



INSTITUTO SUPERIOR MINERO METALURGICO
“Dr. Antonio Núñez Jiménez”.
Facultad de Metalurgia - Electromecánica
Moa, Holguín

Trabajo de diploma
para optar por el título de Ingeniero en Informática

Título:

openPlanner: Sistema para la gestión de los Planes de Trabajo en el JSMM.

Autora: Yanet Serrano Camejo

Tutor: Ing. Meyquel García Avilés

Declaración de Autoría

Yo, Yanet Serrano Camejo, declaro que soy la única autora de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de Junio del 2010.

Yanet Serrano Camejo_____

Nombre completo del primer autor

Ing. Meyquel García Avilés_____

Nombre completo del primer tutor

Agradecimientos

Son muchas las personas que hacen posible que este sueño de toda la vida se haga realidad, son importantes los familiares y amigos que ayudan e inspiran a que uno siga adelante creyendo que todo lo que uno pueda hacer según sus fuerzas puede hacerlo, a ellos mis más sinceros agradecimientos.

Agradezco primeramente a Dios mi padre, mi amigo porque me alentó con su palabra, me dio las fuerzas que necesitaba para alcanzar esta meta y me dio un esposo maravilloso para que estuviese conmigo en todo momento.

A mis padres Ulises y Bárbara por aconsejarme y guiarme por este camino desde pequeña.

A mi hermana Yudenia por todo el apoyo que me dio.

A mi familia por su afecto y aliento en todo momento.

A mis amigos que siempre estuvieron ahí para animarme e impulsarme a seguir adelante, en especial a mi amiga Maidelis (May), a mi amiga y hermana Diani, a Katy, a Isabel, a Robert, a Vivi, a Octavio, a Nerius.

A tía Claribel por inspirarme confianza y a mi suegra Elena que me aconsejó y apoyó en todo momento.

A mis cuñados Michel y Mayelín por su ayuda incondicional.

Al claustro de profesores que me acompañaron durante estos cinco años de estudio.

Y muy especialmente a mi tutor y esposo Meyquel que sin él este sueño no hubiese sido posible.

A todos aquellos que aunque no son mencionados pero se sienten aludidos con estas palabras, a esos que su ayuda incondicional ha sido parte de este trabajo.

Gracias.

Dedicatoria

Dedico este trabajo a Dios porque fue quien me dio la sabiduría que necesitaba para hacerlo.

A mis padres por infundir en mí, ese camino que inició con toda la responsabilidad que representa el término de mi carrera.

A mi esposo Meyquel por su comprensión hasta el último momento, por darme tanto apoyo y fuerzas para seguir adelante con la carrera, por haberme dado la dicha más grande: la de ser madre.

A mi pequeña bebé Diana Esther que me ha hecho sentir los sentimientos maternos más bellos del mundo y por estimularme con su tierna sonrisa infantil.

Resumen

En la actualidad con el desarrollo de la informática, las computadoras han dejado de ser simples equipos de cómputo para convertirse en excelentes medios de intercambio de información. Estos medios son la piedra angular de todas las instituciones de la sociedad. El Instituto Superior Minero Metalúrgico de Moa (ISMMM) está encaminando la aplicación de la informatización en gran parte de sus esferas y procesos, logrando así ser más eficiente.

Este trabajo consiste en el desarrollo de una herramienta informática que permita gestionar todo lo referente al proceso de desarrollo de los Planes de Trabajo en el ISMM. Se basa en una aplicación Web, desarrollada sobre la plataforma JEE 5,+ usando herramientas libres.

Abstract

Currently, with the development of information technologies, computers have not longer been simple calculation equipment to be converted in excellent information interchange media. This media is the cornerstone of all society institutions. The Metallurgical and Mining Higher Institute (in Spanish, ISMM) is directing the application of information technology in most of its spheres and processes, being more efficient.

This research work is aimed at developing of a computer science tool that allows managing everything related to the development process at the ISMM. This system will be a business application with web client, implemented over IEE5 platform using free development tools.

Índice

Introducción	11
Capítulo 1 Fundamentación Teórica	15
Introducción.....	15
1.1 Descripción del objeto de estudio	15
1.1.1 ¿Qué es un Plan de Trabajo?.....	15
1.1.2 Los planes de trabajo en el ISMM.....	15
1.1.3 Análisis crítico de la ejecución del proceso	16
1.2 Antecedentes	16
1.2.1 ¿Qué es un Groupware o Software colaborativo?.....	16
1.2.2 Tipos de colaboración	17
1.2.3 Estudio de algunos Groupware libres	18
1.2.4 Razones para no usar los GroupWare antes mencionados.	27
1.3 Metodología	28
1.3.1 Scrum.....	28
1.3.2 XP.....	29
1.3.3 Scrum-XP.....	32
1.4 Herramientas de desarrollo utilizadas	34
1.4.1 Razones para usar Software Libre	34
1.4.2 Java como lenguaje de programación	35
1.4.3 PostgreSQL como gestor de Base de Datos	36
1.4.4 Netbeans IDE 6.5.1 como entorno de desarrollo.	37
1.4.5 JEE 5 como plataforma de desarrollo	40
1.4.6 Glassfish como servidor de aplicaciones	43
Conclusiones.....	43
Capítulo 2 Planeación y Diseño	44
Introducción.....	44
2.1 Planificación – Definición.....	44
2.1.1 Concepción del sistema.	44
2.1.2 Modelo de Historias de Usuario del negocio.....	47
2.1.3 Lista de reserva del producto (LRP).....	48
2.1.4 Historias de usuario (Ver Anexo 1).....	53
2.1.5 Valoración de riesgos	57
2.2 Arquitectura	60
2.2.1 Aplicaciones en capas	60
2.2.2 MVC (Model-View-Controller).....	62
Conclusiones.....	64
Capítulo 3 Desarrollo y Pruebas	65
Introducción.....	65
3.1 Modelo de Datos	65
3.2 Desarrollo	66
3.2.1 Tarjetas CRC	66
3.2.2 Tareas por Historias de Usuario (Ver Anexo 2)	68

3.2.3 Cronograma de Producción	71
3.3 Pruebas	74
3.3.1 Pruebas de aceptación	75
Conclusiones.....	80
Capítulo 4 Estudio de Factibilidad.....	81
Introducción.....	81
4.1 Efectos Económicos.....	81
4.1.1 Efectos directos.....	81
4.1.2 Efectos indirectos.....	82
4.1.3 Efectos externos.....	82
4.1.4 Intangibles	82
4.2 Beneficios y Costos Intangibles en el proyecto	83
4.3 Ficha de costo.....	84
Conclusiones.....	87
Conclusiones Generales.....	88
Recomendaciones.....	89
Bibliografía.....	90
Glosario de Términos.....	92
Anexos	95

Tabla 1 Concepción del Sistema.....	47
Tabla 2 Actores del Sistema.....	48
Tabla 3 Trabajador del sistema.....	48
Tabla 4 Lista de Reserva del Producto (Requisitos Funcionales)	50
Tabla 5 Lista de Reserva del Producto (Requisitos no Funcionales)	52
Tabla 6 Historia de usuario: Administrar usuario	55
Tabla 7 Historia de Usuario: Crear Plan de Trabajo	55
Tabla 8 Historia de Usuario: Administrar grupos.....	56
Tabla 9 Historia de usuario: Buscar actividad	57
Tabla 10 Lista de Riesgos	60
Tabla 11 Tarjeta CRC: Index1	66
Tabla 12 Tarjeta CRC: pagelog.....	67
Tabla 13 Tarjeta CRC: SessionBean1	67
Tabla 14 Tarjeta CRC: actividadesBean	67
Tabla 15 Tarjeta CRC: dbBean	68
Tabla 16 Tarjeta CRC: nivelesBean.....	68
Tabla 17 Tarjeta CRC: PersonaBean	68
Tabla 18 Tarea de ingeniería: Asignar privilegios	69
Tabla 19 Tarea de ingeniería: Crear cuentas de usuario	69
Tabla 20 Tarea de ingeniería: Eliminar cuenta de usuario	70
Tabla 21 Tarea de ingeniería: Crear actividades	71
Tabla 22 Tarea de ingeniería: Modificar actividades	71
Tabla 23 Cronograma de Producción	74
Tabla 24 Caso de prueba: Autenticar usuario	76
Tabla 25 Caso de prueba: Buscar actividad.....	77
Tabla 26 Caso de prueba: Crear Plan de Trabajo	77
Tabla 27 Caso de prueba: Administrar usuario.....	78
Tabla 28 Caso de prueba: Administrar grupo.....	79
Tabla 29 Caso de prueba: Administrar lugares	79
Tabla 30 Caso de prueba: Administrar niveles	80

Índice ilustraciones

Ilustración 1 Arquitectura Web en tres capas	61
Ilustración 2 Modelo de Datos	65
Ilustración 3 Punto de Equilibrio.....	87

Introducción

Las tecnologías de la información son un elemento fundamental para la superación y desarrollo de un país. Ante el nuevo entorno tecnológico mundial, los países emergentes están obligados a preparar profesionales en áreas de la Informática y las Telecomunicaciones, capaces de enfrentar los retos que se tienen hoy en día. Los actuales intentos del país, de prosperar a partir de bases propias de desarrollo, a fin de lograr una pronta recuperación hacen imprescindible que sean adoptadas tecnologías que se encuentran adecuadas a las necesidades de equilibrio con el estado científico técnico del mundo contemporáneo.

Debido al desarrollo alcanzado en la esfera de la informática y como oportunidad de contribuir a alcanzar la excelencia en la gestión e integración del proceso docente de la universidad contemporánea, la Educación Superior comienza a dar los primeros pasos en aras de sumergirse en las corrientes tecnológicas actuales. En ese sentido se trabaja en el desarrollo de sistemas informáticos que contribuyan al perfeccionamiento del procesamiento y gestión de varias de sus esferas.

En el Instituto Superior Minero Metalúrgico, Dr. Antonio Núñez Jiménez, como en la mayoría de las instituciones del país, se desarrollan diferentes actividades (sociales, políticas, culturales, deportivas, docentes, de gestión organizacional, etc.) a través de las cuales se evidencia el correcto funcionamiento de la Institución. Estas actividades se plasman en el Plan de Trabajo del Centro, el cual se publica y constituye guía y referencia para que las instancias inferiores (facultades y departamentos) elaboren su propio Plan de Trabajo atendiendo a su jerarquía, dando prioridad cada Plan de Trabajo al inmediato superior en la estructura organizacional, es decir, cada nivel jerárquico debe respetar los horarios de los planes superiores y buscar espacios vacantes para planificar sus propias actividades.

Como herramienta informática a lo largo de este proceso se usa Microsoft Word en

la mayoría y OpenOffice Writer en algunos casos, de esta manera se obtiene un documento electrónico el cual se distribuye principalmente vía e-mail. Dicho documento tiene una estructura tabular, donde la obtención de la información resulta tediosa ya que las actividades no están ordenadas cronológicamente y no se pueden hacer búsquedas atendiendo a criterios como lugar, participantes, fecha, hora, etc. Además todo el proceso de elaboración resulta trabajoso debido a que cada nivel debe esperar a su inmediato superior para elaborar su Plan, provocando esto la demora en la planificación y entrega de los mismos.

Atendiendo a lo antes expuesto derivamos el siguiente **problema**: La inexistencia de una herramienta informática para la gestión de los Planes de Trabajo en el ISMM. Este problema se enmarca en el **objeto de estudio**: Gestión de Planes de Trabajo; dentro de este objeto se manifiesta como **campo de acción**: Gestión de los Planes de Trabajo en el ISMM. Para dar solución al problema planteado se propone como **Objetivo General**: Diseñar e Implementar una aplicación Web que favorezca y agilice la gestión de los Planes de Trabajo en el ISMM.

Para guiar nuestra investigación se plantea la siguiente **idea a defender**: El desarrollo de una herramienta informática que gestione los Planes de Trabajo debe contribuir a mejorar al proceso de elaboración de los Planes de Trabajo en el ISMM, logrando así que la interacción con éstos sea más eficiente. De acuerdo a esta propuesta se plantean los siguientes **Objetivos Específicos**:

- Estudiar cómo se lleva a cabo el proceso de elaboración de los Planes de Trabajo en el ISMM.
- Realizar un estudio bibliográfico sobre sistemas informáticos vinculados al objeto de estudio.
- Diseñar e implementar el Sistema para la gestión de los Planes de Trabajo.

Para el logro de los objetivos fue necesario plantearse las siguientes **tareas**:

- La entrevista a diferentes usuarios para conocer el proceso de elaboración de los Planes de Trabajo.
- El estudio de herramientas informáticas usadas para la gestión de Planes de Trabajo.
- La realización del análisis y diseño del sistema.
- La implementación del sistema.
- La elaboración del estudio de factibilidad.

Para cumplir estas tareas se han empleado **métodos teóricos** y **empíricos** de la investigación científica.

Los **métodos teóricos** facilitaron la construcción del modelo e hipótesis de investigación y crearon las condiciones para ir mas allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas.

Para su ejecución nos apoyamos en el **análisis y síntesis** como procesos lógicos del pensamiento, que permitieron interpretar y procesar la información obtenida tanto teórica como empírica y para arribar a los criterios expuestos en la investigación.

La **modelación** permitió predecir la respuesta del sistema a variaciones de algunos de sus parámetros sin tener que ejecutar el sistema en la realidad.

Los **métodos empíricos** describieron y explicaron las características fenomenológicas del objeto, representaron el nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

El **análisis de documentos**, para saber cómo funciona actualmente el proceso de

elaboración de los Planes de Trabajo en el ISMM.

Las **entrevistas** para determinar los principales requerimientos del sistema y las nuevas funcionalidades que necesita, éstas están plasmadas en las historias de usuarios.

La **observación**, fue útil para entender el comportamiento del sistema y sus especificaciones.

Capítulo 1 Fundamentación Teórica

Introducción

En este capítulo se presenta una descripción general del contexto donde radica el objeto de estudio así como del proceso a informatizar además de abordar los conceptos principales asociados al dominio del problema y a la solución planteada. También se justifica la metodología usada para el desarrollo de este trabajo, sus principales características, las tecnologías y herramientas usadas.

1.1 Descripción del objeto de estudio

1.1.1 ¿Qué es un Plan de Trabajo?

Un Plan de Trabajo es un documento que recoge un conjunto de actividades planificadas con el fin de cumplir objetivos específicos definidos por una organización. En el contexto donde se desarrolla esta investigación dicho documento tiene la estructura siguiente: emisor del Plan de Trabajo, intervalo de tiempo que abarca, quién lo elabora, quién lo aprueba, lista de prioridades, lista de actividades, cada actividad contiene fecha, hora, lugar, responsable, quién la dirige, quién la ejecuta, quienes participan, entre otras.

1.1.2 Los planes de trabajo en el ISMM

En el Instituto Superior Minero Metalúrgico de Moa, como la mayoría de las instituciones del país se desarrollan diferentes actividades (sociales, políticas, culturales, deportivas, docentes, de gestión organizacional, etc.) a través de las cuales se evidencia el correcto funcionamiento de la Institución. Estas actividades se plasman en el Plan de Trabajo del Centro el cual se publica y constituye guía y referencia para que las instancias inferiores (facultades y departamentos) elaboren su propio Plan de Trabajo.

Actualmente el Ministerio de Educación Superior (MES) emite un Plan de Trabajo rector el cual constituye la guía para todos los Centros de Educación Superior

(CES), a partir de ahí se elaboran los distintos Planes de Trabajo con un efecto en cascada atendiendo a su jerarquía, dando prioridad cada Plan al inmediato superior en la estructura organizacional, es decir, cada nivel jerárquico debe respetar los horarios de los planes superiores y buscar espacios vacantes para planificar sus propias actividades.

1.1.3 Análisis crítico de la ejecución del proceso

En el Instituto se elabora un Plan de Trabajo general en el que deben participar todas las facultades, departamentos y demás trabajadores del Centro que estén vinculados con las actividades planificadas, luego este Plan es enviado a las facultades donde cada facultad debe evaluar los espacios libres para poder confeccionar su propio Plan de Trabajo, de la misma forma ocurre con los departamentos, éstos deben respetar los horarios de los Planes de Trabajo de sus respectivas facultades para poder elaborar el de ellos. Todo este proceso provoca la demora en la planificación y entrega de éstos ya que cada nivel debe esperar a su inmediato superior para elaborar el suyo.

Antes de que el Centro elabore su Plan de Trabajo los departamentos y facultades envían propuestas de actividades para que sean incluidas en el Plan del Centro, si éstas son aprobadas se incluyen. Hay ocasiones en las que el Plan del Centro ya está elaborado y cuando llegan esas actividades no se pueden publicar o si son aprobadas debe ser cambiado el Plan de Trabajo, todo esto se realiza manualmente o utilizando procesadores de textos trayendo consigo que el trabajo se torne tedioso y lento y no se satisfagan completamente sus necesidades.

1.2 Antecedentes

1.2.1 ¿Qué es un Groupware o Software colaborativo?

Se refiere al conjunto de programas informáticos que integran el trabajo en un solo proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo conectadas a través de una red (Internet o intranet).

1.2.2 Tipos de colaboración

El software colaborativo se puede dividir en tres categorías: herramientas de colaboración-comunicación, herramientas de conferencia y herramientas de gestión colaborativa o en grupo.

➤ Herramientas de comunicación electrónica que envían mensajes, archivos, datos o documentos entre personas y facilitan la compartición de información (colaboración asíncrona), como por ejemplo:

- Correo electrónico
- Correo de voz.
- Publicación en web.

➤ Herramientas de conferencia que facilitan la compartición de información, de forma interactiva (colaboración síncrona), como por ejemplo:

- Conferencia de datos - PC en red que comparten un espacio de presentación compartido que cada usuario puede modificar.
- Conferencias de voz - teléfonos que permiten interactuar a los participantes.

➤ Conferencias de video (o audio conferencia) - PC en red que comparten señales de audio o video.

➤ Salas de chat o mensajería instantánea - una plataforma de discusión que facilita el intercambio inmediato de mensajes.

➤ Sistemas para facilitar reuniones - un sistema de conferencias integrado en una sala. Estas salas suelen disponer de un avanzado sistema de sonido y presentación que permite una mejor interacción entre participantes en una misma

sala o entre salas separadas. Ejemplo de ello son los sistemas de soporte a decisiones.

➤ Herramientas de gestión colaborativa que facilitan las actividades del grupo, como por ejemplo:

- Calendarios electrónicos: para acordar fechas de eventos automáticamente y enviar notificaciones y recordatorios a los participantes.

➤ Sistemas de gestión de proyectos - para organizar y hacer seguimiento de las acciones en un proyecto hasta que se finaliza.

➤ Sistemas de control de flujo de actividad - para gestionar tareas y documentos en un proceso organizado de forma estructurada (burocracia).

➤ Sistemas de gestión del conocimiento - para recoger, organizar, gestionar y compartir varios tipos de información.

➤ Sistemas de soporte a redes sociales - para organizar las relaciones de colectivos.

1.2.3 Estudio de algunos Groupware libres

1.2.3.1 eGroupWare

eGroupware es una solución de trabajo en grupo vía web.

Características principales

- Calendario (programación de horario de grupos, de recursos y de contactos).
- Gestor de contactos con base de datos en SQL o LDAP.
- Cliente de correo integrado tipo webmail usa el protocolo IMAP.

- Infolog, una aplicación para tarea y notas.
- Gestor y proyecto integrado las otras aplicaciones.
- Gestor de recursos (inventario) y una herramienta para su reservación integrado en el Calendario del eGroupWare.
- Gestor de archivos.
- Sistema de autor de web con listas de control de acceso.
- Seguimiento de proyecto integrado con el gestor de proyectos.
- Seguimiento de errores.

1.2.3.2 Open GroupWare

Open Groupware es un conjunto de aplicaciones que permite compartir información entre grupos de trabajo de manera distribuida. Mediante Open Groupware es posible centralizar el punto de acceso para toda la información típicamente utilizada en PC's ("Desktops") y que ésta pueda ser accedida así como clasificada en base a los requerimientos de una organización o grupo de trabajo.

Open Groupware contiene los siguientes módulos:

- Administración de contactos: Permite guardar y organizar un sin número de contactos empresariales y personales con teléfono, fax, dirección, correo electrónico, entre otras variantes más, incluyendo capacidad de búsqueda, categorización y desde luego acceso remoto.
- Calendario para grupos: Utilizado para organizar reuniones o eventos dentro de una organización. Es capaz de agregar notas, relacionar reuniones con contactos y proyectos, además de ofrecer detección de conflictos.

- Administrador de recursos: Es capaz de administrar los recursos de una empresa, tales como: salones de conferencia, proyectores o autos. Incluye búsqueda para horarios de disponibilidad sobre recurso.
- Administración de actividades: Permite organizar las actividades de una persona, grupo o proyecto. También permite organizar listas de actividades "por hacer" en base a fecha de entrega, prioridad y otras variantes más.
- Cliente de correo electrónico: Ofrece un ambiente integral (basado en IMAP) para la lectura de correos electrónicos. Ligado íntegramente a la administración de contactos global para evitar contratiempos al buscar correo electrónicos de contacto.
- Proyectos y documentos: Permite compartir documentos y archivos, localmente y remotamente en grupos o de manera privada por proyecto. Incluye relación de documentos hacia contactos de empleados o clientes, así como actividades de proyectos, siendo también capaz de archivar correo electrónico y documentos externos como faxes.
- Noticias: Ofrece la capacidad de publicar información o artículos a nivel de Intranet.
- Preferencias: A través de este módulo puede ser modificado el comportamiento visual y lógico que experimenta cada usuario al acceder Open Groupware.
- Administrador de usuarios: Permite administrar usuarios, grupos y configurar el ambiente global de Open Groupware.

Aunque los mismos creadores de Open Groupware declaran que el software por sí sólo no contiene todas las funcionalidades que sus contrapartes comerciales, éste si incluye las funcionalidades necesarias para cumplir con los requerimientos de

varias organizaciones buscando un software para grupos de trabajo con licenciamiento libre. Inclusive, vale mencionar que los mismos creadores de Open Groupware poseen una versión comercial llamada SKYRiX para organizaciones con requerimientos más avanzados basada en el mismo *Open Groupware*.

Estructura Cliente - Servidor

Open Groupware está basado en una arquitectura cliente-servidor basada en ambientes "web", esto implica que el acceso de información es llevado a cabo directamente de un "Browser", mientras el depósito central reside en un servidor habilitado como un sitio de Internet.

Lo anterior fomenta un ambiente flexible de acceso al no requerirse instalar o actualizar software en los clientes que deseen acceder al depósito informativo. No obstante, Open Groupware ofrece clientes nativos ("Plug-ins") en caso de requerir acceder información de otro software para PC ("Desktop") distinto a un "Browser", como: Microsoft Outlook, Ximian Evolution, Mozilla, WebDAV, Apple iCal.app y KDE KOrganizer. Como se puede notar, las aplicaciones mencionadas anteriormente también hacen uso común de información que puede residir en Open Groupware como teléfonos, correos de contacto y calendarios, razón por la cual existen este tipo de adaptadores.

El proceso de conexión para cualquier cliente de Open Groupware vía "Browser" o nativo depende simplemente de las configuraciones realizadas en la instalación del servidor para Open Groupware, las mismas serán descritas en su totalidad en los siguientes párrafos.

Pasos Previos - Servidor

Antes de iniciar el proceso de instalación en servidor para Open Groupware, es necesario que las siguientes aplicaciones estén instaladas:

Base de Datos

El modelo de datos utilizado por Open Groupware esta diseñado originalmente para operar en la base de datos Postgres. No obstante, es posible (aunque seguramente hay que realizar modificaciones manuales) instalar el modelo de datos en otra base de datos relacional como MySQL, Oracle o SQL-Server.

Apache o Apache 2

Como fue mencionado anteriormente, dada la naturaleza de acceso vía web al depósito informativo de Open Grupoware, el uso de un servidor de páginas como Apache es una opción natural. El mismo deberá estar configurado para operar módulos externos ya que es necesario utilizar un módulo propio de Open Groupware.

Instalación

El proceso para instalar Open Groupware antes de adaptar la Base de Datos y Apache consta de tres pasos:

- Instalación de módulos o librerías dependientes de SOPE.
- Instalación de SOPE.
- Instalación de Open Groupware en sí.

SOPE es una parte central de Open Groupware que contiene la estructura lógica ("Framework") capaz de operar la serie de módulos correspondientes a cualquier instalación.

1.2.3.3 Kolab

Kolab es una suite de herramienta colaborativa basada en Software Libre con capacidades para las funciones de e-mail, calendario, notas y tareas compartidas.

Kolab puede integrarse con componentes del escritorio como Kmail, Korganizer, Kaddressbook, Kpilot, y Mozilla Thunderbird.

Características:

- Ayuda inconsútil completa de los ambientes mezclados de los clientes (Outlook/KDE/Web).
- Interfaz de administración de la red.
- Idiomas soportados: Alemán, Inglés, Francés, Holandés.
- Una agenda compartida a disposición de los usuarios, así como administración de contactos.
- POP3 así como acceso IMAP4 (rev1) al correo.
- Cifrado de email de S/MIME posible (oficialmente Esfinge-ínter operable).

Componentes:

Kolab está formado por varios componentes preexistentes todos ellos con licencia libre entre los que podemos destacar:

- Apache
- ClamAV
- Cyrus-IMAP
- Horde
- SASL

- Spamassasin
- SQLite
- OpenLDAP
- OpenPKG
- OpenSSL
- Postfix

Arquitectura

Kolab se monta sobre un directorio LDAP en donde se almacena la mayor parte de su configuración, cuentas de usuarios. Este directorio LDAP puede ser único o pueden existir múltiples réplicas esclavas para permitir la existencia de servidores secundarios para balanceo de carga y alta disponibilidad.

1.2.3.4 Open-Xchange

Open-Xchange es una aplicación para el trabajo en equipo (del inglés, workgroup, groupware o collaborative software) que proporciona a sus usuarios un avanzado sistema de comunicaciones y funciones para la colaboración e ínter actuación.

Las características de la mensajería incluyen correo electrónico, filtro anti-spam y detector de virus. Los aspectos más básicos de la colaboración incluyen el calendario, los contactos, la gestión de tareas y carpetas privadas, públicas y compartidas. Aspectos más avanzados son la completa integración mediante enlaces y permisos con la compartición de documentos, el seguimiento de proyectos, el repositorio de marcadores o favoritos, el tablón de anuncios, los foros de debate y el archivo de conocimientos. Todo ello a través de una vista integrada a modo de portal o página de inicio.

El servidor puede accederse desde cualquier navegador web moderno y desde varios clientes de escritorio, como KDE Kontact, Mozilla Calendar, Mozilla Sunbird, Palm o Microsoft Outlook y Outlook Express (mediante el Outlook OXtender, un plug-in que permite interactuar con el servidor Open-Xchange del mismo modo que si fuera un servidor Microsoft Exchange). La mayoría de estas funcionalidades avanzadas requieren la compra de una licencia comercial y estos componentes no son de código abierto. Esto permite el uso de carpetas públicas y otras funciones de trabajo en grupo que normalmente sólo están disponibles a usuarios de los productos de Microsoft. Pero hay otros productos de terceras partes que pueden sincronizarse con el servidor usando protocolos estándar, incluyendo la interfaz WebDAV (XML), LDAP, iCal de Apple, HTTP(S), SMTP, IMAP y POP3.

Open-Xchange ha sido creado sobre estándares abiertos e internacionales, lo que significa el uso de APIs estándar, protocolos estándar y, siempre que sea posible, formatos de datos estándar. Algunos de los estándares que Open-Xchange utiliza son POP3, IMAP4, HTML, XML, JavaScript, LDAP y SQL, entre otros. Open-Xchange está basado en Java, pero tiene algunos componentes de seguridad escritos en C y algunos scripts en Perl. El uso de esta diversidad de tecnologías y protocolos hace que la instalación de Open-Xchange no sea una tarea sencilla, pero los pasos que a continuación se describen deberían de facilitar muchísimo dicho proceso.

Requerimientos del sistema

Los requerimientos básicos de Open-Xchange, sin tener en cuenta las librerías de Java necesarias para compilar las fuentes, nos darán una idea de la complejidad del software y podrían resumirse de la siguiente manera:

- Servidor web con motor de Servlets.
- Servidor de correo electrónico con soporte de los protocolos SMTP e IMAP.

- Base de datos con controlador JDBC disponible y que soporte sub-selects, timestamps con zonas horarias y secuenciadores/autonuméricos.
- Servicio de directorio.

1.2.3.5 Hipergate

Hipergate es una suite de aplicaciones de código abierto completamente gratuita basada en un servidor Web. Tiene la capacidad de trabajar sobre múltiples bases de datos y sistemas operativos, dispone de una red de servicios profesionales de soporte y está bien documentado con manuales, referencias API y casos de uso, así como una completa gama de contenido explicativo en la Web. Este programa tiene capacidad multi-entidad, pudiendo ser empleado para dar servicio a una empresa, a un grupo de empresas, o en modalidad ASP para alojar un número ilimitado de entidades usuarios diferentes.

El software está desarrollado en Java y Tomcat, con una escritura 100% en Java Puro, Unix y Win32. El programa puede correr en cualquier versión de la máquina virtual desde la 1.1.8 a la 1.5. La máquina que aloje el servidor Web debe estar configurada en BSD, Solaris, AIX o Windows 2000. Para su elaboración se han empleado herramientas Open Source como Jakarta Bean Scripting Framework 2.3, Jakarta POI 2.5, Jakarta ORO 2.0.8, Xerces2 XML Java Parser 2.6.2, Xalan XSLT Processor 2.6.2, Enterprise DT Ltd Java FTP Library 1.2.2; otros componentes constitutivos han sido empleados mediante una licencia de SUN Microsystems Inc: JavaBeans™ Activation Framework 1.0.2, JavaMail™ 1.3.2 y Java™ Advanced Imaging 1.1.2. Como complemento para desarrollar algunas aplicaciones se tuvo que utilizar también Infomentum AppletFile 3.0, DipuTree Java tree applet 3.0 y Oracle 9i, 10g, Microsoft SQL Server 2000 y PostgreSQL 7.3, 7.4 para el desarrollo de los sistemas gestores de base de datos relacional soportados. Asimismo una parte de la lógica de proceso emplea procedimientos almacenados emplea PL/SQL, PL/pgSQL o Transact-SQL reescritos manualmente para aprovechar al

máximo las opciones más avanzadas de cada SGBDR. Hipergate cubre un amplio rango de utilidades entre las principales y más comunes están la gestión de correo, calendario, foros, lista de empleados; asimismo es un completo CRM con herramientas para el manejo y gestión de llamadas, individuos, oportunidades y listas. Hipergate 3.0 posee además módulos que permite controlar las ventas, gestionar proyectos, controlar incidencias de la empresa, administrar tus contenidos y newsletter y gestionar dominios.

1.2.3.6 OBM GroupWare

Basado en Postfix, Cyrus IMAP, OpenLDAP, Squid y Apache y una mezcla de lenguajes de desarrollo como PHP y Perl para la interfaz de automatización. Todas las funcionalidades de groupware están presentes y también otras, tales como Customer Relationship Management, gestión de proyectos, la mensajería instantánea con soporte para Jabber o la función de controlador de dominio NT con Samba.

Conector para Outlook y Thunderbird están disponibles, así como la funcionalidad de sincronización para PDA. Este proyecto es apoyado por la Sociedad de Software Libre de servicios LINAGORA y hay una web para los profesionales con una completa oferta de servicios.

1.2.4 Razones para no usar los GroupWare antes mencionados.

- Aunque estos groupware poseen requerimientos parecidos a los del ISMM no satisfacen completamente las necesidades de éste.
- Si bien uno de éstos groupware pudiese adaptarse de acuerdo a las necesidades del ISMM e implantarse no contamos con :
 - El tiempo suficiente para estudiar a fondo las funcionalidades de éstos.
 - Los medios necesarios como Internet y computadoras que soporten estos

sistemas, para realizar este estudio.

- El conocimiento pleno de las herramientas de desarrollo como para implantarlo en el Instituto.

1.3 Metodología

1.3.1 Scrum

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. Scrum no está definida como metodología independiente, sino como un complemento a otras metodologías como lo son XP y MSF. Como método, Scrum enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas. En Scrum se fomentan valores como:

- Equipos auto-dirigidos y auto-organizados. No hay Director que decida; la excepción es el Scrum Master que debe ser 50% programador y que resuelve problemas, pero no manda.
- Una vez elegida una tarea, no se agrega trabajo extra. En caso que se agregue algo, se recomienda quitar alguna otra cosa.
- Encuentros diarios con las tres preguntas siguientes:

¿Qué es lo que se hizo el día anterior?

¿Qué es lo que se va a hacer hoy?

¿Qué impedimentos tengo para realizar mi trabajo?

- Iteraciones de treinta días; se admite que sean más frecuentes.
- Demostración a participantes externos al fin de cada iteración.
- Al principio de cada iteración, planeamiento adaptativo guiado por el cliente.

1.3.2 XP

La metodología XP o Extreme Programming es una de las variantes de las metodologías ágiles con más aceptación en la comunidad internacional de desarrollo. Esta metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

1.3.2.1 Prácticas en las que se fundamenta la metodología XP.

1. Planificación incremental

La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de retroalimentación que permitan conocer con precisión dónde estamos. Como es lógico, la planificación es iterativa: un representante del negocio decide al comienzo de cada iteración qué características concretas se van a implementar.

2. Programación en parejas

La metodología XP incluye, como una de sus prácticas estándar, la programación en parejas. Nadie programa en solitario, siempre hay dos personas delante del ordenador. El hecho de que todas las decisiones las tomen al menos dos personas

proporciona un mecanismo de seguridad enormemente valioso. Es muy difícil que dos personas se salten tareas por descuido o negligencia. El código siempre está siendo revisado por otra persona. La revisión de código es el método más eficaz de conseguir código de calidad, algo corroborado por numerosos estudios, muchos de los cuáles son anteriores a la Programación Extrema. En contra de lo que pueda parecer, los dos desarrolladores no hacen lo mismo: mientras el que tiene el teclado adopta un rol más táctico, el otro adopta un rol más estratégico, preguntándose constantemente si lo que se está haciendo tiene sentido desde un punto de vista global.

3. Refactorización

Uno de los objetivos de la XP es mantener la curva de costes tan plana como sea posible, por lo que existen una serie de mecanismos destinados a mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. A este proceso básico para mantener el código en buena forma se le llama refactorización. La refactorización no sólo sirve para mantener el código legible y sencillo: también se utiliza cuando resulta conveniente modificar código existente para hacer más fácil implementar nueva funcionalidad.

4. Diseño simple

Otra práctica fundamental de la Programación Extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". La metodología XP nos pide que no vivamos bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras.

La XP define un "diseño tan simple como sea posible" como aquél que pasa todos los tests, no contiene código duplicado, deja clara la intención de los programadores (enfatisa el qué, no el cómo) en cada línea de código y contiene el

menor número posible de clases y métodos.

5. Testing

La ejecución automatizada de tests es un elemento clave de la metodología XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor cualificada para decidir cuál es la funcionalidad más valiosa. Un efecto lateral importante de los tests es que dan una gran seguridad a los desarrolladores: es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporcionan una red de seguridad. La existencia de tests hace el código muy maleable.

6. Propiedad colectiva del código

La XP aboga por la propiedad colectiva del código. En otras palabras, todo el mundo tiene autoridad para hacer cambios a cualquier código, y es responsable de ellos. Esto permite no tener que estar esperando a otros cuando todo lo que hace falta es algún pequeño cambio.

7. Integración continua

En muchos casos la integración de código produce efectos laterales imprevistos, y en ocasiones la integración puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La Programación Extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata.

8. Cliente en el equipo

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo de desarrollo toma decisiones de negocio críticas. Esto no debería ocurrir, pero a la hora de la verdad con frecuencia no se obtiene retroalimentación del cliente con la fluidez necesaria. La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para resolver dudas y para decidir qué se hace y qué no se hace en cada momento, en función de los intereses del negocio.

9. Entregas pequeñas

Siguiendo la política de la metodología XP de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Éstas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor como para que resulten valiosas para el cliente.

10. Estándares de codificación

Para conseguir que el código se encuentre en buen estado y que cualquier persona del equipo pueda modificar cualquier parte del código es imprescindible que el estilo de codificación sea consistente. Un estándar de codificación es necesario para soportar otras prácticas de la XP.

1.3.3 Scrum-XP

Como se mencionaba anteriormente cuando se habla de la metodología Scrum, se refería a que Scrum no está definida como metodología independiente, sino como un complemento a otras metodologías como lo son XP y MSF. Para el desarrollo de esta investigación se utilizará como metodología la unión de Scrum y XP (SXP).

1.3.3.1 Aplicación de Scrum-XP (SXP) ¿Por qué?

- Realiza entregas cortas en el tiempo y continuas.

- Da la bienvenida a los cambios.
- Promueve entregas periódicas y frecuentes que funcionen.
- Los clientes forman parte del equipo de desarrollo.
- La simplicidad es algo básico.

1.3.3.2 Fases de la metodología Scrum-XP

Esta metodología está dividida en cuatro fases, éstas son:

- **Planificación-Definición:** Es la primera fase que define la metodología XP – Scrum. Se centra sobre el qué, aquí se intenta que información ha de ser procesada, qué función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué duración se necesita para definir un sistema correcto, en fin, han de identificarse los requisitos claves del sistemas y del software. Cuenta con tres tareas principales: ingeniería de sistemas o de información, planificación del proyecto y análisis de los requisitos.

- **Desarrollo:** Se centra en el cómo, se intenta definir cómo han de diseñarse las estructuras de datos, como ha de implementarse la función dentro de una arquitectura de software, como han de implementarse los detalles procedimentales, cómo han de caracterizarse las interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo ha de realizarse la prueba. Cuenta con tres tareas: diseño del software, generación de código y prueba del software.

- **Entrega:** En esta fase se realiza la entrega del software y su documentación, generándose aquellos documentos que son imprescindibles para el entrenamiento y entendimiento del producto.

- **Mantenimiento:** Se centra en el cambio que va asociado a la corrección de

errores, a las adaptaciones requeridas a medida que va evolucionando el entorno del software y a cambios debido a las mejoras producidas por los requisitos cambiantes del cliente. Tipos de cambios: Corrección, Adaptación, Mejora, Prevención.

1.4 Herramientas de desarrollo utilizadas

1.4.1 Razones para usar Software Libre

Para el desarrollo de este proyecto se opta por usar herramientas totalmente libres, varios de los motivos se exponen a continuación:

El Software Libre se usa porque es libre y porque funciona. Los usuarios pueden usar, estudiar, copiar y mejorar los productos en esta modalidad tecnológica, sin mayores restricciones.

El Software Libre no requiere (no permite) el pago por licencias de uso. De hecho, aborrece el concepto por la falta de ética que implica.

El Software Libre es adaptable a las necesidades y a los recursos disponibles en la institución.

La incidencia de los virus computacionales en las plataformas libres es prácticamente nula. Aún si existiera alguna amenaza por esa vía, la comunidad organizada del software libre ha dado muestras de una clara disposición y una sorprendente capacidad para responder a esas amenazas.

El Software Libre estimula la cooperación y la solidaridad. Comunidades de Software Libre destacan por su increíble productividad colectiva. Incluso, hay experiencias de difusión de conocimiento por vías tradicionales que se han visto potenciadas por el cambio tecnológico y la distribución solidaria.

Usar Software Libre está acorde a la estrategia de informatización que desarrolla el

país.

1.4.2 Java como lenguaje de programación

Java es un lenguaje de programación de alto nivel de propósito general, multiplataforma, semicompilado, orientado a objetos y multihilo. Es multiplataforma ya que Java es independiente de la máquina y del Sistema Operativo lo que permite que se pueda ejecutar en un ordenador con Linux o Windows, en un teléfono móvil, entre otros.

Es semicompilado ya que el código fuente es compilado por un código intermedio denominado "bytecode", el cual es interpretado por una máquina virtual de Java. La máquina virtual hace posible que una aplicación que haya sido implementada en Java se ejecute en cualquier sistema operativo con soporte para la máquina virtual. La máquina virtual proporciona un entorno de ejecución que convierte

Java, como lenguaje de programación es multipropósito ya que reúne todas las características de un ambiente orientado a objetos: es sencillo, cuenta con capacidad de generación de aplicaciones distribuidas, robusto, seguro, de arquitectura neutral, portable, dinámico y de alto rendimiento. Pero esto no lo es todo, la API (Application Program Interface, interfaz de programas de aplicación) de Java es muy versátil, ya que está formada por un conjunto de paquetes de clases que le proporcionan una extensa funcionalidad. El núcleo de la API viene con cada una de las implementaciones de la máquina virtual: tipos de datos, clases y objetos, manejo de red, seguridad, componentes, etc. Estos componentes son llamados Java Beans, los cuales son código reusable que se pueden desarrollar fácilmente para crear aplicaciones sofisticadas. Se puede decir que con Java, Sun Microsystems introdujo en el mercado la primer plataforma de software universal diseñada desde y para el crecimiento de Internet y de las intranets corporativas.

Con Java se pueden programar páginas web dinámicas, con accesos a

bases de datos, utilizando XML, con cualquier tipo de conexión de red entre cualquier sistema. En general, cualquier aplicación que se desee hacer con acceso a través de la web se puede hacer utilizando Java.

1.4.3 PostgreSQL como gestor de Base de Datos

Es un servidor de base de datos relacional orientada a objetos de software libre, está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle, la siguiente es una breve lista de algunas de esas características:

- **DBMS Objeto-Relacional:** PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arrays.
- **Altamente Extensible:** PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.
- **Soporte SQL Comprensivo:** PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.
- **Integridad Referencial:** PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.
- **Lenguajes Procedurales:** PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Esto le permite al desarrollador crear funciones o bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones

básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional. Otros lenguajes soportados son: C, C++, Gambas, Java PL/Java Web, PL/Perl, pIPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme y Lenguaje para aplicaciones estadísticas R through PL/R.

- PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta. Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido.

1.4.4 Netbeans IDE 6.5.1 como entorno de desarrollo.

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 2000 y continúa siendo el patrocinador principal de los proyectos.

La Plataforma NetBeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones. 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.

El IDE NetBeans 6.5.1 es una herramienta para programadores, pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, JavaEE, web, EJB y aplicaciones móviles). Entre sus características se encuentran un sistema de proyectos basado en Ant, control de versiones y refactoring. El IDE NetBeans IDE 6.5.1 corre en diferentes sistemas

operativos entre los que podemos mencionar Linux, Mac OS X, Solaris y Windows. Como se puede observar se ha lanzado con soporte para la mayoría de las plataformas. Como requerimiento se necesita tener previamente instalado el JDK 5.0 o 6.0.

1.4.4.1 Frameworks utilizados

ICEFaces

Es un framework de código abierto para construir aplicaciones web con AJAX (JavaScript Asíncrono y XML) tipo RIA (Rich Internet Application).

ICEFaces aísla completamente al desarrollador de AJAX. No hacen falta etiquetas especiales: se ponen los controles en la pantalla e ICEFaces se encarga de enviar sólo la información necesaria entre cliente y servidor.

Es decir, ya no se envían los formularios a la antigua usanza, en un POST de HTTP, sino que sólo se envían los cambios que ha hecho el usuario del cliente al servidor, y los cambios en la pantalla del servidor al cliente. Además, con la inclusión de la librería Scriptaculous en ICEFaces, se dispone de arrastrar+soltar y de efectos (fundidos, parpadeos, apariciones,...) para los controles.

Las aplicaciones desarrolladas en ICEfaces no necesitan plugins de navegador o applets para ser vistas.

Estas aplicaciones están basadas en JavaServer Faces (JSF), así que permite el desarrollo de aplicaciones Java EE con la posibilidad de utilizar de forma fácil desarrollos basados en JavaScript. Experiencia de usuario enriquecedora: crea una experiencia de usuario superior además de utilizar las ventajas de aplicaciones Java EE. Esto se consigue gracias a los componentes que vienen incluidos dentro de la distribución de ICEfaces.

Las ventajas del uso de ICEfaces para enriquecer las aplicaciones web de

escritorio son numerosas. Debido a la cantidad de frameworks existentes, se procede a mostrar una serie de características diferenciadoras respecto a otros frameworks:

- Está basado en código abierto: ICEfaces es un framework basado en Ajax bajo licencia de código abierto.
- Basado en estándares: ICEfaces es una solución basada en Java, así que los desarrolladores pueden continuar trabajando de la misma forma que lo hacen. Hay multitud de plugins desarrollados para que ICEfaces sea integrado con multitud de IDEs Java.
- El Ajax es transparente: ICEfaces aporta a los programadores un desarrollo con mínimo esfuerzo en la sección JSF.
- Compatibilidad: ICEfaces soporta todos los servidores de aplicaciones, aporta plugins para los distintos IDEs y efectos javascript de librerías de cualquier empresa que haya desarrollado Ajax del mercado.
- Seguridad: ICEfaces es una de las soluciones Ajax más seguras del mercado. Es compatible con SSL, previene los scripts de cross-site, inyección de código malicioso. Es una solución Ajax basada en servidor, la cual no utiliza datos de usuarios, además es especialmente efectivo en la prevención de fallos en los submits de los formularios y el ataque SQL por inyección.
- Escalabilidad y clustering: El servidor asíncrono HTTP (AHS) aporta una alta escalabilidad para aplicaciones ICEfaces y pueden ser utilizadas por un gran número de usuarios concurrentes, además aporta despliegue en clúster (un requisito crítico que algunas soluciones no aportan).
- Carga de páginas incremental con edición de secciones y sin recargas de página completas.

- Se preserva el contexto del usuario durante la actualización de la página, incluyendo posición del foco y scroll.
- En aplicaciones de tiempo real, las recargas de páginas son asíncronas.

JSF (Java Server Faces)

Es un framework para aplicaciones basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Page (JSP) como la tecnología que permite hacer el despliegue de las páginas. JSF incluye:

Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.

Un conjunto por defecto de componentes para la interfaz de usuario.

Dos librerías de etiquetas personalizadas para JavaServer Pages que permiten expresar una interfaz JavaServer Faces dentro de una página JSP.

Un modelo de eventos en el lado del servidor.

Administración de estados.

Beans administrados.

1.4.5 JEE 5 como plataforma de desarrollo

Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JavaEE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc. y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para JavaEE para componentes. Éstas incluyen Enterprise JavaBeans, servlets, portlets, Java Server Pages y varias tecnologías de servicios web. Esto permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes que en las tareas de mantenimiento de bajo nivel.

Uno de los beneficios de Java EE como plataforma es que es posible empezar con poco o ningún coste. La implementación Java EE de Sun Microsystems puede ser descargada gratuitamente, y hay muchas herramientas de códigos abiertos disponibles para extender la plataforma o para simplificar el desarrollo.

1.4.5.1 EJB

Los Enterprise JavaBeans (también conocidos por sus siglas EJB) son una de las API (Application Program Interfaces) que forman parte del estándar de construcción de aplicaciones empresariales JEE 5.0 de Sun Microsystems.

La especificación de Enterprise JavaBean define los papeles jugados por el contenedor de EJB y los EJBs, además de disponer los EJBs en un contenedor.

La API Enterprise JavaBeans define un conjunto de APIs que un contenedor de objetos distribuidos soportará para suministrar persistencia, RPCs (usando RMI o RMI-IIOP), control de concurrencia, transacciones y control de acceso para objetos distribuidos.

Los EJBs proporcionan un modelo de componentes distribuido estándar del lado

del servidor.

El objetivo de los EJBs es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

No hay que confundir los Enterprise JavaBeans con los JavaBeans. Los JavaBeans también son un modelo de componentes creado por Sun Microsystems para la construcción de aplicaciones, pero no pueden utilizarse en entornos de objetos distribuidos al no soportar nativamente la invocación remota (RMI).

EJBs de Sesión (Session EJBs): gestionan el flujo de la información en el servidor. Generalmente sirven a los clientes como una fachada de los servicios proporcionados por otros componentes disponibles en el servidor. Puede haber dos tipos:

Con estado (stateful). Los beans de sesión con estado son objetos distribuidos que poseen un estado. El estado no es persistente, pero el acceso al bean se limita a un solo cliente.

Sin estado (stateless). Los beans de sesión sin estado son objetos distribuidos que carecen de estado asociado permitiendo por tanto que se los acceda concurrentemente. No se garantiza que los contenidos de las variables de instancia se conserven entre llamadas al método.

1.4.5.2 JPA

Java Persistence API, más conocida por su sigla JPA, es la API de persistencia desarrollada para la plataforma JEE e incluida en el estándar EJB3. Esta API busca justificar la manera en que funcionen las utilidades que proveen un mapeo objeto-

relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de orientación a objetos al interactuar con una base de datos.

1.4.5.3 JDBC

Es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

1.4.6 Glassfish como servidor de aplicaciones

GlassFish es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE.

Conclusiones

Al concluir este capítulo se ha podido apreciar todo lo referente a la base teórica que fundamenta la presente investigación. Dando a conocer al lector el flujo actual de los procesos, el análisis crítico de la ejecución de los procesos, los antecedentes, así como la metodología de desarrollo que se usará.

Capítulo 2 Planeación y Diseño

Introducción

Este capítulo expone la fase de planeación y diseño, donde se conocerá la concepción inicial del sistema, se especificarán las Historias de Usuario (user histories o HU), los requisitos funcionales y no funcionales del sistema, así como la lista de riesgos y modelo del diseño.

2.1 Planificación – Definición

La fase de Planificación - Definición, es la primera que define la metodología XP – Scrum. En esta fase se generan todos los documentos que se encuentran relacionados con la concepción inicial del sistema, así como la definición del mismo. También se incluyen algunos que están vinculados a la primera parte de los procesos de Ingeniería de Software, tales como los relacionados con el negocio, los requisitos y el diseño.

2.1.1 Concepción del sistema.

La plantilla de Concepción del sistema, es el primer documento generado en la fase de Planificación-Definición, este se elaboró luego de realizarse la actividad Entrevista con el cliente, momento en el que se puntualizó la concepción inicial del sistema. Este documento refleja la visión general del producto a implementar, también se recogieron los diferentes roles que intervendrán en el desarrollo del software, así como las responsabilidades que tendrán en dicho proceso. Se documenta el tipo de proyecto al que pertenece así como su clasificación. Se plantea además cuáles herramientas serán utilizadas para el desarrollo de la aplicación, el alcance que va a tener, una descripción de los involucrados en el negocio, cuáles son los motivos de la necesidad del desarrollo del software y la propuesta de solución.

Esta plantilla posee gran importancia dentro de la documentación, pues es la base para los demás documentos que se generan durante el ciclo de desarrollo de software. Proporciona ventajas tales como:

- Permite conocer el producto a implementar, pues recoge la concepción inicial del sistema.
- Facilita el proceso de negocio, pues define aspectos relacionados con el mismo.

Para comprender cuáles son los datos que se recogerán en esta plantilla, se muestra la misma gráficamente.

Clasificación del proyecto. Aplicación Web
Tipo de proyecto. Nacional
Resumen: El presente documento tratará el tema relacionado con la elaboración de los Planes de Trabajo en el ISMM, las razones por las cuales surge este proyecto y qué se pretende solucionar con el mismo, abordando además los roles que intervendrán en el desarrollo del trabajo.
Surgimiento. Con el desarrollo de la informática y las comunicaciones los procesos que se manejan en las instituciones se han ido automatizando para lograr un mejor funcionamiento. En aras de seguir el camino de la automatización, se ha estudiado el proceso de elaboración de los Planes de Trabajo en el ISMM, dicha elaboración se realiza de forma manual trayendo consigo que en ocasiones se demore la entrega de los mismos, por lo que surgió la idea de que era necesario la existencia de una herramienta informática que ayudara

en el proceso de gestión de los planes de trabajo.

¿Qué es?

openPlanner (como se llamará el sistema) no es más que una solución de trabajo en grupo vía web, basado en un modelo de software libre. Permite compartir información entre grupos de trabajo de manera distribuida. Mediante openPlanner es posible centralizar el punto de acceso para la información referente a los planes de trabajo y que ésta pueda ser accedida así como clasificada en base a los requerimientos de nuestra universidad.

Este sistema está basado en una arquitectura cliente-servidor, esto implica que el acceso de información es llevado a cabo directamente de un "Browser", mientras el depósito central reside en un servidor de aplicaciones que soporta la tecnología JEE.

Metodología a utilizar.

Scrum-XP (SXP)

Roles.

<i>Rol</i>	<i>Nombre</i>
Gerente	Meyquel García Avilés
Analista	Yanet Serrano Camejo
Diseñador	Yanet Serrano Camejo
Desarrollador	Meyquel García Avilés Yanet Serrano Camejo
Cliente	Universidad

Misión.

La aplicación Web que se desea desarrollar tiene como misión constituir la herramienta básica y fundamental que permita gestionar los planes de trabajo en el ISMM.

Visión.

Se espera contar con una aplicación web robusta e interactiva, donde el cliente pueda personalizar la información que le es permitida, habiéndose

autenticado anteriormente. Se aspira que cuente con un diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema.

Herramientas utilizadas.

- PostgreSQL
- Netbeans IDE 6.5.
- ICEFaces
- JSF(Java Server Faces)
- JEE 5
- EJB
- JDBC
- Metodología Scrum-XP

Tabla 1 Concepción del Sistema

2.1.2 Modelo de Historias de Usuario del negocio.

El Modelo Historias de Usuario del negocio genera una plantilla en la que se precisaron las características específicas del negocio, así como la forma en que interactúan los clientes con el sistema y viceversa. El Modelo de negocio cuando se trabaja con metodologías ágiles, es diferente al ya conocido en el proceso unificado, ya que en este caso se trabaja con historias de usuarios, en vez de con casos de uso. Pero independientemente de los cambios técnicos que puedan existir, el negocio se modela igual en cualquier metodología.

Con la elaboración del Modelo del negocio, se obtuvieron ventajas, tales como:

- ✓ Se concibió mejor la interacción del sistema con los usuarios.
- ✓ Se facilitó la captura de requisitos, pues ya se ha definido el negocio y se

tiene una noción de lo que se va a realizar.

Para comprender cuáles son los datos que se recogerán en esta plantilla, a continuación se muestra la misma.

1. Actores del negocio

<i>Actor</i>	<i>Descripción</i>
Usuario	Es el que elabora el plan de trabajo que le corresponde, puede ejercer rol de invitado.
Invitado	Ve la información que el administrador le permita

Tabla 2 Actores del Sistema

2. Trabajadores del negocio

<i>Trabajador</i>	<i>Descripción</i>
Administrador	Administrador del sistema, configura y actualiza el sistema, puede ejercer rol de usuario.

Tabla 3 Trabajador del sistema

2.1.3 Lista de reserva del producto (LRP).

La plantilla de Lista de reserva del producto, es el primer artefacto generado en la etapa de Captura de Requisitos, está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración.

Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos

acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Esta lista puede estar conformada por requerimientos técnicos y del negocio, funciones, errores a reparar, defectos, mejoras y actualizaciones tecnológicas requeridas. Si se realiza el llenado de esta plantilla con calidad, se garantizarán las ventajas siguientes:

- La organización de los requisitos, tanto funcionales como no funcionales, en dependencia de la prioridad que tengan para el desarrollo del sistema.
- Se facilitará el trabajo al confeccionar las historias de usuarios.

Para comprender cuales son los datos que se recogerán en esta plantilla, se muestra la misma gráficamente.

Asignado a	Item	Descripción	Estimación	Estimado por
Prioridad		Muy Alta		
Yanet Serrano Camejo	1	Administrar usuario	2 semanas	A
Meyquel García Avilés	2	Crear Plan de Trabajo	5 semanas	A
Yanet Serrano Camejo	3	Autenticar usuario	1 semana	A
Prioridad		Alta		
Meyquel García Avilés	4	Administrar grupos	1 semana	A
Meyquel García Avilés	5	Administrar lugares	1 semana	A

Yanet Serrano Camejo	6	Administrar niveles	1 semana	A
Prioridad		Media		
Meyquel García Avilés	7	Buscar Plan de Trabajo	1 semana	A
Yanet Serrano Camejo	8	Buscar actividad	1 semana	A
Meyquel García Avilés	9	Organizar salidas de las actividades	1 semana	A
Yanet Serrano Camejo	10	Mostrar actividades diarias	3 semanas	A
Prioridad		Baja		
Yanet Serrano Camejo	11	Enviar aviso de actividades por correo electrónico	3 semanas	A
Yanet Serrano Camejo	12	Enviar informe de cumplimiento por correo electrónico.	3 semanas	A

Tabla 4 Lista de Reserva del Producto (Requisitos Funcionales)

Requisitos no Funcionales	
Apariencia o Interfaz Externa	Diseño sencillo, con pocas entradas, permitiendo que no sea necesario mucho entrenamiento para utilizar el sistema. Su interfaz agradable logra captar la confianza de los usuarios así como su identificación con los colores y formatos en que realizan

	sus acciones.
Usabilidad	El sistema en general se desarrolla con el objetivo de facilitar el trabajo manual realizado hasta el momento por lo que se espera que cuente con un alto nivel de usabilidad por los usuarios debido a que constituye un forma más flexible de desarrollar los planes de trabajo. Para tener el acceso a cierta información se requiere ser usuario del sistema, pero para acceder a otro tipo de información el usuario no necesita autenticarse. El sistema garantizará un acceso fácil y rápido a los usuarios.
Rendimiento	Como la aplicación está concebida para un ambiente cliente/servidor los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información. El tiempo de respuesta tiene que ser corto pues se deben generar pantallas dinámicas, implicando esto que el acceso a la base de datos tendrá la rapidez suficiente. También el sistema debe identificar con seguridad a los diferentes usuarios que interactúan con él. Estará implementado sobre una tecnología Web, facilitando su uso a través de la red.

Soporte	La base de datos que utiliza el sistema como medio de almacenamiento de la información está soportado sobre un gestor de bases de datos PostgreSQL, su plataforma es JEE y están montados sobre un servidor de aplicaciones Glassfish; logrando así una solidez de los datos a informatizar, realizando mantenimientos a la base de datos orientados a la actualización y corrección de la información.
Portabilidad	Multiplataforma
Seguridad	Garantizar que la información solo pueda ser vista por los usuarios con el nivel de acceso adecuado. Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo. Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.
Hardware	Para el servidor: <ul style="list-style-type: none"> • Pentium(R) 4 con 512 MB de RAM y un microprocesador a 300 MHz, 40 GB de disco duro. Para el cliente (mínimo): <ul style="list-style-type: none"> • Pentium(R) 4 con 128 MB de RAM y un microprocesador a 300 MHz.

Tabla 5 Lista de Reserva del Producto (Requisitos no Funcionales)

2.1.4 Historias de usuario (Ver Anexo 1).

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Son escritas en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto.

Las historias de usuario guían la construcción de las pruebas de aceptación, elemento clave en XP (deben generarse una o más pruebas para verificar que la historia ha sido correctamente implementada) y son utilizadas para estimar tiempos de desarrollo. En este sentido, sólo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas. En el momento de implementar una historia de usuario, se debe detallar a través de la comunicación con el cliente, éstas son la base para las pruebas funcionales. En esta plantilla se describe la historia de usuario, se asigna la iteración y el tiempo en que se desarrollará la misma, así como, su prioridad en el negocio.

Las historias de usuario proporcionan ventajas, tales como:

- Se logró escribirlas en lenguaje del cliente, por lo que es muy fácil su comprensión.
- Se especifican cada uno de los requisitos del sistema, sin necesidad de documentaciones extensas.
- Se reflejan todas las características del sistema.
- Al definir las correctamente, guían el proceso de implementación.

Para comprender cuales son los datos que se recogerán en esta plantilla, se muestra la misma gráficamente.

Historia de Usuario			
Número: 01	Nombre Historia de Usuario: Administrar usuario		
Modificación de Historia de Usuario Número: ninguna			
Usuario: Yanet Serrano Camejo	Iteración Asignada: 2		
Prioridad en Negocio: Muy Alta	Puntos Estimados: 2		
Riesgo en Desarrollo: Bajo	Puntos Reales: 1		
Descripción: Esta HU define los usuarios que tendrán derecho a manipular cierta información, es decir para que una persona pueda trabajar con los Planes de Trabajo directamente debe haberse autenticado anteriormente.			
Observaciones: Administrar usuario incluye: Asignar privilegios, Eliminar cuentas de usuario, Cambiar de usuario y contraseña.			
Prototipo de interfase:			
<div style="border: 1px solid #ccc; padding: 10px;"> <div style="text-align: center; background-color: #e6f2ff; padding: 5px;"> open planner <small>Sistema para la gestión de planes de trabajo en el ISMM</small> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 25%; border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p>Administrar Usuarios</p> <p>Administrar Lugares</p> </div> <div style="width: 50%; border: 1px solid #ccc; padding: 5px;"> <p>Nombre Completo <input type="text"/></p> <p>Email <input type="text"/></p> <p>Nombre de Usuario <input type="text"/></p> <p>Nivel (área) al que Pertenece <input type="text" value="Centro ISMM"/> ▼</p> <p>Contraseña <input type="text"/></p> <p>Repita la Contraseña <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Adicionar"/></p> <p style="font-size: small; text-align: center;">Para modificar o eliminar debe seleccionar un elemento de la tabla...</p> <p style="text-align: center;"> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/> </p> </div> <div style="width: 20%; border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p style="text-align: center; font-weight: bold;">Lista de Usuarios del Sistema</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>Meyquel García Avilés</td></tr> <tr><td>Administrador del Sistema</td></tr> </table> </div> </div> </div>		Meyquel García Avilés	Administrador del Sistema
Meyquel García Avilés			
Administrador del Sistema			

Tabla 6 Historia de usuario: Administrar usuario

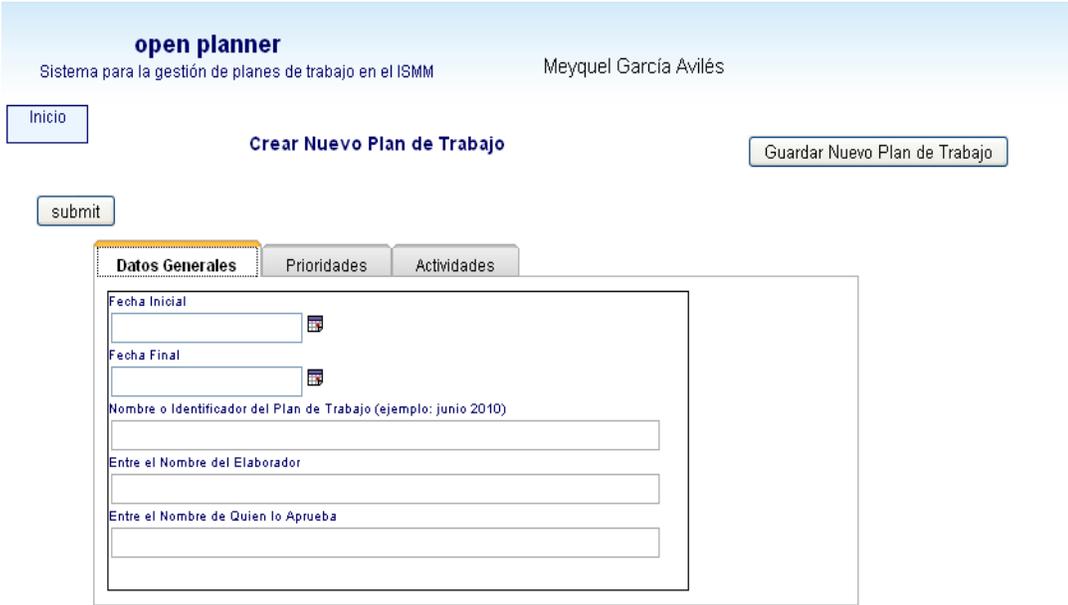
Historia de Usuario	
Número: 02	Nombre Historia de Usuario: Crear Plan de Trabajo
Modificación de Historia de Usuario Número: ninguna	
Usuario: Meyquel García Avilés	Iteración Asignada: 3
Prioridad en Negocio: Muy Alta	Puntos Estimados: 5
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
<p>Descripción: Esta HU no es más que el conjunto de actividades planificadas que tendrán lugar en el mes.</p> <p>Observaciones: Para el desarrollo del Plan de Trabajo hay que tener en cuenta las prioridades de dicho plan.</p> <p>Prototipo de interfase:</p> 	

Tabla 7 Historia de Usuario: Crear Plan de Trabajo

Historia de Usuario	
Número: 04	Nombre Historia de Usuario: Administrar grupos
Modificación de Historia de Usuario Número: ninguna	
Usuario: Meyquel García Avilés	Iteración Asignada: 2
Prioridad en Negocio: Muy Alta (Muy Alta /Alta/ Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.5
Descripción: Cuando se elabora un Plan de Trabajo existe un grupo de personas que participan. Administrar grupos se encarga de agrupar a estas personas según el tipo de grupo al que pertenezcan.	
Observaciones:	
Prototipo de interfase: [Imagen de cada una de las interfaces relacionadas con la HU.]	

Tabla 8 Historia de Usuario: Administrar grupos

Historia de Usuario	
Número: 08	Nombre Historia de Usuario: Buscar actividad
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yanet Serrano Camejo	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.5
Descripción: Esta HU le da la posibilidad al usuario de conocer los detalles de la o las actividades buscadas, que pueden ser: lugar fecha, hora, etc.	
Observaciones:	

Prototipo de interfase:

The screenshot shows a web interface titled "open planner" with the subtitle "Sistema para la gestión de planes de trabajo en el ISMM". On the left, there is a "Buscar actividades" (Search activities) panel containing several search criteria: "Fecha Inicial" (Initial Date) and "Fecha Final" (Final Date) as text input fields; "Selección Nivel" (Select Level) as a dropdown menu with "Centro ISMM" selected; "Selección el Plan de Trabajo" (Select the Work Plan) as a dropdown menu with "junio2010" selected; and "Selección Lugar" (Select Location) as a dropdown menu with "<-Cualquier Lugar->" selected. A "Buscar" (Search) button is at the bottom of this panel. The main content area has two tabs: "Lista de Actividades" (List of Activities) and "Otros datos del Plan de Trabajo" (Other data of the Work Plan). The "Lista de Actividades" tab is active and contains the text "Haga clic sobre una actividad para ver detalles..." (Click on an activity to see details...). Below this text is a table with two columns: "Actividades del día en el Centro" (Activities of the day in the Center) and "Hora" (Hour). To the right of the table is a details panel with labels: "Actividad: Fecha y Hora: Lugar:" (Activity: Date and Time: Location:), "Dirige: Ejecuta: Tipo de Actividad:" (Directs: Executes: Activity Type:), and "Actividad:" (Activity:).

Tabla 9 Historia de usuario: Buscar actividad

2.1.5 Valoración de riesgos

La plantilla Lista de riesgos, es el documento que se genera de la actividad Valoración de riesgos. En ella se definen los posibles riesgos que actuarán sobre el proceso de desarrollo del sistema, así como la estrategia trazada para mitigarlos, además de un plan de contingencia que describe que curso seguirán las acciones si el riesgo se materializa.

Esta plantilla se le atribuye gran importancia, pues a pesar que es imposible definir desde un inicio todos los riesgos que pueda atravesar el proyecto, si se tendrán algunos en cuenta. Propicia algunas ventajas, tales como:

- Se logró definir los posibles riesgos, así como la forma de mitigarlos, lo que disminuye el efecto de los mismos, si ocurrieran.
- Se llevó un control de todos los problemas que han azotado al proyecto, así como la manera que fueron enfrentados y el impacto que tuvieron en el

proceso de desarrollo.

Riesgo	Tipo de riesgo	Impacto	Descripción	Probabilidad	Efectos
1. Precario dominio de la tecnología a utilizar.	Tecnológico	1. Puede atrasar el plazo de entrega del software. 2. Falta de calidad del producto.	Estudiando las tecnologías existentes, llegamos a la conclusión que usar como lenguaje de programación Java y como framework Icefaces era lo más factible por sus características para el desarrollo del software, aunque dominamos el lenguaje usado y conocemos otros frameworks, Icefaces vino a ser nuevo para	Baja	Serios

			nosotros, por lo que consideramos el estudio de este marco de trabajo un riesgo para nuestra investigación.		
2. Falta del fluido eléctrico.	Tecnológico	1. Pérdida de información del software. 2. Atraso en la entrega del producto.	Debido a que el desarrollo del software se realiza utilizando medios informáticos, (computadoras) que al faltar el fluido eléctrico no se puede trabajar.	Alta	Serios
3. Bajo nivel de comunicación con los clientes.	Organización	1. Pobre captura de requisitos	Debido a que los clientes son personas con muchas responsabilidades, tienen poco tiempo para realizar reuniones en pos de discutir los requisitos del software.	Media	Serios

Plan de Contingencia
<p>Plan para el riesgo 1</p> <p>En el caso de que no se logre mitigar este riesgo, como una solución alternativa se acude a otra tecnología que se domine.</p>
<p>Plan para el riesgo 2</p> <p>Hacer salvallas constantes cuando se esté trabajando en el software. Trabajar en otros puntos que no necesiten necesariamente una computadora.</p>
<p>Plan para el riesgo 3</p> <p>Estudiar el proceso del negocio por otras vías.</p>

Tabla 10 Lista de Riesgos

2.2 Arquitectura

2.2.1 Aplicaciones en capas

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad e integración.

Si se establece una separación entre la capa de interfaz gráfica (cliente), replicada en cada uno de los entornos de usuario, y la capa modelo, que quedaría centralizada en un servidor de aplicaciones, se obtiene una potente arquitectura que otorga algunas ventajas:

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.

- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, reduciendo los costes de mantenimiento.
- Mayor sencillez de los clientes.

Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.

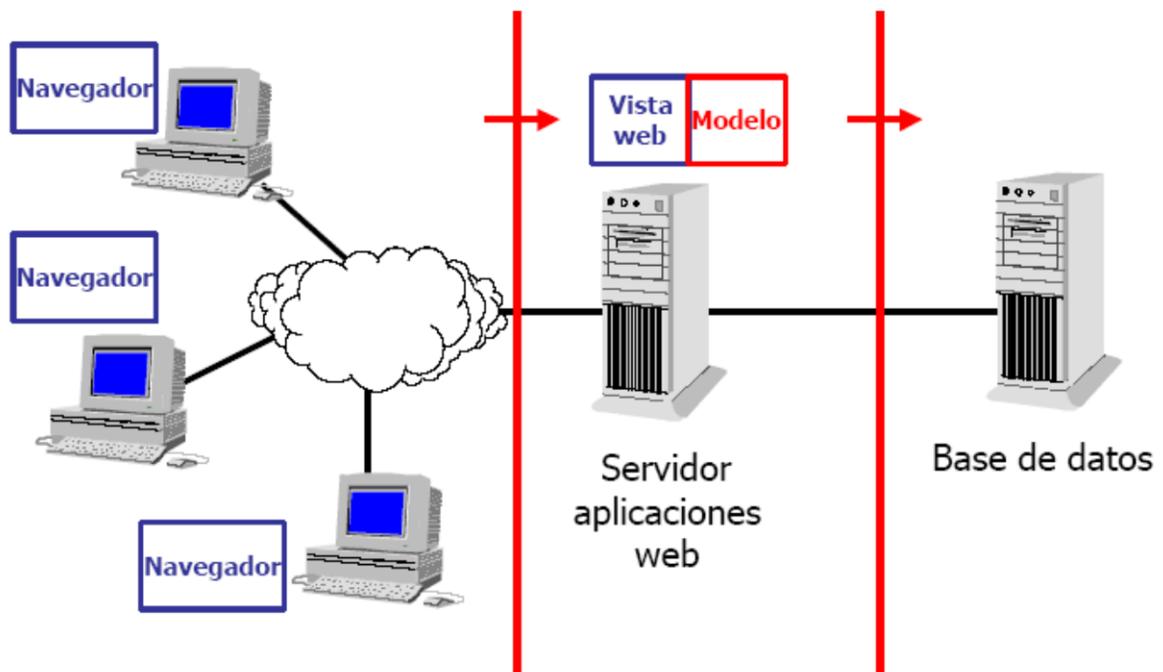


Ilustración 1 Arquitectura Web en tres capas

En este proyecto se mantiene el número de capas en 3, integrando interfaz Web y modelo en un mismo servidor aunque conservando su independencia funcional. Ésta es la distribución en capas más común en las aplicaciones Web.

2.2.2 MVC (Model-View-Controller)

La arquitectura Model-View-Controller surgió como patrón arquitectónico para el desarrollo de interfaces gráficos de usuario en entornos Smalltalk. Su concepto se basaba en separar el modelo de datos de la aplicación de su representación de cara al usuario y de la interacción de éste con la aplicación, mediante la división de la aplicación en tres partes fundamentales:

- El modelo, que contiene la lógica de negocio de la aplicación.
- La vista, que muestra al usuario la información que éste necesita.
- El controlador, que recibe e interpreta la interacción del usuario, actuando sobre modelo y vista de manera adecuada para provocar cambios de estado en la representación interna de los datos, así como en su visualización.

Esta arquitectura ha demostrado ser muy apropiada para las aplicaciones web y especialmente adaptarse bien a las tecnologías proporcionadas por la plataforma JEE, de manera que:

- El modelo, conteniendo lógica de negocio, sería modelado por un conjunto de clases Java, existiendo dos claras alternativas de implementación, utilizando objetos java tradicionales llamados POJOs (*Plain Old Java Objects*) o bien utilizando EJB (*Enterprise JavaBeans*) en sistemas con unas mayores necesidades de concurrencia o distribución. En este trabajo se decidió usar EJB.
- La vista proporcionará una serie de páginas web dinámicamente al cliente, siendo para él simples páginas HTML. Existen múltiples frameworks que generan estas páginas web a partir del formato de páginas JSP (*JavaServer Pages*) que proporciona una gran funcionalidad y versatilidad.

- El controlador en la plataforma JEE hace de intermediarios entre la vista y el modelo, es más versátiles que los JSP para esta función al estar escritos como clases Java normales, evitando mezclar código visual (HTML, XML...) con código Java. Para facilitar la implementación también existe una serie de frameworks que proporcionan soporte a los desarrolladores, entre los que cabe destacar JSF.

Con todo lo anterior, el funcionamiento de una aplicación web JEE que utilice el patrón arquitectónico MVC se puede descomponer en una serie de pasos:

1. El usuario realiza una acción en su navegador, que llega al servidor mediante una petición HTTP y es recibida por un bean (controlador). Esa petición es interpretada y se transforma en la ejecución de código java que delegará al modelo la ejecución de una acción de éste.
2. El modelo recibe las peticiones del controlador, a través de un interfaz o fachada que encapsulará y ocultará la complejidad del modelo al controlador. El resultado de esa petición será devuelto al controlador.
3. El controlador recibe del modelo el resultado, y en función de éste, selecciona la vista que será mostrada al usuario, y le proporcionará los datos recibidos del modelo y otros datos necesarios para su transformación a HTML. Una vez hecho esto el control pasa a la vista para la realización de esa transformación.
4. En la vista se realiza la transformación tras recibir los datos del controlador, elaborando la respuesta HTML adecuada para que el usuario la visualice.

Esta arquitectura de aplicaciones otorga varias ventajas clave al desarrollo de aplicaciones web, destacando:

- Al separar de manera clara la lógica de negocio (modelo) de la vista permite

la reusabilidad del modelo, de modo que la misma implementación de la lógica de negocio que maneja una aplicación pueda ser usado en otras aplicaciones, sean éstas web o no.

- Permite una sencilla división de roles, dejando que sean diseñadores gráficos sin conocimientos de programación o desarrollo de aplicaciones los que se encarguen de la realización de la capa vista, sin necesidad de mezclar código Java entre el código visual que desarrollen.

Conclusiones

Con el desarrollo de este capítulo se comprendió mejor la interacción del sistema con los usuarios. Se facilitó la captura de requisitos pues al definir el negocio ya se tiene una noción de lo que se va a realizar. Se llevó un control de todos los problemas que han azotado al proyecto, así como la manera que fueron enfrentados y el impacto que tuvieron en el proceso de desarrollo. También se puntualizó dentro del diseño la arquitectura utilizada.

Capítulo 3 Desarrollo y Pruebas

Introducción

En este capítulo se presentan las fases desarrollo y pruebas de la metodología XP. Se presentan las tarjetas CRC, los resultados de las iteraciones realizadas así como la descripción de las distintas pruebas. Por otra parte se describen cada una de las tareas confeccionadas para llevar cumplir con el desarrollo de cada una de las historias de usuario detectadas.

3.1 Modelo de Datos

El siguiente diagrama muestra la interacción entre las entidades y constituye infraestructura de datos persistentes del sistema.

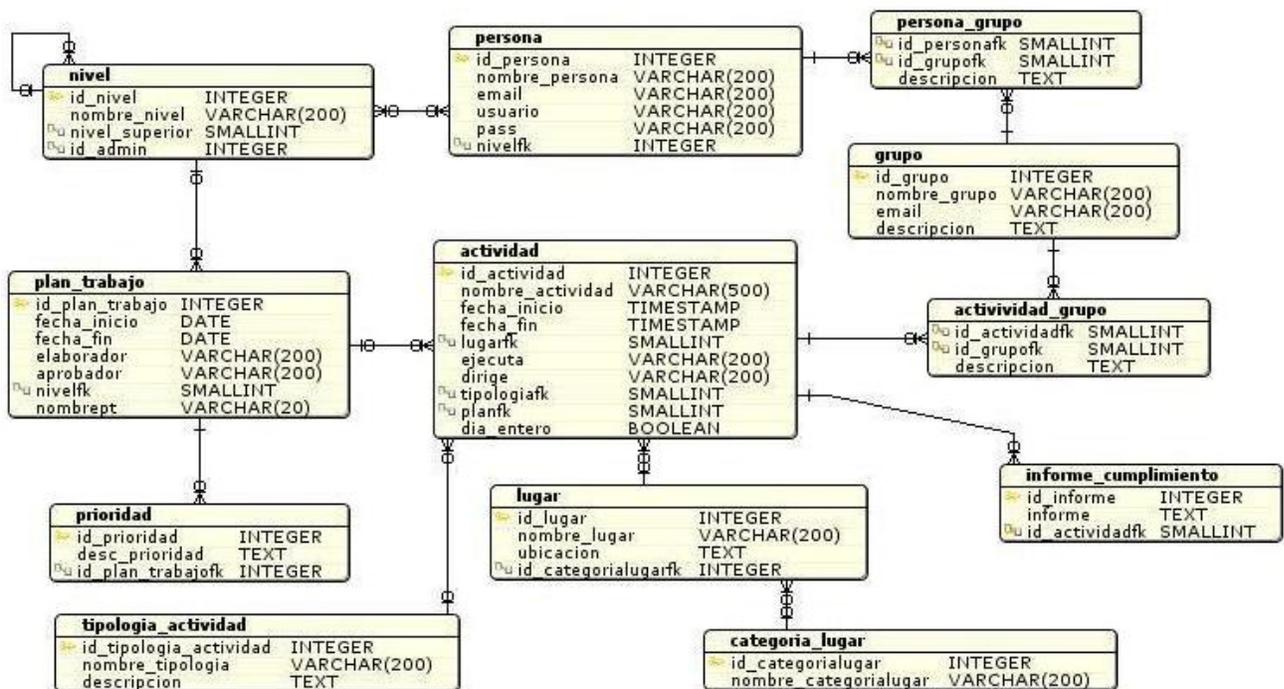


Ilustración 2 Modelo de Datos

3.2 Desarrollo

La fase de Desarrollo, es la segunda que define la metodología XP - Scrum. En esta fase se recogen las principales definiciones que se manejan en la metodología y otros términos de difícil entendimiento para los clientes, así como de las tareas a realizar durante la implementación. Además se genera el código fuente en la etapa de implementación, los documentos relacionados con las pruebas, como última parte de esta etapa.

3.2.1 Tarjetas CRC

Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Una tarjeta CRC representa un objeto. El nombre de la clase se coloca a modo de título en la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda, y las clases que están implicadas en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente. Para una mejor comprensión de las mismas se decide agruparlas por historias de usuarios.

Nombre de la clase: Index1	
Tipo de la clase: Controladora	
Responsabilidades	Colaboradores
Mostrar actividades del día.	
Permitir sistema de búsqueda personalizada por niveles, lugar, fecha y plan de trabajo.	
Permitir autenticación de usuario	

Tabla 11 Tarjeta CRC: Index1

Nombre de la clase: pagelog
Tipo de la clase: Controladora

Responsabilidades	Colaboradores
Mostrar responsabilidades de persona autenticada.	Index1
Permite realizar búsquedas de actividades y planes de trabajo.	

Tabla 12 Tarjeta CRC: pagelog

Nombre de la clase: SessionBean1	
Tipo de la clase: Controladora	
Responsabilidades	Colaboradores
Conecta la interfaz de usuario con la lógica del negocio.	ActividadesBean
	dbBean
	nivelesBean
	personaBean
	lugaresBean

Tabla 13 Tarjeta CRC: SessionBean1

Nombre de la clase: actividadesBean	
Tipo de la clase: Lógica del negocio	
Responsabilidades	Colaboradores
Gestionar actividad (Insertar, Modificar, Eliminar)	dbBean
Buscar actividad	

Tabla 14 Tarjeta CRC: actividadesBean

Nombre de la clase: dbBean	
Tipo de la clase: Lógica del negocio	
Responsabilidades	Colaboradores
Acceso a los datos	

Tabla 15 Tarjeta CRC: dbBean

Nombre de la clase: nivelesBean	
Tipo de la clase: Lógica del negocio	
Responsabilidades	Colaboradores
Gestionar niveles (Insertar, Modificar, Eliminar actividad)	dbBean

Tabla 16 Tarjeta CRC: nivelesBean

Nombre de la clase: PersonaBean	
Tipo de la clase: Lógica del negocio	
Responsabilidades	Colaboradores
Autenticar usuario	dbBean

Tabla 17 Tarjeta CRC: PersonaBean

3.2.2 Tareas por Historias de Usuario (Ver Anexo 2)

La plantilla de Tarea de ingeniería, posee una gran importancia, pues permite definir cada una de las actividades que estarán asociadas a las historias de usuario y que permitirán su implementación. También posibilita conocer qué programador está asignado a cada tarea, así como el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo que se llevará cada historia de usuario en implementarse, de acuerdo a su complejidad.

Esta plantilla proporciona ventajas tales como:

- Permite organizar el proceso de implementación, pues las tareas se van implementando de acuerdo a su prioridad.
- Posibilita conocer el grado de complejidad de cada historia de usuario, teniendo en cuenta la cantidad de tareas asociadas.

Tarea de ingeniería	
Número Tarea: 001	Número Historia de Usuario: 01
Nombre Tarea: Asignar privilegios	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Esta tarea consiste en configurar el sistema de modo que el usuario solo pueda acceder a la información que le corresponde. Solo asigna privilegios el administrador	

Tabla 18 Tarea de ingeniería: Asignar privilegios

Tarea de ingeniería	
Número Tarea: 002	Número Historia de Usuario: 01
Nombre Tarea: Crear cuentas de usuario	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Se insertan los datos de los usuarios que pertenecerán al sistema.	

Tabla 19 Tarea de ingeniería: Crear cuentas de usuario

Tarea de ingeniería	
Número Tarea: 003	Número Historia de Usuario: 01
Nombre Tarea: Eliminar cuenta de usuario.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Si un usuario deja de pertenecer al grupo de usuarios del sistema, el administrador elimina esta cuenta de usuario.	

Tabla 20 Tarea de ingeniería: Eliminar cuenta de usuario

Tarea de ingeniería	
Número Tarea: 005	Número Historia de Usuario: 02
Nombre Tarea: Crear actividades	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	

Descripción: Diseñar una interfaz gráfica que le permita al cliente crear actividades teniendo en cuenta el nivel al que pertenece y así gestionar los datos de los Planes de Trabajo.

Tabla 21 Tarea de ingeniería: Crear actividades

Tarea de ingeniería	
Número Tarea: 006	Número Historia de Usuario: 02
Nombre Tarea: Modificar actividades	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: En el caso de que el usuario necesite modificar una actividad por una razón determinada, esta tarea responderá a su petición, pero solo modificará las actividades que le están permitidas.	

Tabla 22 Tarea de ingeniería: Modificar actividades

3.2.3 Cronograma de Producción

La plantilla de Cronograma de producción, es un documento de suma importancia, pues fue el que guió y controló las actividades que se realizaron en cada iteración. En éste, no sólo se recogen las actividades planificadas, sino el tiempo que durarán y el rol responsable de desarrollarlas.

Con la existencia del Cronograma de producción en la documentación, se

garantizan ventajas, tales como:

- Control y organización de cada una de las actividades a desarrollar por el equipo de trabajo.
- Planificación eficaz de las iteraciones, garantizando una buena producción.

Para comprender los datos que recoge esta plantilla se muestra la misma a continuación:

No	Hito	Descripción	Inicio	Fin	%ejec.	Ejecutor
1	Entrevista con Manolito (Secretario General del ISMM)	Se desarrolló esta entrevista con el objetivo de conocer la estructura de los Planes de Trabajo y cómo funcionan en el Instituto.	27/01/10		100%	Analista
2	Entrevista con Juan Manuel (Decano Facultad Humanidades)	Esta actividad se desarrolló con el objetivo de conocer de qué forma la facultad se introduce al plan de trabajo del centro.	5/02/10		100%	Analista
3	Entrevista con Juan Manuel (Decano Facultad Humanidades)	En esta entrevista se plasmaron algunas de las funcionalidades que se deseaba que incluyera el sistema.	15/02/10		100%	Analista

4	Entrevista con Yoliuba (Jefa del departamento de Socioculturales.)	Esta actividad se desarrolló con el objetivo de conocer de qué forma el departamento se vincula al plan de trabajo de la facultad y por ende del centro.	6/03/10		100%	Analista
5	Entrevista con Ramón (Jefe de departamento de Contabilidad)	Esta entrevista se desarrolló con el propósito de conocer otras funcionalidades del software.	15/03/10		100%	Analista
6	Entrevista con Michel García (Vice-Rector Docente).	Esta entrevista se desarrolló con el propósito de conocer la estructura organizativa del ISMM, además de obtener otras funcionalidades para el desarrollo de la aplicación.	16/04/10		100%	Analista
7	Diseño de la Base de Datos.	Se diseñaron las tablas que se necesitarían para el manejo de la información	10/02/10	25/04/10	100%	Diseñador

8	Diseño del sistema	Se diseñó la interfaz de usuario.			100%	Diseñador
9	Implementación	Se implementó la lógica del negocio.			95%	Desarrolladores
10	Pruebas	Se realizaron pruebas a las diferentes historias de usuario.	5/06/10		100%	Desarrolladores
11	Factibilidad	Se realizó un estudio de factibilidad.	20/05/10		100%	Analista

Tabla 23 Cronograma de Producción

3.3 Pruebas

Una de las mejores características de la metodología XP es el proceso de pruebas. Esta metodología propone probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones. XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente.

Caso de prueba de aceptación	
Código Caso de Prueba: [Inicial del proyecto-número de la HU a la que	Nombre Historia de Usuario: [Nombre de la HU a realizar prueba.]

pertenece la prueba-número de la prueba.]	
Nombre de la persona que realiza la prueba: [Nombre y apellidos.]	
Descripción de la Prueba: [Descripción de la prueba realizada.]	
Condiciones de Ejecución: [Condiciones necesarias para poder realizar la prueba.]	
Entrada / Pasos de ejecución: [Serie de pasos necesarios para lograr la realización de la HU, y así realizar la prueba.]	
Resultado Esperado: [Que cumpla con las restricciones del producto.]	
Evaluación de la Prueba: [Satisfactoria o no satisfactoria.]	

3.3.1 Pruebas de aceptación

En esta plantilla se precisan las pruebas realizadas según la historia de usuario seleccionada para realizar la comprobación y validar las funcionalidades del sistema, y de esta forma saber si está apto para ser liberado. Para realizar las pruebas de aceptación el cliente utiliza la siguiente plantilla:

A continuación se presentan las pruebas llevadas a cabo para verificar el buen funcionamiento del sistema.

Caso de prueba de aceptación	
Código Caso de Prueba: 011	Nombre Historia de Usuario: Autenticar usuario
Nombre de la persona que realiza la prueba: Yanet Serrano Camejo	
Descripción de la Prueba: El usuario se autentica para entrar a su sesión en el sistema.	

Condiciones de Ejecución: Se necesita el usuario y contraseña.
Entrada / Pasos de ejecución: El usuario entra al sistema y en la página de inicio entra su nombre de usuario y contraseña, y oprime el botón entrar.
Resultado Esperado: Si los datos son incorrectos se muestra un mensaje de error, sino se muestra la sesión del usuario, en el caso que sea el administrador se muestra la página de administración.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 24 Caso de prueba: Autenticar usuario

Caso de prueba de aceptación	
Código Caso de Prueba: 072	Nombre Historia de Usuario: Buscar actividad
Nombre de la persona que realiza la prueba: Yanet Serrano Camejo	
Descripción de la Prueba: El usuario escoge los datos necesarios para buscar una o varias actividades, (se puede buscar por intervalo de fechas, lugar, plan de trabajo).	
Condiciones de Ejecución: Se necesita escoger los datos necesarios (fecha, plan de trabajo) para la búsqueda.	
Entrada / Pasos de ejecución: El usuario escoge la fecha inicial y final en la que se desarrollan la o las actividades, el nivel del que se desea buscar la actividad, el lugar donde se desarrollan, (este último es opcional), y el nombre del plan de trabajo.	
Resultado Esperado: Se muestra en una tabla la o las actividades que el usuario escogió y al lado los detalles de esas actividades. También se muestran los datos del plan de trabajo al que pertenecen dichas actividades.	

Evaluación de la Prueba: Prueba satisfactoria.

Tabla 25 Caso de prueba: Buscar actividad

Caso de prueba de aceptación	
Código Caso de Prueba: 023	Nombre Historia de Usuario: Crear Plan de Trabajo
Nombre de la persona que realiza la prueba: Meyquel García Avilés	
Descripción de la Prueba: El usuario introduce los datos necesarios para crear un nuevo plan de trabajo.	
Condiciones de Ejecución: El usuario debe haber entrado los datos del plan de trabajo.	
Entrada / Pasos de ejecución: El usuario introduce los datos generales del plan de trabajo, la lista de prioridades dicho plan y las actividades a desarrollar con sus respectivos datos.	
Resultado Esperado: Se crea y se guarda en la base de datos el plan de trabajo, el cual se podrá visualizar en la página inicial.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 26 Caso de prueba: Crear Plan de Trabajo

Caso de prueba de aceptación	
Código Caso de Prueba: 014	Nombre Historia de Usuario: Administrar usuario
Nombre de la persona que realiza la prueba: Yanet Serrano Camejo	
Descripción de la Prueba: El administrador entra a su sesión para administrar los datos del usuario, donde éste puede insertar, modificar y eliminar al usuario que	

esté gestionando.
Condiciones de Ejecución: Debe haber entrado los datos del usuario.
Entrada / Pasos de ejecución: El administrador entra el nombre completo de la persona, el e-mail, el nombre de usuario, el nivel al que pertenece y la contraseña, luego lo inserta, en el caso que quiera modificar, escoge el usuario en la lista de usuarios y entra los nuevos datos a modificar, si desea eliminar también los escoge de la lista. Si no entra algunos de los datos necesarios, el sistema le da un mensaje de error.
Resultado Esperado: Realiza lo que el administrador decidió hacer, ya sea insertar, modificar o eliminar.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 27 Caso de prueba: Administrar usuario

Caso de prueba de aceptación	
Código Caso de Prueba: 045	Nombre Historia de Usuario: Administrar grupo
Nombre de la persona que realiza la prueba: Yanet Serrano Camejo	
Descripción de la Prueba: El administrador entra a su sesión para administrar los datos de los grupos, donde éste puede insertar, modificar y eliminar el grupo que esté gestionando.	
Condiciones de Ejecución: Debe haber entrado los datos del grupo.	
Entrada / Pasos de ejecución: El administrador entra el nombre del grupo, el e-mail y la descripción luego lo inserta, en el caso que quiera modificar, escoge el grupo en la lista de grupos y entra los nuevos datos a modificar, si desea eliminar también los escoge de la lista. Si no entra algunos de los datos necesarios, el sistema le da un mensaje de error.	

Resultado Esperado: Realiza lo que el administrador decidió hacer, ya sea insertar, modificar o eliminar.

Evaluación de la Prueba: Prueba satisfactoria.

Tabla 28 Caso de prueba: Administrar grupo

Caso de prueba de aceptación	
Código Caso de Prueba: 056	Nombre Historia de Usuario: Administrar lugares
Nombre de la persona que realiza la prueba: Yanet Serrano Camejo	
Descripción de la Prueba: El administrador entra a su sesión para administrar los datos de los lugares, donde éste puede insertar, modificar y eliminar el lugar que esté gestionando.	
Condiciones de Ejecución: Debe haber entrado los datos del lugar.	
Entrada / Pasos de ejecución: El administrador entra el nombre del lugar, la ubicación y la categoría del lugar luego lo inserta, en el caso que quiera modificar, escoge el lugar en la lista de lugares y entra los nuevos datos a modificar, si desea eliminar también los escoge de la lista. Si no entra algunos de los datos necesarios, el sistema le da un mensaje de error.	
Resultado Esperado: Realiza lo que el administrador decidió hacer, ya sea insertar, modificar o eliminar.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla 29 Caso de prueba: Administrar lugares

Caso de prueba de aceptación	
Código Caso de Prueba: 077	Nombre Historia de Usuario: Administrar niveles

Nombre de la persona que realiza la prueba: Yanet Serrano Camejo
Descripción de la Prueba: El administrador entra a su sesión para administrar los datos de los niveles, donde éste puede insertar, modificar y eliminar el nivel que esté gestionando.
Condiciones de Ejecución: Debe haber entrado los datos del nivel.
Entrada / Pasos de ejecución: El administrador entra el nombre del nivel, y el nivel superior luego lo inserta, en el caso que quiera modificar escoge el nivel en la lista de niveles y entra los nuevos datos a modificar, si desea eliminar también los escoge de la lista. Si no entra algunos de los datos necesarios el sistema le da un mensaje de error.
Resultado Esperado: Realiza lo que el administrador decidió hacer, ya sea insertar, modificar o eliminar.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla 30 Caso de prueba: Administrar niveles

Conclusiones

Al concluir este capítulo se pudo precisar la realización de las pruebas según la historia de usuario seleccionada para consumir la comprobación y validar las funcionalidades del sistema, se logró controlar y organizar cada una de las actividades a desarrollar por el equipo de trabajo, se confeccionaron las tareas de ingeniería que propiciaron el proceso de implementación, así como el modelo de datos y las tarjetas CRC.

Capítulo 4 Estudio de Factibilidad

Introducción

Para estudiar la factibilidad de este proyecto se utilizará la Metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo, que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.
- La efectividad, que se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo).

4.1 Efectos Económicos

- Efectos directos
- Efectos indirectos
- Efectos externos
- Efectos Intangibles

4.1.1 Efectos directos

Efectos positivos

- Permitirá compartir información entre los usuarios del sistema.
- Se facilitará la configuración de los datos para la creación de los Planes de Trabajo.
- Mejorará la eficiencia en la elaboración de los Planes de Trabajo.
- Permitirá la búsqueda de información a los usuarios del sistema de acuerdo a sus necesidades.

Efectos negativos

Para usar la aplicación es imprescindible el uso de un ordenador, aparejado a los gastos que este trae de consumo de corriente eléctrica y mantenimiento.

4.1.2 Efectos indirectos

Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

4.1.3 Efectos externos

Se contará con una herramienta disponible que permitirá a los usuarios visualizar y elaborar los datos correspondientes de los Planes de Trabajo en el ISMM.

4.1.4 Intangibles

En la estimación financiera siempre hay elementos visibles por un grupo como perjuicio o beneficio, pero al momento de darle valor en unidades monetarias esto resulta difícil o prácticamente imposible. A fin de medir con precisión los efectos, deberán considerarse dos situaciones:

Situación sin Proyecto

En el Instituto se elabora un Plan de Trabajo general en el que deben participar todas las facultades, departamentos y demás trabajadores del centro que estén vinculados con las actividades planificadas, luego este plan es enviado a las facultades donde cada facultad debe evaluar los espacios libres para poder desarrollar su propio plan, asimismo ocurre con los departamentos, éstos deben respetar los horarios de los planes de trabajos de sus respectivas facultades para poder elaborar el de ellos. Todo este proceso provoca la demora de planificación y entrega de los Planes de Trabajo ya que cada nivel debe esperar a su inmediato superior para elaborar el suyo.

Antes de que el centro elabore su plan de trabajo los departamentos y facultades envían propuestas de actividades para que sean incluidas en el plan del centro, si éstas son aprobadas se incluyen. Hay ocasiones en las que el plan del centro ya está elaborado y cuando llegan esas actividades no se pueden publicar o si son aprobadas debe ser cambiado el plan de trabajo, todo esto se realiza manualmente o utilizando procesadores de textos trayendo consigo que el trabajo se torne tedioso y lento y no se satisfacen completamente sus necesidades.

Situación con Proyecto

Mediante este sistema es posible centralizar el punto de acceso para la información referente a los Planes de Trabajo y que pueda ser accedida así como clasificada en base a los requerimientos de nuestra universidad. Permite compartir información entre grupos de trabajo de manera distribuida.

Los usuarios con solo entrar al sistema podrán ver las actividades diarias así como visualizar los datos relacionados con su información personal y en dependencia del usuario que sea podrá elaborar los Planes de Trabajo que le corresponda.

4.2 Beneficios y Costos Intangibles en el proyecto

Costos:

- Resistencia al cambio.

Beneficios:

- Mayor comodidad para los usuarios.
- Mayor rapidez en la búsqueda de información.

4.3 Ficha de costo

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar Una Ficha De Costo de un Producto Informático [Dra. Ana María Gracia Pérez, UCLV]. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costos en Moneda Libremente Convertible:

▪ **Costos Directos.**

1. Compra de equipos de cómputo: No procede.
2. Alquiler de equipos de cómputo: No procede.
3. Compra de licencia de Software: No procede.
4. Depreciación de equipos: \$ 60.78.
5. Materiales directos: No procede.

Total: \$ 60.78.

▪ **Costos Indirectos.**

1. Formación del personal que elabora el proyecto: No procede.

2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

Total: \$0.00.

Gastos de distribución y venta.

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

Total: \$0.00.

Costos en Moneda Nacional:

▪ Costos Directos.

1. Salario del personal que laborará en el proyecto: \$100.00
2. El 5% del total de gastos por salarios se dedica a la seguridad social: No procede.
3. El 0.09% de salario total, por concepto de vacaciones a acumular: No procede
4. Gasto por consumo de energía eléctrica: \$ 10.94.

5. Gastos en llamadas telefónicas: No procede.

6. Gastos administrativos: No procede.

▪ **Costos Indirectos.**

1. Know How: \$ 108, 75.

Total: \$ 219, 69.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en minutos empleado para introducir los datos de entrada al sistema.

Valores de la variable (Solución manual):

1. Hacer modificación a alguna actividad del Plan de Trabajo (5 min).
2. Buscar actividades que se desarrollan en un tiempo y lugar determinados (10 min).
3. Buscar cada nivel las actividades de su nivel superior para desarrollar su propio Plan de Trabajo (30 min).

Valores de la variable (Solución con el programa):

1. Modificar actividades (1 min).
2. Buscar actividades (0.20 min).

3. Mostrar sugerencias de fechas (0.10 min).

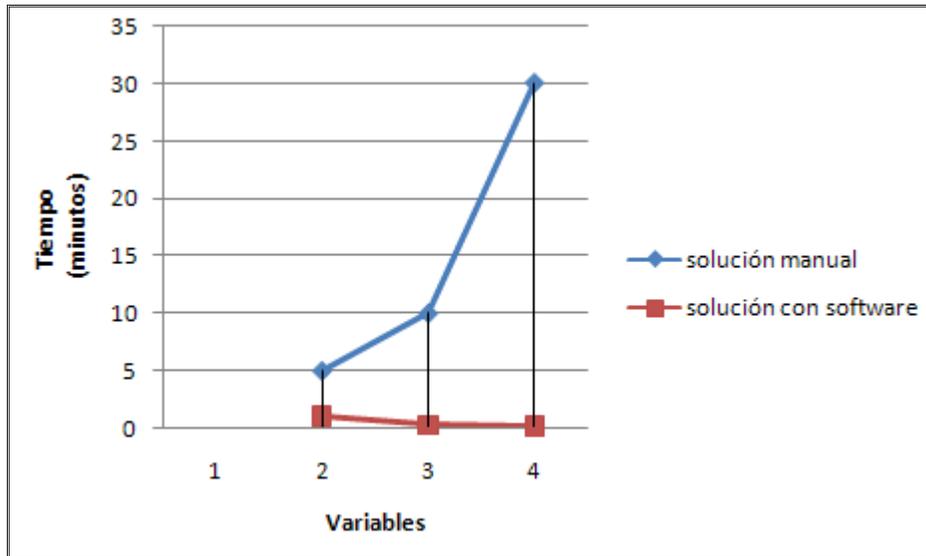


Ilustración 3 Punto de Equilibrio

Conclusiones

En este capítulo se realizó el estudio de factibilidad mediante La Metodología Costo Efectividad (Beneficio), se analizó los efectos económicos, los beneficios y costos intangibles, así como se calculó el costo de ejecución del proyecto mediante la ficha de costo arrojando como resultado \$ 60.78 CUC y \$ 219,69 MN demostrándose la factibilidad del proyecto.

Conclusiones Generales

- ❖ Se estudió cómo se lleva a cabo el proceso de elaboración de los Planes de Trabajo en el ISMM, entendiendo así el flujo actual de este proceso en la Universidad.
- ❖ Se conoció acerca de seis de los sistemas informáticos que en el mundo se utilizan para la gestión de Planes de Trabajo; los mismos no satisfacen las necesidades de nuestro Centro, lo que justifica el análisis, diseño e implementación de openPlanner.
- ❖ Se logró desarrollar un software que permite crear, modificar y eliminar Planes de Trabajo, además de realizar búsquedas personalizadas de actividades y maneja la seguridad del sistema a través de permisos otorgados a los usuarios.

Recomendaciones

- ❖ Utilizar el sistema propuesto como apoyo al proceso de elaboración de los Planes de Trabajo del ISMM, lo cual puede hacerse extensivo a otras instituciones, que así lo consideren necesario.
- ❖ Realizar un curso de entrenamiento para que las personas aprendan a utilizar el software.
- ❖ Incentivar la creación de nuevos módulos para continuar desarrollando el Sistema para la Gestión de los Planes de Trabajo en el ISMM.

Bibliografía

AWARD, J. *Bruce Eckel's, Thinking in Java*. 4 ed. Nueva York: Prentice Hall, 2006.
[en línea]. [Consultado: 2010]. Disponible en: <http://mindview.net/Books/TIJ4>

DAYRON PÉREZ, Roldan. *Sistema de Clonación y Distribución de Imágenes de Sistemas Operativos*. Abel Meneses Abad (tutor). Trabajo de Diploma. Universidad de las Ciencias Informáticas. Ciudad La Habana, 2008.

Extreme Programming: A gentle introduction. [en línea]. 28 de Septiembre 2009.
[Consultado: 2010]. Disponible en: <http://www.extremeprogramming.org>

GARCÍA AVILÉS, M; Pérez Toirac, N. *Implementación del Módulo de Estimación de Costos del ERP Nazim*. Pablo R. López Martínez (tutor). Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, 2008.

MUÑOZ SUBIRÓS, D. R. *Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*. Roiky Rodríguez Noa (tutor). Trabajo de Diploma. Instituto Superior Minero Metalúrgico de Moa “Dr. Antonio Núñez Jiménez”, Moa, 2009.

Portal del IDE Java. [en línea]. 2009 [Consultado: 2010]. Disponible en: <http://www.netbeans.org>

PRESMAN, R. S. *Ingeniería del Software, un enfoque práctico*. La Habana: Editorial Félix Varela, 2005.

ROMERO PEÑALVER, G. M. *Metodología ágil para proyectos de software libre*. Maidely Calderón Montero (tutor). Trabajo de Diploma. Universidad de las Ciencias Informáticas. Ciudad La Habana, 2008.

Wikipedia. La enciclopedia libre. [en línea]. 2009 [Consultado: 2010]. Disponible en:

<http://es.wikipedia.org/wiki/NetBeans>

Glosario de Términos

Plan de Trabajo: Documento que contiene actividades planificadas con el fin de cumplir objetivos específicos definidos por una organización.

Actividad: Es un evento que se realiza para cumplir un objetivo.

Persona: Es la que interactúa directamente con el Plan de Trabajo, elaborando el mismo y participando en las actividades.

Grupo: Es un conjunto de personas.

Lugar: Es la ubicación donde se realizan las actividades.

Nivel: Es la instancia que emite el Plan de Trabajo.

Prioridad: Cuando se elabora el Plan de Trabajo, se definen las prioridades, que no es más que las actividades u objetivos de mayor importancia.

Tipología _ actividad: La tipología es la clasificación de la actividad, por ejemplo, reunión, visitas, actos, etc.

Informe de cumplimiento: Es el documento que contiene la constancia del cumplimiento de las actividades, así como las personas presentes y ausentes en dicha actividad.

XML: Extensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

RMI: Mecanismo ofrecido en Java para invocar un método remotamente. Al ser RMI parte estándar del entorno de ejecución Java, usarlo provee un

mecanismo simple en una aplicación distribuida que solamente necesita comunicar servidores codificados para Java.

Servicios Web: Conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet.

Servlets: Objetos que corren dentro del contexto de un contenedor de servlets (Ej.: Tomcat) y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones (Ej.: OC4J Oracle) que además de contenedor para servlet tendrá contenedor para objetos más avanzados como son los EJB.

Sun Microsystems: Sun Microsystems es una empresa informática de Silicon Valley, fabricante de semiconductores y software. Fue constituida en 1982 por el alemán Andreas von Bechtolsheim y los norteamericanos Vinod Koshla, Bill Joy, Scott McNealy y Marcel Newman.

Portlets: Los portlets son componentes modulares de interfaz de usuario gestionados y visualizados en un portal web. Los portlets producen fragmentos de código de marcado que se agregan en una página de un portal.

JMS: La API de Servicios de Mensajería de Java (también conocida por sus siglas JMS) es la solución creada por SUN para el uso de colas de mensajes. Este es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma de Java 2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona.

Framework: En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de

programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

API: Una API (del inglés Application Programming Interface) es el conjunto de funciones y que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

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

Mapeo Objeto-Relacional: El mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo). Hay paquetes comerciales y de uso libre disponibles que desarrollan el mapeo relacional de objetos, aunque algunos programadores prefieren crear sus propias herramientas ORM.

Anexos

Anexo1: Plantilla Historia de Usuario del Negocio

Historia de Usuario	
Número: 03	Nombre Historia de Usuario: Autenticar usuario
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yanet Serrano Camejo	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.5
Descripción: Cuando se entra al sistema el usuario debe identificarse con su nombre de usuario y contraseña.	
Observaciones:	
Prototipo de interfase:	
	

Historia de Usuario	
Número: 05	Nombre Historia de Usuario: Administrar lugares
Modificación de Historia de Usuario Número: ninguna	
Usuario: Meyquel García Avilés	Iteración Asignada: 2

Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.5
Descripción: Cuando se elabora un Plan de Trabajo, cada actividad tiene un lugar donde se han de desarrollar, esta HU se encarga de gestionar esos lugares de modo que el usuario pueda ver las actividades que se desarrollarán en un lugar determinado.	
Observaciones:	
Prototipo de interfase:	

Historia de Usuario	
Número: 06	Nombre Historia de Usuario: Administrar niveles
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yanet Serrano Camejo	Iteración Asignada: 2

Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: Cuando se elabora un Plan de Trabajo, hay que tener en cuenta el nivel al que pertenece, que puede ser centro, facultad o departamento, esta HU le permite al administrador, gestionar esos niveles.	
Observaciones:	
Prototipo de interfase:	

Historia de Usuario	
Número: 07	Nombre Historia de Usuario: Buscar Plan de trabajo
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yanet Serrano Camejo	Iteración Asignada: 1

Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 1
Riesgo en Desarrollo: Bajo	Puntos Reales: 1
Descripción: Esta HU le da la posibilidad al usuario de conocer las prioridades que se encuentran en el plan de trabajo buscado.	
Observaciones:	
Prototipo de interfase:	

Historia de Usuario	
Número: 09	Nombre Historia de Usuario: Organizar salidas de las actividades.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Meyquel García Avilés	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 1

(Alta / Media / Baja)	
Riesgo en Desarrollo: Medio	Puntos Reales: 0.5
Descripción: Esta HU permite que cuando el usuario desee buscar una o varias actividades el sistema sea capaz de ordenarlas según la opción que el usuario escoja, que puede ser cronológicamente o por la tipología de la actividad.	
Observaciones:	
Prototipo de interfase: [Imagen de cada una de las interfaces relacionadas con la HU.]	

Historia de Usuario	
Número: 10	Nombre Historia de Usuario: Mostrar actividades diarias.
Modificación de Historia de Usuario Número: ninguna	
Usuario: Yanet Serrano Camejo	Iteración Asignada: 1
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 2
Riesgo en Desarrollo: Medio	Puntos Reales: 1
Descripción: El sistema se encarga de actualizar las actividades diarias en la página inicial.	
Observaciones:	
Prototipo de interfase: [Imagen de cada una de las interfaces relacionadas con la HU.]	

Anexo 2: Tareas de Ingeniería por Historias de Usuario

Tarea de Ingeniería	
Número Tarea: 004	Número Historia de Usuario: 01
Nombre Tarea: Cambiar de usuario y contraseña.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
<p>Descripción: El sistema debe permitir que se pueda modificar el nombre de usuario y contraseña de una determinada persona. Esta tarea solo la puede ejecutar el administrador del sistema.</p>	

Tarea de Ingeniería	
Número Tarea: 007	Número Historia de Usuario: 02
Nombre Tarea: Eliminar actividades	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
<p>Descripción: En el caso de que el usuario necesite eliminar una actividad por una razón determinada, esta tarea responderá a su petición, pero solo eliminará las actividades que le están permitidas.</p>	

Tarea de Ingeniería

Número Tarea: 008	Número Historia de Usuario: 02
Nombre Tarea: Crear prioridades	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: Diseñar en la interfaz gráfica una opción que permita definir las prioridades del plan de trabajo que se esté elaborando en ese momento.	

Tarea de Ingeniería	
Número Tarea: 009	Número Historia de Usuario: 02
Nombre Tarea: Modificar prioridades	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: En el caso de que el usuario necesite modificar una o varias prioridades por una razón determinada, esta tarea responderá a su petición, pero solo modificará las prioridades que le están permitidas.	

Tarea de Ingeniería	
Número Tarea: 010	Número Historia de Usuario: 02
Nombre Tarea: Eliminar prioridades	

Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: En el caso de que el usuario necesite eliminar una o varias prioridades por una razón determinada, esta tarea responderá a su petición, pero solo eliminará las prioridades que le están permitidas.	

Tarea de Ingeniería	
Número Tarea: 011	Número Historia de Usuario: 04
Nombre Tarea: Insertar grupos	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: En las actividades que se planifican en el Plan de Trabajo participan grupos de personas que están definidos según un criterio, ejemplo, los que pertenecen a la UJC, PCC, etc. Esta tarea se encarga de insertar esos grupos.	

Tarea de Ingeniería	
Número Tarea: 012	Número Historia de Usuario: 04
Nombre Tarea: Modificar grupos	

Tipo de Tarea : Desarrollo	Puntos Estimados: 0.25
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: Esta tarea será implementada para el caso de que se necesite modificar uno o varios grupos, solo lo podrá realizar el administrador.	

Tarea de Ingeniería	
Número Tarea: 013	Número Historia de Usuario: 04
Nombre Tarea: Eliminar grupos	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.25
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Meyquel García Avilés	
Descripción: Esta tarea será implementada para el caso de que se necesite eliminar uno o varios grupos, solo lo podrá realizar el administrador.	

Tarea de Ingeniería	
Número Tarea: 014	Número Historia de Usuario: 05
Nombre Tarea: Insertar lugar	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	

Descripción: Cada actividad se desarrolla en un determinado lugar, teniendo cada lugar un nombre y una ubicación diferentes, y podemos conocer dado un lugar cuales son las actividades que se desarrollarán en ese lugar durante el mes, la semana o en el día. Esta tarea permite insertar esos lugares, que solo lo puede realizar el administrador del sistema.

Tarea de Ingeniería	
Número Tarea: 015	Número Historia de Usuario: 05
Nombre Tarea: Modificar lugar	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.25
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Esta tarea será implementada para el caso de que se necesite modificar uno o varios lugares.	

Tarea de Ingeniería	
Número Tarea: 016	Número Historia de Usuario: 05
Nombre Tarea: Eliminar lugar	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.25
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	

Descripción: Esta tarea será implementada para el caso de que se necesite eliminar uno o varios lugares.

Tarea de Ingeniería	
Número Tarea: 017	Número Historia de Usuario: 06
Nombre Tarea: Insertar nivel	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.5
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Esta tarea será implementada para el caso de que se necesite insertar uno o varios niveles.	

Tarea de Ingeniería	
Número Tarea: 018	Número Historia de Usuario: 06
Nombre Tarea: Modificar nivel	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.25
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Esta tarea será implementada para el caso de que se necesite modificar el nivel.	

Tarea de Ingeniería	
Número Tarea: 019	Número Historia de Usuario: 06
Nombre Tarea: Eliminar nivel	
Tipo de Tarea : Desarrollo	Puntos Estimados: 0.25
Fecha Inicio:	Fecha Fin:
Programador Responsable:: Yanet Serrano Camejo	
Descripción: Esta tarea será implementada para el caso de que se necesite eliminar el nivel.	