



UNIVERSIDAD DEL BÍO-BÍO

FACULTAD DE CIENCIAS EMPRESARIALES

INGENIERÍA CIVIL INFORMÁTICA

“Plataforma web para el Control y gestión de Proyecto de la Dirección de Extensión de la universidad del Bío-Bío”

Taller de Desarrollo de Proyecto para Optar al Título de Ingeniero Civil en Informática.

INTEGRANTE

Ramón Villagrán Azocar

PROFESOR GUÍA

Sergio Bravo Silva

CONCEPCION, febrero de 2013

Introducción

Los sistemas informáticos ya se han implantado en nuestra sociedad actual , sin embargo cada día los sistemas son más específicos en sus requerimientos, ya no solo basta con registrar a funcionarios , si no que este debe tener un plus mas con el fin de por otorgar mas funcionalidades al funcionarios, claro!, esto sin dejar de lado el hecho que los sistemas sean amigables e intuitivos ,esto quiere decir que sean fácil de usar y que su respuesta a nuestras consultas sean rápidas.

Esto es lo que se basa el presente informe de título en un proyecto que sea capaz aparte de registrar funcionarios tenga el plus de poder llevar el control y gestionar cada etapa de un proyecto. Además de tener la capacidad de registrar eventos y que sea capaz de validar cada uno de los campos , con el fin que sea un sistema seguro y estable, para que el usuario final tenga plena confianza.

A continuación se describen los capítulos de nuestro informe.

Capitulo1: Control y Gestión de de Procesos Administrativos.

Capítulos 2: Descripción de la organización.

Capitulo 3: DESCRIPCION DE LA PROBLEMÁTICA ACTUAL.

Capitulo4:DESCRIPCIÓN EL PROYECTO.

Capitulo 5 REQUERIMIENTOS..

Capitulo 6:Herramientas Utilizadas.

Capitulo 7:Factibilidades.

Capitulo 8: Analisis

Capitulo9: Diseño del sistema

Capitulo10: Pruebas y respaldos

CAPITULO I

Control y Gestión de de Procesos Administrativos.

1.1 Definición De Control

El control es la función administrativa por medio de la cual se evalúa el rendimiento. Para Robbins (1996) el control puede definirse como "el proceso de regular actividades que aseguren que se están cumpliendo como fueron planificadas y corrigiendo cualquier desviación significativa".

Sin embargo Stoner (1996) lo define de la siguiente manera: "El control administrativo es el proceso que permite garantizar que las actividades reales se ajusten a las actividades proyectadas".

Mientras que para Fayol, citado por Melinkoff (1990), el control , "Consiste en verificar si todo se realiza conforme al programa adoptado, a las órdenes impartidas y a los principios administrativo. Tiene la finalidad de señalar las faltas y los errores a fin de que se pueda repararlos y evitar su repetición".

Analizando todas las definiciones citadas notamos que el control posee ciertos elementos que son básicos o esenciales:

- En primer lugar, se debe llevar a cabo un proceso de supervisión de las actividades realizadas.
- En segundo lugar, deben existir estándares o patrones establecidos para determinar posibles desviaciones de los resultados.
- En un tercer lugar, el control permite la corrección de errores, de posibles desviaciones en los resultados o en las actividades realizadas.
- Y en último lugar, a través del proceso de control se debe planificar las actividades y objetivos a realizar, después de haber hecho las correcciones necesarias. Gráficamente, el proceso de control pueda adoptar la forma circular identificando la acción continua entre sus fases y de retroalimentación permanente:

En conclusión podemos definir el control como la función que permite la supervisión y comparación de los resultados obtenidos contra los resultados esperados originalmente,

asegurando además que la acción dirigida se esté llevando a cabo de acuerdo con los planes de la organización y dentro de los límites de la estructura organizacional.

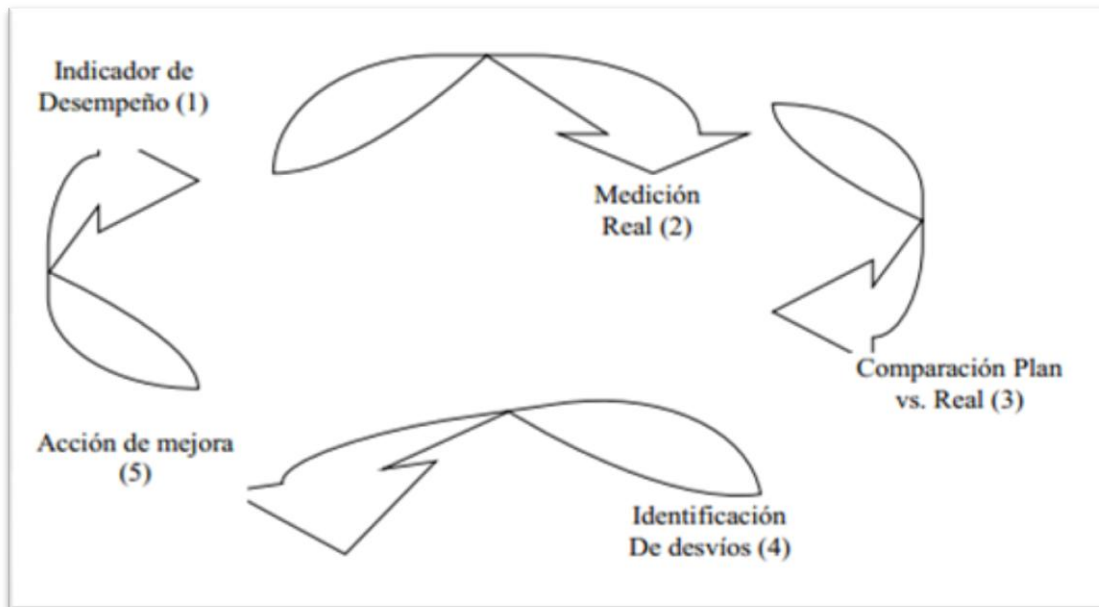


Figura 1. Ciclo de control administrativo.

1.2 Importancia Del Control Dentro Del Proceso Administrativo

El control se enfoca en evaluar y corregir el desempeño de las actividades de los subordinados para asegurar que los objetivos y planes de la organización se están llevando a cabo. De aquí puede deducirse la gran importancia que tiene el control, pues es solo a través de esta función que lograremos precisar si lo realizado se ajusta a lo planeado y en caso de existir desviaciones, identificar los responsables y corregir dichos errores.

Sin embargo es conveniente recordar que no debe existir solo el control a posteriori, sino que, al igual que el planteamiento, debe ser, por lo menos en parte, una labor de previsión. En este caso se puede estudiar el pasado para determinar lo que ha ocurrido y porque los estándares no han sido alcanzados; de esta manera se puede adoptar las medidas necesarias para que en el futuro no se cometan los errores del pasado.

Además siendo el control la última de las funciones del proceso administrativo, esta cierra el ciclo del sistema al proveer retroalimentación respecto a desviaciones significativas contra el

desempeño planeado. La retroalimentación de información pertinente a partir de la función de control puede afectar el proceso de planeación.

1.3 Tipos De Control

Terry (1999) en su libro "Principios de Administración" expone que existen 3 tipos de control que son: El control preliminar, el concurrente y el de retroalimentación.

- Control preliminar, este tipo de control tiene lugar antes de que comiencen las operaciones e incluye la creación de políticas, procedimientos y reglas diseñadas para asegurar que las actividades planeadas serán ejecutadas con propiedad. En vez de esperar los resultados y compararlos con los objetivos es posible ejercer una influencia controladora limitando las actividades por adelantado.

Son deseables debido a que permiten a la administración evitar problemas en lugar de tener que corregirlos después, pero desafortunadamente este tipo de control requiere tiempo e información oportuna y precisa que suele ser difícil de desarrollar.

Por ejemplo, un gerente de ventas de una determinada tienda puede tener la política de que todo cambio en el precio, respecto a los precios publicados, debe ser autorizados por escrito por el gerente, es decir, a ningún vendedor de campo se le permite que altere algún precio. Con esto se puede observar que el gerente de ventas lleva un control en su departamento a través de las políticas existentes, cuyos empleados deben cumplir para un mayor funcionamiento del mismo.

- Control concurrente, este tipo de control tiene lugar durante la fase de la acción de ejecutar los planes e incluye la dirección, vigilancia y sincronización de las actividades según ocurran, en otras palabras, pueden ayudar a garantizar que el plan será llevado a cabo en el tiempo específico y bajo las condiciones requeridas.

La forma mejor conocida del control concurrente es la supervisión directa. Cuando un administrador supervisa las acciones de un empleado de manera directa, el administrador puede verificar de forma concurrente las actividades del empleado y corregir los problemas que puedan presentarse.

Por ejemplo, la mayor parte de los computadores están programados para ofrecer a los operadores respuestas inmediatas si se presenta algún error. Si se introduce un comando equivocado, los controles del programa rechazan el comando y todavía así pueden indicarle por qué es el error.

- Control de retroalimentación, este tipo de control se enfoca sobre el uso de la información de los resultados anteriores para corregir posibles desviaciones futuras de estándar aceptable. El control de retroalimentación implica que se han reunido algunos datos, se han analizado y se han regresado los resultados a alguien o a algo en el proceso que se está controlando de manera que puedan hacerse correcciones. El principal inconveniente de este tipo de control es que en el momento en que el administrador tiene la información el daño ya está hecho, es decir, se lleva a cabo después de la acción. Por ejemplo, se tiene una empresa que tiene 3 sucursales distribuidas por todo el país: Sucursal A, Sucursal B y Sucursal C. El gerente general ha detectado que la sucursal A tiene serios problemas financieros, mientras que sus otras dos sucursales están funcionando correctamente. Es aquí cuando el gerente debe decidir si esta información es causa suficiente para cerrar dicha sucursal o deberá cambiar las estrategias que han venido implementando.

1.4 CONTROL DE GESTIÓN

Este tipo de control ha sido incorporado al lenguaje usual de la administración a partir de la década del 60. Los cambios producidos en el contexto, han exigido esfuerzos de adaptación y de comportamiento a los directivos de las organizaciones tendientes a incrementar su capacidad competitiva; estos cambios también se trasladaron a la concepción de los sistemas de control, y de allí la incorporación del control de gestión.

Con una mirada retrospectiva se puede advertir que en los sistemas de control tradicionales no se incluían indicadores no financieros, internos y externos, de corto y largo plazo para poder medir los aspectos cualitativos de la gestión.

La nueva concepción del control, propone dar énfasis a los aspectos cualitativos y no contables (exclusivamente cuantitativos) del control, sumando también, aspectos motivacionales y culturales.

Esta visión resulta de la necesaria adaptación de las herramientas de gestión al uso por parte de organizaciones que viven en entornos emergentes e imprevisibles, donde indefectiblemente se deben sumar a las variables de control tradicionales (contables y operativas), otras más informales, para abarcar así todos los procesos. De esta manera, el control de gestión pasa a ser clave en el desarrollo del proceso administrativo, pues adquiere alta relevancia en entornos cambiantes, convirtiéndose en la brújula de los directivos que ayuda a diagnosticar dónde estamos, definir el rumbo, comunicarlo para alinear los comportamientos de la organización y monitorear posibles desvíos.

El control de gestión se puede definir entonces como un sistema de información que permite al nivel directivo efectuar una revisión crítica, cuanti y cualitativa, del planeamiento, determinando el grado en que se logran los distintos planes para permitir realizar los ajustes y corregir los desvíos producidos pasando por todos los niveles de la organización.

En síntesis, debe entenderse que el control de gestión:

- Es un medio para desplegar la estrategia en toda la organización
- Desarrolla actividades de diagnóstico, planificación y evaluación
- Sirve para evaluar el desempeño de la organización, entendida como la medición y Análisis de los resultados, desde múltiples ángulos o criterios, para decidir qué acción tomar a partir de los recursos disponibles, con una orientación hacia su mejora permanente en todos los niveles de la organización.
- Es un medio para movilizar el talento y la energía del colectivo hacia el logro de los objetivos de la organización.
- Es un medio para gestionar el cambio

CAPITULO II

DESCRIPCION DE LA ORGANIZACIÓN.

2.1 Un poco de historia.

La Universidad del Bío-Bío es heredera de la más antigua tradición de la educación superior estatal y pública en la Región del Bío Bío. Sus orígenes se remontan a la creación de la Universidad Técnica del Estado, UTE, el 9 de abril de 1947, bajo la presidencia de Gabriel González Videla.

Dependiente del Ministerio de Educación Pública, la nueva institución fusionó en su interior la Escuela de Ingenieros Industriales y los grados técnicos de la Escuela de Artes y Oficios de Santiago; Escuela de Minas de Antofagasta, Copiapó y La Serena y las Escuelas Industriales de Concepción, Temuco y Valdivia.

La UTE abrió oficialmente sus puertas en 1952, luego que el Senado aprobó su Estatuto Orgánico, contrariando la férrea oposición de la Universidad de Chile, cuyas autoridades consideraban que la nueva casa de estudios superiores debía funcionar bajo su tuición. En Concepción, el plantel jugaría un importante papel para responder a las necesidades y desafíos que plantea la Región como uno de los polos del desarrollo industrial del país, no sólo a través de la docencia de pregrado sino que también mediante la investigación científica y tecnológica.

A las carreras técnicas de Electricidad, Mecánica y Textil que se impartían en 1959 se sumaron, en 1969, las de Ingeniería de Ejecución en Electricidad, Mecánica y Madera. Ese mismo año se creó la carrera de Arquitectura, hecho que constituye un hito significativo: De acuerdo con la nueva Ley de Universidades dictada en 1980, la Sede Concepción de la UTE pasó a ser una universidad autónoma –la Universidad de Bío-Bío- por impartir Arquitectura, definida entonces como una de las 12 carreras universitarias.

No corrió igual suerte la Universidad de Chile, Sede Ñuble, que venía funcionando en Chillán desde 1966. La nueva legislación, que obligó a convertir en universidades a las sedes regionales, de acuerdo con el concepto de regionalización y desconcentración, dio lugar al

surgimiento del Instituto Profesional de Chillán, IPROCH, ya que entre sus carreras no figuraba ninguna de las consideradas universitarias.

La creación de la Sede Ñuble de la Universidad de Chile, primero como Colegio Regional, había sido el fruto de un amplio movimiento ciudadano destinado a evitar que los jóvenes egresados de la Enseñanza Media tuvieran que emigrar a Santiago u otras ciudades para continuar la enseñanza superior. En sus inicios, ocupó las dependencias cedidas por la Sociedad Musical Santa Cecilia, además de un edificio en avenida Libertad, donde funcionó la Escuela de Idiomas. Posteriormente, en 1973, recibió la donación de 33 hectáreas del fundo El Mono, propiedad de Fernando May Didier, comenzando la construcción del actual Campus Fernando May. El año 1981, en tanto, el patrimonio del naciente IPROCH se incrementó con la incorporación de las antiguas instalaciones de la Escuela Normal de Chillán donde ahora se encuentra el Campus La Castilla.

Más tarde, en 1988, la fusión de la Universidad de Bío Bío y el Instituto Profesional de Chillán dio origen a la que es hoy la Universidad del Bío-Bío, uniendo a dos instituciones que asumieron el desafío de caminar juntas y construir una historia en común.

2.2 Área de Desarrollo.

El área de desarrollo será centrada en la *Unidad de Extensión* de la Universidad del Bio-Bio, a continuación se describe en detalle en qué consiste, su función y organigrama de esta unidad dentro de la universidad.

2.2.1 Función

De acuerdo con el PGDU, la extensión constituye parte de la misión institucional, la que deberá articularse con la Docencia y la Investigación, partiendo de las fortalezas propias de la Universidad, esta actividad ofrece desafíos y oportunidades extraordinarias para que la Universidad se inserte en la vida social y económica del país, por lo que es necesario contar con un diagnóstico preciso de esas necesidades y requerimientos de las comunidades e instituciones de desarrollo, para así articular de acuerdo a nuestras fortalezas y potencialidades los programas y proyectos académicos de acuerdo a esas exigencias.

En este contexto, podemos definir la actividad de extensión universitaria como aquella función propia de la educación superior, indeclinable y de servicio público, que tiene como propósito divulgar los logros alcanzados por la institución como resultado de su vinculación e intervención en los procesos de desarrollo social, contribuyendo así a la consolidación de la misión y visión institucional. Asimismo fomenta actividades complementarias en lo cultural, deportivo, artístico y recreativo con el propósito de aportar soluciones de desarrollo social y potenciar el talento humano y el uso adecuado de los recursos en la permanente búsqueda del bienestar colectivo de la comunidad en la que interviene.

2.2.2 Objetivos de Unidad de Extensión.

- Orientar y gestionar las acciones y programas de extensión universitaria tendientes a establecer relaciones interactivas en la comunidad universitaria y con los sectores sociales, económicos, profesionales e institucionales de la región y el país.
- Garantizar que sus actividades, en especial aquellas denominadas de extensión relevante se enmarquen en los principios que orientan las demás acciones universitarias. En este sentido, los estándares de calidad y excelencia académica estarán presentes de tal manera que se incorporen a las acciones de extensión los más altos niveles de conocimiento.
- Fomentar actividades complementarias en lo cultural, deportivo, artístico y recreativo con el propósito de aportar soluciones de desarrollo social, potenciando el talento humano y el uso adecuado de los recursos en una permanente búsqueda del bienestar colectivo de la comunidad en la que interviene.
- Estimular el espíritu y la vocación de servicio social, promoviendo la participación de todos los actores de la comunidad universitaria, invitándolos a involucrarse en el desarrollo y puesta en marcha de los diferentes programas de extensión.

2.2.3 Organigrama

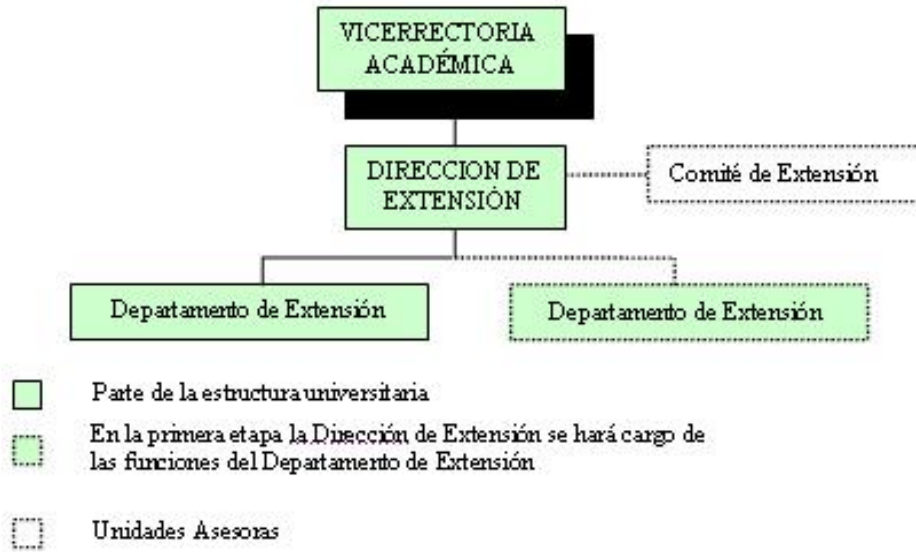


Figura 2.

CAPITULO III

DESCRIPCION DE LA PROBLEMÁTICA ACTUAL.

3.1 Origen del Tema

El tema nace ante la necesidad de la Universidad del Bío Bío ,de no tener un sistema de apoyo a la Planificación y Control de Proyectos, que incluya la posibilidad de visualizar a los proyectos a través de una página WEB, describiendo cada una de sus etapas, los diferentes ítems que se presentan en cada una de estas y sus respectivos eventos en los que se da cuenta a la comunidad de los logros de nuestra Universidad. La Dirección de Extensión de la Universidad, en conversaciones con el académico de la FACE Sergio Bravo Silva, ha manifestado gran interés en contar con un sistema como el mencionado. Por dicha razón se integrará al equipo de desarrollo

3.2 Descripción del área de Estudio.

El área de estudio cómo se menciona anteriormente, es la Dirección de Extensión. Debido a que esta es la encargada de ver, asignar, controlar y gestionar .La mayoría de los proyectos y eventos que se desarrollan dentro de nuestra casa de estudio.

3.3 Descripción situación actual

En la actualidad la Dirección y Extensión, lleva la responsabilidad de gestionar diferentes proyectos, cada uno de estos dirigidos por diferentes docentes de las diferentes facultades dentro de la universidad.

Este proceso comienza cuando las postulación a los concursos a proyectos anuales y en ocasiones semestrales, se abren o son publicadas en la página de Extensión, por medio de un formulario el cual los docentes deben llenar los datos requeridos para la postulación de estos. El siguiente paso es enviar, la información requerida para su posterior evaluación, para optar a un monto para desarrollar el o los proyectos.

Durante esta etapa la dirección de extensión procura un análisis a cada una de las propuestas, no solamente clasificando o evaluando la propuesta por el monto de dinero, el contenido, la diversidad del proyecto, el impacto dentro de la comunidad universitaria o el aporte educacional del posible proyecto a realizar, sino, que también deben analizar a cada uno de sus participantes que en muchas ocasiones o casi siempre son docentes de nuestra universidad, para cerciorarse si van ser capaces de llevar su carga académica y además el proyecto en juego, cabe destacar que un académico puede tener más de un proyecto. Esto se hace a través de los jefes de departamento, consultando sus horarios y respectivas clases. Por esta razón es fundamental el análisis, para prevenir cualquier sobre carga al docente y que este no sea capaz de cumplir con alguna de sus obligaciones ya sea como docente o como participante de un proyecto. Finalmente cuando se ha visto todos los factores y analizado cada uno de esto se asigna un puntaje, para poder aprobar, rechazar o en ocasiones aprobar con correcciones los diferentes proyectos, para que en una posible segunda vuelta sean aprobados.

En la siguiente etapa, es cuando los proyectos ya fueron aprobados, de aquí se nombra por parte de extensión a un encargado, el cual tendrá el rol, de ser supervisor de los diferentes proyectos que se van desarrollando durante el año escolar. Este lleva una carpeta física de todos los proyectos, de modo que a medida que pase el año, van ocurriendo pequeñas reuniones con cada uno de los representantes para ver el estado del proyecto, según su programación la maya de tareas y si han invertido lo justificado en sus respectivos gastos.

Luego que el supervisor reúne un cumulo de información importante este, se reúne con el jefe de extensión, para dar a conocer los diferentes estado de los proyectos, si van cumpliendo las etapas o lamentablemente en ocasiones el termino anticipado del proyecto, ya sea, por enfermedad del jefe del proyecto o otras causas que muchas veces escapan del manejo de los jefes de proyectos.

Finalmente cuando los proyectos son finalizados, muchas veces estos se vuelven eventos ya sea dentro de nuestra comunidad universitaria dando a conocer los hallazgos obtenidos, logros alcanzados o en forma de congresos, exponiendo sobre nuevas tendencias en áreas de investigación, líneas de negocios, administración, tecnologías, etc. o en el exterior por medio de publicaciones en revistas, radio o televisión, dando a conocer la información de manera local, nacional o internacional.



3.3.1 Ciclo de Concurso Proyecto.

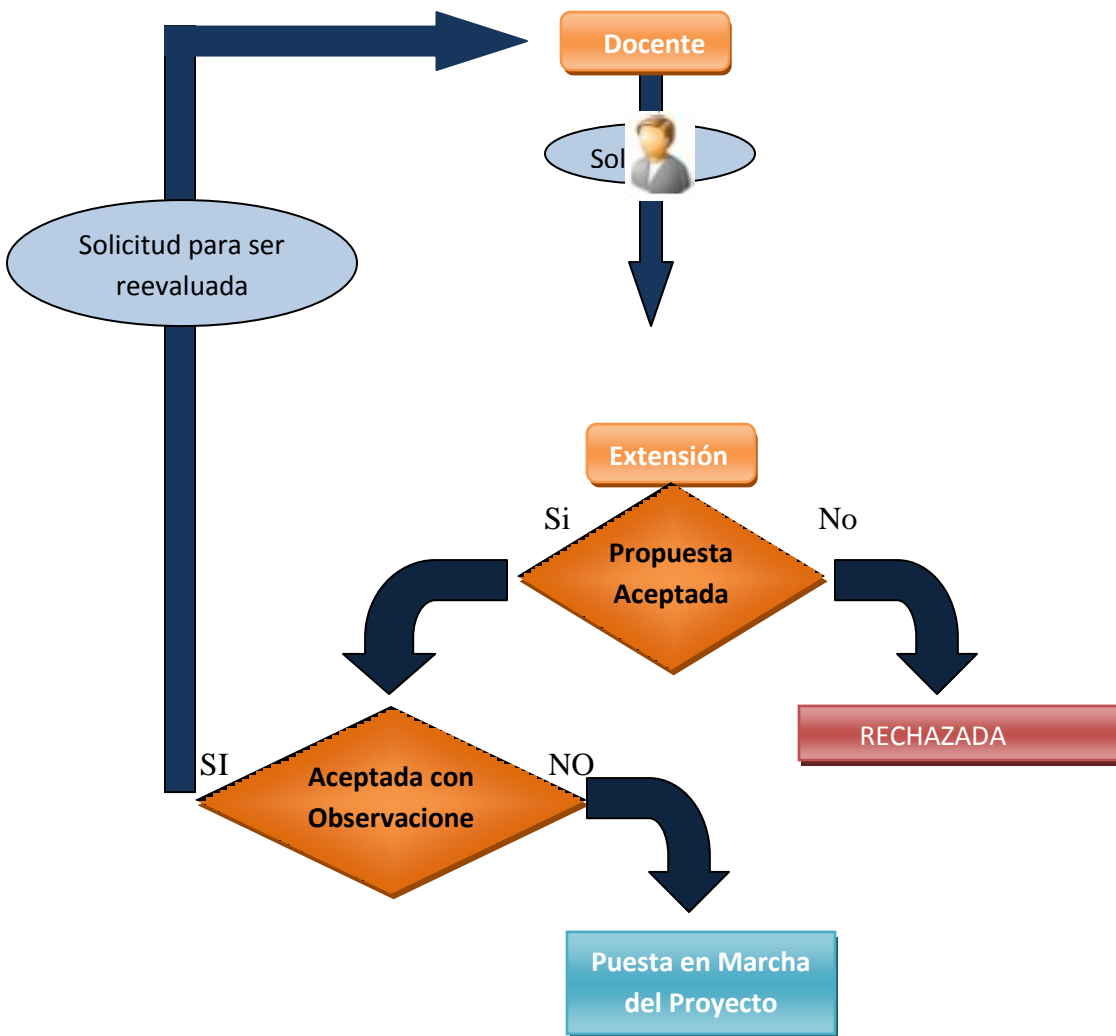


Figura 2 .Ciclo de concurso a proyecto por extensión.

3.3 Problemática de la situación actual.

Para presentar la problemática debemos enfocarnos , principalmente en la etapa número 2 del desarrollo de un proyecto , con más exactitud no estamos refiriendo , al momento en que el proyecto está evaluado y aprobado sin ninguna observación pendiente. Una vez que el supervisor ha sido nombrado por parte de extensión ,es este ,como se menciona anteriormente ,él que debe velar por cada uno de los proyectos , siendo en ocasiones una tarea casi imposible , pues llevar un análisis en detalle y actualizado de cada uno de los proyectos , considerando que muchas veces estos llegan a números como 70 o más proyectos rodando.

Éste debe analizar de cada una de las tareas que se expusieron en la entrega de de la solicitud del proyecto, el supervisor con el fin de cumplir con su misión debe cuestionarse lo siguiente ¿cómo se han ido ejecutando? , ¿Se cumplieron los tiempos estimados? ,¿!Fracasaron!? ,¿ La tareas están atrasadas? , ¿ Los montos asignados a cada tarea son congruentes con los plasmados en los documentos oficiales? , ¿Los montos de la tareas realizadas sobrepasan el presupuesto total del proyecto? , si es así, ¿Por qué? , en fin un sin número de detalles que dificultan mucho el trabajo y si aun mas le sumamos a esto la cantidad de proyectos por la cantidad de personas con que debe entrevistarse con el fin de obtener esta información y más aun que sea actualizada, nos encontramos con una labor heroica por parte del supervisor.

La otra gran Problemática que aqueja , es en la siguiente etapa ,la publicación del proyecto o como le hemos llamado el evento .En muchas ocasiones en nuestra universidad , nos vemos abrumados por la cantidad de eventos atractivos que se realizan y al no tener un calendario visible para los realizadores de estos eventos muchas veces se topan con otros o simplemente dejan sin audiencia algunos por no publicar a tiempo estos eventos. Sabemos que actualmente existe un sistema para reservar los lugares de publicación como salas de aula magna , paraninfo, etc..Sin embargo esto no basta ,pues el constante crecimiento de eventos dentro de

nuestra casa de estudio , ha dificultado mucho este sistema , ya que se podría decir que es prácticamente un registro a mano y muchas veces falla.

3.4 Problemas resumidos.

A continuación se presentaran en forma de punto cada uno de los problemas expuestos anteriormente con el fin de divisar y exponer de una forma más sencilla , los problemas.

- Flujo lento de retroalimentación para el supervisor.
- Posibles pérdidas en la documentación del supervisor.
- Carpeta física de proyectos.
- Problemas para reunirse con los encargados de proyectos por temas de horarios.
- Perdida de la noción de las etapas de los proyectos.
- Al no llevar un detalle , se pasan del presupuesto aprobado.
- Choques de eventos , dentro de la universidad.
- Poco público en eventos por falta de información de los organizadores.

3.5 Alternativas de solución

En el campo de la administración de proyectos son variadas las alternativas , para llevar un análisis de estos , sin embargo, esto siempre y cuando se hable solo de 1 proyecto, a continuación expondremos 3 soluciones para ser implementadas por este problemas dando ventajas de desventajas de cada una de ellas.

3.5.1 Sistema de escritorio.

Esta solución plantea usar la herramienta de Microsoft Project, este permite generar un acta Gantt , llevando en detalle cada una de las tareas el tiempo que esta tendrá, la sucesión de cada una, los costos por tarea, etc...

Ventajas

El programa corre en forma nativa en el S.O. disminuyendo tiempos de respuesta.

Desventajas

- Costos por licencias en cada PC que se requiera su instalación.
- El software no es transportable.
- Trabaja de forma offline

3.5.2 Aplicación Web.

Esta solución plantea el desarrollo de un sistema WEB de acceso restringido, montado sobre un servidor, con posibilidades de acceso múltiples a este, a través del internet.

Ventajas

- Amplias opciones de implementación, ya que los lenguajes para el desarrollo dinámico de páginas WEB son variados (PHP, JSP, ASP, Ruby).
- Permite de manera muy sencilla el acceso desde el exterior, ya que el usuario sólo necesitará un navegador Web.
- Sistema multiplataforma.
- Sistema Multiusuario

Desventajas

- Dependencia absoluta para la conexión al sistema.

3.5.3 Hoja de Cálculo

Esta solución en una especie de hoja Excel, programada a través de macros, para un accesos más rápido en cuanto a los proyectos.

Ventajas

- Rápida implementación.
- Archivos de poco tamaño

Desventajas

- Informes poco personalizables.
- Escalabilidad limitada.
- Problemas para replicar la información
- Difícil mantención.

CAPITULO IV

DESCRIPCIÓN EL PROYECTO.

4.1 Presentación del Proyecto.

Este capítulo abordara principalmente sobre la descripción del proyecto, tomando base la solución escogida de las propuestas en el capítulo anterior, esta será explicada y justificada en breve. Además se listan los objetivos que se intentaran satisfacer dentro de la misma solución, la metodología y cronología en el tiempo de esta, estimando tareas y fechas.

4.2 Sobre el Proyecto.

Actualmente la Universidad del Bío Bío no cuenta con un sistema de apoyo a la Planificación y Control de Proyectos, que incluya la posibilidad de visualizar a través de una página WEB, los de cada proyecto y los eventos en los que se da cuenta a la comunidad de los logros. La Dirección de Extensión de la Universidad en conversaciones con el académico de la FACE *Sergio Bravo Silva*, ha manifestado gran interés en contar con un sistema como el mencionado. Por dicha razón se integrará al equipo de desarrollo.

A continuación se presentan los objetivos del proyecto, basados en la solución elegida como propuesta, para la resolución del problema descrito en capítulo anterior.

4.3 Solución escogida

De las 3 posibles soluciones listadas en el capítulo anterior, se llegó a la determinación de optar por la segunda esto por los motivos que se enumeran a continuación.

- La opción A si bien es una solución esperable a este tipo de problemas, no se comporta de buena forma al momento de permitir accesos externos como lo son los accesos desde fuera, además; El hecho que sea un software privativo como lo es

Proyecto de Microsoft, hace que sea más difícil el poder usarlo y ver libremente los archivos de los proyectos, es más para que funcione en cada PC, debería estar instalada esta aplicación.

- La opción C es una buena solución pero para aplicaciones con menos exigencias y de menor envergadura, que no manejen tipos de acceso ni tanta cantidad de datos, puesto que se hará obligatorio compartir este tipo de archivos (.xls) lo que formará un cuello de botella entre quienes deseen ocupar la planilla, o en su defecto, replicar la información en diferentes archivos o fuentes, haciendo prácticamente imposible una sincronización entre ellos.
- La opción B se comporta bien en todos los casos esperables mencionados anteriormente lo que sumado a la facilidad técnica de utilización (solo requiere un navegador WEB y conexión a internet por parte del usuario) y la facilidad de controlar los accesos, hacen de esta opción la mejor dentro de las listadas.

Como la opción elegida es un sistema WEB, el lenguaje a ocupar será PHP5, Javascript y AJAX, para el servidor será montado en un sistema Linux con servicio de apache y el motor de la base de datos será Postgresql. Esto se basa en la premura de la comunidad existente, numerosos manuales/tutoriales y excelentes frameworks que facilitan aun más el desarrollo de aplicaciones (esto último es una de las grandes razones de optar por PHP5).

4.4 Objetivos Generales.

“Desarrollar un Sistema de Planificación y Control de proyectos, que incluya una plataforma web que facilite la Planificación y Control de Proyecto y Eventos de la UBB, evitando, dentro de lo posible, coincidencia de eventos en la misma fecha y hora.”

4.5 Objetivos Específicos.

1.-Analizar las variables que intervienen en planificación y control de Proyectos detectando deficiencias actuales y posibles mejoras requeridas, estableciendo el modelo de gestión de proyecto y control de eventos para nuestro software.

2.-Diseñar un sistema de apoyo a la gestión de planificación y control de eventos, que permitiendo a las Direcciones patrocinadoras y a los Directores de Proyecto hacer un seguimiento del avance y conocer oportunamente las desviaciones al Plan.

3.-Construir una plataforma web de apoyo a la gestión de eventos universitarios, utilizables por los diferentes niveles de la organización, con el fin conocer los espacios libres y registrar fecha, hora, lugar y detalles de cada evento y evitar en lo posible los traslapos con otros eventos.

4.6 Justificación del Proyecto.

En la Universidad las distintas Direcciones respaldan proyectos cuya Planificación y Control resultan engorrosas (Productos, fechas, recursos). El Sistema a desarrollar despejará la complejidad del actual control. Por otra parte las distintas personas invitadas a asistir a eventos, con frecuencia ven privadas de asistir a muchos por traslapos de fecha y hora, que se producen en muchos casos. Esto representa una pérdida importante para la institución en términos de recursos invertidos y la baja convocatoria.

La imagen de Universidad del Bío-Bío, se menoscaba cuando tienen lugar eventos con muy baja convocatoria, lo cual debería reducirse con un adecuado apoyo a la planificación.

La baja convocatoria de eventos redundará en una pérdida económica, de oportunidad de transferencia cultural y de imagen de la universidad.

Existen todas las capacidades en la universidad y en particular en la FACE para concebir, diseñar y construir soluciones a problemas como el planteado.

4.7 Metodología a utilizar

La metodología elegida para desarrollar este proyecto es el modelo iterativo incremental, es por ello que para comprender las ventajas, desventajas y cuando es conveniente su uso, se cita a continuación un capítulo del autor Ian Sommerville de su libro la ingeniería del software, quien explica estas características y detalla el modelo iterativo incremental.

El desarrollo incremental, implica producir y entregar el software en incrementos más que en un paquete único. Cada iteración del proceso produce un nuevo incremento del software. Las dos ventajas principales de adoptar un enfoque incremental para el desarrollo del software son:

- **Entrega acelerada de los servicios del cliente.** En los incrementos iniciales del sistema se pueden entregar las funcionalidades de alta prioridad para que los clientes puedan aprovechar el sistema desde el principio de su desarrollo. Los clientes pueden ver sus requerimientos en la práctica y especificar cambios a incorporar en entregas posteriores del sistema.
- **Compromiso del cliente con el sistema.** Los usuarios del sistema tienen que estar implicados en el proceso de desarrollo incremental debido a que tienen que proporcionar retroalimentación sobre los incrementos entregados al equipo de desarrollo. Su participación no sólo significa que es más probable que el sistema cumpla sus requerimientos, sino que también los usuarios finales del sistema tienen que hacer un compromiso con él y conseguir que éste llegue a funcionar.

En la figura 2.1 se ilustra un modelo de proceso general para el desarrollo incremental. Observe que las etapas iniciales de este proceso se centran en el diseño arquitectónico. Si no se considera la arquitectura al principio del proceso, es probable que la estructura general del sistema sea inestable y se degrade conforme se entreguen nuevos incrementos.

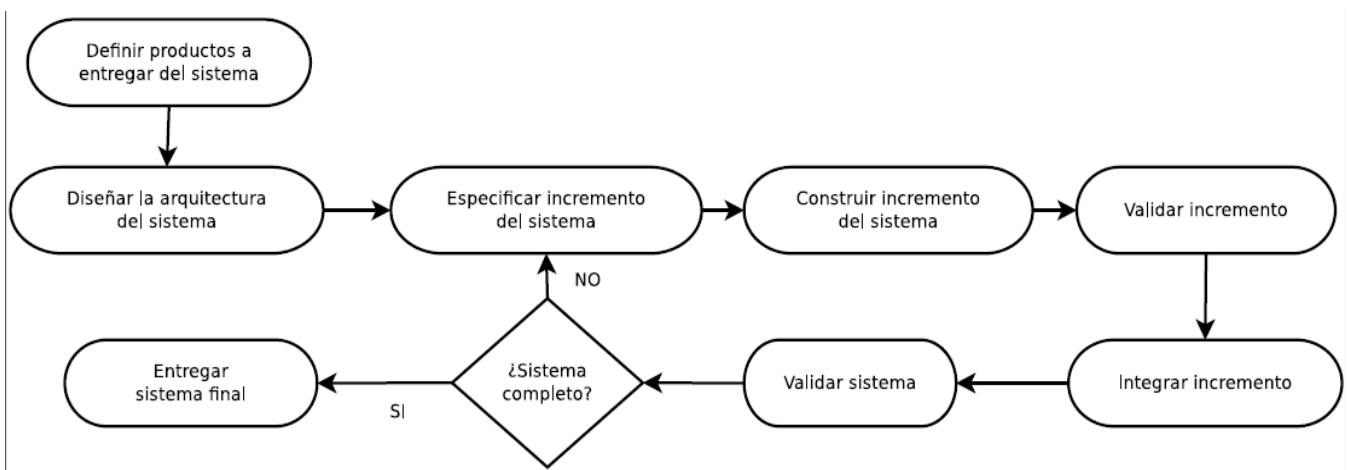


Figura 2.1: Proceso de desarrollo iterativo incremental

El desarrollo incremental del software es un enfoque mucho mejor para el desarrollo de la mayoría de los sistemas de negocio, comercio electrónico y personales porque refleja el modo fundamental al que todos nosotros tendemos al resolver problemas. Rara vez encontramos una solución completa a un problema por adelantado, peor nos movemos hacia una solución en una serie de pasos, dando marcha atrás cuando nos damos cuenta de que hemos cometido un error [18].

Sin embargo, puede haber verdaderos problemas con este enfoque, particularmente en las grandes compañías con procedimientos bastantes regidos y en organizaciones donde el desarrollo del software Normalmente se subcontrata con un contratista exterior. Los principales problemas con el desarrollo iterativo y la entrega incremental son:

- **Problemas de administración.** Las estructuras de administración del software para sistemas grandes se diseñan para tratar con modelos de proceso del software que generan entregas periódicas para evaluar el progreso. Los sistemas desarrollados incrementalmente cambian tan rápido que no es rentable producir una gran cantidad de documentación del sistema. Además, el desarrollo incremental muchas veces puede requerir el uso de tecnologías desconocidas para asegurar una entrega más rápida del software. A los administradores puede resultarles difícil utilizar el personal existente en los procesos de desarrollo incremental puesto que carecen de las habilidades requeridas.
- **Problemas contractuales.** El modelo contractual normal entre un cliente y un desarrollador de software se basa en la especificación del sistema. Cuando no existe tal especificación, puede ser difícil diseñar un contrato para el desarrollo del sistema. Los clientes pueden estar descontentos con un contrato que simplemente pague a los desarrolladores por el tiempo invertido en el proyecto, ya que puede conducir a que el sistema se desarrolle lentamente y se sobrepase el presupuesto; los desarrolladores probablemente no aceptarían un contrato con precio fijo debido a que no pueden controlar los cambios requeridos por los usuarios finales.
- **Problemas de validación.** En un proceso basado en la especificación, la verificación y la validación están pensadas para demostrar que el sistema cumple su especificación.

Un equipo independiente de V & V puede empezar a trabajar tan pronto como esté disponible la especificación y puede preparar pruebas en paralelo con la implementación del sistema. Los procesos de desarrollo iterativo intentan minimizar la documentación y entrelazan la especificación y el desarrollo. Por lo tanto, la validación independiente de los sistemas desarrollados incrementalmente es difícil.

- **Problemas de mantenimiento.** Los cambios continuos tienden a corromper la estructura del cualquier sistema software. Esto significa que cualquiera, aparte de los desarrolladores originales, puede tener dificultades para entender el software. Una forma de reducir este problema es utilizar refactorización, donde se mejoran continuamente las estructuras del software durante el proceso de desarrollo. Además, si se utiliza tecnología especializada, como los entornos RAD puede convertirse en obsoleta. Por lo tanto, puede ser difícil encontrar personas que tengan los conocimientos requeridos para dar mantenimiento al sistema [18].

4.9 Sobre el sistema.

A continuación detallaremos, varios puntos importantes de nuestro sistema, ya sea, el nombre, sus objetivos, el objetivo general que quiere satisfacer y los específicos.

4.9.1 Nombre del Sistema.

Plataforma Web para el control de Gestión de Eventos Institucionales.

4.9.2 Objetivo general.

“Dar apoyo al proceso de vida o desarrollo de un proyecto institucional, a través, de una herramienta eficaz, que sea amigable y de nivel profesional.”

4.9.3 Objetivos Específicos.

- Amigable al usuario.
- Rapidez en la consulta de información.
- Estructura jerárquica.
- Perfiles de sistema cerrados.
- Información protegida.

CAPITULO V

REQUERIMIENTOS.

5.1 Introducción

El presente capítulo se define como un documento de especificación de requerimientos del software (ERS), requerimientos que son acordados por el cliente (usuario), por lo que contiene en forma detallada todos los aspectos y funcionalidades esenciales para poder diseñar y desarrollar el sistema propuesto como solución.

Además se profundizará en los aspectos mismos de la relación que tendrán los usuarios con el software, y se detallarán los niveles de accesibilidad.

5.1.1 Propósito

El documento de especificación de requerimientos tiene por finalidad el describir de forma clara y lo más detalladamente posible las funciones que el cliente requiere para el sistema.

Esta dirigido tanto a los desarrolladores, diseñadores del sistema como a los mismos clientes (usuarios del sistema) ya que éste es un documento dinámico sujeto a diversas actualizaciones y/o correcciones futuras las cuales serán llevadas a cabo por cada uno de los grupos involucrados anteriormente nombrados, hasta alcanzar una versión final.

Sin embargo, para aspectos de éste proyecto se definirá sólo un breve ciclo de toma y levantamiento de requerimientos, esto para estar dentro de los plazos ya estipulados.

5.1.2 Alcance.

Los requerimientos de este documento tienen directa relación con el sistema a desarrollar, por ende, también se relacionan al proceso mismo de recepción de solicitudes de ayudas, es decir sólo contendrá descripciones a las funcionalidades que se encuentren en el sistema y según se requiera.

5.1.3 Definiciones, siglas y abreviaciones.

Definiciones.

Cliente	Es la persona o entidad a la cual se le desarrollará la solución, puede coincidir o no con los usuarios del sistema.
Usuario	Persona o grupo de personas que utilizarán el sistema, existen diversos, cada uno con distinto tipo de acceso.
Solicitante	Son los docentes de la universidad que solicitaran o generaran automáticamente un lugar dentro del sistema.

Siglas.

IEEE	Institute of Electrical and Electronics Engineers.
ERS	Especificación de requerimientos de software.

5.1.4 Referencias

Estándar IEEE 830[6].

5.2 Apreciación Global

Hasta ahora se ha hablado básicamente del ámbito del documento incluyendo la terminología necesaria para poder comprenderlo a cabalidad, las secciones siguientes describirán los factores generales que influyen y conciernen al sistema junto los requerimientos específicos que explicaran en detalle las funcionalidades exigidas por el cliente.

5.2.1 Descripción Global.

El sistema surge para ofrecer una forma más rápida y más confiable de obtener información, permitiendo hacer un seguimiento sencillo de los proyectos de la nuestra comunidad universitaria, siendo una alternativa moderna al sistema manual actualmente ocupado en la Dirección de Extensión.

El sistema está pensado para ser autónomo, sin dejar de lado por ello, la posibilidad de agregar módulos o también anexar futuros sistemas a su base de datos.

5.2.2 Funciones del Sistema.

A modo de vistazo global de nuestro sistema, las funcionalidades requeridas son:

- ❖ Administración de usuarios y miembros de los proyectos existentes y pasados.
- ❖ Administración de los Proyectos.

- ❖ Historial de Proyectos.
- ❖ Administración de comunicados de Prensa.
- ❖ Administración de Eventos.

Administración de usuarios y miembros de los proyectos existentes y pasados.

Modulo encargado de llevar el control detallado de cada persona que participe en los proyectos, ya sea, docente, alumnos o alguna persona externa a la universidad pero que tenga un rol dentro de éstos. Las funciones básicas en esta sección son: ingresar miembros de proyectos, modificar miembros, ingresar usuarios del sistema, modificar usuarios del sistema, desplegar usuario, dar de baja usuario. Ver a todos las personas que participan dentro de un proyecto o ver todos los participantes de los diferentes proyectos.

Se hace la diferencia entre usuarios y miembros del proyecto, pues un miembro , puede estar contenido en el proyecto , pero no necesariamente debe ser usuario del sistema.

Al momento de dar de baja un proyecto los usuarios o participantes no se eliminan del sistema, pues como son docentes y pueden postular nuevamente a nuevos proyectos solamente se deja inactivo el proyecto dejando al docente en calidad de que le asignen un nuevo proyecto.

Administración de los Proyectos.

Sección encargada principalmente de velar por la administración e los proyectos en curso. En esta sección se define funciones como:

- *Agregar datos sobre el proyecto:* nombre, fechas de inicio, fechas de termino, presupuestos, también se puede agregar miembros de equipo si fuese necesario.
- *Agregar tareas:* nombre las diferentes etapas o tareas del proyecto asignando, fechas de inicio, fechas de términos, presupuestos para esta, con el fin de no pasarse del presupuesto establecido inicialmente.

- *Agregar subtareas:* nombres , fechas inicio, fechas de termino, asignar la etapa a la que pertenece la subtarea y si hubiese un gasto su ítem presupuestario.
- *Informe de tareas:* este consta de una filtro de búsqueda para ver el estado de las tareas existentes dentro del proyecto , esto quiere decir, si las tareas están en curso , no han sido comenzadas , si fueron terminadas o en el caso que la tarea estuviese atrasada.
- *Ver datos del proyecto:* de una maneja de proteger la integridad de los datos , y ver la situación actual , esta sección se encarga de ver presupuestos reales versus los presupuestos generados durante el desarrollo , el numero de tareas comenzadas , finalizadas y atrasadas, las respectivas fechas de inicio y de termino , para que el usuario lleve un completo detalle del estado del proyecto.
- *Redactar Comunicado de prensa :* esto es mas generar un comunicado de aviso al departamento de preense en el caso que fuese necesario , para dar a conocer a la comunidad sobre el evento generado o los resultado obtenidos del proyectos.

Administración de comunicados de Prensa.

Este modulo es el encargado de generar un listado de comunicados para que los usuarios, con el rol adecuado puedan ver y generar la comunicación entre el docente y el departamento de prensa.

Historial de Proyectos.

Modulo encargado de ver todos los proyectos y la forma en que finalizaron, si estos fueron ejecutados completamente o fueron abortados antes de tiempo, con el fin de llevar una un detalle de los proyectos realizados dentro de la universidad.

Administración de Eventos.

Este modulo es el cargado , de la administración de los diferentes eventos que ocurren al finalizar los proyectos , su principal función es el ingreso de los eventos , restringiendo horarios ,fechas, lugares de eventos y estableciendo una relación directa entre el evento y el proyecto.

Dentro de esta sección también se encuentra el visor de eventos , para mirar si el día planificado existe otro evento ,dentro de los datos que se publican esta; el día , hora, el lugar y el nombre el proyecto encargado del evento.

5.2.3 Características de los roles usuario

Rol Administrador

El administrador tendrá todos los privilegios, poseerá el acceso a todas las secciones del sistema, como característica única, podrá gestionar quien puede acceder al sistema (también el tipo de acceso) y además podrá ingresa datos importantes ,como los presupuestos del proyectos y las fechas de inicio y termino.

Rol Docente

El docente será tendrá casi los mismos privilegios del rol administrador. Esto quiere decir que podrá ingresar miembros a su equipo de trabajo, ingresar tareas, ingresar subtareas ,ingresar eventos , ver los diferentes ítem mencionados anteriormente y ver los datos del proyecto, sin embargo, éste no podrá modificar los datos de presupuesto del proyecto , fechas de inicio del proyecto y fecha de termino del proyecto.

Rol Miembro del equipo.

Este usuario estará restringido solo podrá acceder a las vistas de las subtareas, los datos del proyecto (sin modificar estos),ver los datos del evento, redactar comunicados de prensa.

5.2.4 Restricciones

Es necesario un manejo de sesiones por cada usuario lo que implica que el acceso a este sistema es restringido a previa \"suscripción\" o registro de dicho usuario en el sistema (por parte del administrador).

Además debido a que la arquitectura es cliente/servidor la implementación se ve obligada realizarse mediante un servidor local (para tener un mayor control sobre el respaldo de las bases de datos y el sistema mismo).

5.2.5 Atención y dependencias

Como la solución es un sistema basado en web, se debe tener (y se asume) una red local bien configurada y administrada, además queda en evidencia que para poder ocupar eficientemente el sistema, el usuario debe estar más o menos familiarizado con el manejo básico del PC.

5.3 Requerimientos específicos.

En esta sección se realiza una descripción en detalle de 5 tipos de requisitos observados para el sistema, los cuales son:

- Requerimientos funcionales.
- Requerimientos no funcionales.
- Requerimientos de información.
- Requerimientos de técnicos.
- Requerimientos de seguridad y auditoria.

5.3.1 Requerimiento Funcionales.

Administración de usuarios y miembros de Proyectos.

R1.- El sistema deberá gestionar usuarios.

R 1.1.- El sistema deberá manejar usuarios con diferentes tipos de rol.

R 1.2.- Deberá permitir solo al usuario con rol administrador ingresar los presupuestos de los proyectos.

R.1.3.- Deberá poder asignar a usuarios a un o más proyectos.

R 1.4.- Deberá poder ingresar un usuario.

R 1.5.- Deberá poder modificar un usuario.

R 1.6.- Deberá poder desplegar listado de usuarios.

R 1.7.- Deberá poder dar de baja (inactivo) a un usuario.

R 2.- El sistema deberá restringir el acceso.

R 2.1.- Deberá poder validar si un usuario puede o no entrar al sistema.

R 2.2.- Deberá poder desplegar la información restringida según corresponda (por cada nivel de acceso)

R 2.3.- Deberá poder manejar límites en los tiempos de vida de las sesiones.

R 3.- El sistema deberá gestionar los miembros de cada proyecto.

R 3.1.- Deberá poder ingresar datos relevantes de los miembros.

R 3.2.- Deberá poder modificar miembros.

R 3.3.- Deberá poder desplegar miembros.

R 3.4.- Deberá poder permitir búsqueda mediante filtros.

Administración de Proyectos.

R.4 EL sistema deberá gestionar y administrar los proyectos.

R.4.1-Deberá manejar los datos de los Proyectos.

R.4.2-Deberá modificar los datos del Proyectos.

R.4.3-Deberá restringir las modificaciones de los datos del Proyecto.

R.4.4- Deberá Ligar los miembros de los equipos a los Proyectos.

Administración de Tareas.

R.5- El sistema deberá manejar las tareas o etapas de los proyectos.

R.5.1-Deberá Ingresar tareas.

R 5.2- Deberá modificar Tareas.

R 5.3 –Deberá ver Tareas por proyecto.

Administrar Subtareas.

R.6- El sistema deberá manejar las de los proyectos.

R.6.1-Deberá Ingresar subareas.

R 6.2- Deberá modificar subareas.

R 6.3–Deberá ver subareas por proyecto.

Administrar y Gestionar Eventos.

R.7 El sistema deberá gestionar y manejar los eventos.

R.7.1 Deberá ingresar Eventos.

R.7.2 Deberá modificar Eventos.

R.7.3 Deberá Gestionar Eventos, previniendo topes.

R7.4 Deberá mostrar un listado de eventos por medio de filtros.

Administrar Prensa.

R8. El sistema deberá administrar los comunicados de prensa.

R 8.1 Deberá ingresar comunicados.

R 8.1 Deberá Enviar aviso de comunicados Nuevos al departamento de prensa.

R 8.1 Informe de los comunicados existentes dentro del sistema.

Administrar historial de proyectos y usuarios.

R.9 El sistema deberá manejar los usuarios y proyectos que no estén en curso .

R9.1 Por medio de filtros generar informe de proyectos con estado: terminado o acabado antes de tiempo.

R9.2 Por medio de filtro generar informe de personal que ha intervenido en los proyectos.

5.3.2. Requerimientos no funcionales

1. El sistema deberá tener un logo identificativo de la Universidad del Bío-Bío.
2. El sistema deberá ser amigable y en gran medida de uso intuitivo.
3. El sistema deberá tener tiempos de respuesta en promedio inferiores a 7 segundos.
4. El lenguaje a ocupar para el desarrollo del proyecto será PHP, junto a html y javascript.

5. El enfoque a ocupar en el proyecto se guiara mediante el patrón de diseño modelo, vista, controlador.
6. Para abaratar costos se ocupará como administrador de base de datos Postgresql y apache 2
7. Para agilizar el desarrollo se optó por la integración de PHP a Neatbens 7.0 un framework MVC
8. Como capa de abstracción de BD se ocupará Doctrine 1.2
9. Como framework de javascript enriquecido se ocupará ajax.

5.3.3. Requerimientos de información

A continuación se detallan los datos que el sistema necesita registrar y administrar.

Usuario

- Nombre de usuario
- Contraseña de acceso
- Tipo de usuario
- Nombres
- Apellidos
- Correo electrónico
- Departamento
- Dirección
- Cargo dentro del Proyecto.

Proyectos.

- Nombre del Proyecto.
- Fecha Inicio.
- Fecha termino.
- Presupuesto.

Miembros.

- Nombre de usuario
- Contraseña de acceso

- Tipo de usuario
- Nombres
- Apellidos
- Correo electrónico
- Departamento
- Dirección.
- Cargo dentro del Proyecto.

Tareas .

- Nombre Subtarea.
- Proyecto asociado.
- Fecha Inicio.
- Fecha termino.
- Presupuesto.

Subtarea.

- Nombre Subtarea.
- Tarea asociada.
- Fecha Inicio.
- Fecha termino.
- Ítem Presupuestario

Evento.

- Nombre Evento.
- Proyecto asociado.
- Día
- Fecha
- Hora
- Lugar

Comunicado de Prensa.

- Título
- Comunicado
- Proyecto Asociado.

5.3.4. Requerimientos Técnicos

En esta sección se describen los requerimientos tanto de la parte cliente sistema, como de la parte servidora, detallando los requisitos mínimos para poder obtener una buena experiencia de usabilidad por parte del usuario final.

Requerimientos de desarrollo

Hardware

Estos son los requerimientos de hardware **mínimos** para poder realizar el desarrollo.

- ✓ Procesador Intel o compatible de 1.5 GHz
- ✓ 512 MB de memoria RAM
- ✓ Disco duro 50 GB
- ✓ Periféricos (Teclado, mouse, monitor)

Software

- ✓ Sistema operativo GNU/Linux (Ubuntu 9.0.4)
- ✓ Herramientas CASE: umbrello, dia, gedit
- ✓ Power Designer.
- ✓ Servidor XAMPP ,Apache
- ✓ PostgreSQL.
- ✓ Neatbeans 7.0

Requerimientos de puesta en marcha (Cliente)

Estos son los requerimientos de hardware mínimos para poder usar la parte cliente del sistema con un mínimo usabilidad.

Hardware

- ✓ Procesador Intel o compatible de 1 GHz
- ✓ 256 MB de memoria RAM
- ✓ Disco duro 50 GB

- ✓ Periféricos (Teclado, mouse, monitor)

Software

- ✓ Windows o Sistema operativo GNU/Linux
- ✓ Última versión de navegador Firefox (recomendado), Chrome, Internet Explorer.

Requerimientos de puesta en marcha (Servidor)

Hardware

- ✓ Procesador Intel o compatible de 2.0 GHz
- ✓ 1 GB. de memoria RAM
- ✓ Disco duro 100 GB

Software

- ✓ Windows o Sistema operativo GNU/Linux
- ✓ Servidor WAMP o XAMPP y en el caso que fuese Linux el sistema operativo Apache server.
- ✓ PostgreSQL.

5.3.5. Requerimientos de Seguridad y Auditoría

Integridad de la Base de Datos

Como toda aplicación que maneja grandes cantidades de datos, es completamente necesario el constante respaldo de datos, se recomiendan realizar esta acción cada 15 días, pero esto queda a completa responsabilidad por parte de él o los encargados del sistema en la Universidad (departamento de informática).

Control de acceso al sistema y base de datos

El acceso al sistema estará restringido a sólo algunos usuarios, como se describió anteriormente, sin embargo, existen 2 tipos de acceso al mismo.

Acceso de usuarios al sistema

Este tipo de acceso es el descrito en secciones anteriores, y hace referencia a los usuarios "físicos" del mismo, es decir, a las personas que usarán la aplicación su acceso está permitido mediante el login del mismo, y las validaciones son por lenguaje de programación.

Acceso del sistema a la base de datos

Este tipo de acceso hace referencia al usuario de la base de datos; y es el usuario con el que el sistema se conecta al administrador de BD y realiza los cambios necesarios en ella, por seguridad sólo debe tener permisos para ejecutar acciones básicas sobre el conjunto de datos, como por ejemplo. select, delete, insert, update entre otras, y no permitir acciones como borrado del esquema (base de datos), agregar permisos o eliminar usuarios, esto se logra en conjunto con el administrador de base de datos y es indivisible para el usuario final.

Control de acceso al servidor

El control de acceso a los sistemas es la primera forma de evitar la intrusión de individuos ajenos a la institución, pero esto debe ser llevado en conjunto con la restricción al acceso del servidor mismo, pues si un individuo "mal intencionado" accede físicamente al servidor, todos los esfuerzos de restricción por tipos de usuarios serían en vano, por ello se recomienda un análisis y posible reestructuración de las políticas de seguridad actuales.

CAPITULO VI

Herramientas Utilizadas.

6.1. Introducción

El presente capítulo expondrá las herramientas, sistemas y utilidades más importantes utilizadas en el desarrollo e implementación del sistema. Muchas de ellas se consideraron a partir de los requerimientos expuestos anteriormente.

Para cada una de ellas se detallará según convenga un poco de historia y una breve explicación de que es y para qué sirve, si bien la lista podrá parecer un poco extensa, el lector debe saber que no se quiso profundizar en todas las herramientas ocupadas, sino que solamente en las más relevantes involucradas con el diseño e implementación, dejando fuera algunas como las herramientas CASE ocupadas principalmente para el diagramado y creación de documentación anexa.

Las herramientas omitidas fueron: Gedit, Dia, Umbrello, Power Designer ,entre otras.

6.2. GNU/Linux (Ubuntu 9.04).

Antes de hablar sobre el sistema operativo GNU/Linux es necesario introducir al lector al sistema Unix, ya que Linux es un Clon de este _ultimo [10].

6.2.1. Historia de Unix

En 1969, Ken Thompson, desarrollador de los laboratorios Bell (que forman parte de AT&T), invento el sistema Unix.

K. Thompson había trabajado previamente en el sistema Multics que permitía, por ejemplo, ocultar el sistema al usuario e incluso al programador; el objetivo era poder desarrollar nuevas aplicaciones haciendo abstracción del sistema. Cuando Bells Labs se retiró del proyecto Multics, empezó a desarrollar su propio sistema en un equipo DEC PDP-7 salvado de la

quema. Conservó ciertas ideas desarrolladas por Multics y añadió sus propias innovaciones: acababa de nacer Unix.

El mismo tiempo, Dennis Ritchie inventa un nuevo lenguaje de programación: el lenguaje C. En la línea de Unix, este lenguaje pretendía ser amigable, flexible y sin restricciones. Los laboratorios Bell, que conocían las herramientas desarrolladas, dieron apoyo a este dúo con una prima en 1971 para la elaboración de un sistema de automatización de escritorio de uso interno.

En realidad, los sistemas desarrollados hasta entonces estaban codificados en lenguaje ensamblador, próximo a la arquitectura de hardware, lo que les confería un alto rendimiento, pero obligaba también a reescribir el programa cuando el hardware obsoleto era reemplazado por equipos más recientes. Comprendiendo que los avances tecnológicos podían paliar una pérdida de velocidad y que las técnicas de compilación habían experimentado grandes progresos, D. Ritchie y K. Thompson dieron prioridad a la portabilidad del sistema. Al escribir la totalidad del sistema en Unix en lenguaje C, consiguieron hacer funcionar el sistema en muchos equipos de tipos diferentes desde 1974.

Así, un compilador de C desarrollado para cada equipo bastaba para hacer posible el uso del mismo entorno en diferentes equipos; esto no se había hecho nunca abría grandes posibilidades. El desarrollo de aplicaciones liberadas de la restricción del hardware disminuiría considerablemente los costes de diseño para el usuario cuando era necesario renovar un equipo que se había quedado anticuado, Además, los programadores podían fabricarse cajas de herramientas de software y transferirlas de un equipo a otro sin tener que reinventarlo todo.

Bajo el impulso de AT&T Unix se desarrolló rápidamente. En 1980, este sistema estaba ya muy extendido en las universidades y en los centros de investigación.

La universidad de California en Berkeley (que utilizaba Unix desde hacia varios años), aportó diferentes mejoras a este sistema operativo, actualmente más conocido con el acrónimo de BSD (Berkeley Software Distribution). Los desarrollos "Berkeley" se publicaron paralelamente a los realizados por el grupo de trabajo de Unix de AT&T. Actualmente, la

mayor parte de sistemas operativos de tipo Unix en el mercado consisten en una mezcla del código original de AT&T y de las "mejoras de Berkeley".

6.2.2. GNU

Debido a la imposibilidad de obtener y modificar el código fuente del controlador de la nueva impresora que llegó a su departamento, Richard Matthew Stallman, entonces joven investigador en el laboratorio de Inteligencia Artificial del MIT (Massachusetts Institute of Technology), se concienció de los riesgos vinculados con software propietario; para oponerse a la comercialización del software y más particularmente a la falta de disponibilidad del código fuente, inició entonces un movimiento para el desarrollo de software libre de derechos 1984.

Este movimiento se tradujo en la creación de un proyecto que consistía en reescribir completamente un sistema operativo libre. el modelo seguido era Unix, y Richard Stallman llamó a su proyecto GNU, que significa GNU's Not Unix.

El proyecto GNU conoció rápidamente un gran éxito y muchas herramientas y aplicaciones de Unix se desarrollaron a partir de cero ("from scratch").

Sin embargo, el núcleo libre de este sistema, llamado "Hurd", no se desarrolló tan rápidamente. El proyecto GNU se limitó durante cierto tiempo a ser una gama de herramientas completa de Unix sin núcleo.

6.2.3. Linux

Utilizando el sistema operativo Minix, desarrollado por Andrew S. Tanenbaum con el objetivo de mostrar a sus estudiantes el funcionamiento de un sistema de tipo Unix, en un proyecto escolar sobre el modo protegido de los procesadores Intel 386, Linus Torvalds empezó a desarrollar su propio núcleo Unix para añadirle nuevas funcionalidades.

Linux (Linus's Unix) nació, pues, en 1991, gracias a un estudiante de la universidad de Helsinki. El éxito de Linux se basa en una idea ingeniosa de su creador, L. Torvalds: inscribir su proyecto bajo los términos de la licencia GPL y proponer a todos los programadores y otros hackers de Internet que le ayudaran.

El impulso de Linux proviene en gran parte del hueco que llenó en términos de núcleo en el proyecto GNU.

Características

Multitarea: Está diseñado para ejecutar varios programas al mismo tiempo (PID - process ID, codificados en 32 bits). Utiliza un asignador para ejecutar varias acciones con un mismo procesador y también puede sacar partido de arquitecturas multiprocesador (hyperthreading, SMP - Symmetric Multi-Processors, NUMA - Non-Uniform Memory Access).

Multiusuario: Este sistema permite el uso por parte de varias personas de los recursos que administra. Estas personas se distribuyen en grupos de usuario y es necesaria una autenticación para asegurar los derechos individuales. Los UID (User's ID) y GID (Group's ID) están codificados en 32 bits.

Multiplataforma: Linux ha sido llevado a un gran número de arquitecturas de hardware, como Intel y sus clones (AMD, Cyrix), SPARC, MIPS, Dec Alpha, PowerPC, PalmPilot... Encontramos también Linux en sistemas embebidos, como autómatas industriales o autorradios MP para el particular.

Sistema de archivos : Linux soporta un gran número de sistemas de archivo, además de los de tipo Unix, incluyendo ISO9660 (CD-Rom), Windows 9x, NTFS y Macintosh. Soporta también los ACL (Access List) y las cuotas.

Administración de memoria: En vez de efectuar un "swap" de procesos completos, el núcleo pagina la memoria virtual, lo que limita los accesos al disco cuando falta memoria física. Además, la memoria compartida con copia en escritura permite que varios procesos utilicen el mismo espacio de memoria; cuando uno de ellos intenta escribir en la memoria, se realiza una copia de página de memoria en otra parte. Este mecanismo optimiza el uso y la cantidad de memoria necesaria en el sistema. Además, Linux soporta hasta 64 GB de RAM con las funciones de hardware PAE (Physical Address Extension) de Intel.

Redes: Como cualquier otro Unix, Linux posee una capa de red muy fiable y rápida. Soporta un amplio abanico de protocolos (TCP/IP) versiones 4 y 6, Netware, Appletalk Lan Manager, IPX, IPsec ...) y dispositivos de red (Ethernet, TokenRing, FDDI, Wi-Fi ...).

Compatibilidad: Linux es compatible con Unix System V (AT&T) y BSD a nivel de código fuente, soporta las bibliotecas Unix en formato COFF y ELF y los binarios SCO SVR3 y SVR4 con los módulos de emulación IBCS2.

4.3. Apache

(Acrónimo de "a patchy server"). Servidor web de distribución libre y de código abierto, siendo el más popular del mundo desde abril de 1996, con una penetración actual del 50% del total de servidores web del mundo (agosto de 2007) [9].

La principal competencia de Apache es el IIS (Microsoft Internet Information Services) de Microsoft.

Apache fue la primera alternativa viable para el servidor web de Netscape Communications, actualmente conocido como Sun Java System Web Server.

Apache es desarrollado y mantenido por una comunidad abierta de desarrolladores bajo el auspicio de la Apache Software Foundation.

La aplicación permite ejecutarse en múltiples sistemas operativos como Windows, Novell NetWare, Mac OS X y los sistemas basados en Unix.

Historia de Apache

La primera versión del servidor web Apache fue desarrollada por Robert McCool, quien desarrollaba el servidor web NCSA HTTPd (National Center for Supercomputing Applications). Cuando Robert dejó el NCSA a mediados de 1994, el desarrollo de httpd se detuvo.

Robert McCool buscó otros desarrolladores para que lo ayudaran, formando el Apache Group. Algunos miembros del grupo original fueron Brian Behlendorf, Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau, Andrew Wilson, Eric Hagberg, Frank Peters y Nicolas Pioch.

La versión 2 del servidor Apache fue una reescritura sustancial de la mayor parte del código de Apache 1.x, enfocándose en una mayor modularización y el desarrollo de una capa de portabilidad, el Apache Portable Runtime.

Apache 2.x incluyó multitarea en UNIX, mejor soporte para plataformas no Unix (como Windows), una nueva API Apache y soporte para IPv6.

La versión 3.4 estable de Apache, fue lanzada el 6 de abril de 2008.

Características de Apache

- ✓ Soporte para los lenguajes perl, python, tcl y PHP.
- ✓ Módulos de autenticación: mod access, mod auth y mod digest.
- ✓ Soporte para SSL y TLS.
- ✓ Permite la configuración de mensajes de errores personalizados y negociación de contenido.
- ✓ Permite autenticación de base de datos basada en SGBD.

Uso de Apache

Apache es principalmente usado para servir páginas web estáticas y dinámicas en la WEB. Apache es el servidor web del popular sistema XAMP, junto con MySQL y los lenguajes de programación PHP/Perl/Python. La "X" puede ser la inicial de cualquier sistema operativo, si es Windows: WAMP, si es el Linux: LAMP, etc.

6.4 PostgreSQL.

El uso de caracteres en mayúscula en el nombre PostgreSQL puede confundir a algunas personas a primera vista. Las distintas pronunciaciones de "SQL" pueden llevar a confusión. Los desarrolladores de PostgreSQL lo pronuncian /po:st gres kju: el/. Es también común oír abreviadamente como simplemente "Postgres", el que fue su nombre original. Debido a su soporte del estándar SQL entre la mayor parte de bases de datos relacionales, la comunidad consideró cambiar el nombre al anterior Postgres. Sin embargo, el PostgreSQL Core Team anunció en 2007 que el producto seguiría llamándose PostgreSQL. El nombre hace referencia a los orígenes del proyecto como la base de datos "post-Ingres", y los autores originales también desarrollaron la base de datos Ingres.

6.4.1 Historia

PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto Ingres en la Universidad de Berkeley. Este proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un motor de base de datos relacional. Después de haber trabajado un largo tiempo en Ingres y de haber tenido una experiencia comercial con él mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 1980. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos. Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones - las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos «comprendía» las relaciones y podía obtener información de tablas relacionadas utilizando reglas. Postgres usó muchas ideas de Ingres pero no su código.

La siguiente lista muestra los hitos más importantes en la vida del proyecto Postgres.

- 1986: se publicaron varios papers que describían las bases del sistema.
- 1988: ya se contaba con una versión utilizable.
- 1989: el grupo publicaba la versión 1 para una pequeña comunidad de usuarios.
- 1990: se publicaba la versión 2 la cual tenía prácticamente reescrito el sistema de reglas.
- 1991: publicación de la versión 3, esta añadía la capacidad de múltiples motores de almacenamiento.
- 1993: crecimiento importante de la comunidad de usuarios, la cual demandaba más características.
- 1994: después de la publicación de la versión 4, el proyecto terminó y el grupo se disolvió.

Después de que el proyecto POSTGRES terminara, dos graduados de la universidad, Andrew Yu y Jolly Chen, comenzaron a trabajar sobre el código de POSTGRES, esto fue posible dado que POSTGRES estaba licenciado bajo la BSD, y lo primero que hicieron fue añadir soporte para el lenguaje SQL a POSTGRES, dado que anteriormente contaba con un intérprete del lenguaje de consultas QUEL (basado en Ingres), creando así el sistema al cual denominaron Postgres95.

Para el año 1996 se unieron al proyecto personas ajenas a la Universidad como Marc Fournier de Hub.Org Networking Services, Bruce Momjian y Vadim B. Mikheev quienes proporcionaron el primer servidor de desarrollo no universitario para el esfuerzo de desarrollo de código abierto y comenzaron a trabajar para estabilizar el código de Postgres95.

En el año 1996 decidieron cambiar el nombre de Postgres95 de tal modo que refleje la característica del lenguaje SQL y lo terminaron llamando PostgreSQL, cuya primera versión de código abierto fue lanzada el 1 de agosto de 1996. La primera versión formal de PostgreSQL (6.0) fue liberada en enero de 1997. Desde entonces, muchos desarrolladores entusiastas de los motores de base de datos se unieron al proyecto, coordinaron vía Internet y entre todos comenzaron a incorporar muchas características al motor.

Aunque la licencia permitía la comercialización de PostgreSQL, el código no se desarrolló en principio con fines comerciales, algo sorprendente considerando las ventajas que PostgreSQL ofrecía. La principal derivación se originó cuando Paula Hawthorn (un miembro del equipo original de Ingres que se pasó a Postgres) y Michael Stonebraker conformaron Illustra Information Technologies para comercializar Postgres.

En 2000, ex inversionistas de Red Hat crearon la empresa Great Bridge para comercializar PostgreSQL y competir contra proveedores comerciales de bases de datos. Great Bridge auspició a varios desarrolladores de PostgreSQL y donó recursos de vuelta a la comunidad, pero a fines de 2001 cerró debido a la dura competencia de compañías como Red Hat y pobres condiciones del mercado.

En agosto de 2007 EnterpriseDB anunció el Postgres Resource Center y EnterpriseDB Postgres, diseñados para ser una completamente configurada distribución de PostgreSQL

incluyendo muchos módulos contribuidos y agregados. EnterpriseDB Postgres fue renombrado Postgres Plus en marzo de 2008.

El proyecto PostgreSQL continúa haciendo lanzamientos principales anualmente y lanzamientos menores de reparación de bugs, todos disponibles bajo la licencia BSD, y basados en contribuciones de proveedores comerciales, empresas aportantes y programadores de código abierto mayormente.

6.5 PHP

PHP (acrónimo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP [7].

6.5.1 Historia

Rasmus Lerdorf, miembro del equipo de desarrollo de Apache, creó PHP (Personal Home Page) en 1994. Su única intención fue la de crear un pequeño sistema de control para verificar el número de personas que leían su currículum vitae en la Web [13].

En los meses siguientes a su creación, PHP se desarrolló en torno a un grupo de programadores que comprobaban el código y sus revisiones. Para dar más potencia al sistema, Rasmus creó funciones en lenguaje C para permitir conexión a bases de datos. Este fue el comienzo de la potencia real del lenguaje.

En 1995, apareció un conjunto de herramientas sobre PHP. Esta biblioteca se llamó "Herramientas para páginas personales" y contenían un analizador de código muy sencillo, un libro de visitas, un contador y algunas macros que facilitaban el trabajo de los diseñadores.

A mediados de 1995, apareció una revisión pública llamada PHP/FI 2.0. Esta nueva versión contaba con un analizador sintáctico reescrito desde 0, además de unas herramientas escritas para el tratamiento de datos desde un formulario (de ahí el nombre Fi, Form Interpreter) y conectividad con MySQL (Gestor de bases de datos).

Hacia 1997, PHP/FI se estaba usando en más de 50.000 páginas en todo el mundo. En este período de tiempo, Zeev Suraski y Andi Gutmans decidieron crear una nueva versión de PHP/FI para solventar unos problemas con una aplicación de comercio electrónico que estaban desarrollando.

PHP 3.0 nació con suculentas innovaciones como la conectividad con varios gestores de bases de datos, protocolos y una API ampliada. La versión oficial de PHP 3.0 vio la luz en junio de 1998, donde se contemplaba ya la programación orientada a objetos.

En 1999 se realizó la primera revisión del motor Zend (Zend Engine), que aportaba modularidad, claridad y herramientas de optimización para páginas de gran escala. Zend viene de la unión de Zeev y Andi.

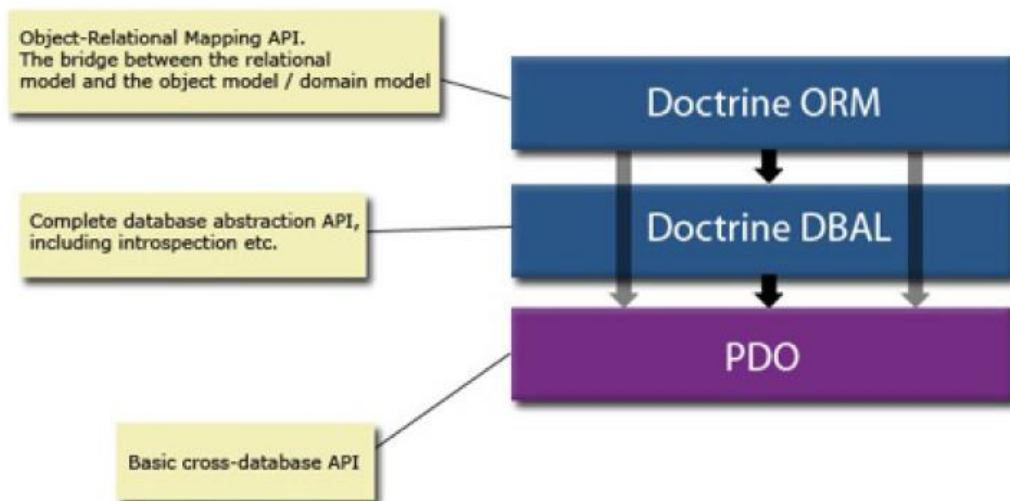
PHP 4.0 vio la luz en mayo de 2000, dividida en 3 partes: El motor Zend, la API de servidor y los módulos de funciones. El motor Zend es el responsable de analizar el código PHP, definir la sintaxis y del lenguaje de programación. La API permite la comunicación con el servidor. Con esta API es posible utilizar PHP desde varios servidores. Los módulos contienen funciones para el manejo de cadenas, archivos XML o tratamiento de imágenes. La orientación a objetos no está muy lograda en PHP 4.0. Los objetos tienen un tratamiento muy pobre e ilógico. La definición de las variables miembro (propiedades) y los métodos son siempre públicos, por lo que la encapsulación es nula. Todos los objetos se pasan por valor por defecto cuando deberían pasarse por referencia. Todas estas propuestas realizadas por el equipo de desarrollo de PHP han desembocado en la creación del motor Zend 2.0. y su consecuencia PHP 5.

PHP 5 incorpora una verdadera orientación a objetos. Añadiendo las palabras reservadas public, protected y private a la definición de las propiedades y métodos de los objetos, se permite una verdadera encapsulación. Además del considerable avance con respecto a los objetos, PHP 5 incorpora un control de errores muy mejorado, al estilo de los lenguajes de programación más avanzados.

6.6. Doctrine

Doctrine es un mapeador de objeto relacional (ORM - Object Relational Mapper) para PHP 5.2.3+ Un ORM es una técnica usada en lenguajes de programación como capa de abstracción que permite negociar con una base de datos relacional cuando ocupamos tipos de datos incompatibles con ella (como son los objetos) lo que permite tener una "base de datos orientada a objetos virtual".

Una de las características claves que posee Doctrine es la opción de escribir consultas de base de datos en un dialecto SQL propietario orientado a objetos llamado DQL - Doctrine Query Language, inspirado en Hibernate HQL, este proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene flexibilidad sin necesidad de la duplicación innecesaria de código. Se sitúa sobre PDO y se divide en dos capas principales, la capa de abstracción a la base de datos (DBAL - Database Abstraction Layer) y el ORM. como lo ilustra la figura



6.7 Netbeans

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE1 es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun

MicroSystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

6.7.1 Historia.

NetBeans comenzó como un proyecto estudiantil en la República Checa (originalmente llamado Xelfi), en 1996 bajo la tutoría de la Facultad de Matemáticas y Física en la Universidad Carolina en Praga. La meta era escribir un entorno de desarrollo integrado (IDE) para Java parecido a Delphi. Xelfi fue el primer IDE escrito en Java; tuvo su primer prelanzamiento en 1997.

Xelfi fue un proyecto divertido para trabajar, ya que los IDE escritos en Java eran un territorio desconocido en esa época. El proyecto atrajo suficiente interés, por lo que los estudiantes, después de graduarse, decidieron que lo podían convertir en un proyecto comercial. Prestando espacios web de amigos y familiares, formaron una compañía alrededor de esto. Casi todos ellos siguen trabajando en NetBeans.

Tiempo después, fueron contactados por Roman Stanek, un empresario que ya había estado relacionado con varias iniciativas en la República Checa. Estaba buscando una buena idea en la que invertir, y encontró en Xelfi una buena oportunidad. Así, tras una reunión, el negocio surgió.

El plan original era desarrollar unos componentes JavaBeans para redes. Jarda Tulach, quien diseñó la arquitectura básica de la IDE, propuso la idea de llamarlo NetBeans, a fin de describir este propósito. Cuando las especificaciones de los Enterprise JavaBeans salieron, decidieron trabajar con este estándar, ya que no tenía sentido competir contra él, sin embargo permaneció el nombre de NetBeans.

En la primavera de 1999, Netbeans DeveloperX2 fue lanzado, soportando Swing. Las mejoras de rendimiento que llegaron con el JDK 1.3, lanzado en otoño de 1999, hicieron de NetBeans una alternativa realmente viable para el desarrollo de herramientas. En el verano de 1999, el

equipo trabajó duro para rediseñar DeveloperX2 en un NetBeans más modular, lo que lo convirtió en la base de NetBeans hoy en día.

Algo más ocurrió en el verano de 1999. Sun Microsystems quería una herramienta mejor de desarrollo en Java, y comenzó a estar interesado en NetBeans. En otoño de 1999, con la nueva generación de NetBeans en Beta, se llegaría a un acuerdo.

Sun adquirió otra compañía de herramientas al mismo tiempo, Forté, y decidió renombrar NetBeans a Forté for Java. El nombre de NetBeans desapareció por un tiempo.

Seis meses después, se tomó la decisión de hacer a NetBeans open source. Mientras que Sun había contribuido considerablemente con líneas de código en varios proyectos de código abierto a través de los años, NetBeans se convirtió en el primer proyecto de código abierto patrocinado por ellos. En junio del 2000 NetBeans.org fue lanzado.

6.7.2 Aplicaciones.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones.

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)
- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato)
- Administración de ventanas
- Framework basado en asistentes (diálogos paso a paso)

CAPITULO VII

Factibilidad.

7.1 Introducción.

A continuación en este capítulo se muestra un estudio de factibilidad, de la plataforma de control y gestión de proyecto, para la Dirección de Extensión de la Universidad del Bío-Bío.

Para la construcción de la aplicación, se ha realizado un estudio para determinar si la infraestructura tecnológica, los costos, la capacidad técnica, que el sistema necesita para su construcción, pueda ser implementado sin mayores restricciones.

Los aspectos tomados en cuenta para éste estudio se clasifican en factibilidad técnica, operativa y económica, que se detallan a continuación.

7.2 Factibilidad Técnica

En esta fase se Identifican y especifican los dispositivos, equipamiento y software necesarios para el desarrollo y operación del sistema a desarrollar.

En cuanto a la viabilidad técnica, la organización debe contar o adquirir las siguientes herramientas de Hardware y Software necesario que hace posible realizar el nuevo sistema.

- **Hardware**

Para estimar los requerimientos mínimos del hardware, se han considerado los siguientes requisitos en base a las exigencias mínimas de los programas que necesita la aplicación, para que este funcione correctamente.

Programa	Procesador	RAM	Espacio en Disco Duro
PHP	450 Mhz	64 Mb	20 Mb
Java	1 GHz	512 Mb	850 Mb
PostgreSql	500 Mhz	64 MB	2GB
Apache	1,0 GHz	256 Mb	1700 Mb
Ubuntu	300 Mhz	64 Mb	1,5 Gb

Resumiendo de una manera más compacta y didáctica la tabla anterior , sería algo como :

Equipo	Descripción	Cantidad
Servidor	Computador procesador Intel Pentium III 1,0 GHz, 512 Mb RAM, 200 Mb memoria virtual mínimo, espacio en Disco Duro 7,5 GB (NTFS) con tarjeta de red Ethernet 100mbps, o superior.	1
Computador	Computador procesador Intel Pentium III 1,0 GHz, 64 Mb RAM, espacio en Disco Duro 1,5 GB (NTFS) con tarjeta de red Ethernet 100mbps, o superior.	4
Monitor	Benq Monitor LED 15.6	4
Dispositivo de Entrada	Teclado y Mouse	4

- **Arquitectura Cliente-Servidor Web**

Para la ejecución de esta aplicación es necesaria la arquitectura cliente-servidor mediante la utilización de tecnologías de servidor. Esta arquitectura consiste en la utilización de 1 o más clientes, normalmente exploradores de Internet (Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari, Opera, etc.), que mediante una conexión en red se conectan a un servidor en el cual reside la aplicación. En donde estos se comunican y realizan diferentes transacciones.

Debido a que la institución cuenta con las herramientas de hardware, software, y la arquitectura cliente-servidor, se considera factible técnicamente.

7.3 Factibilidad operativa

Como el proyecto nace de la necesidad de la universidad en mejorar su proceso actual de control y gestión de proyectos, y según los requerimientos tomados se puntualizan los siguientes términos del sistema relacionados con el nivel operativo del mismo:

- El sistema está pensado para ser intuitivo y amigable con el usuario.
- El sistema reducirá los tiempos en los procesos de canalización de información por parte de los docentes, al personal de Extensión.
- El sistema permite recopilar la información relevante según los casos.
- El sistema permite acceso desde diferentes plataformas y equipos.
- El sistema permite llevar un control de los presupuestos pactados para los proyectos.

En definitiva las mejoras que otorga el sistema, sumado a la facilidad de uso (por ende rápida curva de aprendizaje) y los planes de capacitación y puesta en marcha del sistema prevén que el sistema es factible en términos operacionales.

7.4 Factibilidad Económica.

A continuación, se presenta un análisis para identificar las necesidades de inversión de recursos tecnológicos y humanos que demanda la aplicación. También se incluye un estudio de los beneficios en cuanto a los ahorros tras la implantación del sistema.

7.4.1 Costos

- **Costos por Hardware**

Equipo	Descripción	Precio	Cantidad	Costo Total
Servidor	AMD Turion II Neo N40L 1,5 GHz, 2 GB RAM, Disco Duro 250 Gb.	\$231.590	1	\$241.590
Computador Mouse Teclado	<ul style="list-style-type: none"> ▪ Intel Celeron 743 CULV 1,3 GHz, 1GB SoDIMM DDR3, 320 GB 5400rpm. ▪ Teclado: Gear Teclado USB. ▪ Mouse: Mouse Óptico USB. 	\$168.390	4	\$643.560
Monitor	<ul style="list-style-type: none"> ▪ Benq Monitor LED 15.6" G615HDPL. ▪ Modelo: G615HDPL. ▪ Resolución: 1366 x 768. 	\$45.290	4	\$151.160

Los costos de hardware ascienden a \$1.036.310.

- **Costos por Software**

Nombre	Descripción	Precio	Cantidad	Costo Total
Netbeans.	Marco de trabajo de programación.	\$0	1	\$0
Apache	Servidor web HTTP.	\$0	1	\$0
PHP	Software para tomar decisiones, basada en cubo de datos (Data Warehouse).	\$0	1	\$0
PostreSql	Sistema de gestión de base de datos.	\$0	1	\$0

Ubuntu 9.04	Sistema Operativo.	\$ 0	1	\$ 0
-------------	--------------------	------	---	------

Los costos por hardware corresponden a \$0, debido a que la institución cuenta con las herramientas necesarias para la instalación y uso de la aplicación. Con respecto al Software de la aplicación su costo es \$0, pues todas las herramientas utilizadas son open Source (Licencia Libre).

- **Costos por Mano de Obra.**

Como es bien sabido en la ingeniería de software, es bastante difícil estimar gastos lo más exacto posible, por este motivo es imprescindible tener un buen plan, ya que, en muchas ocasiones la planificación es una tarea subjetiva. Uno de los aspectos que dificultan la labor de administradores y jefes de proyecto es la difícil tarea de realizar una estimación de costos y plazos realista.

Para llevar a cabo esta tarea se utiliza el principal manejador de costo, es decir el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad [1].

Para obtener los costos del desarrollo y el precio de venta del producto se utilizan las siguientes formulas.

CF	Costo Fijo = Suma de los Costos Fijos Totales por Actividad
CV	Costo Variable = Transporte + Manuales + Luz
Q	Cantidad a Producir
CT	Costo Total = CV * Q + CF
CU	Costo Unitario = Costo Total/ Q
h	Porcentaje Ganancia
PV	Precio Venta = Cu * (1+h)

- Costos por Funcionario.

Productividad.

En las siguiente tablas se muestra información, de los gastos directos que significa producir nuestro sistema. Definiendo los gatos de los diferentes profesionales , como son el arquitecto del sistema, el jefe de proyecto y el desarrollado. Para cumplir a cabo nuestros objetivos y lograr un cálculo efectivo y exacto de los costos de desarrollo. Supondremos que contamos con un equipo encargado para este proyecto.

A continuación una tabla de valores con los profesionales que deberían intervenir en un proyecto, como el que se desarrolla.

		Mensual	Hora
A	Arquitecto	\$ 2.000.000	\$ 10.370
JP	Jefe Proyecto	\$ 1.600.000	\$ 8.296
D	Desarrollador	\$ 900.000	\$ 4.667

Nota: Valores obtenidos de la Tabla Sueldos Líquidos de Cargos TIC 2012 | IT Hunter [2].

Según Estimaciones previas, expuestas en carta Gantt.

Actividad	Duración			Recursos	Costos Total Horas
	Días	Horas Diaria	Horas * Día		
Estudio Situación Actual	10	6	60	1 JP, 1 A	\$ 1.119.960
Especificación de Requisitos y Prototipo	25	6	150	1D	
Análisis y Especificación	10	6	60	1D	\$280.020
Diseño y Construcción	5	6	30	1 D	\$ 140.010
Evaluación y Modificación	5	6	30	1 D	\$ 140.010
Documentación de Requerimientos	5	6	30	1 JP, 1 A	\$ 559.980
Diseño Técnico	40		240		
Análisis Base Datos	10	6	60	1 A, 1 D	\$902.220
Diseño Base Datos	15	6	90	1 D	\$ 420.030
Diseño Sistema (Diagramas)	10	6	60	1 D	\$ 560.000
Diseño Interfaz de Usuario	5	6	30	1 D	\$ 140.010
Programación (Codificación)	40		165		
Construcción Base de Datos	5	6	30	1 D	\$ 280.000
Construcción Plataforma	30	4	120	1 D	\$ 560.040
Implementación y Mantenimiento	5	3	15	1 D	\$ 140.000
Total					\$ 5.242.281

➤ Costos Variables

	Total
Transporte	\$ 20.000
Luz	\$ 16.000
Materiales	\$ 10.000

○ Resultado

	Resultado
<i>CF</i>	\$ 5.242.281
<i>CV</i>	\$ 46.000
<i>Q</i>	1
<i>CT</i>	\$ 5.288.281
<i>CU</i>	\$ 5.288.281
<i>h</i>	20%
<i>PV</i>	\$ 6.345.937

Resumiendo, nuestra plataforma tiene un costo de desarrollo de \$ 6.345.937, sin embargo, el desarrollo de este proyecto actualmente tiene un costo \$0.

7.5 Conclusión de la factibilidad

Según los estudios realizados anteriormente pasando por cada uno de los análisis que implica el desarrollo de un software tenemos lo siguiente:

El proyecto es claramente sustentable desde el punto de vista técnico, pues la mayoría de requerimiento de hardware están presente en la universidad y con respecto al software necesario para el uso del sistema, éste está presente en la mayoría de los PC's actuales y en el caso que no estuviese , no habría problema , pues todo el software es gratuito.

La factibilidad operativa claramente, puede un punto débil , ya que esta depende principalmente de los usuario que usan el sistema, sin embargo, nuestro sistema presentará grafica amistosa al usuario .Con el fin de que este se adapte rápidamente al uso continuo del la plataforma.

Durante el estudio de la factibilidad económica se expuso de una manera clara y concisa, basando en las tablas de sueldos publicadas, por *IT Hunter*. A pesar de su costo desarrollo , es te viable , pues el costo final es cero.

Finalmente podemos decir con bases solidas que nuestro proyecto es factible , pues como se demostró anteriormente ,a pesar de demostrar ciertas debilidades , supera con creces las fortalezas que presenta.

CAPÍTULO VIII

Análisis.

8.1 Introducción.

Para comenzar con la etapa de análisis, debemos tener claro y bases solidas de los que comprende nuestro sistema, para no cometer errores durante su construcción. Debido a este motivo se puso mucho énfasis en el trabajo de tomar los requerimientos con cuidado para evitar errores que pudiesen retrasar la entrega del software. Una vez comprendido todo lo anterior y realizado una descripción de nivel de sistema, de lo que este puede o debe hacer, para luego pasar a la codificación del software.

8.2 Diagrama de Casos de Uso

Los casos de usos son la clave para comprender lo que quieren los usuarios, también ofrecen un buen vehículo para la planificación de proyectos, ya que controlan el desarrollo. Además sirven para la comunicación de los elementos superficiales y las cuestiones más profundas .De hecho se les considera como una de las tareas principales durante la fase de elaboración. Dado su importancia se ha diseñado el siguiente caso de uso.

8.2.1 Actores

- **Administrador:** Persona que tiene la responsabilidad de ejecutar, mantener, operar y asegurar el correcto funcionamiento del sistema. Posee todos los privilegios de los usuarios (actores).
 - Privilegios dentro del sistema.
 - Registrar funcionario.
 - Registrar Proyecto.
 - Registrar evento.
 - Registrar tarea.
 - Registrar Subtarea.
 - Ver tareas
 - Ver historial de proyectos.
 - ver Miembros de los proyectos presentes y pasados.
 - Ver datos del proyecto.
 - Modificar tareas.
 - Modificar subtareas.

- Ver eventos.
 - Registrar nuevo usuario.
 - Ver comunicados de prensa.
-
- **Docentes(Jefe Proyectos):** Persona de la universidad encargado del proyecto, para un entendimiento mejor se le llamo docente, en vez de jefe de proyecto.
 - Privilegios dentro del sistema.
 - Registrar funcionario.
 - Registrar evento.
 - Registrar tarea.
 - Registrar Subtarea.
 - Ver tareas
 - Ver Miembros de su proyecto.
 - Ver datos del proyecto.
 - Modificar tareas.
 - Modificar subtareas.
 - Ver eventos.
 - Redactar comunicado a la prensa
-
- **Equipo de Proyecto:** Personas participantes dentro del proyecto.
 - Privilegios dentro del sistema.
 - Registrar tarea.
 - Registrar Subtarea.
 - Ver tareas
 - Ver datos del proyecto.
 - Ver eventos.
 - Redactar comunicado a la prensa.
-
- **Personal de Extensión:** Este es el encargado de ir monitoreando cada una de las etapas del o los proyectos en curso.

- Privilegios dentro del sistema.
 - Ver tareas
 - Ver historial de proyectos.
 - ver Miembros de los proyectos presentes y pasados.
 - Ver datos del proyecto.
 - Ver eventos.
 - Ver comunicados de prensa.

Prensa: departamento de comunicación y prensa encargado de ver los comunicados de los proyectos en curso.

- Privilegios dentro del sistema.
 - Ver comunicados de prensa.

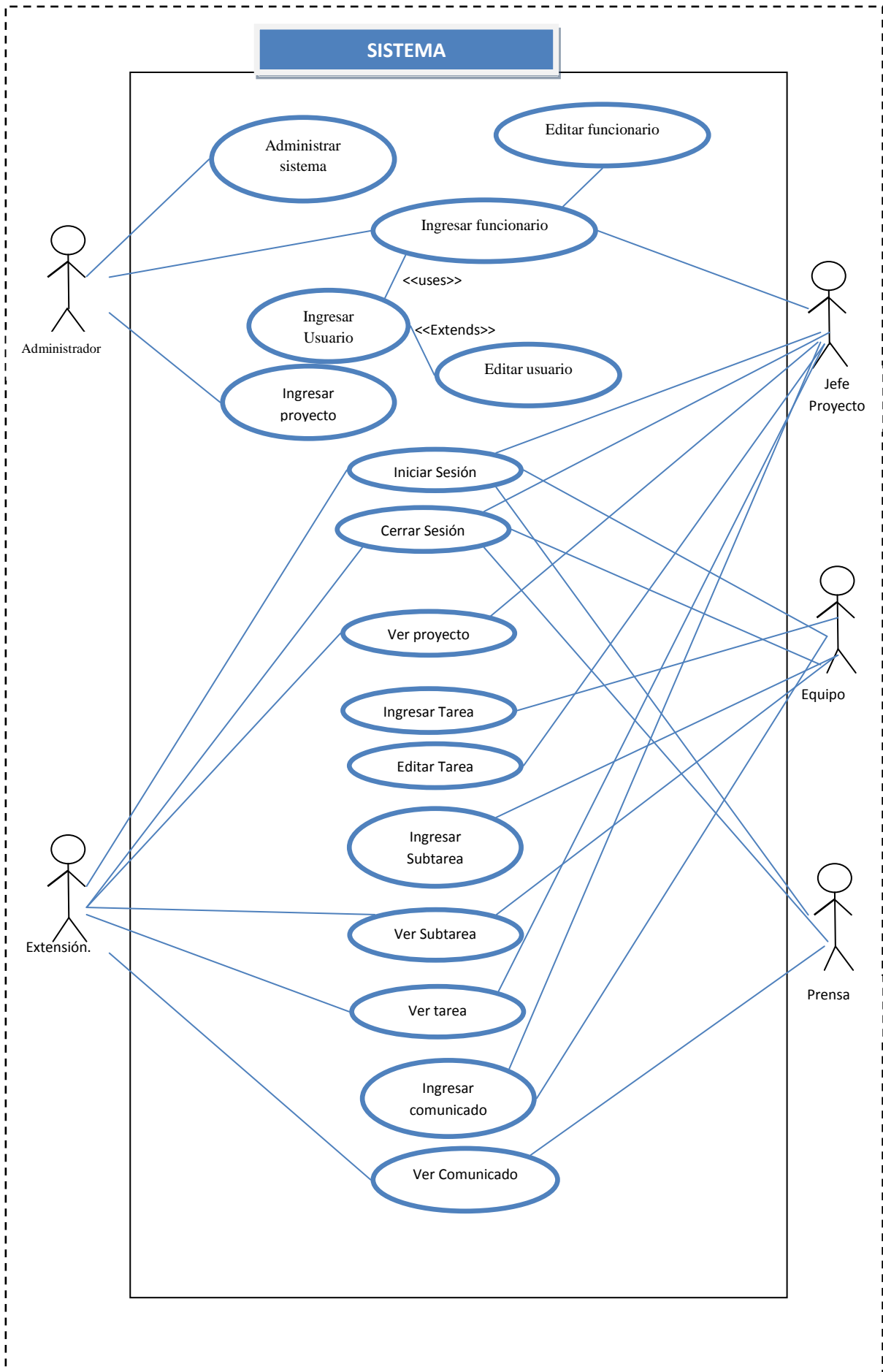
8.2.2 Diagrama de Casos de Uso y Descripción

El presente Diagrama de Casos de Uso representa las funcionalidades destacadas dentro del sistema.

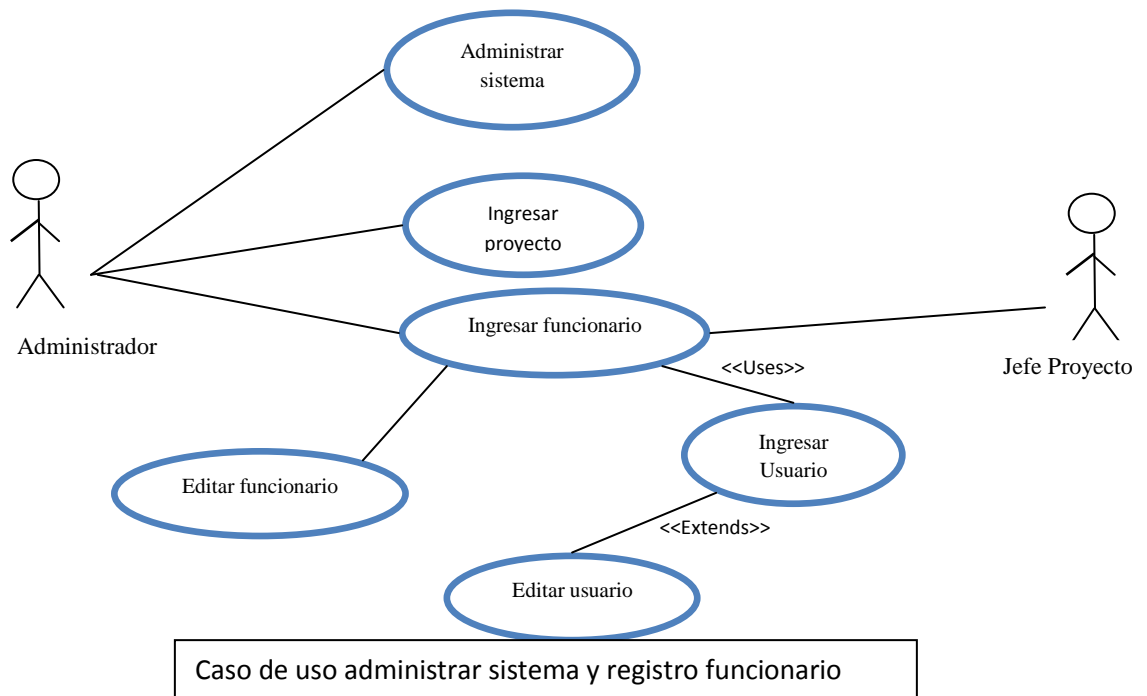
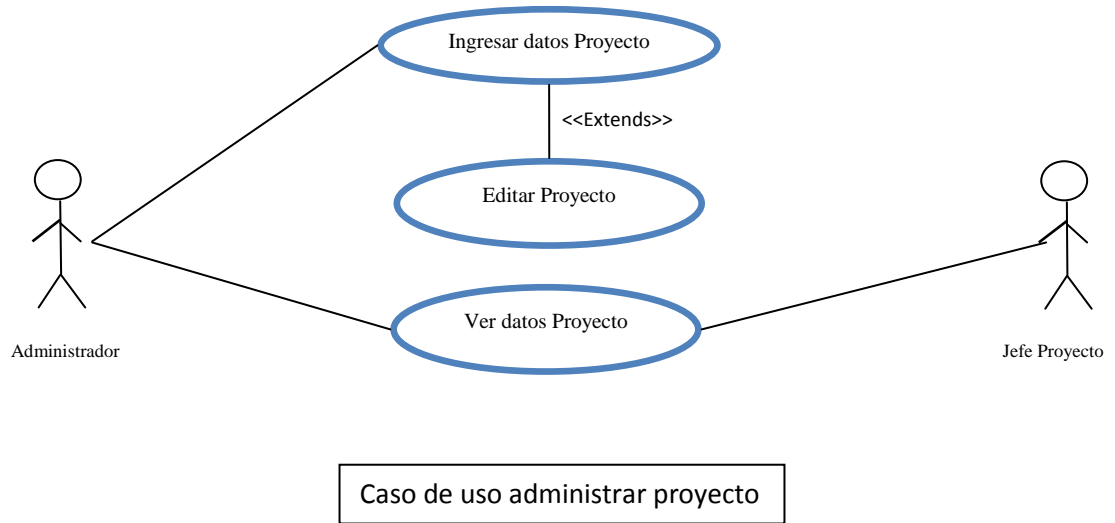
Para la construcción del diagrama se ha considerado el uso de los “Extends” y “Uses” de la siguiente manera, basados en [3].

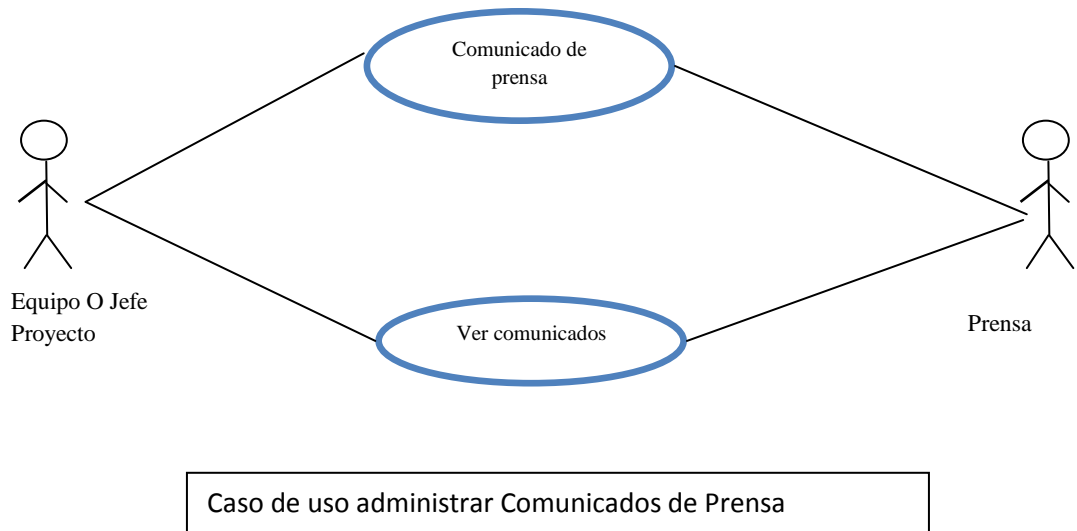
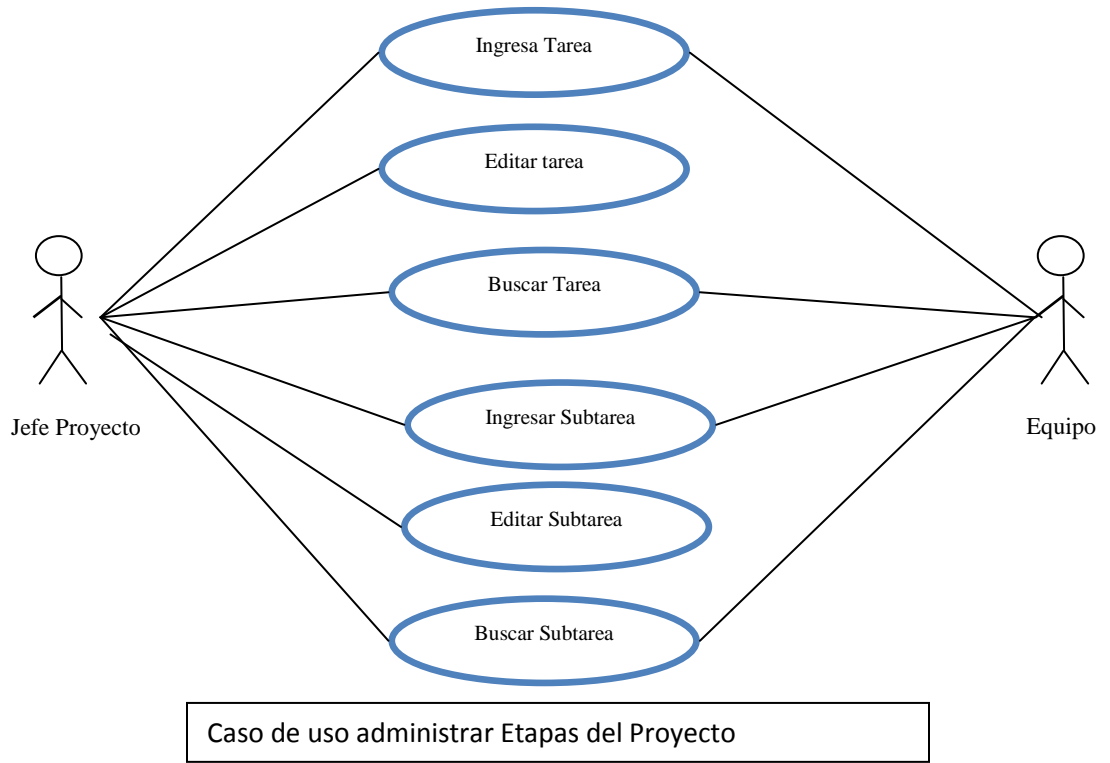
Extends: Cuando se tiene un caso de uso que es similar a otro, pero que hace un poco más, es decir, donde existe una relación de dependencia entre dos casos de uso que denota que un caso de uso es una especialización de otro.

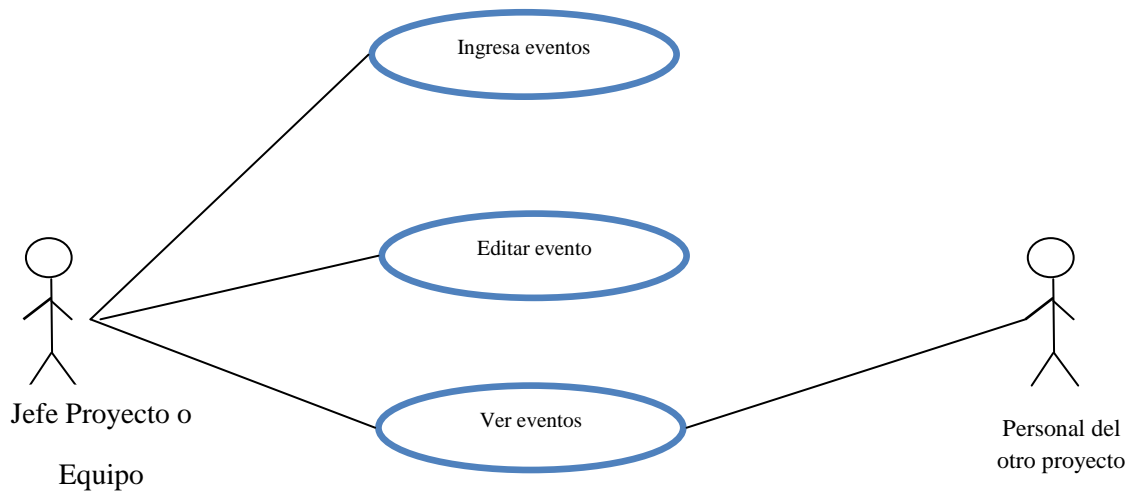
Uses: Cuando se tienen una porción de comportamiento que es similar en más de un caso de uso y no se quiere copiar la descripción de tal conducta, este implica obligación entre los caso de uso utilizados.



8.2.2.1 Diagrama de Casos específicos.







Caso de uso administrar Eventos

8.2.3 Especificación de los Caso de Uso.

Identificación	CU_01
Caso de Uso	Iniciar Sesión.
Descripción	Permite ingresar al sistema
Pre-Condiciones	1. Estar Registrado en el sistema.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor ingresa nombre de usuario, contraseña. 2. El sistema valida los datos ingresados por el actor. 3. El sistema muestra los perfiles (roles) que el usuario posee. 4. El actor escoge un perfil. 5. El sistema muestra los módulos que tiene el perfil seleccionado. 	
Flujo de Eventos Alternativos	
<ol style="list-style-type: none"> 1. Paso 2 del FEA: Usuario no registrado. <ol style="list-style-type: none"> 1.1. Sistema muestra un mensaje donde se le indica que debe estar registrado por el Administrador. 2. Paso 2 del FEA: Usuario Inactivo. <ol style="list-style-type: none"> 2.1. Sistema muestra un mensaje donde se le informa su estado en el sistema y le indica que debe ser activado por el administrador. 	
Post-Condiciones	Ingreso con éxito.

Tabla: Caso de Uso (Iniciar Sesión)

Identificación	CU_02
Caso de Uso	Cerrar Sesión.
Descripción	Permite cerrar el sistema .
Pre-Condiciones	<ol style="list-style-type: none"> 1. Estar Registrado en el sistema. 2. Estar dentro del sistema.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. Actor selecciona la opción cerrar sesión 2. El sistema cierra sesión correctamente. 3. Sistema muestra un mensaje de despedida. 	
Flujo de Eventos Alternativos	
<ol style="list-style-type: none"> 1. Paso 1 del FEA: Actor cierra sección sin guardar información. <ol style="list-style-type: none"> 1.1. Sistema borra la información ingresada. 2. Paso 1.1 del FEA: Actor decide cancelar el cierre de sesión <ol style="list-style-type: none"> 2.1. Sistema cancela el cierre de sesión. 	
Post-Condiciones	---

Tabla: Caso de Uso (Cerrar Sesión) .

Identificación	CU_03
Caso de Uso	Ingresar Jefe Proyecto
Descripción	Ingresar al Jefe Proyecto
Pre-Condiciones	1. Estar en el sistema como Administrador.
Flujo de Eventos Básicos.	
<ol style="list-style-type: none"> 1. Se ingresa al docente. 2. Asigna como el rol jefe Proyecto. 	
Flujo de Eventos Alternativos	
<p>Paso 1 del FEA: El Docente ya se encuentra dentro del sistema, pero desactivado</p> <ol style="list-style-type: none"> 2.2 Se activa el docente. <p>Paso 2 del FEA : El docente ya se encuentra dentro del sistema pero sin el cargo de jefe de Proyecto.</p> <ol style="list-style-type: none"> 2.2 Se asigna el rol de jefe de Proyecto dentro del sistema. 	
Post-Condiciones	Docente Ingresado.

Tabla: Caso de Uso (ingresar Jefe Proyecto).

Identificación	CU_04
Caso de Uso	Ingresar Proyecto
Descripción	Permite asignar Proyecto a un docente
Pre-Condiciones	1. Estar ingresado el docente
Flujo de Eventos Básicos	
1. Asignar el proyecto y sus datos a un docente.	
Flujo de Eventos Alternativos	
Post-Condiciones	---

Tabla: Caso de Uso (ingresar Proyecto).

Identificación	CU_05
Caso de Uso	Crear Usuario.
Descripción	Permite crear cuenta de usuario.
Pre-Condiciones	<ol style="list-style-type: none"> 2. Estar ingresado en el sistema como Administrador. 3. Estar en el módulo de Administración.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la opción registrar cuentas de usuario. 2. Sistema muestra la interfaz de crear cuentas de usuario. 3. El actor ingresa al usuario 4. Sistema verifica los datos. 5. Sistema almacena los datos ingresados. 6. Sistema indica que el usuario ha sido creado correctamente. 	
Flujo de Eventos Alternativos	
<ol style="list-style-type: none"> 1. Paso 1 del FEA: Información errónea. <ol style="list-style-type: none"> 1.1. El sistema invalida el ingreso del usuario. 1.2. El sistema muestra un mensaje indicando el error del ingreso. 	
Post-Condiciones	El usuario a ingresar no exista en el sistema.

Tabla: Caso de Uso (Crear Usuario).

Identificación	CU_06
Caso de Uso	Crear nueva Tarea.
Descripción	Permite crear Tareas del proyecto.
Pre-Condiciones	<ol style="list-style-type: none"> 1. .Estar ingresado en el sistema como jefe Proyecto. 2. Estar en el módulo de crear Tarea. 3. El proyecto al cual se asigna la tarea, debe estar ingresado
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la Modulo Administración de Proyectos. 2. El actor selección la opción de crear nueva tarea. 3. El actor ingresa la nueva tarea 4. Sistema verifica los datos. 5. Sistema almacena los datos ingresados. 6. Sistema indica que el usuario ha sido creado correctamente. 	
Flujo de Eventos Alternativos	
<p>Paso 1 del FEA: Información errónea.</p> <ol style="list-style-type: none"> 1.3. El sistema invalida el ingreso del usuario. 1.4. El sistema muestra un mensaje indicando el error del ingreso de la tarea. <p>Paso 2 del FEA: Presupuesto Adjudicado supera al presupuesto sumado de las tareas.</p> <ol style="list-style-type: none"> 1.3.-Se modifica el o los gastos de la tarea con el fin de que sea igual o menor al presupuesto del proyecto. 	
Post-Condiciones	La tarea existe dentro del sistema.

Tabla : Caso de Uso (Crear Tarea).

Identificación	CU_07
Caso de Uso	Crear nueva Subtarea.
Descripción	Permite crear subtareas del proyecto.
Pre-Condiciones	<ol style="list-style-type: none"> 1. Estar ingresado en el sistema como jefe Proyecto o miembro . 2. Estar en el módulo de crear subtarea. 3. La tarea o etapa en que se realiza la subtarea, debe estar registrada
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la Modulo Administración de Proyectos. 2. El actor selección la opción de crear nueva Subtarea. 3. El actor ingresa la nueva Subtarea 4. Sistema verifica los datos. 5. Sistema almacena los datos ingresados. 6. Sistema indica que el usuario ha sido creado correctamente. 	
Flujo de Eventos Alternativos	
Paso 1 del FEA: Información errónea. <ol style="list-style-type: none"> a. El sistema invalida el ingreso del usuario. b. El sistema muestra un mensaje indicando el error del ingreso de la Subtarea. 	
Post-Condiciones	La Subtarea existe dentro del sistema.

Tabla : Caso de Uso (Crear Subtarea).

Identificación	CU_08
Caso de Uso	Crear Comunicado.
Descripción	Permite crear Comunicados.
Pre-Condiciones	<ol style="list-style-type: none"> 1. Estar ingresado en el sistema como jefe Proyecto o miembro .
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la Modulo Administración de Prensa. 2. El actor selección la opción de crear Comunicado. 3. El actor ingresa el nuevo Comunicado 	
Flujo de Eventos Alternativos	
Post-Condiciones	---

Tabla : Caso de Uso (Crear Comunicado Prensa).

Identificación	CU_09
Caso de Uso	Ver Historial de Proyecto
Descripción	Permite ver los proyectos registrados
Pre-Condiciones	1.El usuario debe estar registrado como administrado o personal o Supervisor .
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la Modulo Administración de Proyectos. 2. El actor selección la opción de historial. 	
Flujo de Eventos Alternativos	
Post-Condiciones	---

Tabla: Caso de Uso (ver historial de Proyectos).

Identificación	CU_10
Caso de Uso	Ver datos del Proyecto
Descripción	Permite ver los datos de los proyecto en que esta participando el usuario.
Pre-Condiciones	1. El usuario debe estar registrado como Jefe Proyecto.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la Modulo Administración de Proyectos. 2. El actor selección la opción de historial de proyectos. 	
Flujo de Eventos Alternativos	
Post-Condiciones	---

Tabla: Caso de Uso (ver datos del Proyecto).

Identificación	CU_11
Caso de Uso	Ver Tareas
Descripción	Permite ver los datos de las tareas
Pre-Condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado. 2. La Tarea debe estar ingresada.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor selecciona la Modulo Administración de Proyectos. 2. El actor selección la opción ver Tareas. 	
Flujo de Eventos Alternativos	
Paso 1 del FEA: El proyecto no tiene tareas ingresadas 1.1 El Jefe de proyecto o el Equipo del Proyecto debe ingresar las Tareas del Proyecto.	
Post-Condiciones	---

Tabla: Caso de Uso (ver Tareas del Proyecto).

Identificación	CU_12
Caso de Uso	Ver Subtareas
Descripción	Permite ver los datos de las Subtareas
Pre-Condiciones	<ol style="list-style-type: none"> 3. El usuario debe estar registrado. 4. La Subtarea debe estar ingresada.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 3. El actor selecciona la Modulo Administración de Proyectos. 4. El actor selección la opción ver Subtareas. 	
Flujo de Eventos Alternativos	
Paso 1 del FEA: El proyecto no tiene Subtareas ingresadas 1.1 El Jefe de proyecto o el Equipo del Proyecto debe ingresar las Subtareas de las Tareas del proyecto..	
Post-Condiciones	---

Tabla: Caso de Uso (ver Subtareas del Proyecto).

Identificación	CU_13
Caso de Uso	Redactar Comunicado
Descripción	Permite Redactar comunicados para la Prensa de la Universidad.
Pre-Condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado como jefe Proyecto o Equipo. 2. Debe existir un proyecto al cual asignar el comentario.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 5. El actor selecciona la Modulo Administración de Proyecto. 6. El actor selección la opción redactar Comunicado de Prensa. 	
Flujo de Eventos Alternativos	
Post-Condiciones	--

Tabla: Caso de Uso (Redactar comunicado de prensa).

Identificación	CU_14
Caso de Uso	ver Comunicado
Descripción	Permite ver comunicados para la Prensa de la Universidad.
Pre-Condiciones	<ol style="list-style-type: none"> 1. El usuario debe estar registrado como administrador o Prensa.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1.- El actor selecciona la Modulo Administración de Prensa. 2.- El actor selección la opción ver Comunicado de Prensa. 	
Flujo de Eventos Alternativos	
Post-Condiciones	--

Tabla: Caso de Uso (ver comunicado).

Identificación	CU_15
Caso de Uso	Registrar un evento.
Descripción	Permite registrar el evento del proyecto
Pre-Condiciones	1. El usuario debe estar registrado como Jefe Proyecto o Equipo .
Flujo de Eventos Básicos	
1.-El actor selecciona la Modulo Administración de Eventos. 2.- El actor selección la opción registrar evento.	
Flujo de Eventos Alternativos	
Paso 1 del FEA: Ya existe un evento registrado con el mismo lugar ,fecha y hora 1.1 El usuario debe buscar en el modulo ver eventos, para verificar la disponibilidad. 1.2 El usuario registra el evento con la disponibilidad, requerida para este.	
Post-Condiciones	El evento queda registrado acaparando fecha, lugar y hora.

Tabla: Caso de Uso (registrar evento).

Identificación	CU_16
Caso de Uso	Ver eventos.
Descripción	Permite visualizar los evento del proyecto
Pre-Condiciones	1.- El usuario debe estar registrado en el sistema.
Flujo de Eventos Básicos	
1.-El actor selecciona la Modulo Administración de Eventos. 2.- El actor selección la opción ver eventos.	
Flujo de Eventos Alternativos	
Post-Condiciones	-----

Tabla: Caso de Uso (ver eventos evento).

Identificación	CU_17
Caso de Uso	Administrar Sistema.
Descripción	Permite ingresar, modificar, y eliminar información del sistema.
Pre-Condiciones	1. Estar ingresado en el sistema como Administrador.
Flujo de Eventos Básicos	
<ol style="list-style-type: none"> 1. El actor modifica los datos. 2. Sistema verifica los datos. 3. Sistema actualiza los datos modificados. 4. Sistema muestra un mensaje confirmando los cambios realizados. 	
Flujo de Eventos Alternativos	
<ol style="list-style-type: none"> 1. Paso 2 del FEB: Faltan datos. <ol style="list-style-type: none"> 1.1. Sistema muestra un mensaje indicando los campos que son obligatorios. 2. Paso 2 del FEB: Datos erróneos. <ol style="list-style-type: none"> 2.1. Sistema muestra un mensaje con el error cometido. 	
Post-Condiciones	---

Tabla : Caso de Uso (Administrar Sistema)

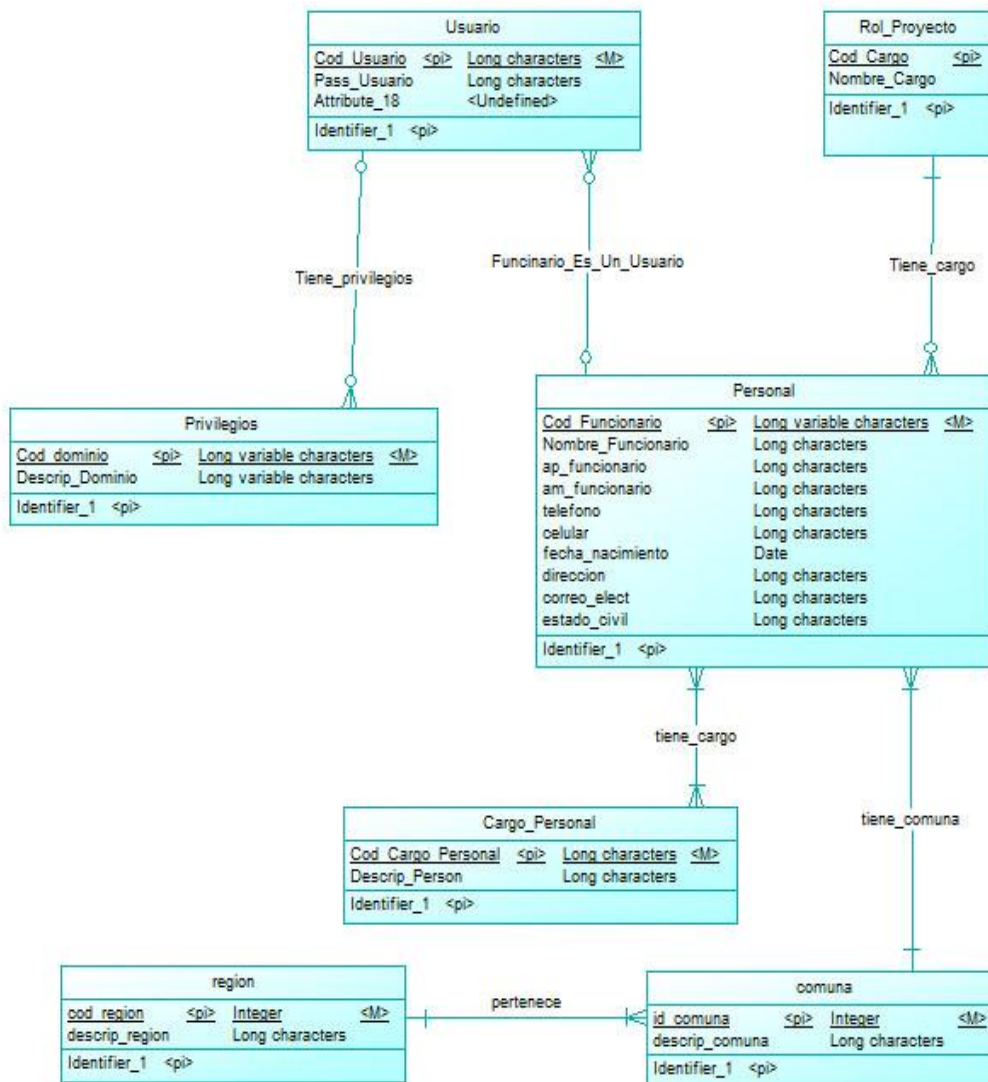
8.3 Modelamiento de datos.

El Modelo Entidad Relación está diseñado según la situación que se presenta a continuación:

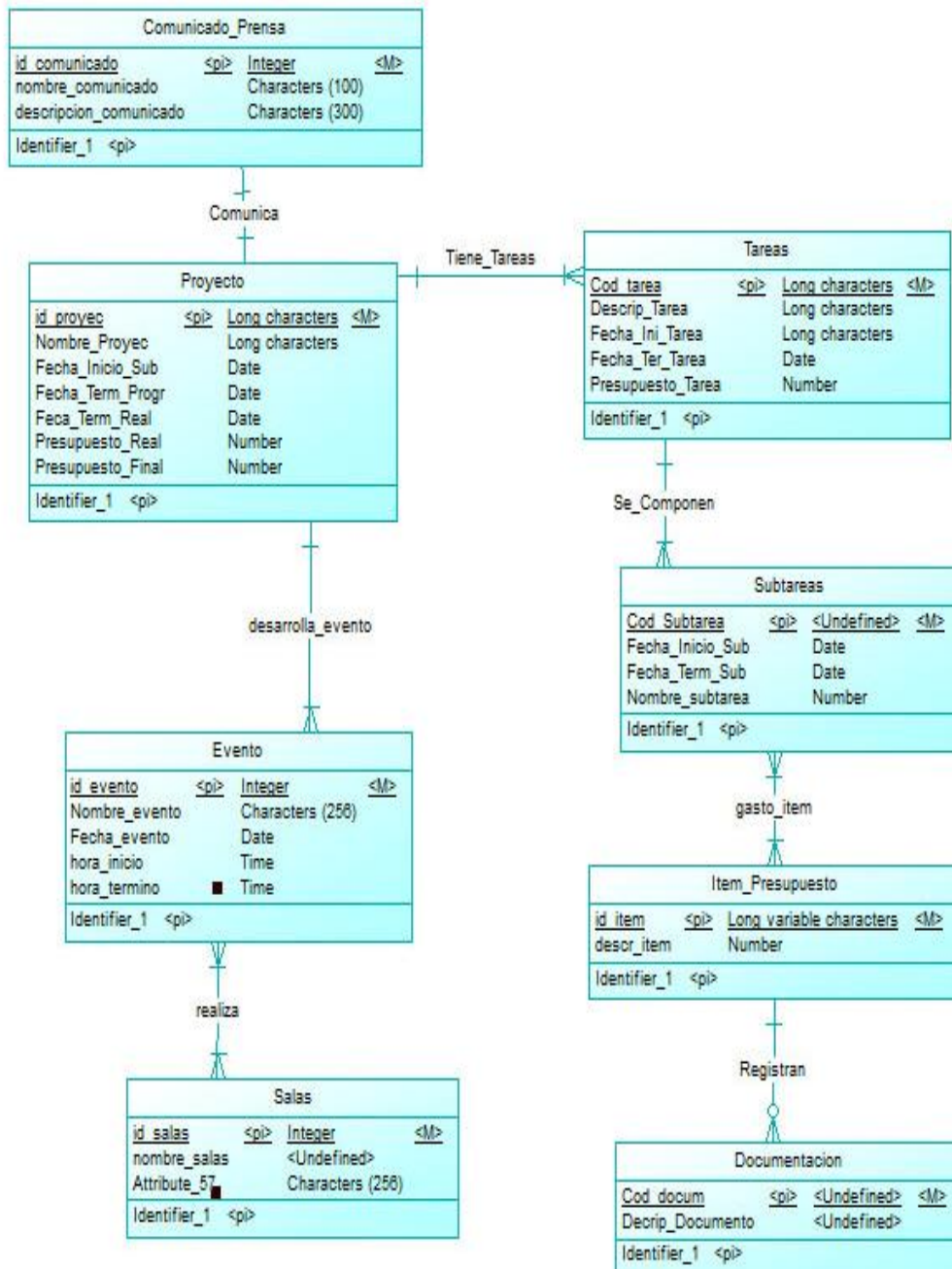
Actualmente, la universidad del Bío-Bío y en especial la Unidad de Extensión, administra de forma física los diferentes proyecto que en ella se desarrollan, esto quiere decir que se lleva una carpeta con todos los documento de los diferentes proyectos.

Este proceso se ha llevado de la mejor forma posible hasta ahora, sin embargo , cada vez es más difícil ya que es solo una persona la encargada de llevar un control total de los diferentes proyectos ,sus respectivas etapas y el gasto que se producen ellas.

Modelo Entidad Relación (MER)



Modelo Entidad Relación (MER) Parte 2.



8.3.1 Descripción de Entidades y Relaciones del MER

Entidades:

Personal: Son los docentes que participan dentro de los proyectos.

Usuario: Es donde se guardan los datos de acceso al sistema.

Privilegios: Son los roles dentro del sistema.

Cargo Personal: El cargo de la persona ya sea docente o alumno.

Comuna: Corresponde al nombre de las comunas, cada comuna pertenece a una ciudad.

Región: Corresponde al nombre de las regiones, cada región posee muchas ciudades.

Rol del Proyecto: Es el cargo que ocupa dentro del proyecto(jefe proyecto, secretario, etc).

Proyecto: Es el proyecto que se realiza .

Comunicado Prensa: Párrafo que se escribe con el fin de publicar ,sus resultado obtenidos por los proyectos a la comunidad.

Evento: Es Suceso que se realiza, en ciertas ocasiones por fin del proyecto o es el mismo proyecto.

Lugar: Es el lugar físico adaptado para realizar eventos a la comunidad universitaria o afuerina.

Tareas: Son las diferentes etapas que tienen los proyecto, estas pueden ser limitadas por tiempo o presupuestos.

Subtareas: Son las pequeñas misiones que ocurren dentro de las etapas o tareas de los proyectos.

Ítem Presupuesto: Es el tipo de gasto que se realiza durante la Subtarea.

Documentación: Son los documentos que puede generar el ítem presupuesto , ya sea ,boletas, facturas, etc.

Relaciones.

Desarrolla evento: Relación que identifica al evento y al proyecto.

Realiza: Es en la encargada de unir los eventos y los lugares donde estos ocurren.

Comunica: Es para identificar que comunicado es de que proyecto.

Tiene tareas: Relación encargada para asignar a tarea a sus respectivo proyecto.

Se compone: relación encargada de unir las Subtareas a sus respectivas tareas.

Gasto Ítem: Es la encargada de la identificación del tipo de ítem asociada a un gasto que se produjo en la Subtarea.

Registran: Es la encargada de unir los ítem presupuestario con sus respectivos documentos que los certifiquen.

Tiene Privilegios: Es la encargada de asignar los roles dentro del sistema a los respectivos usuarios.

Funcionario es un : Relación encargada de unir al funcionario con su respectivo usuario.

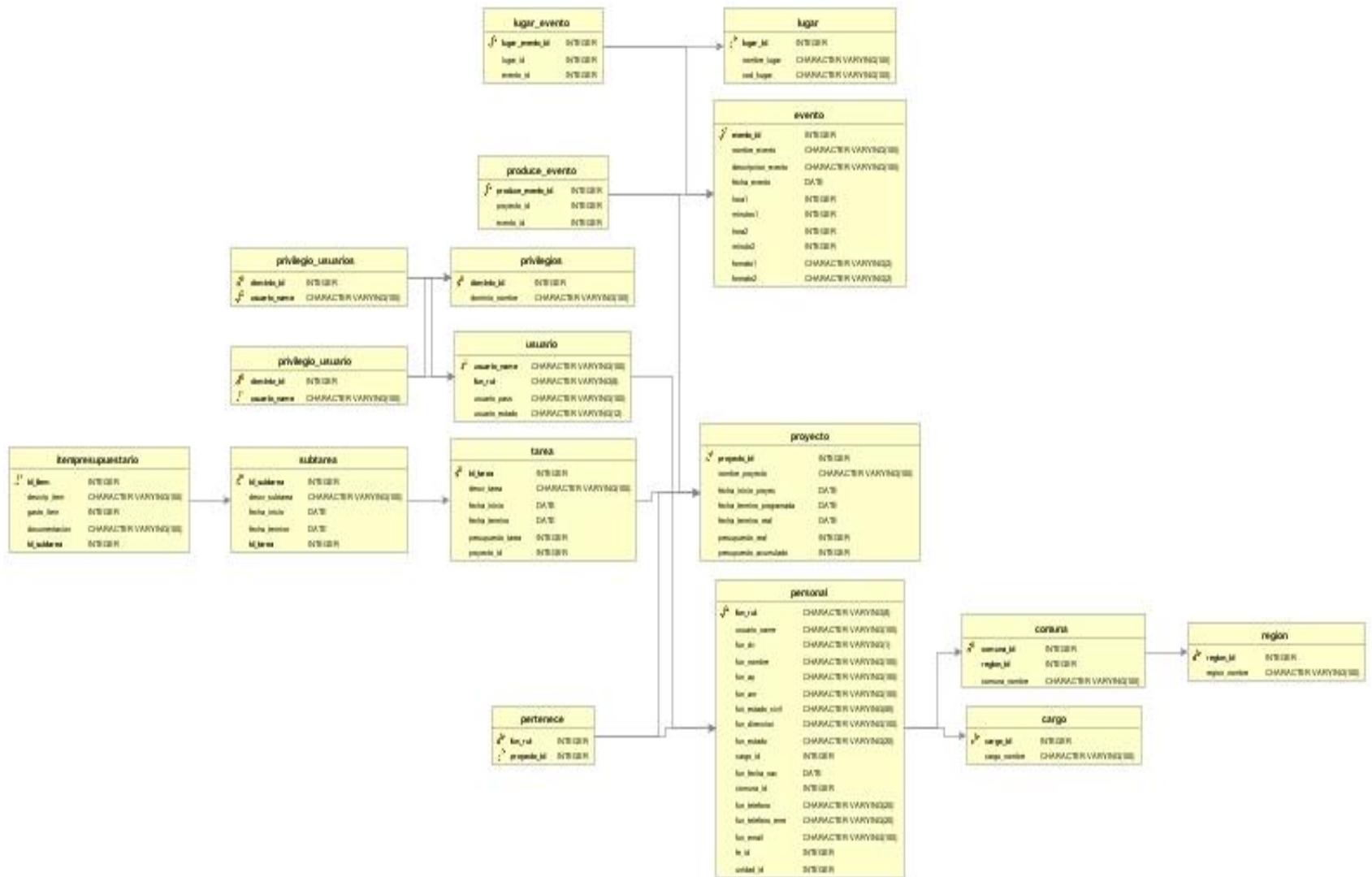
Tiene cargo: relación encargada de unir al funcionario con su cargo dentro de la universidad.

Tiene comuna: Relación que identifica la comuna del funcionario.

Pertenece: Relación que identifica la comuna a su respectiva región.

Tiene cargo _rol :Rol del personal dentro del proyecto.

Modelo Relacional.



8.3.2 Diseño Físico de la Base de datos.

El modelo físico de la base de datos (Modelo Relacional) se ha elaborado a partir del Modelo Entidad Relación.

A continuación se justifica el modelo construido, según las cardinalidades obtenidas en el MER.

Justificación de las relaciones:

- Las relaciones; **Personal, Usuario, Privilegios, Cargo Personal, Comuna, Región, Rol del Proyecto, Proyecto, Comunicado Prensa, Evento, Lugar, Tareas, Subtareas, Ítem Presupuesto, Documentación** se debe a sus respectivas entidades en el MER, ya que la regla [4] indica que cada entidad pasa a relación en el modelo relacional.
- La relaciones; **Desarrolla evento, Realiza, Tiene tareas, Se compone, Gasto Ítem, Tiene Privilegios** pasaron al modelo relacional debido a que la cardinalidad entre sus entidades es de "1:n" (uno muchos). Estas relaciones heredan las claves primarias de sus entidades.
- Las relación **Funcionario es un ,Comunica, Tiene cargo, Tiene comuna, Pertenece, Tiene cargo _rol , Registran** no pasa al modelo relacional debido a que su cardinalidad entre sus entidades es de "1:1", donde una de las entidades tiene participación opcional (cardinalidad (0,1))

CAPÍTULO IX

Diseño del Sistema.

9.1 Introducción.

Este capítulo detallaremos los niveles de diseño del sistema, desde el modelo arquitectónico, los diagramas de clase, las vistas de las pantallas o diseños de las interfaces de usuarios. Cada uno de estos modelos representa un diferente del software y una etapa distinta de implementación, aunque la mayoría se entrelazan entre sí, detallaremos estas, para hacer de manera fácil el entendimiento del sistema.

9.2 Arquitectura del sistema

Para comprender que es y el papel que desempeña la arquitectura en un sistema se cita un extracto del autor Ian Sommerville [18]:

El diseño arquitectónico es un proceso creativo en el que se intenta establecer una organización del sistema que satisfaga los requerimientos funcionales y no funcionales del propio sistema. Debido a que es un proceso creativo, las actividades dentro del proceso difieren radicalmente dependiendo del tipo de sistema a desarrollar, el conocimiento y la experiencia del arquitecto del sistema, y los requerimientos específicos del mismo. Es, por tanto, más útil pensar en el proceso de diseño arquitectónico desde una perspectiva de decisión en lugar de una perspectiva de actividades. Durante el proceso de diseño arquitectónico, los arquitectos del sistema tiene que tomar varias decisiones fundamentales que afectan profundamente al sistema y a su proceso de desarrollo. Basándose en su conocimiento y experiencia, los arquitectos del sistema tienen que responder a las siguientes cuestiones fundamentales:

Existe una arquitectura de aplicación genérica que pueda actuar como una plantilla para el sistema

- Que se está diseñando?
- Cómo se distribuirá el sistema entre varios procesadores?
- Qué estilo o estilos arquitectónicos son apropiados para el sistema?
- Cuál será la aproximación fundamental utilizada para estructurar el sistema?
- Cómo se descompondrán en módulos las unidades estructurales del sistema?
- Qué estrategia se usará para controlar el funcionamiento de las unidades del sistema?
- Cómo se evaluará el diseño arquitectónico?
- Cómo debería documentarse la arquitectura del sistema?

Para el sistema, en el caso de la arquitectura física el modelo más apropiado es el cliente servidor y la arquitectura lógica se optó por el modelo en capas, específicamente el "Modelo Vista Controlador" (MVC - Model View Controller) esto debido a que como se expuso previamente, la solución a implementar es basada en la WEB.

9.2.1 Arquitectura Física

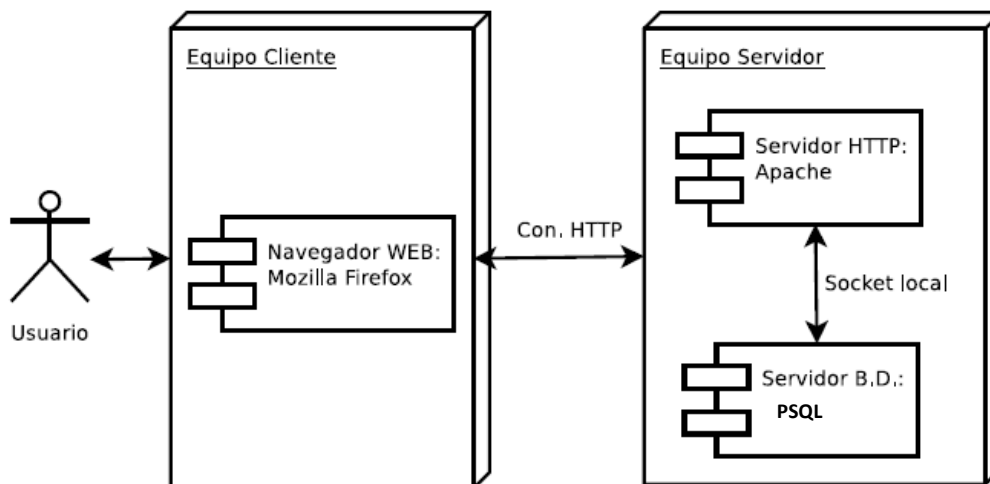


Figura : Arq. Cliente-servidor ocupada en el sistema.

9.2.2 Arquitectura Lógica.

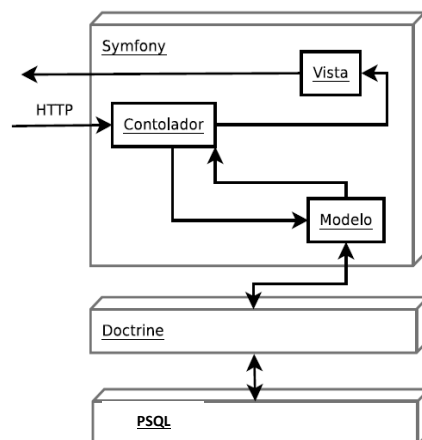


Figura 6.2: Arq. Modelo Vista Controlador ocupada en el sistema

9.3 Diagrama de Clases.

En esta sección se expone el diagrama de clases el cual será presentado en 2 partes, primero el diagrama general y luego por cada clase listada, se incluirá un diagrama exclusivo de la respectiva clase, incluyendo los métodos que contenga. Cabe destacar que el diagrama de clases expuesto está adaptado para integrarse de mejor forma al plugin sfGuard (que facilita el manejo de autenticación), es por ello que se encuentran separados los datos de acceso (sfGuardUser) de los datos del perfil (Usuario).

9.3.1 Modelo Entidades

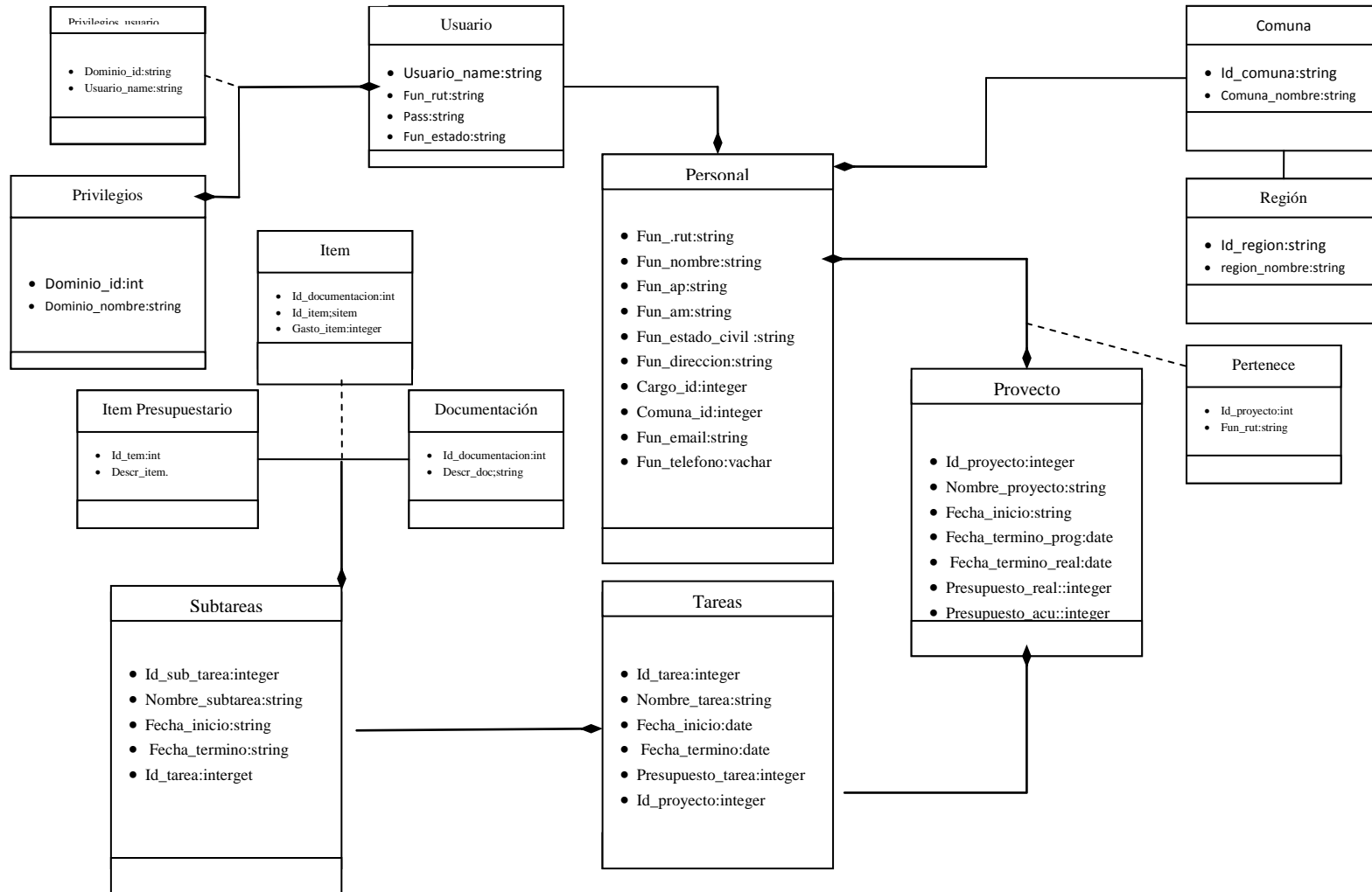


Figura modelo Entidades.

Personal
<ul style="list-style-type: none"> • Fun_rut:string • Fun_nombre:string • Fun_ap:string • Fun_am:string • Fun_estado_civil :string • Fun_direccion:string • Cargo_id:integer • Comuna_id:integer • Fun_email:string • Fun_telefono:string
<ul style="list-style-type: none"> • getfun_rut(): integer • getfun_nombre:string • getfun_ap:string • getfun_am:string • getfun_estado_civil:string • getfun_direccion:string • getcargo_id:string • getfun_direccion:string • getComuna_id:string • getfun_email:string • getfun_telefono:string • setfun_nombre:string • setfun_ap:string • setfun_am:string • setfun_estado_civil:string • setfun_direccion:string • setcargo_id:string • setfun_direccion:string • setComuna_id:string • setfun_email:string • setfun_telefono:string

Figura: clase personal

Proyecto
<ul style="list-style-type: none"> • Id_proyecto:integer • Nombre_proyecto:string • Fecha_inicio:string • Fecha_termino_prog:string • Fecha_termino_real:string • Presupuesto_real:string • Presupuesto_acu::integer
<ul style="list-style-type: none"> • Getid_proyecto:integer • Getnombre_proyecto:string • Getfecha_inicio:date • Getfecha_termino:date • getpresupuesto_real:integer • getpresupuesto_acu:integer • setid_proyecto:integer • setnombre_proyecto:string • setfecha_inicio:date • setfecha_termino:date • setpresupuesto_real:integer • setpresupuesto_acu:integer

Figura: clase Proyecto

Tareas
<ul style="list-style-type: none"> • Id_tarea:integer • Nombre_tarea:string • Fecha_inicio:date • Fecha_termino:date • Presupuesto_tarea:integer • Id_proyecto:integer
<ul style="list-style-type: none"> • getid_tarea:integer • getnombre_tarea:string • getfecha_inicio:date • getfecha_termino:date • getpresupuesto_tarea:integer • getid_proyecto:integer • setid_tarea:integer • setnombre_tarea:string • setfecha_inicio:date • setfecha_termino:date • setpresupuesto_tarea:integer • setid_proyecto:integer

Figura: clase Tareas

Subtareas
<ul style="list-style-type: none"> • Id_sub_tarea:integer • Nombre_subtarea:string • Fecha_inicio:string • Fecha_termino:string • Id_tarea:interget
<ul style="list-style-type: none"> • setid_sub_tarea:integer • setnombre_subtarea:string • setfecha_inicio:string • setfecha_termino:string • setid_tarea:interget • getid_sub_tarea:integer • getnombre_subtarea:string • getfecha_inicio:string • getfecha_termino:string • getid_tarea:interget

Figura: clase Subtareas

Usuario
<ul style="list-style-type: none"> • Usuario_name:string • Fun_rut:string • Pass:string
<ul style="list-style-type: none"> • setusuario_name:string • setfun_rut:string • setpass:string • getusuario_name:string • getfun_rut:string • getpass:string

Privilegios
<ul style="list-style-type: none"> • Dominio_id:int • Dominio_nombre:string
<ul style="list-style-type: none"> • setdominio_id:int • setdominio_nombre:string • getdominio_id:int • getdominio_nombre:string

Privilegios usuario
<ul style="list-style-type: none"> • Dominio_id:string • Usuario_name:string
<ul style="list-style-type: none"> • setdominio_id:string • setusuario_name:string • getdominio_id:string • getusuario_name:string

Comuna
<ul style="list-style-type: none"> • Id_comuna:string • Comuna_nombre:string
<ul style="list-style-type: none"> • setid_comuna:string • setcomuna_nombre:string • getid_comuna:string • getcomuna_nombre:string

Figura: clase usuario

Figura: clase privilegios

Figura: clase privilegios usuarios

Figura: clase comuna

Región
<ul style="list-style-type: none"> • Id_region:string • region_nombre:string
<ul style="list-style-type: none"> • setid_region:string • setregion_nombre:string • getid_region:string • getregion_nombre:string

Pertenece
<ul style="list-style-type: none"> • Id_proyecto:int • Fun_rut:string
<ul style="list-style-type: none"> • getid_proyecto:int • getfun_rut:string • setid_proyecto:int • setfun_rut:string

Documentación
<ul style="list-style-type: none"> • Id_documentacion:int • Descr_doc:string
<ul style="list-style-type: none"> • setId_documentacion:int • setescr_doc:string • getescr_doc:string • getid_documentacion:int

Item
<ul style="list-style-type: none"> • Id_documentacion:int • Id_item;sitem • Gasto item:integer
<ul style="list-style-type: none"> • setid_documentacion:int • setid_item;sitem • setgasto_item:integer • getid_documentacion:int • getid_item;sitem • getgasto_item:integer

Item Presupuestario
<ul style="list-style-type: none"> • Id_tem:int • Descr_item.
<ul style="list-style-type: none"> • setid_tem:int • setdescr_item. • getid_tem:int • getdescr_item.

Figura: clase región

Figura: clase pertenece

Figura: clase Documentación

Figura: clase item

Figura: clase ítem
Presupuesto

9.4 Diseño Interfaz y Navegación

El diseño de la interfaz para el software fue pensado para la interacción con el usuario mediante plataforma Web, es por ello que se debe tener acceso a él mediante un principalmente con Mozilla y en algunos casos con *Chrome*, Internet Explorer, etc. Siendo recomendado una resolución 1024x768 pixeles a 32 bit para la optima visualización.

Los colores utilizados para el diseño fueron definidos en base a los colores auténticos o representativos de la universidad, de tal manera obtener una completa armonía. La estructura del Diseño está definida a través de una Pagina Maestra, permitiendo mantener el mismo formato para todas las paginas hijas.

Como se puede apreciar, las pantallas del Sistema, estarán divididas en 4 principales secciones: Cabecera, Menú Vertical, Contenido y Pie de Página.

- **Cabecera:** Esta parte tiene la labor de contener un Banner diseñado para el Sitio, que a su vez tiene el Logo de la Institución. Además la cabecera posee el Menú principal que esta posicionado de forma horizontal.
- **División Menú-Vertical:** Esta posición solo cumple misión de contener a un submenú destinado a cada uno de los módulos del Sistema.
- **División Contenido:** Es en esta parte donde va el denominado “Content Place Holder”, que permite la edición del contenido.
- **Pie de Página:** Sección donde se encontrara información de contacto.

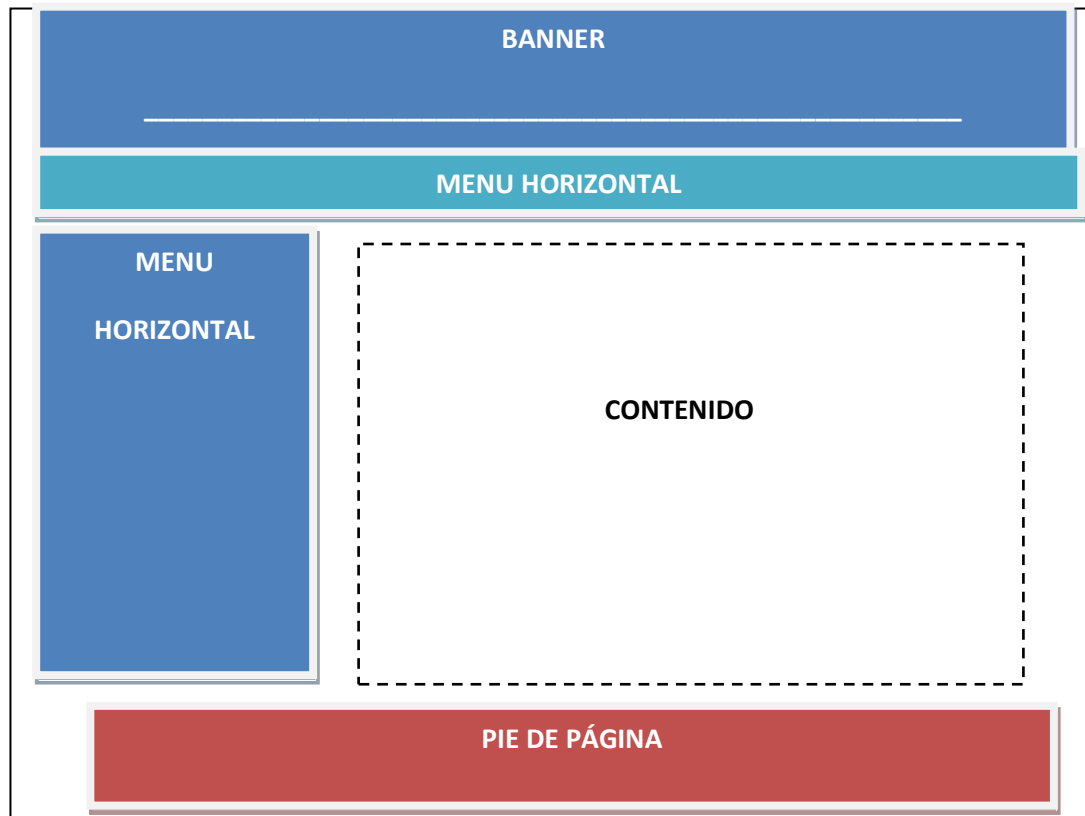


Figura : división de la pantalla para la organización de la información .

9.4.1 Diseño de Pantallas del Sistema

A continuación se presenta algunas pantallas mostrando funcionalidades del sistema (las pantallas están más detalladas en el Anexo: Manual de Usuario).

Solo se mostraran las organización de los módulos o mostrando la navegabilidad del sistema .

Inicio del sistema.

Pantalla de inicio del sistema, se ingresa el nombre de usuario e inmediatamente este devuelve los roles que tiene el usuario en el sistema o bien si no está ingresado.



Plataforma Virtual para
el Control de Proyectos

Bienvenido a

Por favor, introduzca su nombre de usuario y contraseña:

Usuario:

Contraseña:

Rol de Usuario:

Tiempo de carga del servidor: 0.0009 segundos.
Sitio optimizado para resolución de 1280x1024 píxeles. Diseñado para Mozilla Firefox®, Internet Explorer® 8.
© Todos los derechos reservados.



A continuación se muestra los diferentes roles del sistema.

Bienvenido a

Por favor, introduzca su nombre de usuario y contraseña:

Usuario:

Contraseña:

Rol de Usuario:

- Seleccione Dominio
- ADMINISTADOR
- JEFE PROYECTO
- EQUIPO
- PRENSA

Tiempo de carga del servidor: 0.0009 segundos.
Sitio optimizado para resolución de 1280x1024 píxeles. Diseñado para Mozilla Firefox®, Internet Explorer®
© Todos los derechos reservados.



Pantalla de la Bienvenida.

En esta pantalla se muestran las distribuciones de los módulos en el sistema.

The screenshot shows the top navigation bar of the system. On the left is the logo of the Universidad del Bío-Bío. The main title is 'Plataforma Virtual para el Control de Proyectos'. On the right, it says 'Bienvenido, RAMON ANTONIO [Logout] | 28 de Febrero de 2013'. Below the title is a horizontal menu with five items: 'HOME', 'ADMINISTRACIÓN DE PROYECTOS', 'PRENSA', 'ADMINISTRACIÓN PERSONAL', and 'ADMINISTRACIÓN EVENTOS'. The 'ADMINISTRACIÓN EVENTOS' item is highlighted with a blue background.

Tiempo de carga del servidor: 0.0015 segundos.
Sitio optimizado para resolución de 1280x1024 píxeles. Diseñado para Mozilla Firefox®, Internet Explorer® 8.
© Todos los derechos reservados.



Administración de eventos.

En este modulo esta la posibilidad de generar un evento y buscar los diferentes eventos que producen los proyectos, con el fin de no toparse en su organización(en el manual de usuario se hondará mas en ellos).

This screenshot shows the 'Administración de eventos' module page. The top navigation bar is identical to the previous screenshot, but 'ADMINISTRACIÓN EVENTOS' is now the active page. Below the navigation bar, there are two buttons: 'Agregar Nuevo Evento' and 'Consultar Eventos'. The rest of the page content is mostly blank, with a footer area containing server load information and technology logos, similar to the previous screenshot.

Administración de Proyectos.

Este es el modulo que actualmente posee la mayor cantidad de módulos dentro del sistema y su organización es la siguiente.

The screenshot shows the 'Administración de Proyectos' module interface. At the top left is the Universidad del Bío-Bío logo and the title 'Plataforma Virtual para el Control de Proyectos'. On the top right, it says 'Bienvenido, RAMON ANTONIO [Logout] | 28 de Febrero de 2013'. Below the title is a navigation bar with tabs: HOME, ADMINISTRACIÓN DE PROYECTOS (selected), PRENSA, ADMINISTRACIÓN PERSONAL, and ADMINISTRACIÓN EVENTOS. A sidebar on the left contains a list of menu items: Nuevo Integrante de Equipo, Ver Datos del Proyecto, Nueva Tarea, Ver Tareas, Nueva Subtarea, Ver Subtarea, Redictar Comunicado de Prensa, and Nuevo Item Presupuestario. The main content area is empty. At the bottom, there is a footer with server load information: 'Tiempo de carga del servidor: 0.0054 segundos. Sitio optimizado para resolución de 1280x1024 pixeles. Diseñado para Mozilla Firefox®, Internet Explorer® 8. © Todos los derechos reservados.' and browser icons for Firefox, WS, XHTML, CSS, JS, and AJAX.

Administración del Personal.

A continuación se muestra la información , sobre la administración del personal o los diferentes integrantes de los equipos de trabajo.

The screenshot shows the 'Administración del Personal' module interface. It has the same header and navigation bar as the previous screenshot. The sidebar on the left contains a list of menu items: Crear Perfil Acceso and Ver Funcionarios. The main content area is empty. The footer is identical to the previous screenshot, showing server load information and browser icons.

Administración de Prensa.

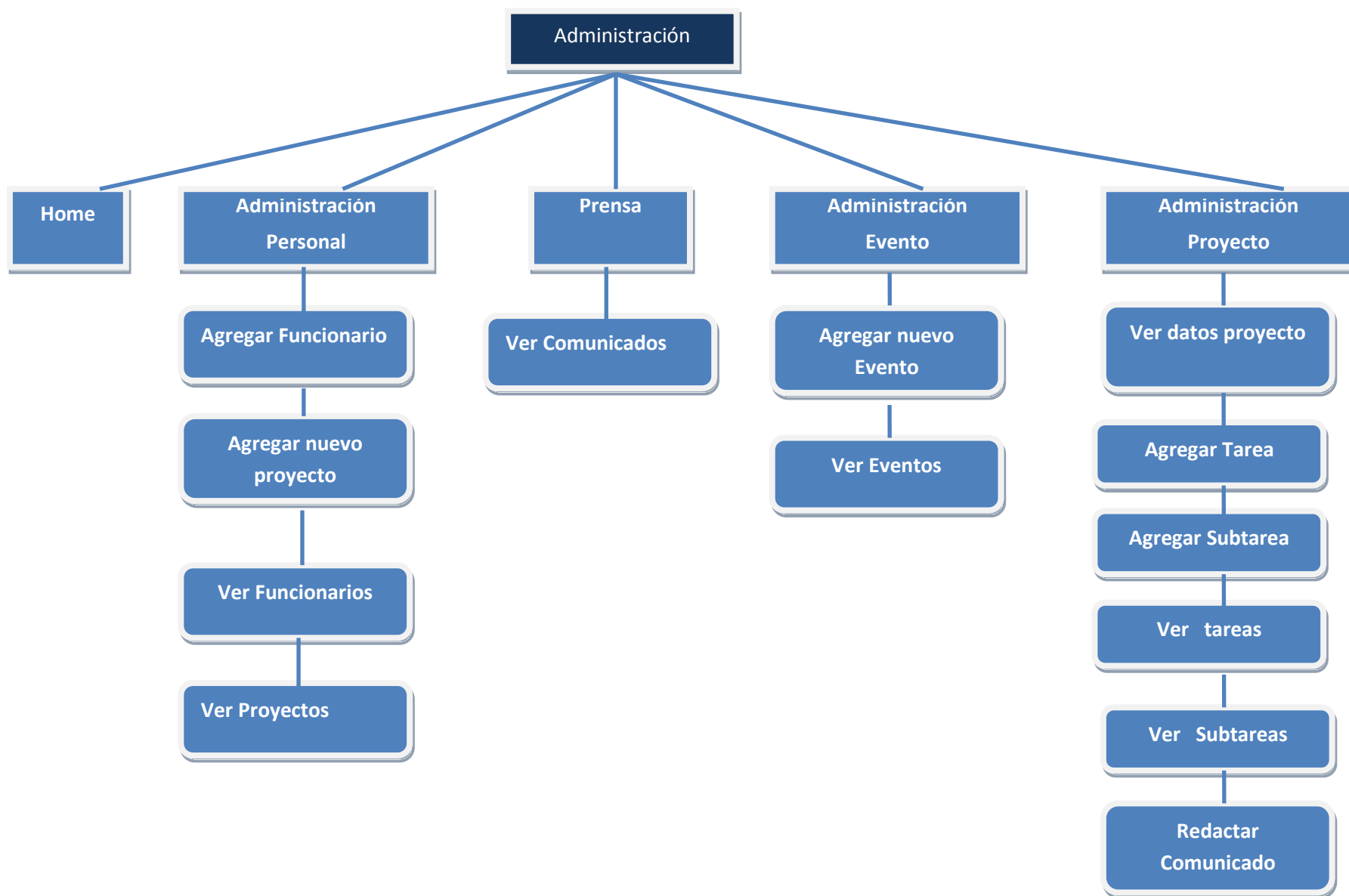
Actualmente solo tiene un modulo , el cual consiste en listar y mostrar el estado de los comunicado de prensa generados por los miembros de los equipos de trabajo.



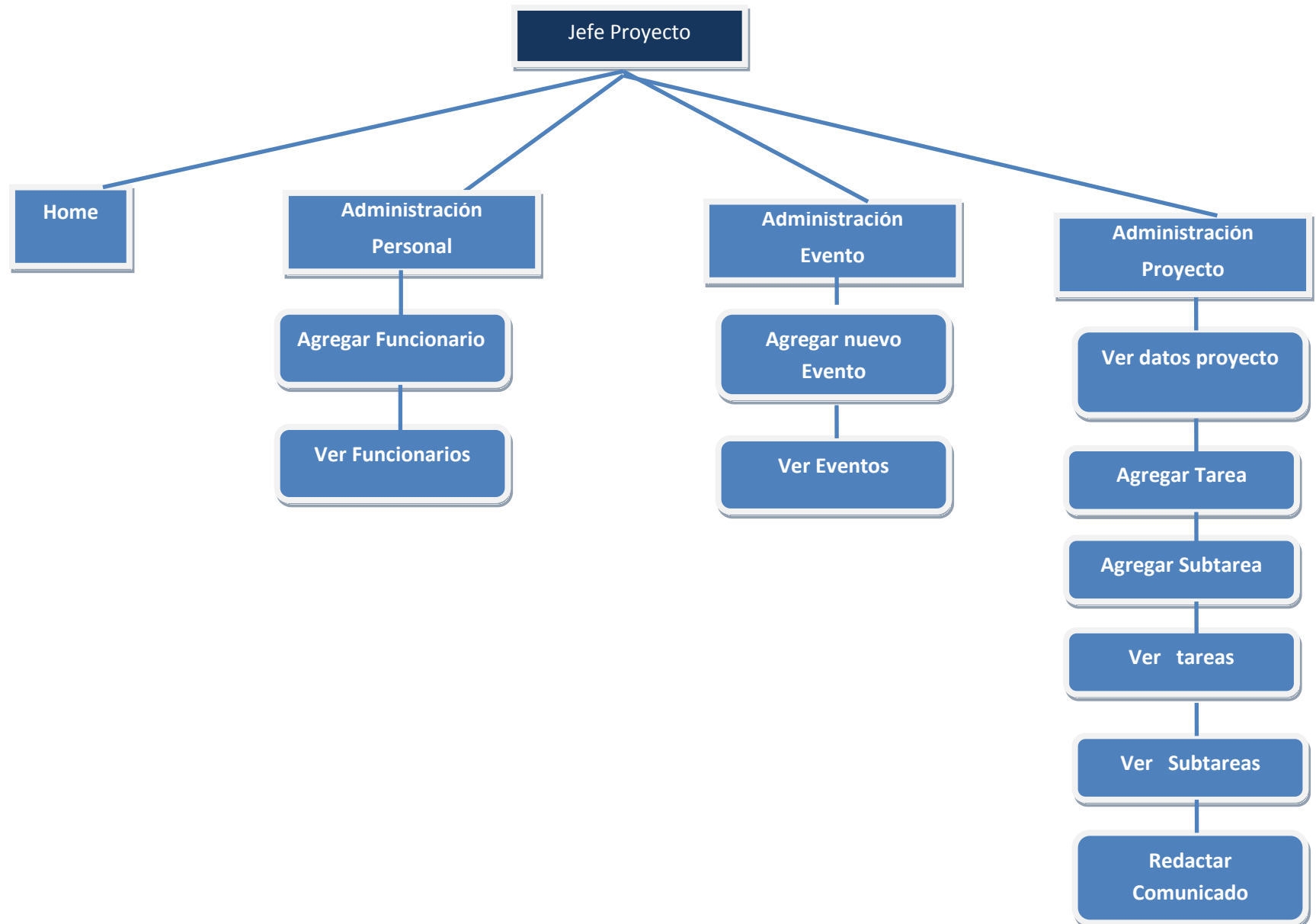
9.4.2 Mapas de Navegación

A continuación los distintos Mapas de Navegación para cada usuario dependiendo de perfil.

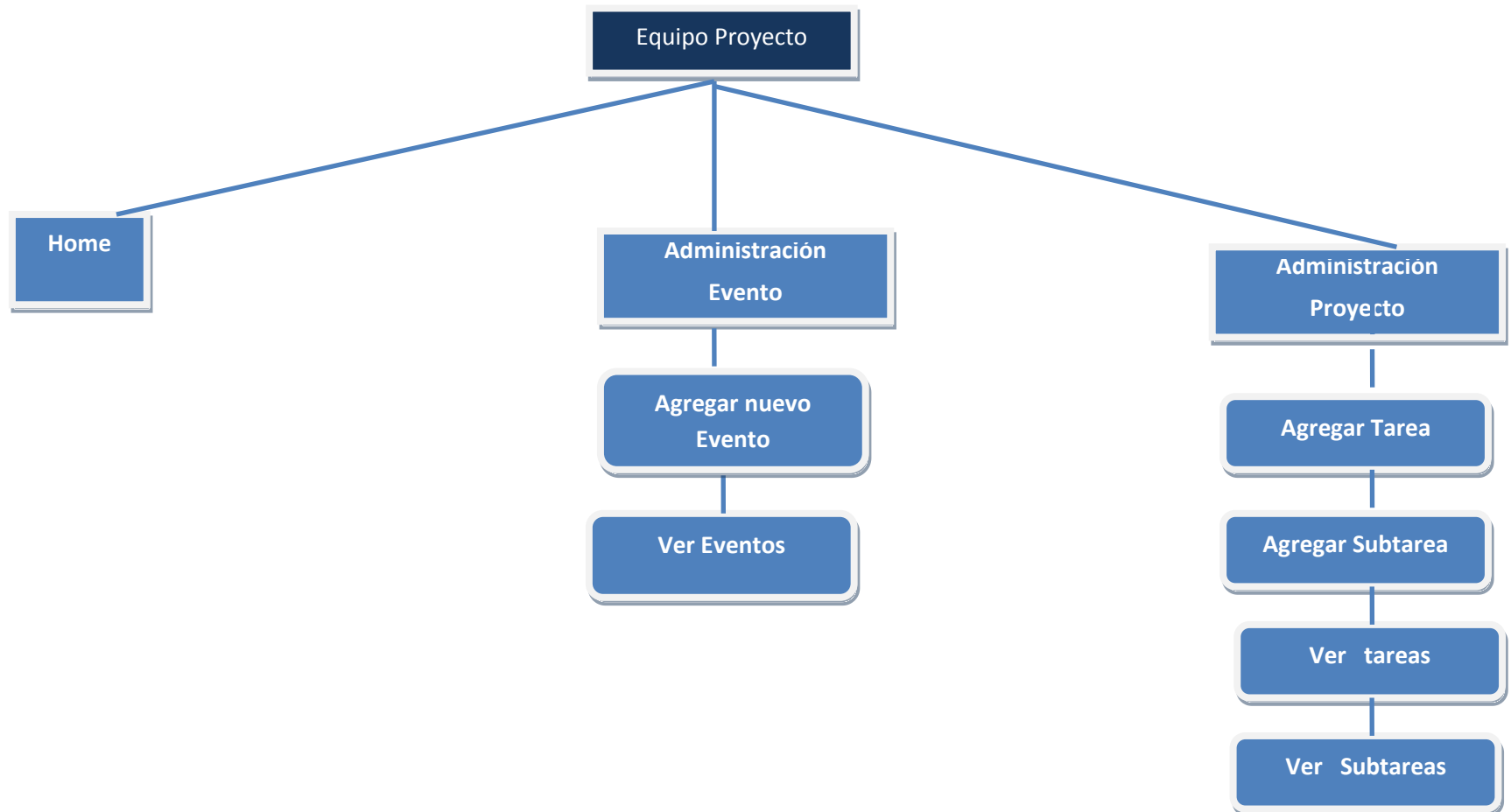
Administrador de personal.



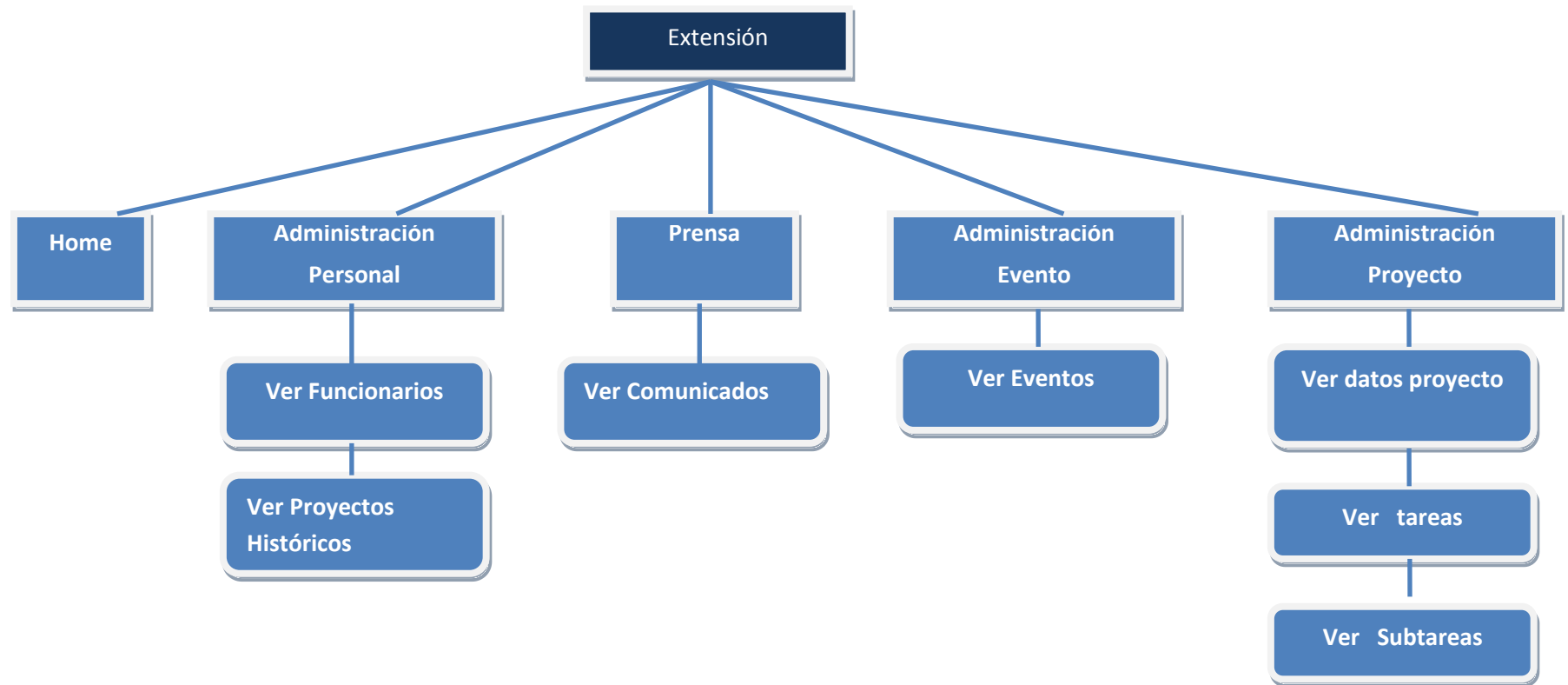
Usuario: Jefe Proyecto.



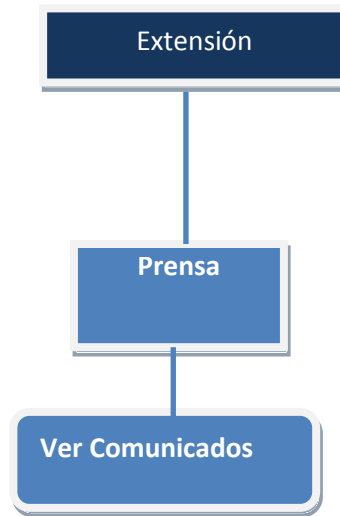
Usuario: Equipo de Proyecto.



Usuario: Personal de extensión.



Usuario : Departamento de comunicación y prensa.



CAPÍTULO X

Pruebas y respaldos.

10.1 Introducción.

El presente capítulo detalla cada uno de los elementos de prueba o pruebas que se realiza antes que el sistema se ponga a marcha, para ello se seguiremos un plan de pruebas, las herramientas que se utilizan y los tipos de prueba que se ejecutaron, también se incluye una breve explicación de cada tipo de prueba.

10.2 Elementos de prueba

Los elementos que se consideran para las pruebas están basados según los requerimientos funcionales del sistema (5.3.1 Requerimientos funcionales del sistema) y el módulo de gestión.

- **Iniciar Sesión:** Permite a los usuarios ingresar al sistema, dando acceso y permisos en función al perfil que posea.
- **Cerrar Sesión:** Correcto funcionamiento del cierre de sesión.
- **Ingresar usuario:** El funcionario debe ingresarse correctamente.
- **Generar los usuarios del sistema:** Correcto visualizar todos los funcionarios que han participado en proyectos.
- **Asignar el proyecto al Jefe Proyecto:** Correcto la asignación de un nuevo proyecto a un funcionario convirtiéndolo en jefe del proyecto ingresado.
- **Generar un nuevo evento:** Correcto los eventos son plenamente ingresados y validados al momento de ser ingresados al sistema.
- **Generar una nueva Tarea:** modulo correcto y validando las tareas con respecto a los presupuestos.
- **Generar nueva Subtarea:** Correcto, este genera la validación para asignar correctamente la Subtarea a la tarea a la que pertenece y el correspondiente ítem presupuestario , entre otras cosas.

- **Ver tareas:** correcto para analizar el estado de las tareas desarrolladas dentro del sistema.
- **Ver datos del proyecto:** correcto es debidamente asignado el funcionario a el o sus proyecto dentro del sistema y ver los demás proyecto que podrían estar dentro del sistema.
- **Ver eventos:** correcto la visualización de los diferentes evento y sus respectivos filtros de búsquedas.

10.3 **Características a probar**

- **Funcionalidad:** Considerando que las pruebas anteriores (en la construcción del sistema) se concentraron en los componentes y sus interacciones, este primer paso de prueba de sistema ignora la estructura del sistema y se enfoca directamente en la funcionalidad.
- **Desempeño:** Cuando se tiene certeza de que el sistema realiza las funciones exigidas por los requerimientos, el esfuerzo de prueba se vuelca a la manera en que se realizan esas funciones. Por lo tanto, la prueba funcional se ocupa de los requerimientos funcionales.
- **Seguridad de acceso:** Con el objeto de proteger la información y datos del producto de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas autorizadas.

10.4 **Objetivo de la Prueba**

El objetivo de la prueba del software es presentar al cliente/usuario un sistema de calidad, que satisfaga los requisitos funcionales y no funcionales, además de comprobar si el usuario está conforme con la navegabilidad y rendimiento.

Para este propósito el proceso se centra en la búsqueda de defectos a través de una prueba individual e integral. Además, de verificar las estructuras internas de datos, la lógica y las condiciones de límites para los datos de entrada y salida.

10.5 Criterios de Cumplimiento

Según las pruebas realizadas el sistema será aprobado, si los errores encontrados son leves, ahora bien si estos son tolerables se considera aceptable ya que se pueden resolver sin mayores dificultades. Pero en el caso que se encuentren errores críticos no se podrá dar por finalizado, puesto que dichos errores pueden poner el riesgo el funcionamiento y por ende registrar información errónea la cual conlleva diversos problemas.

10.6 Respaldo Periódicos

Es recomendable que la base de datos sea respaldada en medios extraíbles o en sistemas tipo backups, por lo menos cada 15 días, esto con el fin de minimizar en lo posible la pérdida de datos en casos de catástrofes, corrupción de archivos o cualquier eventual problema que pudiera surgir con la integridad de la base de datos.

Los puntos anteriores son solo recomendaciones y deben tomarse como formas básicas de minimizar riesgos, puesto que estas decisiones se dejan

Conclusión.

Según las pruebas de navegación ,las pruebas se dan por aprobadas , se encontraron errores leves y fueron corregido en el momento , sin embargo las pruebas finales serán durante la marcha de este , pues es aquí es donde se ve el sistema enfrentado al stres y llevado a su punto de tolerancia máxima.

Conclusiones.

Para finalizar debemos exponer que los objetivos planteados al principio del proyecto, fueron alcanzados en su totalidad, creemos que se logro un resultado coherente con los objetivos generales y específicos definidos en nuestro informe. Sin embargo dada la fecha de entrega, no nos permitió realizar un exhaustivo análisis de los objetivos durante las pruebas en el software , ya que estas pruebas solo se realizaron por el desarrollador sin intervención de otras personas que participen en el proyecto.

Durante el desarrollo de este proyecto hemos dejado en evidencia, la importancia de automatizar procesos dentro de las organizaciones, no importa cuán pequeña o grande sea la institución , un sistema siempre puede alivianar la carga de los funcionarios que participan en ella y de esta forma procurar que los proyectos que se desarrollan en ella lleguen a un buen puerto.

El software logra agilizar los procesos de control de los proyecto en desarrollo dentro de la universidad permitiendo a la persona encargada de llevar este control de una manera más fácil y rápida.

El desarrollo implicó, la planificación y ejecución de varias etapas con sus respectivos plazos y recursos, los cuales se tuvieron que coordinar y distribuir con los diferentes involucrados en el proyecto. Para ello se escogió la metodología de iteración con la cual se logro reducir el riesgo de la insatisfacción del producto, debido a que da la oportunidad de que el usuario pueda ver el proyecto a medida que se van generando nuevos módulos o dicho de otra forma agregando nuevas funcionalidades al sistema, dependiendo de la iteración en la que nos encontremos.

Podemos destacar el uso de herramientas poderosas , en el área de la programación web como son Netbeans,PostreSql, Symfony 1.4, utilizadas en el desarrollo de este sistema, pues nos brindan apoyo en la codificación o estructuración de los sistema, de manera grafica y a través de sus propias ayudas. Lo más rescatable de esto , sin embargo es el hecho que estas herramientas son completamente gratuitas como los lenguajes utilizados para desarrollar el sistema;PHP, Javascript y sus respectivas tecnologías de aplicación como son AJAX, JQUERY.

Como conclusión personal y profesional me gustaría decir; que muchas veces cuando planeamos algo no sale como queremos, aun así debemos enfrentarnos a estos retos, para lograr nuestros objetivos, el hecho de haber construido algo de la nada como este sistema, me ha brindado una mayor confianza como profesional del área informática, asumiendo responsabilidades, trabajar bajo la supervisión de otras personas y en ocasiones discrepando con ellas acerca de sus opiniones, sobre nuestro trabajo.

Bibliografía.

- 1.-Fowler, Martin y Scott Kendall. UML gota a gota. México. Pearson, 1997. 203 p. ISBN: 968-444-364-1.
- 2.-Larman, Craig. UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2a ed. México. Pearson, 2003. 590 p. ISBN: 84-205-3438-2. Título original: Applying UML patterns: and introduction to object-oriented analysis and design and the unified process.
- 3.- Sueldos profesionales de la informática. <<http://www.biobiochile.cl/2012/02/20/sueldos-de-cargos-tecnologicos-en-chile-oscilan-entre-850-mil-y-11-millones-segun-estudio.shtml> > [Consultada 22 febrero 10:30 am].
- 4.-Marcela Varas. Una Experiencia con la Estimación del Tamaño del Software. [en línea] <<http://www.inf.udec.cl/~revista/ediciones/edicion1/mvaras.PDF>> [consulta: 08 junio 2012].
- 5.-Ingeniería de Software :tercera edición.(Roger S.Pressman).
- 6.- IEEE std 830-1998, \IEEE Recommended Practice for Software Requieriemnts Specifications".
- 7.- Documentación Oficial de PHP [En Línea].
- 8.- <http://docs.php.net/manual/es/index.php> [consulta: 10 Enero 2013].
9. -Definición de apache - >Que es Apache? [En Linea].
- 10.- <http://www.alegsa.com.ar/Dic/apache.php> [consulta: 30 enero 2013].
- 11.-<http://www.w3schools.com/js/default.asp> [consulta: 03 febreo 2013].

Anexo diccionario de datos.

Tabla: Personal.

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Fun_rut	Varchar	8	PK	Se le puso código en vez de Rut al funcionario para proteger más la integridad de este.	
Usuario_nombre	Varchar	100		Es el nombre de usuario del funcionario	
Fun_dv	Varchar	1		Es el dígito verificador del funcionario sirve para comprobar que el Rut ingresado es correcto.	
Fun_nombre	Varchar	100		Es el nombre del funcionario.	
Fun_ap	Varchar	100		Es el apellido del funcionario.	
Fun_am	Varchar	100		Es el apellido materno.	
Fun_estado_civil	Varchar	50		Es el estado civil del funcionario.	
Fun_direccion	Varchar	100		Es la dirección del funcionario.	
Fun_estado	Varchar	20		Es el cargo dentro del sistema si es activo o inactivo.	
Cargo_id	int			Es el cargo que tiene en el sistema el funcionario.	
Fun_fecha_nacimiento	Date			Es la fecha de nacimiento del funcionario.	

Comuna_id	int			Es la comuna del funcionario.	
Fun_telefono	Varchar	20		Es el teléfono del funcionario	
Fun_telefono_eme	Varchar	20		Es el teléfono de emergencia que tiene el funcionario.	
Fun_email	Varchar	100		Es el correo electrónico del funcionario.	
Fe_id	int			Es la profesión del funcionario.	
Unidad_id	int			Es la unidad del funcionario.	

Tabla: cargo

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Cargo_id	Int		PK	Es el identificador del cargo.	
Cargo_nombre	Varchar	100		Es el nombre del cargo.	

Tabla: evento

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Evento_id	int		PK	Es el identificador del evento.	
Nombre_evento	Varchar	100		Es el nombre del evento.	
Descripción_evento	Varchar	100		Es la descripción del evento.	
Fecha_evento	Date			Es la fecha de cuando es el evento.	
Hora1	int			Es la hora de inicio del evento.	
Minutos1	int			Son los minutos de inicio del evento.	
Hora2	int			Es la hora de término del evento.	

Minuto2	int			Son los minutos de término del evento.	
Formato1	Varchar	2		Es el horario AM o PM de inicio del evento	
Formato2	Varchar	2		Es el horario AM o PM de término del evento.	

Tabla: proyecto

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Id_proyec	Varchar	5	PK	Es el código del proyecto.	
Nombre_Proyec	Varchar	15		Es el nombre del proyecto.	
Fecha_Inicio_sub	Date			Es la fecha de inicio del proyecto.	
Fecha_Term_Progr	Date			Es la fecha de termino programada.	
Fecha_Term_Real	Date			Es la fecha en que el proyecto realmente termino.	
Presupuesto_Real	int			Es el real presupuesto del proyecto.	
Presupuesto_Final	int			Es el presupuesto final del proyecto.	

Tabla: Usuario

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Usuario_nombre	Varchar	100	PK	Es el nombre de usuario del funcionario.	
Fun_rut	Varchar	8		Es el rut del funcionario.	
Usuario_pass	Varchar	100		Es la contraseña del usuario.	
Usuario_estado	Varchar	12		Es el estado del usuario.	

Tabla: privilegio_usuario

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Dominio_id	Int		PK	Es el identificador de dominios o privilegios.	
Usuario_nombre	Varchar	100	PK	Es el nombre de usuario del funcionario.	

Tabla: Privilegios

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Dominio_id	int		PK	Es el identificador de dominio.	
Dominio_nombre	Varchar	100		Es el nombre del dominio.	

Tabla: Rol_proyecto

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Cod_Cargo	Varchar	15	PK	Es el código del rol.	
Nombre_Cargo	Varchar	15		Es el nombre del rol.	

Tabla: Cargo_personal

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Cod_Cargo_Personal	Varchar	15	PK	Es el código del cargo.	
Descripcion	Varchar	15			

Tabla: comuna

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Comuna_id	int		PK	Es el identificador de la	

				comuna.	
Región_id	int			Es el identificador de región.	
Comuna_nombre	Varchar	100		Es el nombre de la comuna.	

Tabla: región

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Cod_region	Int		PK	Es el código de la region	
Descrip_region	Varchar	15		Es la descripción de la región.	

Tabla: ítem_presupuestario.

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Id_item_pre	Int		PK	Es el identificador ítem presupuesto.	
Descr_item	Varchar	100		Es la descripción del ítem.	

Tabla: itempresupuestario

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Id_item	Int		PK	Es el identificador de ítem.	
Descrip_item	Varchar	100		Es la descripción del ítem	
Gasto_item	int			Es el gasto que se hizo en el ítem.	
documentacion	Varchar	100		Son las facturas del ítem	
Id_subtarea	int			Es el identificador de subtareas	

Tabla: lugar

Atributo	Tipo de Dato	Largo	Llave	Descripción	Tabla Foránea
Lugar_id	Int		PK	Es el identificador de lugar	
Nombre_lugar	Varchar	100		Es el nombre del lugar	
Cod_lugar				Es una abreviación al código del lugar.	

