasado el tiempo límite que el administrador del sistema permite que una tarea esté sin comentarios.

El último campo de la lista de tareas presenta un menú (Ver Figura 6), su elemento raíz es una imagen en forma de flecha (↵). El menú despliega las siguientes opciones:

- Eliminar: Al dar clic en esta opción se procede a eliminar la tarea tras la confirmación del usuario.
- Ver Historial: Esta opción abre una ventana en la cual se muestra la historia de lo que ha sucedido con la tarea, desde el momento que se la creó, hasta cada comentario ingresado o archivo agregado.
- Ver Archivos: Al escoger esta opción, en una ventana adicional se muestran los archivos adjuntos que tiene cada tarea.
- Asignar Tema: Esta opción muestra una ventana en la cual aparecen los temas que ha creado cada usuario, de manera que pueda asignar uno de los temas a la tarea.

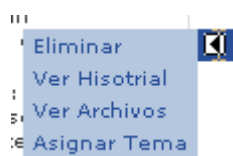


Figura 6. Opciones de menú por tarea