



Universidad de las Ciencias Informáticas

Facultad 5

Sistema para la planificación de las actividades de los
individuos en la comunidad universitaria de la Universidad de
las Ciencias Informáticas

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN
CIENCIAS INFORMÁTICAS

AUTOR:

Pedro Alejandro Hernández Jiménez

TUTOR:

Dra. Karina Pérez Teruel

La Habana, 2015
“Año 57 de la Revolución”

Declaración de Autoría

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes ____ de del año 2015.

Pedro Alejandro Hernández Jiménez

Autor

Dra. Karina Pérez Teruel

Tutor

Dedicatoria

A mi mamá por ser todo para mí, por estar ahí en todo momento. Por su apoyo, dedicación y cariño que me ha dado. Por ayudarme a convertirme en el hombre que soy. Por ser mi ejemplo en la vida. Por indicarme el camino a seguir, pero dejarme elegir por mí mismo y apoyarme en mis decisiones. Por hacerme sentir orgulloso cada vez que hablo de ella.

A mis abuelos (papi y mami) por darme todo su apoyo y ayuda incondicional, por brindarme cariño, amor y quererme como su niño pequeño. Por estar siempre preocupándose por mí y ayudarme en todo lo que puedan.

A mi tía Tata que es la tía de mi vida, por estar siempre apoyándome en todo. Por brindarme su amor desde el primer día en que nací, por cuidarme mucho como si fuese su hijo.

A mis tíos Pedrito y Ale, que son los mejores tíos del mundo, por escucharme, darme consejos y ayudarme en todo lo que puedan.

A mis primos que son los hermanos que nunca tuve, gracias por ayudarme y apoyarme en todo.

A toda mi familia y amigos cercanos.

Agradecimientos

A mi mamá, por estar siempre ahí cuando más la necesito, por depositar toda su confianza en el empeño de hacer de mí la persona que soy, por guiarme en todo momento, por su infinito amor y comprensión... gracias de todo corazón... estoy muy orgulloso de ser su hijo. Te quiero PEPA.

A mi familia por apoyarme siempre y creer en mí.

A Fredy, que es ese hermano que nunca tuve, por ayudarme siempre que se me antojaba algo, por enseñarme que una nota mala no define un persona, y que ser alumno de biblioteca tampoco te convierte en inteligente, por tus consejos y tus críticas, sabes muy bien que siempre te escucho.

A Dyan, por creer desde primer año en mí, por ayudarme en todo momento a ver las vida de forma lógica, por decirme MAL cuando todos me decían BIEN, por decirme SI PUEDES cuando hasta yo mismo creía que NO PODÍA. Por ser una persona sacrificada, yo creo en ti.

A Victor, por enseñarme a creer, en creer en algo que quieres sin importar el trabajo que cueste y lo que puedan decir la gente, al final cuando vean que te gusta y que te apasiona ellos también van a creer en ti.

A Gonzalo, que es esa persona que te enseña a disfrutar cada instante de la vida, y te hace olvidar de todos los problemas por un tiempo.

A Naidy y Leo, que desde el primer momento me estrecharon su mano para ayudarme sin apenas conocerme. Por sus increíbles comidas y dulces que preparaban y por escucharme cada vez que me sucedía algo.

A Tamara que se ha convertido en esa persona que está siempre ahí cuando la necesitas, espero algún día devolverte todo lo que has hecho por mí.

Al grupo de Nerds del 88203 (L, Nexo, Yugi, White, Pizza y Canito), son personas muy talentosas que puedes conversar con ellos de cualquier tema, siempre te entienden, te

dan su criterio y después te buscan en internet para validar las respuestas, gracias por ser mi familia durante un curso completo.

A los que vinieron desde primer año conmigo en la misma aula, y aquellos que se unieron en el camino.

A Sonia, mamá de Fredy, que dijo SI y me brindó su ayuda.

A la UCI, donde he pasado los mejores 5 años de mi vida, y donde me he convertido en el profesional que soy hoy.

A los tribunales que me han atendido, gracias por sus críticas constructivas y consejos que me han ayudado a prepararme para este momento tan importante

A todos ustedes, **MUCHAS GRACIAS.**

Resumen

Los estudiantes y profesores encargados de la planificación de actividades dentro de la Universidad dedican su esfuerzo para brindar a la comunidad universitaria información sobre eventos culturales, educativos y científicos que están próximos a una fecha determinada. Esta información es enviada mediante correo electrónico o publicado en algún sitio de la universidad donde dedican un artículo a la difusión de dicho evento. Sin embargo, debido al dinamismo de la UCI es natural que surjan nuevas actividades y las anteriores a esta queden en algún lugar dentro de la bandeja de entrada y el sitio donde fue publicada ya cuenta con noticias más frescas, pues no existe un mecanismo centralizado para la planificación y divulgación de actividades.

Este trabajo propone el desarrollo de un sistema para la planificación de las actividades que permita ubicar en calendarios las actividades de la comunidad universitaria, agregando la capacidad para planificar tareas en grupo.

Palabras claves: aplicación web, calendario, planificación, tarea.

Índice de Contenido

Introducción.....	1
Capítulo 1 Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Planificación	5
1.3 Planificación colaborativa.....	6
1.3.1 Pasos para la planificación colaborativa	6
1.4 Calendarios	6
1.4.1 Calendarios de actividades.....	7
1.4.2 Calendarios electrónicos	7
1.5 Análisis de otras soluciones existentes	8
1.5.1 iCal	8
1.5.2 Chandler	9
1.5.3 Lightning Calendar	9
1.5.4 Yahoo Calendar.....	10
1.5.5 Google Calendar	12
1.5.6 Korganizer	13
1.5.7 Microsoft Office Outlook 2013	14
1.5.8 Zimbra Calendar.....	15
1.5.9 Conclusiones sobre el estudio de las comunidades	16
1.6 Metodología y tecnologías	18
1.6.1 Metodología.....	18
1.6.2 Tecnologías	21
1.7 Conclusiones parciales	28
Capítulo 2 Modelación del Sistema.....	30
2.1 Introducción.....	30
2.2 Diagnóstico del campo de acción.....	30
2.3 Descripción de la propuesta de solución.....	30
2.3.1 Personas relacionadas con la aplicación.....	31
2.4 Requisitos del sistema	31
2.4.1 Historias de usuario	33

2.4.2 Estimación de esfuerzos por historia de usuario	36
2.4.3 Requisitos no funcionales	37
2.5 Plan de iteraciones	39
2.5.1 Planificación de la duración de iteraciones	40
2.5.2 Planificación de las entregas	41
2.6 Tarjetas CRC	41
2.7 Diseño	43
2.8 Conclusiones parciales	47
Capítulo 3 Modelación del Sistema	48
3.1 Introducción	48
3.2 Tareas de ingeniería	48
3.3 Modelo de datos	51
3.4 Prueba	51
3.4.1 Pruebas de Aceptación	52
3.5 Conclusiones del Capítulo	66
Conclusiones	67
Recomendaciones	68
Bibliografía	69
Trabajos Citados	72
Anexos	75

Introducción

En el mundo de hoy las tecnologías informáticas son muy usadas tanto en países del primer mundo como del 3er mundo para el desarrollo de software con el objetivo de brindar grandes beneficios (Aiteco, 2012). En la actualidad parece ser la tarea de primer orden el uso intensivo de las nuevas tecnologías y la computación para lograr conectar todas las actividades y que estas queden completamente insertadas en los ordenadores y así manipularlas empleando el menor tiempo posible.

A través de los años en Cuba, la industria del software ha ido ascendiendo logrando alcanzar producciones de alta calidad, lo que ha permitido avanzar en las más diversas esferas de la economía y la sociedad. Se han desarrollado soluciones informáticas que facilitan las labores en las oficinas y entidades, reduciendo el gasto de materiales, y aumentando las facilidades de mantener la información almacenada.

En el 2002 surge la Universidad de las Ciencias Informáticas (UCI) como parte de la Batalla de Ideas llevada a cabo en el país. Su objetivo es formar profesionales eficientes y comprometidos, capaces de llevar el desarrollo a cada rincón del país de la mejor manera posible. Esta universidad, por sus características particulares, establece nuevas formas de control y evaluación del proceso docente.

En la UCI a través de los años se han venido desarrollando comunidades virtuales que se han convertido en un lugar donde el individuo puede desarrollarse y relacionarse con los demás miembros. Estas comunidades ofrecen un espacio donde grupos de personas se unen para de forma colaborativa mostrar sus habilidades y conocimientos.

La planificación es un tema de vital importancia dentro de una comunidad ya que facilita que se puedan realizar las actividades necesarias en el tiempo requerido, destinando a cada actividad según su prioridad los recursos de tiempo necesarios (Barkley, 2012).

Para llevar a cabo el proceso de planificación de las actividades en una comunidad se envía a los miembros, por parte de la dirección de la misma, un plan de actividades mensuales, o en algunos casos se redacta un artículo en su espacio en la web informando los venideros eventos. Sin embargo, debido al dinamismo de la UCI es

natural que surjan actividades que no han sido contempladas inicialmente y se vea afectado el desarrollo de la misma por no quedar con la calidad requerida. Actualmente esta información es enviada por correo electrónico y en algunos casos planificados en software de gestión de actividades que posibilitan compartir dichas actividades con los miembros de la comunidad, independientemente si la actividad puede ser o no de acceso público.

La capacidad para agrupar, organizar jerárquicamente y dar prioridad a los eventos (docentes, culturales, científicos, productivos y otros), constituye uno de los principales problemas del docente universitario. Es común la pérdida de tiempo en la replanificación de actividades que han sido organizadas con anterioridad y la poca capacidad de emplear efectivamente el tiempo por la ausencia de una planificación. Consecuentemente se afecta el tiempo dedicado a las estrategias o estilos de aprendizaje tanto individual como colectivo.

Derivado de la situación problemática expuesta previamente, se plantea como **problema a resolver**: ¿Cómo facilitar la gestión del tiempo de los individuos en la comunidad universitaria de la Universidad de las Ciencias Informáticas permitiendo una planificación coherente de las actividades?

El **objeto de estudio** para darle solución a este problema, está enfocado hacia los sistemas de planificación de actividades, el **campo de acción** se enmarca en los sistemas de gestión de las actividades curriculares y extracurriculares de los individuos.

Dadas las condiciones mencionadas anteriormente se propone como **objetivo general**: Desarrollar un sistema para la planificación del tiempo curricular y extracurricular de los individuos en la comunidad universitaria de la Universidad de las Ciencias Informáticas.

Para alcanzar el objetivo propuesto se definieron las siguientes **tareas de la investigación**:

1. Caracterización de las tendencias existentes para el desarrollo de los procesos de planificación de software.
2. Fundamentación de los aspectos teóricos que sustentan la investigación.

3. Análisis de las estrategias de planificación colaborativa de actividades.
4. Selección de las tecnologías informáticas y metodología de desarrollo de software apropiada para el desarrollo de la solución.
5. Generación de los artefactos ingenieriles de acuerdo a la metodología seleccionada.
6. Documentación del proceso de desarrollo del sistema.
7. Diseño de la solución.
8. Implementación de las bibliotecas de clases correspondientes para el sistema de planificación del tiempo de los individuos.
9. Validación del sistema.

Esta propuesta ofrece un sistema web para la planificación del tiempo de los individuos en la comunidad universitaria de la Universidad de las Ciencias Informáticas, además de la especificación de requisitos de un sistema que permita la gestión y planificación de las actividades.

Los métodos utilizados en la investigación se explican a continuación:

Los métodos científicos en los cuales se basa la investigación son los métodos teóricos y empíricos. Los mismos permiten organizar el trabajo, lo cual posibilita el desarrollo de un amplio conocimiento referente al estado del arte tratado, su evolución en etapas determinadas, así como adquirir de bibliografías elementos importantes para el desarrollo de mismo.

Métodos Teóricos:

Análisis-Sintético: Permitió la revisión bibliográfica que se realizó sobre un conjunto de libros, publicaciones, monografías y documentos en soporte electrónico, que se encuentran situados en Páginas Web, Trabajos de Curso, Revistas, etc. y se usó para sintetizar todas las citas, apuntes y datos tomados al respecto.

Modelación: Método que permite la expresión de la realidad mediante modelos que facilitan el proceso investigativo. Se utilizó para modelar el flujo de datos en el proceso de gestión de la planificación de actividades de los individuos.

Métodos Empíricos:

Encuesta: Se utilizó como forma de obtener la percepción del encuestado sobre el proceso de planificación de sus actividades, de manera escrita para garantizar que la investigación este basada en datos reales.

Observación: El uso de este método permitió la extracción de información obtenida de diversas fuentes sobre los mecanismos utilizados para el desarrollo del proceso de gestión y planificación de las actividades.

Para una mejor representación y distribución de la información inherente a esta investigación, se propone la siguiente estructura del documento:

Capítulo 1: Fundamentación teórica

Se refleja en este capítulo el estado del arte referente a la investigación. De igual manera muestra aspectos generales de los sistemas de control y planificación de actividades. Se exponen las tecnologías que se utilizan y su justificación de uso, así como la metodología de desarrollo a emplear.

Capítulo 2: Modelación del sistema

En este capítulo se traza como objetivo principal el proceso de construcción de la solución, en aras de obtener como resultado los artefactos que propone la metodología de desarrollo que se identificó. Además se definen las tarjetas CRC (Contenido, Responsabilidad, Colaboración) para diseñar el sistema en conjunto.

Capítulo 3: Implementación y Validación de la solución propuesta

Se detallan las iteraciones para el desarrollo del sistema, exponiéndose las tareas generadas en cada una de ellas por cada historia de usuario. Además se describen las pruebas realizadas a la aplicación. Para ello se diseñan los casos de pruebas de aceptación a los que será sometida la aplicación en cada una de las iteraciones.

Capítulo 1 Fundamentación Teórica

1.1 *Introducción.*

En este capítulo se pretende abordar los aspectos y conceptos generales relacionados con el tema de la planificación de actividades, una descripción del estado del arte del tema a tratar y las tendencias actuales que se usan para el desarrollo de productos como el que se quiere modelar. Por otra parte se ofrecen detalles sobre la metodología y las tecnologías que darán soporte a la investigación.

1.2 *Planificación*

La planificación es una lista organizada y priorizada de lo que se quiere conseguir en un plazo dado. Puede ser general o parcial, pero siempre es conveniente que exista una de orden general para que no se desequilibren los distintos aspectos de la vida. La planificación consiste en decidir con anticipación lo que hay que hacer, quién tiene que hacerlo, y cómo deberá hacerse (Murdick, 1994). Se rige como puente entre el punto en que se esté y aquel donde se quiera ir.

El hecho de planificar implica una serie de ventajas:

- Contribuye a actividades ordenadas y con un propósito. Todos los esfuerzos están apuntados hacia los resultados deseados y se logra una secuencia efectiva de tales esfuerzos (Bravo, 2007).
- Señala la necesidad de cambios futuros. Dirige la atención hacia los objetivos. Ayuda a determinar funcionalidades necesarias.
- Proporciona una base para el control. La planificación y el control son inseparables, ya que son como los gemelos de la administración. Cualquier intento de controlar sin planes carece de sentido, puesto que no hay forma de que las personas sepan si van en la dirección correcta, a no ser que primero tengan en claro a donde ir. Así, los planes proporcionan los estándares de control.
- Obliga a la visualización de un todo. Se obtiene una identificación constructiva con los distintos problemas y las diversas potencialidades de la organización en

general. Esta forma de abarcar todo es valiosa, pues capacita para obtener una comprensión más plena de cada actividad y a apreciar las bases sobre las cuales están apoyadas las acciones administrativas.

- Ayuda a tener siempre presente, por parte de todos los componentes de la organización, los objetivos de esta y la adecuación de ellos al medio, cuando es necesario (RAE, 2010).

1.3 Planificación colaborativa

La planificación colaborativa es una lista de actividades establecidas dentro de un equipo de personas (Johnson, 1997). Cada equipo de planificación está formado por las personas responsables de cada tarea en particular o de una tarea genérica capaz de involucrar a todos sus miembros.

En su sentido básico, planificación colaborativa (PC) se refiere a la actividad de pequeños grupos desarrollados previamente (Salinas, 2000). Aunque la PC es más que el simple trabajo en equipo por parte de sus miembros, la idea que lo sustenta es que dentro de cada equipo los miembros intercambian información y trabajan en una tarea común hasta que esta sea concluida.

1.3.1 Pasos para la planificación colaborativa (Aiteco, 2012).

1. Decidir cómo se formarán los grupos para ello se toma en cuenta las metas y actividades asignadas.
2. Decidir el número de integrantes que conformarán un grupo. Lo ideal sería de cinco personas por grupo.
3. Planificar la actividad que permita el logro de los objetivos.
4. Reunir los materiales necesarios.
5. Calcular el tiempo que durará la actividad.
6. Explicar los objetivos de la actividad.

1.4 Calendarios

El calendario tuvo su origen en la necesidad de los pueblos, aún de aquellos cuyo nivel cultural era muy primitivo, de dividir el tiempo, inicialmente por fines religiosos y después por fines civiles. Esta división estuvo basada en fenómenos naturales, principalmente astronómicos, como se puede advertir si se considera que las divisiones más importantes del calendario, el año, el mes y el día, corresponden, teóricamente, al tiempo que tarda la Tierra en dar una vuelta alrededor del Sol, la Luna en dar un giro completo alrededor de la Tierra, y ésta en dar un giro sobre su propio eje. El calendario que se utiliza en la vida cotidiana está basado en el romano, el cual en un principio era sólo de diez meses dejando un espacio vacío de sesenta días al final del año correspondiente a la estación muerta del invierno y, posteriormente, probablemente durante la época etrusca, bajo el reinado de Numa, se constituyó el calendario de doce meses (Bakulin, 1992)

1.4.1 Calendarios de actividades

Con el avance de las tecnologías y la necesidad de llevar el control de actividades surgen los calendarios de actividades. Estos permiten programar las mismas durante un tiempo así como recordatorios de tareas y difusión de estas. En fin, los calendarios permiten definir los días de trabajo y las horas de todo un proyecto, los grupos de recursos, cada recurso individualmente y las tareas (Barkley, 2012). Los mismos ofrecen múltiples ventajas ya que contienen eventos que pueden ser vistos por usuarios individuales, sus grupos definidos y por sus cursos. En general este facilita la planificación de todos los recursos a cualquier nivel o lugar que se quiera usar.

1.4.2 Calendarios electrónicos

Un calendario electrónico es un programa de cómputo que permite a los usuarios utilizar versiones electrónicas de herramientas de oficina diversas tales como un calendario, una agenda, un directorio telefónico y una lista de contactos (RAE, 2010). Estas herramientas son una extensión de muchas de las características que ofrecen los programas para la administración del tiempo, tales como los paquetes de accesorio para escritorio y los sistemas centrales automatizados de oficina.

Estos dispositivos verifican, de manera automática, el calendario electrónico de los miembros de un equipo, abren ventanas de tiempo, proponen horarios alternativos para las reuniones, programan las juntas y hacen citas y avisan/recuerdan a los participantes a través de mensajes de correo electrónico.

1.5 Análisis de otras soluciones existentes

En el mundo existen varias aplicaciones que se dedican específicamente a la planificación de actividades. Es por ello que se necesita realizar un profundo estudio crítico de algunos ejemplos que permita a su vez lograr un mayor entendimiento de lo que se pretende conseguir.

1.5.1 iCal

iCal es una aplicación de calendario personal hecha por Apple Inc¹. que se ejecuta en el sistema operativo Mac OS X². Está integrado con MobileMe³, por lo que los calendarios pueden ser compartidos por Internet. También se pueden compartir calendarios a través del protocolo WebDAV⁴.

iCal es un programa de Apple que permite a los usuarios que administren múltiples calendarios y las compartan a través de Internet. Permite crear y administrar actividades grupales o individuales, o múltiples calendarios (Mueller, 2000). Posibilita publicar calendarios en Internet para que colegas, amigos y familiares puedan observarlos. Permite la actualización automática vía Internet de su calendario para mantenerse al día con fechas laborales y eventos. Permite llevar los calendarios a cualquier lugar dentro

¹ Empresa multinacional que diseña y produce equipos electrónicos y software.

² Sistema operativo desarrollado y comercializado por Apple Inc.

³ Herramienta de sincronización en Internet diseñado por Apple.

⁴ Protocolo de comunicación.

de los teléfonos celulares que operan con Bluetooth⁵, dispositivos Palm OS ⁶e iPod ⁷con iCal. Con él se pueden ver todos los calendarios en una sola ventana o elegir ver sólo los calendarios que desee el usuario. Este brinda la posibilidad de avisar de una cita de varios modos, de alarma o por email.

1.5.2 Chandler

Chandler es un organizador personal. Cuenta con un calendario, un planificador de tareas y un gestor de notas. Es una aplicación de escritorio y cuenta con un servicio de apoyo llamado Chandler Hub⁸.

El objetivo que tiene es mostrar la forma en que se planifica en realidad, de forma independiente y en conjunto, especialmente en los grupos pequeños. Su visión principal representa el trabajo de información personal y de colaboración de carácter iterativo.

Chandler para escritorio y el servidor Chandler son multiplataforma y basado en estándares, para que la colaboración no quede atrapada dentro de un mismo sistema. El servidor Chandler proporciona acceso web a la información compartida que facilita a los colaboradores revisar flujos de trabajo de los calendarios sin tener que descargar la aplicación de escritorio.

1.5.3 Lightning Calendar

Lightning Calendar es un programa de agenda basado en el “Calendar” de Mozilla⁹. Este ya es una aplicación independiente basada en el módulo de calendario de Mozilla y que cumple con holgura en sus funciones de agenda, lista de tareas y calendario con alarmas.

⁵ Especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos.

⁶ Sistema operativo móvil desarrollado inicialmente por Palm Inc.

⁷ Reproductor de audio digital portátil diseñado y comercializado por Apple Inc.

⁸ Sistema para integrar y compartir tareas independientemente a la plataforma que se encuentren sus usuarios.

⁹ Organización sin ánimo de lucro que produce software libre

En su configuración básica el usuario dispone de tres ventanas. La ventana principal muestra una vista general de los eventos por día, semana o mes. Dos ventanas más pequeñas permiten visualizar las tareas definidas y un calendario.

Los eventos se definen mediante una interfaz muy simple. Como en cualquier agenda, es posible asignarle a cada evento un período de tiempo de validez (en días y horas), establecer si se repetirá en el futuro y definirlo dentro de una categoría. Se puede agregar una descripción breve del evento y una lista de direcciones de correo electrónico de las personas interesadas. Cada evento puede definirse como público o privado, con su prioridad y su estado (tentativo, confirmado o cancelado). Es posible configurar una alarma para recordar cada evento, en forma de un sonido o una ventana que aparecerá en la pantalla.

La definición de las tareas a realizar es idéntica a la mencionada para los eventos. La única diferencia es que se puede asignar a cada tarea un porcentaje de progreso, que indica cuanto nos falta para terminarla.

El usuario puede definir diferentes tipos de eventos, cada uno con un color que lo identifica. Se pueden editar los tipos predefinidos, cambiándoles el nombre y el color con el cual aparecerán en la agenda.

Se pueden manejar diferentes agendas dentro del mismo programa. Esto puede ser útil para mantener los eventos y tareas familiares separados de los del trabajo. Dado que es posible compartir la agenda personal con quienes se desee, se puede evitar que los compañeros de trabajo tengan acceso a la agenda personal, mientras que pueden consultar los eventos y tareas laborales que tengan en común.

1.5.4 Yahoo Calendar

Yahoo Calendar es una de las opciones que brinda la mensajería Yahoo. Este trae asociado disímiles funcionalidades tales como añadir evento en el cual a la hora de su creación debemos tener en cuenta especificar el nombre del evento, tipo de evento,

fecha de inicio, hora de inicio y duración, ubicación, notas, compartir el cual puede ser privado o público, establecer una recurrencia limitada por no permitir recurrencias complejas, especificar fecha fin y asignar calendario. Estos dos últimos son opcionales para los usuarios. Además permite crear recordatorios, los cuales pueden ser por email, mensajería instantánea o SMS¹⁰, además contiene informaciones opcionales.

Otra de las opciones que permite es añadir evento rápido en el cual se debe especificar nombre del evento, fecha inicio y hora de inicio. Este gestor permite añadir tareas en la que debemos tener en cuenta título, fecha tope que puede ser opcional o sin definir, prioridad que está definida de 1 a 5, estado de la tarea la cual está definida en proceso, finalizada y no iniciada, compartir que puede ser privado o público y notas.

Yahoo Calendar tiene integrado varios tipos de búsquedas tales como búsqueda rápida, búsqueda avanzada, pudiendo hacerse por un filtrado de título, nota y tipo de evento. Este calendario da la opción de decidir qué día empieza la semana para cada usuario que haga uso del mismo.

El calendario de Yahoo facilita la forma de visualizar la información ya que está estructurado por vistas las cuales dan la opción de agregar eventos en la hora o día específico y la opción de imprimir. Estas vistas son por día (dividida por horas brindando la posibilidad de navegar hacia el día anterior y siguiente), por semana (dividida por días brindando la posibilidad de navegar hacia la semana anterior y siguiente), por mes (vista de calendario clásica que posibilita navegar hacia el mes anterior y siguiente), por año (contiene todas las vistas de mes en una sola página dando la posibilidad de acceder a una semana y navegar hacia el año anterior y siguiente), por evento (muestra un listado con todos los eventos dando la opción de filtrar por eventos pendientes o pasados) y por último por tareas (muestra un listado con todas las tareas dando la opción de filtrar por tareas sin acabar o terminadas). Además de todas las opciones que tiene integrado este gestor permite modificar la zona horaria según desee el usuario que haga uso del mismo.

¹⁰ Mensajería instantánea de la telefonía móvil.

1.5.5 Google Calendar

Google Calendar es una agenda y calendario electrónico desarrollado por Google¹¹. Permite sincronizarlo con los contactos de Gmail¹² de manera que se pueda invitarles y compartir eventos. Está disponible desde el 13 de abril de 2006. Los usuarios no están obligados a tener una cuenta de Gmail, sí deben disponer de un Google Account para poder usar el software. Google Calendar el 7 de julio de 2009 dejó su calidad de Beta y pasó a ser producto terminado. La interfaz de Google Calendar tiene un rediseño a Material Design¹³ que aporta nuevas visualizaciones y detalles curiosos (Bryan, 2015). Además de que también se pueden ver nuevas funciones que se integran a la perfección con la cuenta de Google: sincronización con Gmail, integración con Google Street View¹⁴ para todos los eventos con localización.

Este gestor de calendario está estructurado por cuatro vistas. La primera vista, por día, muestra el día dividido en horas con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista dando la opción de acceder a los detalles de este y brinda la posibilidad de navegar al próximo o día anterior. La segunda vista, por semana, muestra la semana dividida en días y estos en horas, con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista, dando la opción de acceder a los detalles de este, además posibilita navegar a la próxima o semana anterior. La tercera vista, por mes, muestra la vista clásica de un calendario, con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista, dando la opción de acceder a los detalles de este, además brinda la posibilidad de navegar al próximo o mes anterior. La última vista, muestra 4 días divididos en horas donde el primero es el día actual, con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista, dando la opción de acceder a los detalles de este y navegar a los 4

¹¹ Empresa multinacional dedicada al desarrollo de soluciones en internet.

¹² Servidor de correo electrónico

¹³ Interfaz de diseño para las aplicaciones nativas

¹⁴ Herramienta de mapas que proporciona panorámicas a nivel de calle (360 grados de movimiento horizontal y 290 grados de movimiento vertical).

próximos o días anteriores. En cada una de las vistas brinda la opción de imprimir el pedazo de programación mostrada.

Google Calendar ofrece múltiples opciones entre las que se encuentra crear evento. Para poder llevar a cabo esta acción hay que tener en cuenta el nombre, fecha y hora de inicio, duración, recurrencia la cual está limitada porque no permite hacer programaciones de alta complejidad, fecha fin el cual es opcional para los usuarios, lugar, descripción, recordatorio el cual estará determinado por distintas vías y minutos de antelación y asignación a otras personas (Fedderen-Beckan, 2008). Otras de las opciones que ofrece es crear tarea, mostrando una ventana con un listado de tareas con la posibilidad de agregar tareas de forma rápida, modificarla o especificar informaciones opcionales, así como visualizarla en las vistas del calendario en un área distinta a la de los eventos. Este gestor brinda la posibilidad de exportar e importar calendarios externos a través del estándar iCalendar y permite visualizar varios calendarios a la vez identificándolos por colores. Esta agenda muestra eventos próximos y trae consigo una búsqueda rápida de fecha la cual busca una fecha más rápida con un calendario pequeño separado por meses con sus días.

1.5.6 Korganizer

Korganizer es un programa de calendario y planificación personal del proyecto KDE¹⁵. Este proporciona la gestión de eventos, tareas así como alarmas de notificación, importación y la exportación de archivos de calendario. Korganizer trabaja junto a servidores en grupo, por ejemplo Kolab, Open-Xchange, OpenGroupware o Citadel. Es personalizable a las necesidades. Permite el intercambio de datos de calendario, de vCalendar¹⁶ y de iCalendar. Este software utiliza además de iCalendar, la versión anterior de este estándar, denominada vCalendar, para tratar los problemas de compatibilidad con aplicaciones que no han evolucionado a iCalendar.

¹⁵ Comunidad Internacional que desarrolla software libre.

¹⁶ Término definido para referirse al estándar vCalendar.

1.5.7 Microsoft Office Outlook 2013

Microsoft Outlook es un programa de organización ofimática y cliente de correo electrónico de Microsoft, y forma parte de la suite Microsoft Office. Con el paso de los años ha presentado varias versiones dándole mejoramiento a la programación de calendarios del cliente o usuario, con la novedad en su versión del año 2013 de permitir compartimiento de calendarios o información de los mismos, utilizando el estándar internacional para la comunicación entre calendarios, dando la posibilidad de integrar con otra herramientas utilizadas con este fin.

Este calendario permite la gestión de varios calendarios a la vez, todos estos identificados por colores, compartimiento de calendario dando la opción de publicarlos en la web, enviarlo por correo, salvarlo en un fichero externo (extensión .ics) con el objetivo de trasladarlo o ejecutarlo con otros gestores. Además de estas opciones brinda la posibilidad de programar eventos, tareas, citas y lanzar convocatorias de grupo a las cuales permite establecer una prioridad e identificarlas con colores.

Este gestor entre sus múltiples opciones permite añadir evento en el cual a la hora de su creación hay que tener en cuenta que puede hacerse desde la vista de calendario o desde una opción general recogiendo los datos del asunto, descripción, fecha y hora inicio, fecha y hora fin del mismo, además de especificar la categoría o ámbito y prioridad.

Este permite la programación de una alarma con tiempo de antelación para el aviso de la ejecución próxima del evento. En esta acción se puede establecer una periodicidad, de la misma hay que especificar una recurrencia en la cual hay que especificar fecha y hora de inicio, fecha y hora fin, esta última siendo opcional para el usuario; además brinda la posibilidad de seleccionar la frecuencia en la que se repetirá el evento la cual está dividida en una repetición diaria, semanal, mensual, anual y cada una de estas con sus especificaciones respectivamente. Se puede también optar por la cantidad de veces que se quiere redundar el evento, característica opcional para el usuario, a su vez se le puede detallar la duración de la misma.

Este gestor de calendario trae consigo integrado la opción de añadir tareas la cual puede hacerse desde una opción general recogiendo los datos tales como asunto, descripción, fecha y hora inicio, fecha y hora fin de la tarea. Esta acción a su vez trae consigo la opción de alarma con tiempo de antelación para el aviso de ocurrencia de la misma, una periodicidad en la cual hay que detallar fecha y hora de inicio, fecha y hora fin, esta última opcional para los usuarios que utilicen la aplicación, además de una frecuencia que estará dividida en diaria, semanal, mensual, anual, cada uno con sus especificaciones), repeticiones las cual es opcional también para los usuarios, categoría o ámbito, duración y cumplimiento el cual estará dado por proceso no iniciado o cumplido.

Además de todas las anteriores opciones que engloba este gestor también trae incluida la posibilidad de añadir citas las cuales pueden hacerse desde una opción general teniendo en cuenta datos tales como asunto, descripción, fecha y hora inicio, fecha y hora fin de la misma, categoría o ámbito y prioridad. Trae consigo la opción de alarma con tiempo de antelación como método de aviso de ocurrencia de la misma, periodicidad en la cual hay que especificar fecha y hora de inicio, fecha y hora fin esta última opcional para los usuarios, frecuencia la cual estará determinada por diaria, semanal, mensual, anual y cada una de ellas con sus especificaciones, repeticiones que es opcional para los usuarios y duración de la cita.

En general este gestor trae consigo integradas las opciones de búsqueda rápida de fecha, permitiendo buscar una fecha de forma más rápida con un calendario pequeño separado por meses con sus días. Brinda vistas por días (va a estar separada por horas permitiendo agregar eventos y citas desde la propia vista), semanas (está separada por semanas permitiendo agregar eventos y citas desde la propia vista) y meses (esta muestra la vista clásica de calendarios permitiendo agregar eventos y citas desde la propia vista).

1.5.8 Zimbra Calendar

Zimbra Calendar nace como un servidor y cliente Open Source¹⁷ de mensajería y colaboración, diseñado para proveer herramientas de correo electrónico, calendarios, mensajería instantánea y documentos compartidos; facilitando la interacción de usuarios finales y administradores de sistemas. La empresa creadora de ZCS26, Zimbra Inc., desarrolla el software desde 2005, y fue comprada por Yahoo! en 2007. Hoy en día Zimbra es propiedad de VMWare¹⁸ el cual la compro después de la decisión de venta de Yahoo.

Este gestor de calendario está estructurado por múltiples vistas, en cada una da la opción de agregar eventos en la hora o día específico y la opción de imprimir. La primera de las vistas es por día, la cual está dividida por horas, con la posibilidad de navegar hacia el día anterior y siguiente. La segunda vista es la de semana, la cual está dividida por días con la posibilidad de navegar hacia la semana anterior y siguiente. La tercera vista es por mes; esta es una vista de calendario clásica que posibilita navegar hacia el mes anterior y siguiente. La cuarta vista es la de semana laboral; esta muestra una vista de los días de la semana laborables. La próxima vista es la vista por agenda, la cual muestra todas las actividades contenidas en la agenda y por último la vista de hoy que muestra las actividades planificadas para el día actual.

Zimbra Calendar brinda la posibilidad de añadir cita, para la cual hay que tener en cuenta campos tales como asunto, ubicación, asistentes, descripción y hora que está definido por evento, todo el día, fecha y hora de comienzo, fecha y hora final, repetir y recordatorio. Otra de las opciones de esta agenda es añadir tareas en la cual hay que definir el asunto, ubicación, prioridad, estado, fecha de inicio, fecha de entrega, porcentaje completado y descripción.

1.5.9 Conclusiones sobre el estudio de los sistemas de planificación

¹⁷ Se utiliza para definir un movimiento nuevo de software libre.

¹⁸ Empresa de software que proporciona software de virtualización disponible para ordenadores compatibles.

Después del análisis realizado a las diferentes soluciones existentes se llegó a la conclusión de que todos estos sistemas de planificación están centrados en la planificación de actividades tanto individual como colectiva en individuos, la posibilidad de envío de notificaciones y la gestión de notas son elementos comunes en cada sistema de planificación.

Como se muestra en la Tabla 1.1 algunos aspectos esenciales en estos sistemas es que no brindan la capacidad de comunicarse con calendarios públicos y la opción de suscripción a diferentes tareas de acceso general.

Sistema	Planificación actividades colaborativas	Creación grupos	Calendarios públicos	Múltiples calendarios	Suscripción tareas.	Gestor notas	Notificaciones
iCal	X			X		X	X
Chandler	X					X	X
Lightning Calendar	X	X		X			X
Yahoo Calendar	X						X
Google Calendar	X	X				X	X
Korganizer	X						X
Microsoft Office Outlook 2013	X			X		X	X
Zimbra Calendar	X					X	X

Tabla 1.1 Características comunes y distintivas de las soluciones existentes.

1.6 Metodología y tecnologías

Terminado el análisis de los sistemas de planificación de actividades, el siguiente paso es la selección de la metodología y las tecnologías que brindarán soporte a la solución, por lo que en los segmentos desarrollados a continuación se describen las principales características inherentes al tema de este epígrafe.

1.6.1 Metodología

XP

La Programación Extrema (XP), es una de las metodologías de desarrollo de software con más éxito en la actualidad, es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Se utiliza en proyectos con equipos de desarrollo pequeños y con plazos de entrega corto. La metodología consiste en una programación rápida o extrema. Una particularidad es tener como miembro del equipo al usuario final. Esta metodología tiene las siguientes características (Jeffries, 2001):

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se pueda hacer pruebas de las fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores.
- **Refabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento (Pressman, 2005).

XP es la más destacada dentro de las metodologías ágiles y una de las más utilizadas en el mundo por sus características pues se basa en una programación rápida y extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Las características fundamentales son:

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Programación por parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta forma, es revisado y discutido mientras se escribe.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

Proceso Unificado de Desarrollo de Software (RUP)

Se caracteriza por ser iterativo e incremental donde cada fase se desarrolla en iteraciones. También está dirigido por los casos de uso que no son más que el instrumento para validar la arquitectura del software y extraer los casos de prueba (Sommerville, 2005). Esta metodología se centra en la arquitectura, aquí los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar. Se utiliza principalmente para proyectos largos y en los que son necesarios establecer una detallada documentación (Jacobson, 2000).

Selección de la metodología de desarrollo a utilizar

Para la selección de la metodología de desarrollo de software se utilizó el método de estrella de Boehm, el cual consiste en evaluar cada uno de los parámetros representados en las situaciones y características particulares de un proyecto.

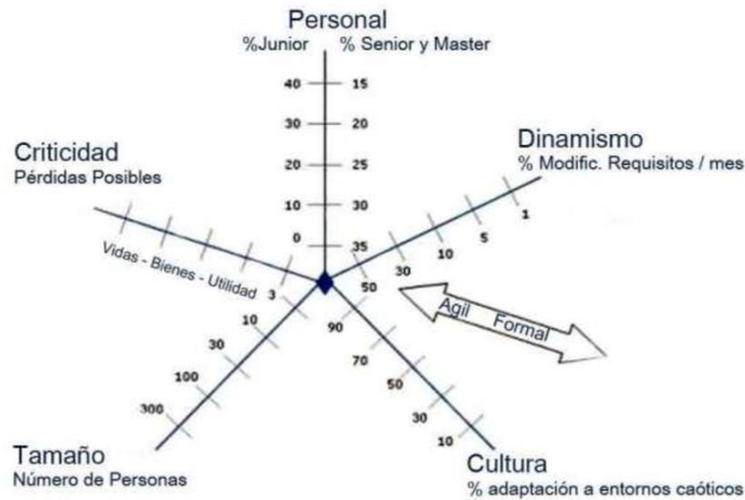


Figura 1. Criterio para la selección apropiada de una metodología de desarrollo (Argota, 2012).

Una vez culminado este proceso se obtuvo como resultado que todos los parámetros se comportaron convergiendo hacia el centro, lo cual propició la adopción de una metodología ágil. Se decidió que para el Sistema para la planificación de las actividades de los individuos en la comunidad universitaria de la UCI se utilizará XP, ya que es adaptada al software a desarrollar, así como a las condiciones de trabajo, de forma general. Propone que la comunicación y satisfacción del cliente es lo principal. No se hace mucho énfasis en la documentación, sólo es más importante definir los requerimientos y las pruebas de calidad.

Es importante destacar de esta metodología, que los requerimientos actuales pueden estar ajustados a futuros cambios, este es un punto donde la metodología es flexible, ya que permite administrar estos cambios de forma óptima.

Además, uno de los objetivos de importancia que aplica esta metodología, es potenciar al máximo el trabajo en grupo, donde los jefes de proyecto, los clientes y desarrolladores son parte del equipo y están involucrados en el desarrollo del software.

1.6.2 Tecnologías

Junto a la metodología de desarrollo, las tecnologías que se emplean para la solución del problema, son parte importante para la obtención del resultado esperado. Dichas tecnologías se describen a continuación.

PHP versión 5.6.9

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML (Ciberaula, 2010). Además, es un lenguaje de programación de estilo clásico con variables, sentencias condicionales, bucles, funciones, etc.

PHP es un lenguaje multiplataforma. Presenta gran capacidad de conexión con la mayoría de los manejadores de base de datos que se utiliza en la actualidad y destaca su conectividad con MySQL. Posee gran capacidad de expandir su potencial utilizando la enorme cantidad de módulos llamados extensiones (Alvarez, 2009). Puede leer y manipular datos desde diversas fuentes. Permite las técnicas de programación orientada a objetos, así como crear los formularios para la web.

Se presenta como una alternativa de fácil acceso, debido a que es libre. La mayoría de los sitios web publicados en internet se basan en este lenguaje de programación (Marcías, 2007). Esta versión de PHP posee una serie de nuevas características como son:

- Se ha añadido la sintaxis corta de array.
- Se ha añadido la posibilidad de referenciar la función de array.
- Los cierres ahora soportan `$this`.
- `<?=` ahora está siempre disponible, sin tener en cuenta la opción de `php.ini` `short_open_tag`.
- Se ha añadido el acceso a miembro de clase en la instanciación. o Ahora está soportada la sintaxis `Clase::{expr}()`.
- Se han mejorado los mensajes de error de análisis y las advertencias de argumentos incompatibles.
- La extensión de sesiones ahora puede rastrear el progreso de subida de ficheros.

JavaScript versión 1.8.5

JavaScript es un lenguaje de programación que permite añadir interactividad a las páginas. Es ideal para agregar ciertas funciones rápidas a una página web. Es un lenguaje muy potente de alto nivel. Es cierto que no se puede hacer nada directamente al nivel de la máquina, pero es capaz de trabajar con muchas propiedades de los exploradores web, páginas web y, a veces, con el propio sistema donde se ejecuta el explorador. No necesita una fase de compilación como C y el explorador no ha de cargar ninguna máquina virtual para ejecutar el código (Peterson, 2014). Solo hay que crear el código y cargarlo. JavaScript también utiliza una arquitectura orientada a objetos parecida a la de Java¹⁹ o C++²⁰. También se trabaja con propiedades como las funciones del constructor o la estratificación en jerarquías. Con lo que hay más opciones para utilizar el código.

Tipos de JavaScript (Negrino, 2012):

- JavaScript cliente. Se desarrolla y aplica en el navegador del cliente.
- JavaScript de servidor. Se desarrolla y ejecuta en el servidor, lo mismo que cualquier Interfaz de Entrada Común (CGI), pero con una sencillez que estos últimos desconocen.

JavaScript permite (Issi, 2012):

- Máxima interactividad entre el usuario y la página.
- Verificación de los datos introducidos por el usuario, antes de enviar el formulario al servidor.
- Ejecución de carritos de compra en el navegador.
- Ejecución de pequeñas cantidades de información al igual que en una base de datos. Manejo de applets²¹ y plug-ins²² dentro de múltiples marcos de HTML.

¹⁹ Lenguaje de programación de propósito general, concurrente y orientado a objetos.

²⁰ Lenguaje de programación.

²¹ Componente de una aplicación que se ejecuta en el contexto de otro programa.

²² Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

- Pre-procesado de información antes de enviarla al servidor.

Se selecciona este lenguaje por su sencillez y su compatibilidad con la mayoría de los navegadores. Además tiene la ventaja que mediante él se puede realizar gran parte de las funciones del cliente, lo que reduce la carga del servidor.

JavaScript también es un lenguaje muy potente de alto nivel, es dinámico y responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página, puede cambiar totalmente el aspecto de la misma. Permite realizar cálculos en base a variables cuyo valor es determinado por el usuario.

JQuery 2.0.2

JQuery es una biblioteca de JavaScript, que puede ser usada en proyectos libres y privativos, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código (Bibeault, 2013). Logra simplificar la manera de interactuar con los documentos HTML²³. Permite manejar eventos, desarrollar animaciones, y agregar interacción con la tecnología Ajax²⁴ a páginas web. Entre las ventajas de utilizar esta librería se encuentran reducción y reutilización de código, código más conciso y de fácil lectura, desarrollo de componentes dinámicos que enriquecen las aplicaciones, fácil manejo de eventos y animaciones. Ésta permite (Chaffer, 2010):

- Acceder a las partes de una página. Ofrece un mecanismo selector robusto y eficaz para recuperar exactamente la parte del documento que se vaya a inspeccionar o manipular.
- Modificar la apariencia de una página. Proporciona el mismo soporte de estándares en todos los navegadores.
- Modificar el contenido de una página. Puede modificar el contenido de un documento en sí mismo con sólo pulsar unas teclas.
- Responder a la interacción de un usuario con una página. Ofrece una forma elegante para interceptar una amplia variedad de eventos. Al mismo tiempo, su

²³ Lenguaje de marcado para la programación web.

²⁴ Técnica de desarrollo web para crear aplicaciones interactivas.

Interfaz de Programación de Aplicaciones (API) de gestión de eventos elimina inconsistencias de navegación que a menudo plagan los desarrolladores web.

- Agregar animación a una página. Proporciona información visual al usuario al ofrecer una serie de efectos como fundidos y cortinillas, así como un conjunto de herramientas para la elaboración de otros nuevos.
- Recuperar información de un servidor sin actualizar la página.
- Simplifica las tareas comunes de JavaScript. Ofrece mejoras en las construcciones básicas de JavaScript, como iteración y la manipulación de matrices.

HTML5

HTML es un lenguaje de marcas hipertextuales que está diseñado para estructurar textos para generar páginas web. El lenguaje HTML es extensible, se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas web, generando un producto vistoso, rápido y sencillo (Harris, 2010). Un documento hipertexto no sólo se compone de texto, puede contener imagen, sonido, vídeo, etc.

Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML (Frain, 2012).

Entre sus nuevas características se tiene que:

- Incorpora etiquetas (canvas6 2D y 3D, audio, video) con codecs (codificador - decodificador) para mostrar los contenidos multimedia.
- Etiquetas para manejar grandes conjuntos de datos. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios. Nuevos tipos de datos y facilidades para validar el contenido sin Javascript.

CSS3

CSS es el acrónimo de Cascading Style Sheets (es decir, hojas de estilo en cascada) es un lenguaje que tiene como objetivo separar la estructura de la presentación, este es muy usado en los documentos escritos en formato HTML, pues este nos muestra el contenido, pero el CSS define la forma en que lo hace, dándole estética al texto. Los estilos CSS aportan gran flexibilidad y control sobre el aspecto exacto que se busca en una página, desde la colocación precisa de elementos hasta el diseño de fuentes y estilos concretos (Pérez, 2009).

Esta tecnología es bastante nueva, por lo que no todos los navegadores la soportan. En concreto, sólo los navegadores de Netscape versiones de la 4 en adelante y de Microsoft a partir de la versión 3 son capaces de comprender los estilos en sintaxis CSS. Además, cabe destacar que no todos los navegadores implementan las mismas funciones de hojas de estilos, por ejemplo, Microsoft Internet Explorer 3 no soporta todo lo relativo a capas (Angel, 2010).

Para la propuesta de solución se usará el CSS3 ya que en diferencia a versiones anteriores, este permite:

- Nuevas propiedades para crear diseños de texto más atractivos.
- Bordes formados por imágenes con border-image, y sombras con box-shadow, entre otras opciones.
- La posibilidad de cambiar el tamaño y posición de una figura con la utilización de diversos métodos y la propiedad transform.
- Transiciones que permiten cambiar el estilo de un objeto de manera progresiva.

MySQL

MySQL es un gestor de bases de datos SQL²⁵ en una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Se puede agregar, acceder, y procesar datos grabados en una base de datos. Actualmente el gestor de

²⁵ Lenguaje de consulta estructurado, declarativo de acceso a bases de datos.

base de datos juega un rol central en la informática, como única utilidad, o como parte de otra aplicación (Torres, 2005).

MySQL en su diseño, su principal objetivo es la velocidad, tanto al conectarse al servidor como al responder a alguna consulta, consume muy pocos recursos tanto de Unidad de Proceso Central (CPU) como de memoria aleatoria por lo que tiene mayor rendimiento. Tiene mayores utilidades de administración, copia de seguridad, recuperación de errores y no suele perder información ni corromper datos.

La seguridad de MySQL es muy reconocida, cuenta con un sistema de privilegios y contraseñas seguro y flexible. Su tráfico de claves es encriptado en la conexión al servidor. Tiene una alta conectividad, puesto que los clientes se pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. (Hellinger 2005).

Justificación de su uso

- Veloz en la accesibilidad a los datos introducidos en las distintas tablas independientes que forman las bases de datos de este lenguaje.
- Es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños (siempre, claro está, comparada con las de su categoría).
- Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios.
- Es software libre, por tanto, se puede obtener y trabajar sin preocupaciones, utiliza la licencia GPL²⁶ para definir que se puede o no hacer.
- Es muy utilizado para el desarrollo de aplicaciones y páginas Web.
- Es multiplataforma.

El lenguaje de Modelado

²⁶ Licencia Pública General de GNU

Un modelo es una representación, en cierto medio, de algo sobre un medio. El modelo capta los aspectos importantes de lo que estamos modelando, desde cierto punto de vista, y simplifica u omite el resto. La ingeniería, la arquitectura y muchos otros campos creativos usan modelos (Rumbaugh, 2000).

El Lenguaje Unificado de Modelado (UML) permite modelar sistemas de información, y su objetivo es lograr modelos que, además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los clientes o usuarios de aquello que se modela. UML es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. No es un lenguaje de programación. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos. Es común utilizarlo para desarrollar diagramas en proyectos realizados con XP (Escribano, 2002).

UML es un método formal de modelado que aporta las siguientes ventajas (Larman, 1999):

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa.

Las principales funciones de UML son (Rumbaugh, 2000):

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

CodeIgniter 3.0

CodeIgniter es un marco de trabajo que ofrece herramientas y clases que reducen el tiempo de desarrollo de una aplicación web compleja. Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Es compatible con la mayoría de los gestores de bases de datos.

Es un completo marco de trabajo php diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación (CodeIgniter, 2010).

Se elige este marco de trabajo para el desarrollo del sistema por lo fácil de instalar y configurar en la mayoría de las plataformas, es independiente del gestor de base de datos. El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente, es capaz de trabajar en la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo se tiene un acceso por FTP para enviar los archivos al servidor.

1.7 Conclusiones parciales

La metodología de investigación posibilitó realizar el estudio del arte del objeto que se estudia. Luego del estudio realizado se pudo puntualizar los conceptos más importantes para dar cumplimiento al problema de la investigación que se realiza. Se pudo determinar que los sistemas de planificación similares no cumplen con las especificidades del negocio inherente a esta investigación. Por lo que justifica la intención del presente trabajo de llevar a cabo el desarrollo de una solución para el problema planteado. De

igual manera se investigó sobre los principales conceptos que se manejan, logrando un mejor entendimiento del negocio y la comunicación con el cliente. El estudio de las herramientas para el desarrollo de la propuesta de solución estuvo apoyado en el criterio de selección de herramientas libres y competentes para la obtención del producto final con la calidad requerida, facilitando sentar las bases para el futuro desarrollo del software.

Capítulo 2 Modelación del Sistema

2.1 Introducción

En este capítulo se describe la propuesta de solución del sistema, se definen los requerimientos funcionales y no funcionales. Además se detallan dos de las fases de la metodología seleccionada, las cuales son Exploración y Planificación, donde se confeccionan las Historias de Usuarios (HU) para cada iteración, definiendo los tiempos de entrega de cada una. De igual manera se detallan los patrones de diseño empleados.

2.2 Diagnóstico del campo de acción

Los estudiantes y profesores encargados de la planificación de actividades dentro de la Universidad dedican su esfuerzo para brindar a la comunidad universitaria información sobre eventos culturales, educativos y científicos que están próximos a una fecha determinada. Esa información es mostrada tanto por correo electrónico como en algunos portales temáticos donde dedican un artículo a la divulgación de dicho evento.

Con frecuencia se escuchan preguntas como: ¿Cuándo es el vespertino? ¿Qué día se va a realizar la actividad “X”? o preguntas retóricas del tipo: ¿Era para hoy? Pues el correo donde se envió dicha información quedó en algún lugar dentro de la bandeja de entrada y los sitios que lo divulgaron ya cuentan con noticias más frescas, por lo que en la actualidad no existe ningún mecanismo centralizado que posibilite la divulgación y planificación de dichas actividades.

2.3 Descripción de la propuesta de solución

Después del análisis expuesto anteriormente se ha llegado a la conclusión que existen varios procesos que necesitan ser informatizados de manera que se contrarreste la descentralización que sufren debido a su creación manual.

La solución propuesta por la presente investigación define la creación de un sistema web desarrollado mediante el marco de trabajo CodeIgniter, que permita concentrar toda la

información referente a la planificación de las actividades tanto individual como colectiva. El mismo debe permitir un manejo adecuado y una gestión eficiente de la información referente a estos procesos. Incluye funcionalidades como compartir actividades con un grupo de personas, creación de calendarios públicos, gestión de las diferentes actividades, y la gestión de grupos de personas. Se desarrollan otras como el envío de notificaciones de actividades y grupos a usuarios que pertenecen al sistema. En el sistema se habilitan dos tipos de calendarios que cuentan con funciones diferentes los cuales tienen como objetivo la planificación de actividades públicas y la gestión las actividades individuales.

2.3.1 Personas relacionadas con la aplicación

Son aquellas personas que interactúan con la aplicación y obtienen un resultado de valor de uno o varios procesos que se ejecutan en la misma. Los roles que pueden desempeñar los usuarios en la aplicación y una breve descripción de los mismos se muestra en la Tabla 2.1.

Roles que pueden desempeñar los usuarios	Justificación
Administrador	Gestiona la información del sistema y da soporte y mantenimiento a la aplicación.
Usuario	Gestiona la planificación de tareas, actividades colaborativas y grupos.

Tabla 2.1 Personas relacionadas con la aplicación.

2.4 Requisitos del sistema

Un requisito es una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema (Sommerville, 2005).

Los requisitos de software son las capacidades o condiciones que debe tener un sistema y están encaminados a cumplir con las exigencias que quedan plasmadas en un documento formal o contrato.

En el epígrafe 2.4.1 se describen detalladamente, mediante Historias de Usuario como lo establece la metodología a utilizar.

- Autenticar

RF1 Autenticar persona

- Gestionar persona

RF2 Insertar persona

RF3 Modificar persona

RF4 Eliminar persona

RF5 Mostrar persona

- Gestionar tarea

RF6 Insertar tarea

RF7 Modificar tarea

RF8 Eliminar tarea

RF9 Mostrar tarea

- Gestionar actividad colaborativa

RF10 Insertar actividad colaborativa

RF11 Modificar actividad colaborativa

RF12 Eliminar actividad colaborativa

RF13 Mostrar actividad colaborativa

- Gestionar grupo

RF14 Insertar grupo

RF15 Modificar grupo

RF16 Eliminar grupo

RF17 Mostrar grupo

- Gestionar notificaciones

- RF18 Insertar notificaciones**
- RF19 Modificar notificaciones**
- RF20 Eliminar notificaciones**
- RF21 Mostrar notificaciones**

- Gestionar calendario

- RF22 Insertar calendario**
- RF23 Modificar calendario**
- RF24 Eliminar calendario**
- RF25 Mostrar calendario**

Luego del proceso de captación de requisitos, el próximo paso es la especificación de los requisitos funcionales, los cuales se convertirán en las principales funcionalidades del software a desarrollar, la metodología adoptada se apoya en el artefacto (Historia de usuario) para la especificación de los mismos.

2.4.1 Historias de usuario

Las historias de usuarios (HU) son una técnica utilizada en la metodología ágil XP para especificar los requisitos de software. Las HU son el primer paso de cualquier proyecto que siga la metodología XP (Escribano, 2002). Tienen la misma finalidad que los casos de uso (CU) utilizados por la metodología RUP, pero con la diferencia que constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin muchos detalles, existiendo diferencias entre éstas y la especificación de requisitos, como por ejemplo el nivel de detalles, ya que las HU solo contienen información sobre la estimación del riesgo y el tiempo que requerirá su implementación.

Como resultado de la fase de exploración aparecen en las Tablas 2.2, 2.3, 2.4 y 2.5 descritas 4 HU, las cuales representan el flujo de actividades que permite la Gestión de tarea; en total se definieron 25 HU (13 con prioridad Alta, 8 Media y 4 Baja). Las restantes pueden ser consultadas en el documento “Historias de usuario” en la carpeta de anexos:

Historia de Usuario	
Número: 6	Nombre: Insertar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir insertar una tarea verificando que la tarea cumpla con los datos establecidos, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea • Notas • Lugar 	
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>	

Tabla 2.2 HU Insertar tarea.

Historia de Usuario	
Número: 7	Nombre: Modificar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir modificar una tarea ya existente, verificando que la tarea cumpla con los datos establecidos para la modificación, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea 	

<ul style="list-style-type: none"> • Notas • Lugar
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>

Tabla 2.3 HU Modificar tarea.

Historia de Usuario	
Número: 8	Nombre: Eliminar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir eliminar una tarea.</p>	
<p>Observaciones:</p>	

Tabla 2.4 HU Eliminar tarea.

Historia de Usuario	
Número: 9	Nombre: Mostrar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1

<p>Descripción: El sistema debe permitir mostrar una tarea:</p> <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea • Notas • Lugar
<p>Observaciones:</p>

Tabla 2.5 HU Mostrar tarea.

2.4.2 Estimación de esfuerzos por historia de usuario

Para la realización de la solución propuesta se realizó la estimación de esfuerzo la cual se presenta en la Tabla 2.6:

No.	Historia de usuario	Estimación(días)
1	Autenticar persona	1.5
2	Insertar persona	1
3	Modificar persona	1
4	Eliminar persona	1
5	Mostrar persona	1
6	Insertar tarea	1
7	Modificar tarea	1
8	Eliminar tarea	1
9	Mostrar tarea	1
10	Insertar actividad colaborativa	2.5
11	Modificar actividad colaborativa	2.5
12	Eliminar actividad colaborativa	2.5
13	Mostrar actividad colaborativa	2.5
14	Insertar grupo	1
15	Modificar grupo	1
16	Eliminar grupo	1
17	Mostrar grupo	1.5

18	Insertar notificación	1.5
19	Modificar notificación	1.5
20	Eliminar notificación	1.5
21	Mostrar notificación	1.5
22	Insertar calendario	1
23	Modificar calendario	1
24	Eliminar calendario	1
25	Mostrar calendario	1.5

Tabla 2.6 Estimación de esfuerzos por historia de usuario.

2.4.3 Requisitos no funcionales

Los requisitos no funcionales son las condiciones que necesita un software para que su rendimiento sea óptimo. Los mismos no se capturan en la metodología XP mediante historias de usuario, puesto que los clientes generalmente no conocen la terminología técnica que se utiliza para describirlos. A continuación se muestran los RNF definidos para el desarrollo del sistema:

Usabilidad

La aplicación debe ser diseñada de forma, que los usuarios que hagan uso de la misma sean capaces de aprender su funcionamiento en el menor tiempo posible, asegurando que el aspecto de las claves visuales en las páginas representen claramente y de forma adecuada la relación entre todas ellas: que cosas están relacionadas entre sí y cuales son parte de otras, definiendo una navegación consistente, clara y natural.

Confidencialidad

Toda la información manejada en el sistema está protegida de acceso no autorizado. El usuario se autentica con sus datos de acceso, en caso contrario la aplicación debe mostrar un error de autenticación.

Integridad

La integridad se refiere a la corrección y completitud de los datos en la base de datos, garantizando la calidad de los mismos. Se debe garantizar que al eliminar o actualizar un tipo de contenido se modifiquen igualmente los asociados a él. El sistema no podrá dar acceso a usuarios que no estén autenticados en el mismo.

Restricciones de diseño

Son aquellos que especifican o restringen la codificación o construcción de un sistema, son restricciones que han sido ordenadas y deben ser cumplidas estrictamente. Se deben realizar validaciones de las entradas de los datos para que el usuario cometa el mínimo de errores, mostrar mensajes de envío y hacer uso del estándar de codificación que propone el marco de trabajo de php CodeIgniter.

Interfaz

El diseño de la interfaz visual no debe tener animaciones, que repercutan en la rapidez de la aplicación. Debe ser intuitiva para los usuarios que interactúan con la aplicación, permitiendo el fácil entendimiento de las funcionalidades que el mismo brinda, además de poseer armonía entre los diferentes colores como el rojo, verde y amarillo que brindan sensación de energía, dinamismo e inteligencia de manera que se pueda crear entre ellos un puente visual para una mejor interacción entre el usuario y la aplicación.

Interfaz externa

Es importante destacar que no se trata del diseño de la interfaz en detalle, sino que especifican cómo se pretende que sea la interfaz externa del producto. Se podrá realizar las funciones en cualquier página pudiendo llegar siempre a la página principal. A todas las funcionalidades se podrá acceder desde cualquier parte del sistema. El color de los textos debe contrastar con el del fondo y el tamaño de fuente debe ser legible, entre 10–12 puntos como mínimo para que la información contenida en la misma sea fácil de leer.

Requisitos de software

En las computadoras de los usuarios se requiere un navegador Web moderno que soporte CSS3 y HTML5 (Navegador web Mozilla Firefox 10.0 o superior), pueden usar

cualquier sistema operativo. Es necesario un gestor de base de dato MySQL 5.0. En el servidor Web se requiere Apache 2 o superior y PHP 5.5.12.

Requisitos de hardware

En el cliente se requiere una máquina con 128 MB de RAM como mínimo, el servidor web junto con el servidor de base de datos debe tener 1 GB de RAM y 10 GB de disco duro mínimo debido a que el sistema solo permite por usuario la subida de imagen hasta 3 MB, todas las máquinas implicadas en el funcionamiento de la aplicación deben estar conectadas a la red.

Portabilidad

Son aquellos que especifican los atributos que debe presentar el software para facilitar su traslado a otras plataformas o entornos de desarrollo. Permitir que el sistema se ejecute sobre el Sistema Operativo Windows XP o superior y GNU/LINUX 2.6 o superior.

Rendimiento

La aplicación debe ejecutarse con el mínimo de recursos para el que fue prevista y además el tiempo de respuesta a las diferentes peticiones de los usuarios debe ser menor de 30 segundos.

2.5 Plan de iteraciones

La metodología XP se divide en iteraciones relativamente cortas. Las HU son divididas en tareas de entre 1 y 3 días, creándose el plan de iteraciones donde se especifican el orden y la duración. En la primera iteración se crea un sistema que abarca los aspectos más importantes de la arquitectura global. Esto se logra seleccionando las HU que hagan referencia a la construcción de la estructura de todo el sistema. Para el desarrollo del sistema propuesto se han definido 3 iteraciones, las cuales se describen en la Tabla 2.11:

Iteración	Orden de las historias de usuario a implementar
1	Tendrá como objetivo, la implementación de las funcionalidades básicas para la puesta en marcha del mismo. Se tendrán en cuenta las historias de usuario 1-13. De manera general al finalizar esta iteración se habrá podido cumplir con

	funcionalidades como la gestión de personas con la autenticación de las mismas y la gestión de las tareas, obteniéndose la primera versión del software.
2	Está centrada en la implementación de las HU que tienen prioridad media para el cliente; estas son de 14-21. Al finalizar esta iteración se habrán desarrollado las funcionalidades gestionar las actividades colaborativas y gestionar grupo. Una vez realizada esta iteración se puede intentar establecer una arquitectura del sistema.
3	Prioriza la implementación de las HU que tienen prioridad baja para el cliente; estas son los números 22, 23, 24 y 25.

Tabla 2.11 Plan de iteraciones.

2.5.1 Planificación de la duración de iteraciones

Después de haber sido estimado el esfuerzo dedicado a la realización de cada una de las HU identificadas, se procede a la creación del plan de iteraciones el cual se expone en la Tabla 2.12 y tiene como objetivo mostrar la duración y el orden en que serán implementadas las mismas.

Iteración	Orden de las Historias de Usuario a Implementar	Cantidad de tiempo de trabajo(días)
1	Autenticar persona Insertar persona Modificar persona Eliminar persona Mostrar persona Insertar tarea Modificar tarea Eliminar tarea Mostrar tarea Insertar actividad colaborativa Modificar actividad colaborativa Eliminar actividad colaborativa Mostrar actividad colaborativa	18.5
2	Insertar grupo Modificar grupo Eliminar grupo	10.5

	Mostrar grupo Insertar notificación Modificar notificación Eliminar notificación Mostrar notificación	
3	Insertar calendario Modificar calendario Eliminar calendario Mostrar calendario	4.5

Tabla 2.12 Plan de duración de iteraciones.

2.5.2 Planificación de las entregas

En el plan de entrega se agrupan las HU de manera ordenada con el objetivo de conformar una entrega. El plan se conforma en base a las estimaciones de tiempo que se emplea en el desarrollo de las funcionalidades. A continuación se muestra en la Tabla 2.13 el plan de entrega para cada iteración:

Sistema	Final Iteración 1	Final Iteración 2	Final Iteración 3
Versión 1.0	24/4/2015	14/5/2015	28/5/2015

Tabla 2.13 Planificación de las entregas.

2.6 Tarjetas CRC

Las tarjetas CRC (Clase, Responsabilidad y Colaboración) se dividen en tres secciones que contienen la información del nombre de las clases, sus responsabilidades y sus colaboraciones. La mayor ventaja de las tarjetas CRC es permitir reducir el modo de pensar procedural y apreciar la tecnología de objetos (Escribano, 2002). Las tarjetas CRC permiten contribuir al diseño a todo el equipo del proyecto.

A continuación se presentan las tarjetas CRC del sistema.

Tarjeta: Acceso	
Funcionalidades	Colaboraciones
Autenticar usuario	Acceso

Tabla 2.14. Tarjeta 1 CRC Acceso.

Tarjeta: Persona	
Funcionalidades	Colaboraciones
Crear persona	Acceso
Eliminar persona	Grupo
Modificar persona	Notificaciones
Obtener persona	

Tabla 2.15. Tarjeta 2 CRC Persona.

Tarjeta: Tarea	
Funcionalidades	Colaboraciones
Crear tarea	Persona
Eliminar tarea	Acceso
Modificar tarea	Notificaciones
Obtener tarea	Calendario
Obtener lista tarea	

Tabla 2.16. Tarjeta 3 CRC Tarea.

Tarjeta: Actividad colaborativa	
Funcionalidades	Colaboraciones
Crear actividad	Grupo
Eliminar actividad	Persona
Modificar actividad	Acceso
Obtener actividad	
Obtener lista actividades	
Asignar actividad a grupo	
Eliminar actividad de un grupo	
Enviar notificación	

Tabla 2.17. Tarjeta 4 CRC Actividad colaborativa.

Tarjeta: Grupo

Funcionalidades	Colaboraciones
Crear grupo	Persona
Eliminar grupo	Acceso
Modificar grupo	
Obtener grupo	
Obtener lista grupo	
Eliminar persona de grupo	
Agregar persona a grupo	
Enviar notificación	

Tabla 2.18. Tarjeta 5 CRC Grupo.

Tarjeta: Notificación	
Funcionalidades	Colaboraciones
Crear notificación	Acceso
Eliminar notificación	Persona
Modificar notificación	
Obtener notificación	

Tabla 2.19. Tarjeta 6 CRC Notificación.

Tarjeta: Calendario	
Funcionalidades	Colaboraciones
Crear calendario	Acceso
Eliminar calendario	Persona
Modificar calendario	Grupo
Obtener calendario	Tarea
	Actividades colaborativas

Tabla 2.20. Tarjeta 7 CRC Calendario.

2.7 Diseño

Patrones de diseño

Los patrones de diseño son los que expresan esquemas para definir estructuras de diseño con las que se pueden construir sistemas de software. CodeIgniter proporciona

varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. Seguidamente se mencionan los patrones más significativos que utiliza CodeIgniter.

Modelo Vista Controlador

CodeIgniter está basado en el patrón de diseño modelo vista controlador conocido como MVC, que está formado por tres niveles:

Modelo: representa las estructuras de datos. Típicamente las clases de tipo modelo contienen funciones que permiten insertar, eliminar y actualizar información en tu base de datos. Define la lógica del negocio.

Vista: es la información que es mostrada al usuario. Una vista es normalmente una página web.

Controlador: funciona como intermediario entre el Modelo y la Vista. Es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente.

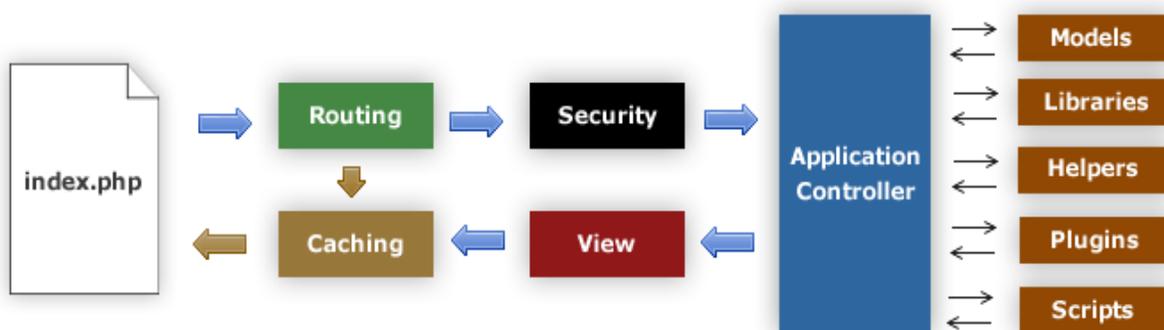


Figura 2 Patrón MVC en CodeIgniter.

El patrón representa un mecanismo de mejora de procesos de desarrollo de software, fácil de comprender y aplicar.

Permite obtener un sistema claro y bien especificado. La separación de la interfaz del resto del código elimina problemas como:

- Que a la complejidad de los cálculos del programa se le sume la complejidad de la interfaz de usuario, lo que proporcionaría un código de mayor calidad y de menor dificultad.
- Menor rigidez en la interfaz con el usuario, dado que el código HTML no estaría entremezclado con el lenguaje de programación y sería una tarea más fácil cambiar el diseño de la interfaz.

Patrones GRASP (General Responsibility Assignment Software Patterns)

Experto: Se evidencia este patrón puesto que CodeIgniter utiliza librerías para realizar una capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades. Esto nos indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Controlador: Todas las peticiones web son manejadas por un solo controlador frontal (Principal_ci), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento (Librería URI en CodeIgniter) para asociar el nombre de una acción y el nombre de un módulo con la URL entrada por el usuario.

Creador: El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase.
- Contiene o agrega la clase.

El uso de este patrón se evidencia en las clases controladoras como Tarea_ci, Persona_ci, permitiendo el acceso a la información almacenada en la clase entidad.

Bajo Acoplamiento: El bajo acoplamiento fomenta el aumento de la reutilización y la eliminación de las redundancias, creando clases más independientes y con mayor resistencia al impacto de los cambios, que aumentan la productividad y la posibilidad de reutilización. Este patrón se basa en la idea de tener las clases lo menos ligadas entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Alta cohesión: La cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase, además de que una alta cohesión garantiza que clases con responsabilidades estrechamente relacionadas no realicen un trabajo enorme, y que cada elemento de nuestro diseño debe realizar una labor única dentro del sistema.

El sistema destinado a la planificación de las actividades de los individuos en la comunidad universitaria de la UCI sigue los principios de alta cohesión ya que no se hace uso de clases saturadas de métodos.

Patrones GOF (Gang of Four)

Abstrac Factory (Fábrica Abstracta): Patrón que se utiliza para construir y devolver objetos. El patrón de la fábrica se utiliza en los casos en donde diferentes clases, por lo general derivan de una clase abstracta, son instanciadas dependiendo de los parámetros. Es utilizado por el marco de trabajo cuando este necesita acceder de forma sencilla a instancias de los objetos globales.

Singleton (Instancia Única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. El acceso a la Base de Datos en el sistema será mediante una única instancia para así impedir que se creen múltiples conexiones innecesarias.

Decorator (Decorador): Añade funcionalidad a una clase, dinámicamente. Los archivos Footer_template y Navigation_Template, que también se denomina plantillas globales, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. La Figura 3 muestra el uso del patrón como ejemplo del contenido del sistema integrándose en las plantillas.

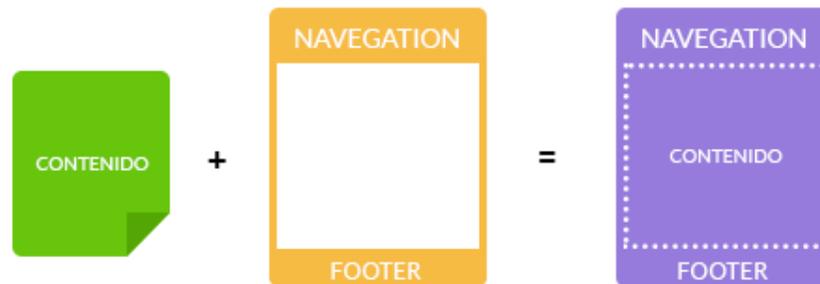


Figura 3 Patrón Decorador.

2.8 Conclusiones parciales

En este capítulo se plasman las bases para el desarrollo del sistema. Se definieron tres iteraciones que abarcan un total de veinticinco historias de usuarios, que describen los aspectos principales a tener en cuenta para el desarrollo de la solución, brindando una visión futura de las funcionalidades a implementar. Asociado a estas historias de usuarios se construyó el plan de entregas y siete tarjetas CRC que traducen los requerimientos a entidades a implementar para, de esta manera, pasar a la siguiente fase. Los principios, prácticas y técnicas propuestos por la metodología XP aportan los artefactos necesarios para la implementación de la solución. Además se describen los patrones arquitectónicos y de diseño para la implementación del producto.

Capítulo 3 Implementación y Validación de la solución propuesta

3.1 Introducción

En este capítulo se describen las tareas de ingeniería y las pruebas realizadas a la aplicación con el objetivo de verificar si se cumplieron las tareas de ingeniería planificadas para el desarrollo de la aplicación.

3.2 Tareas de ingeniería

Las tareas de ingeniería son un conjunto de acciones que se realizan con el objetivo de resolver las historias de usuario, una historia de usuario puede tener una o más tareas de ingeniería en dependencia de la complejidad de la funcionalidad a desarrollar (Beck, 1999).

En las Tablas 3.1 – 3.7 se muestran las tareas de ingeniería correspondientes a la HU Gestionar Calendario de la tercera iteración, las restantes pueden ser consultadas en el documento “Tareas de ingeniería” en la carpeta de anexos:

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 22
Nombre tarea: Insertar calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de insertar calendario en la aplicación.	

Tabla 3.1 Tarea 26 Insertar calendario.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 23
Nombre tarea: Modificar calendario	

Tipo de tarea: Desarrollo
Programador responsable: Pedro Alejandro Hernández Jiménez
Descripción: Implementar la funcionalidad de insertar calendario en la aplicación, teniendo en cuenta que los parámetros de entrada estén correctos.

Tabla 3.2 Tarea 27 Modificar calendario.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 24
Nombre tarea: Eliminar calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de eliminar calendario en la aplicación.	

Tabla 3.3 Tarea 28 Eliminar calendario.

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 24
Nombre tarea: Eliminar notificaciones del calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten eliminar todas las notificaciones del calendario a eliminar.	

Tabla 3.4 Tarea 29 Eliminar notificaciones del calendario.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 25

Nombre tarea: Mostrar información del calendario
Tipo de tarea: Desarrollo
Programador responsable: Pedro Alejandro Hernández Jiménez
Descripción: Se implementan las funcionalidades que permiten mostrar la información básica de un calendario

Tabla 3.5 Tarea 30 Mostrar calendario

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 25
Nombre tarea: Mostrar tareas del calendario en el mes	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades para mostrar todas las tareas del mes de un calendario tanto público como privado.	

Tabla 3.6 Tarea 31 Mostrar tareas del calendario en el mes

Tareas de Ingeniería	
Número de tarea: 3	Número de HU: 25
Nombre tarea: Mostrar total de tareas del calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades para mostrar la cantidad de tareas planificadas en un calendario tanto público como privado.	

Tabla 3.7 Tarea 32 Mostrar total de tareas del calendario

3.3 Modelo de datos

En el proceso de abstracción que conduce a la creación de una base de datos, desempeña una función prioritaria el modelo de datos. El modelo de datos, como abstracción del universo de discurso, es el enfoque utilizado para la representación de las entidades y sus características dentro de la base de datos. Es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos, semántica asociada a los datos y restricciones de consistencia. (Wiederhold, 1977).

Como se muestra en la figura 4, la base de datos cuenta con 8 entidades persistentes, siendo la entidad Persona la de mayor importancia, debido a que prácticamente la mayoría de las restantes tienen una relación directa con ella.

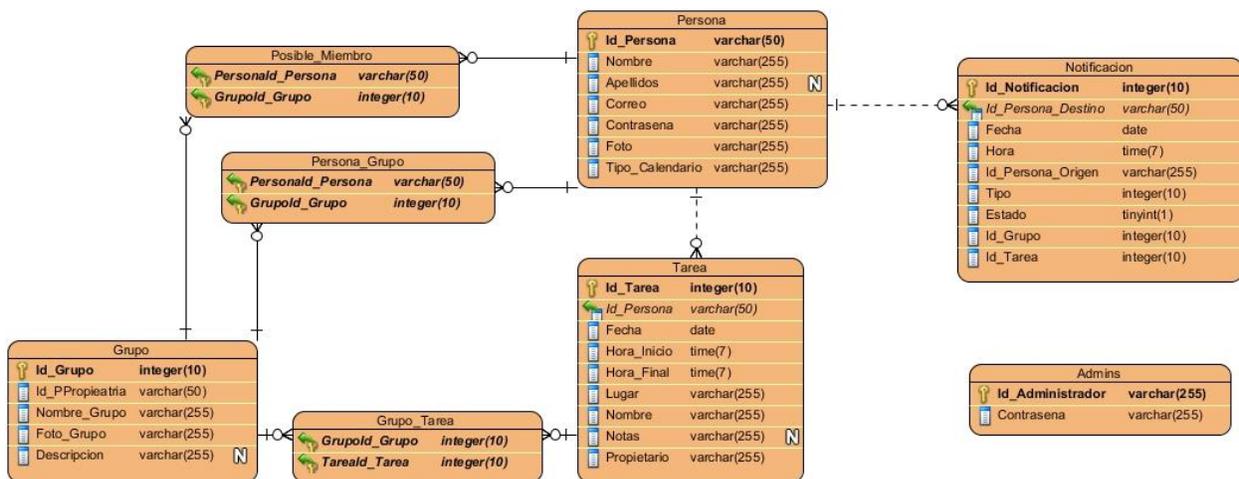


Figura 4. Modelo de Datos.

3.4 Prueba

Uno de los pilares fundamentales de XP es el proceso de prueba, lo cual constituye el último bastión desde el que se puede evaluar la calidad de forma pragmática y descubrir los errores.

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón se debe definir en el proceso de la ingeniería del software. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones (Pressman, 2005).

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (Beck, 1999).

3.4.1 Pruebas de Aceptación

Las pruebas de aceptación son realizadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada, y es responsable de verificar que los resultados de estas pruebas sean correctos (Allende, 2006). Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

A continuación se muestran las pruebas de aceptación propuestas a realizarse por cada iteración para una mayor organización.

Iteración 1

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Autenticar persona	
Descripción: Prueba para la funcionalidad autenticar persona, en caso que el usuario inserte todos los datos correctamente.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario y contraseña deben ser válidos.	
Pasos de Ejecución:	

<ul style="list-style-type: none"> - Acceder al botón “Entrar” en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página “Acceso”. - Introducir datos correctamente. - Acceder al botón “Entrar”
<p>Resultados Esperados: El usuario se autentifica correctamente en el sistema.</p>

Tabla 3.8 Caso de Prueba: HU1_P1

Caso de Prueba de Aceptación	
Código: HU1_P2	Historia de Usuario: 1
Nombre: Autenticar persona	
Descripción: Prueba para comprobar que el sistema no permite autenticar persona, en caso que el usuario inserte datos incorrectamente.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario y contraseña deben ser válidos.	
<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al botón “Entrar” en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página “Acceso”. - Introducir datos correctamente. - Acceder al botón “Entrar” 	
<p>Resultados Esperados: El sistema muestra un mensaje de error.</p>	

Tabla 3.9 Caso de Prueba: HU1_P2

Caso de Prueba de Aceptación	
Código: HU2_P1	Historia de Usuario: 2
Nombre: Insertar persona	
Descripción: Prueba para la funcionalidad insertar persona en el sistema, en caso que el usuario inserte todos los datos correctamente.	
Condiciones de Ejecución: El usuario no debe estar previamente registrado.	
<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al botón “Entrar” en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página “Acceso”. - Acceder en el vínculo “Crear una cuenta”. - Introducir los datos correctamente. - Acceder al botón “Listo” 	
<p>Resultados Esperados: El sistema inserta la persona dentro de la base de datos.</p>	

Tabla 3.10 Caso de Prueba: HU2_P1

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Insertar persona	
Descripción: Prueba para comprobar que el sistema no permite insertar persona con el mismo identificador (Id Persona).	
Condiciones de Ejecución: El usuario debe estar previamente registrado.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al botón "Entrar" en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página "Acceso". - Acceder en el vínculo "Crear una cuenta". - Introducir los datos correctamente. - Acceder al botón "Listo" 	
Resultados Esperados: El sistema muestra un mensaje indicando el error.	

Tabla 3.11 Caso de Prueba: HU2_P2

Caso de Prueba de Aceptación	
Código: HU3_P1	Historia de Usuario: 3
Nombre: Modificar persona	
Descripción: Prueba para la funcionalidad modificar persona.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado en el sistema.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Perfil". - Introducir datos correctamente. - Acceder al botón "Listo" 	
Resultados Esperados: La persona es modificada en el sistema. Mostrando un mensaje de éxito,	

Tabla 3.12 Caso de Prueba: HU3_P1

Caso de Prueba de Aceptación	
Código: HU4_P1	Historia de Usuario: 4

Nombre: Eliminar persona
Descripción: Prueba para la funcionalidad eliminar persona del sistema.
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario autenticado debe tener rol de Administrador
Pasos de Ejecución: - Acceder al botón “Entrar” en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página “Acceso”. - Seleccionar la opción “Entrar como administrador” - Introducir datos correctamente. - Ir al menú de navegación y seleccionar la opción “Persona” - Acceder al botón “Eliminar”
Resultados Esperados: La persona es eliminada del sistema.

Tabla 3.13 Caso de Prueba: HU4_P1

Caso de Prueba de Aceptación	
Código: HU5_P1	Historia de Usuario: 5
Nombre: Mostrar persona	
Descripción: Prueba para la funcionalidad mostrar persona	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado	
Pasos de Ejecución: - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Perfil”.	
Resultados Esperados: El usuario puede ver la información sobre su perfil.	

Tabla 3.14 Caso de Prueba: HU5_P1

Caso de Prueba de Aceptación	
Código: HU6_P1	Historia de Usuario: 6
Nombre: Insertar tarea	
Descripción: Prueba para la funcionalidad insertar tarea	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado.	

<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al botón “Crear nueva tarea” en la parte superior derecha por debajo de la barra de navegación. - Introducir datos correctamente. - Acceder al botón “Crear”
<p>Resultados Esperados:</p> <p>El sistema crear una tarea para el usuario autenticado.</p>

Tabla 3.15 Caso de Prueba: HU6_P1

Caso de Prueba de Aceptación	
Código: HU6_P2	Historia de Usuario: 6
Nombre: Insertar tarea	
Descripción: Prueba para comprobar que el sistema no permite insertar tarea, en caso que el usuario inserte datos incorrectamente.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado.	
<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al botón “Crear nueva tarea” en la parte superior derecha por debajo de la barra de navegación. - Introducir datos incorrectamente. - Acceder al botón “Crear” 	
Resultados Esperados: El sistema muestra un mensaje indicando el error.	

Tabla 3.16 Caso de Prueba: HU6_P2

Caso de Prueba de Aceptación	
Código: HU7_P1	Historia de Usuario: 7
Nombre: Modificar tarea	
Descripción: Prueba para comprobar que el sistema permite modificar tarea	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser propietario de la tarea.	
<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Calendario”. - Seleccionar la fecha para ver las tareas. - Acceder al vínculo “Modificar” en la parte superior derecha de las tareas, en el área del calendario. Esta acción redirecciona a la página “Modificar tarea”. - Insertar los datos a modificar - Acceder al botón “Aceptar” 	

Resultados Esperados:

El sistema modifica la tarea seleccionada.

Tabla 3.17 Caso de Prueba: HU7_P1

Caso de Prueba de Aceptación	
Código: HU8_P1	Historia de Usuario: 8
Nombre: Eliminar tarea	
Descripción: Prueba para comprobar que el sistema permite eliminar tarea	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser propietario de la tarea.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Calendario". - Seleccionar la fecha para ver las tareas. - Acceder al vínculo "Eliminar" en la parte superior derecha de las tareas, en el área del calendario. Esta acción muestra una ventana de confirmación. - Acceder al botón "Aceptar". 	
Resultados Esperados: El sistema eliminar la tarea seleccionada.	

Tabla 3.18 Caso de Prueba: HU8_P1

Caso de Prueba de Aceptación	
Código: HU9_P1	Historia de Usuario: 9
Nombre: Mostrar tarea	
Descripción: Prueba para comprobar que el sistema permite mostrar tarea	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser propietario de la tarea.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Calendario". - Seleccionar la fecha para ver las tareas. 	
Resultados Esperados: El sistema muestra las tareas planificadas.	

Tabla 3.19 Caso de Prueba: HU9_P1

Caso de Prueba de Aceptación	
Código: HU10_P1	Historia de Usuario: 10
Nombre: Insertar Actividad Colaborativa	
Descripción: Prueba para comprobar que el sistema permite insertar actividad colaborativa	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al botón "Crear nueva tarea" en la parte superior derecha por debajo de la barra de navegación. - Introducir datos correctamente. - Seleccionar un grupo con el que se desea compartir la tarea. - Acceder al botón "Crear" 	
Resultados Esperados: El sistema crea la actividad colaborativa.	

Tabla 3.20 Caso de Prueba: HU10_P1

Caso de Prueba de Aceptación	
Código: HU11_P1	Historia de Usuario: 11
Nombre: Modificar Actividad Colaborativa	
Descripción: Prueba para comprobar que el sistema permite modificar actividad colaborativa	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario autenticado debe ser propietario de la actividad colaborativa.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Calendario". - Seleccionar la fecha para ver las tareas. - Seleccionar una actividad colaborativa y acceder al vínculo "Modificar" en la parte superior derecha de la tarea, en el área de tareas. Esta acción redirecciona a la página "Modificar Actividad Colaborativa". - Insertar los datos a modificar. - Acceder al botón "Aceptar" 	
Resultados Esperados: El sistema modifica la actividad colaborativa.	

Tabla 3.21 Caso de Prueba: HU11_P1

Caso de Prueba de Aceptación

Código: HU12_P1	Historia de Usuario: 12
Nombre: Eliminar Actividad Colaborativa	
Descripción: Prueba para comprobar que el sistema permite eliminar actividad colaborativa	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario autenticado debe ser propietario de la actividad colaborativa.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Calendario". - Seleccionar la fecha para ver las tareas. - Seleccionar una actividad colaborativa y acceder al vínculo "Eliminar" en la parte superior derecha de la tarea, en el área de tareas. Esta acción muestra una ventana de confirmación. - Acceder al botón "Aceptar" 	
Resultados Esperados: El sistema elimina la actividad colaborativa.	

Tabla 3.22 Caso de Prueba: HU12_P1

Caso de Prueba de Aceptación	
Código: HU13_P1	Historia de Usuario: 13
Nombre: Mostrar Actividad Colaborativa	
Descripción: Prueba para comprobar que el sistema permite mostrar actividad colaborativa	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario autenticado debe ser propietario de la actividad colaborativa o pertenece al grupo que se le fue asignado esta tarea.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Calendario". - Seleccionar la fecha para ver las tareas. 	
Resultados Esperados: El sistema muestra la actividad colaborativa.	

Tabla 3.23 Caso de Prueba: HU13_P1

Caso de Prueba de Aceptación	
Código: HU14_P1	Historia de Usuario: 14
Nombre: Insertar Grupo	

Descripción: Prueba para comprobar que el sistema permite Insertar Grupo
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado.
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Mis Grupos”. - Seleccionar el vínculo “Nuevo Grupo”. - Introducir datos correctamente. - Acceder al botón “Guardar”
Resultados Esperados: El sistema inserta un grupo.

Tabla 3.24 Caso de Prueba: HU14_P1

Iteración 2

Caso de Prueba de Aceptación	
Código: HU14_P2	Historia de Usuario: 14
Nombre: Insertar Grupo	
Descripción: Prueba para comprobar que el sistema no permite Insertar Grupo con datos incorrectos.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Mis Grupos”. - Seleccionar el vínculo “Nuevo Grupo”. - Introducir datos incorrectamente. - Acceder al botón “Guardar” 	
Resultados Esperados: El sistema muestra un mensaje del error correspondiente.	

Tabla 3.25 Caso de Prueba: HU14_P2

Caso de Prueba de Aceptación	
Código: HU15_P1	Historia de Usuario: 15
Nombre: Modificar Grupo	
Descripción: Prueba para comprobar que el sistema permite Modificar Grupo	

Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser el propietario del grupo.
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Mis Grupos”. - Seleccionar el vínculo en forma de icono “Modificar”. - Introducir datos correctamente. - Acceder al botón “Guardar”
Resultados Esperados: El sistema modificar el grupo seleccionado.

Tabla 3.26 Caso de Prueba: HU15_P1

Caso de Prueba de Aceptación	
Código: HU16_P1	Historia de Usuario: 16
Nombre: Eliminar Grupo	
Descripción: Prueba para comprobar que el sistema permite Eliminar Grupo	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser el propietario del grupo.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Mis Grupos”. - Seleccionar el grupo que se desee eliminar - Seleccionar el vínculo en forma de icono “Eliminar”. Esta acción muestra una ventana de confirmación. - Acceder al botón “Aceptar” 	
Resultados Esperados: El sistema eliminar un grupo.	

Tabla 3.27 Caso de Prueba: HU16_P1

Caso de Prueba de Aceptación	
Código: HU17_P1	Historia de Usuario: 17
Nombre: Mostrar Grupo	
Descripción: Prueba para comprobar que el sistema permite Eliminar Grupo	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser el propietario del grupo.	

<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Mis Grupos”.
<p>Resultados Esperados:</p> <p>El sistema muestra una lista de grupos.</p>

Tabla 3.28 Caso de Prueba: HU17_P1

Caso de Prueba de Aceptación	
Código: HU18_P1	Historia de Usuario: 18
Nombre: Insertar Notificación	
Descripción: Prueba para comprobar que el sistema permite Insertar Notificaciones.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser el propietario de al menos un grupo.	
<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Calendario”. - Acceder al vínculo “Nueva Tarea” en la parte superior derecha. - Introducir los datos correctamente. - Compartir la tarea con un grupo. - Acceder al vínculo “Crear”. 	
<p>Resultados Esperados:</p> <p>El sistema inserta las notificaciones.</p>	

Tabla 3.29 Caso de Prueba: HU18_P1

Caso de Prueba de Aceptación	
Código: HU19_P1	Historia de Usuario: 19
Nombre: Modificar Notificación	
Descripción: Prueba para comprobar que el sistema permite Modificar una Notificación.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario tiene que ser el propietario de al menos un grupo. El usuario tiene que haber insertado al menos una notificación.	
<p>Pasos de Ejecución:</p> <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción “Calendario”. - Acceder a una tarea colaborativa - Acceder al vínculo “Modificar” en la parte superior derecha de la tarea. - Introducir los datos correctamente. 	

- Acceder al vínculo "Listo".
Resultados Esperados: El sistema modifica la notificación en el sistema.

Tabla 3.30 Caso de Prueba: HU19_P1

Caso de Prueba de Aceptación	
Código: HU20_P1	Historia de Usuario: 20
Nombre: Eliminar Notificación	
Descripción: Prueba para comprobar que el sistema permite Eliminar Notificación.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario debe tener notificaciones activas.	
Pasos de Ejecución: - Acceder al vínculo donde se encuentra la imagen que representa las notificaciones en el parte superior derecha del menú de navegación. Esta acción despliega una lista de notificaciones. - Acceder al vínculo de la información de la notificación.	
Resultados Esperados: El sistema eliminar la notificación en el sistema.	

Tabla 3.31 Caso de Prueba: HU20_P1

Caso de Prueba de Aceptación	
Código: HU21_P1	Historia de Usuario: 21
Nombre: Mostrar Notificación	
Descripción: Prueba para comprobar que el sistema permite Mostrar Notificación.	
Condiciones de Ejecución: El usuario debe estar previamente registrado. El usuario tiene que estar autenticado. El usuario debe tener notificaciones.	
Pasos de Ejecución: - Acceder al vínculo donde se encuentra la imagen que representa las notificaciones en el parte superior derecha del menú de navegación. Esta acción despliega una lista de notificaciones.	
Resultados Esperados: El sistema muestra las notificaciones para el usuario autenticado.	

Tabla 3.32 Caso de Prueba: HU21_P1

Iteración 3

Caso de Prueba de Aceptación

Código: HU22_P1	Historia de Usuario: 22
Nombre: Insertar Calendario	
Descripción: Prueba para comprobar que el sistema permite Insertar Calendario.	
Condiciones de Ejecución: El usuario debe estar registrado.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al botón "Entrar" en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página "Acceso". - Acceder en el vínculo "Crear una cuenta". - Introducir los datos correctamente. - Seleccionar la opción "Publico" en Tipo de Calendario. - Acceder al botón "Listo". 	
Resultados Esperados: El sistema inserta un calendario en el sistema.	

Tabla 3.33 Caso de Prueba: HU22_P1

Caso de Prueba de Aceptación	
Código: HU23_P1	Historia de Usuario: 23
Nombre: Modificar Calendario	
Descripción: Prueba para comprobar que el sistema permite Modificar Calendario.	
Condiciones de Ejecución: El usuario debe estar registrado. El usuario tiene que estar autenticado en el sistema.	
Pasos de Ejecución: <ul style="list-style-type: none"> - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Perfil". - Introducir datos correctamente. - Acceder al botón "Listo" 	
Resultados Esperados: El sistema modifica un calendario en el sistema.	

Tabla 3.34 Caso de Prueba: HU23_P1

Caso de Prueba de Aceptación	
Código: HU24_P1	Historia de Usuario: 24
Nombre: Eliminar Calendario	
Descripción: Prueba para comprobar que el sistema permite Eliminar Calendario.	

Condiciones de Ejecución: El usuario debe estar registrado. El usuario tiene que estar autenticado en el sistema. El usuario debe tener permisos administrativos.
Pasos de Ejecución: - Acceder al botón "Entrar" en el parte superior derecha del menú de navegación. Esta acción redirecciona a la página "Acceso". - Seleccionar la opción "Entrar como administrador" - Introducir datos correctamente. - Ir al menú de navegación y seleccionar la opción "Persona" - Seleccionar el Calendario a eliminar. - Acceder al botón "Eliminar"
Resultados Esperados: El sistema eliminar un calendario del sistema.

Tabla 3.35 Caso de Prueba: HU24_P1

Caso de Prueba de Aceptación	
Código: HU25_P1	Historia de Usuario: 25
Nombre: Mostrar Calendario	
Descripción: Prueba para comprobar que el sistema permite Mostrar Calendario.	
Condiciones de Ejecución: El usuario debe estar registrado. El usuario debe estar autenticado.	
Pasos de Ejecución: - Acceder al vínculo donde se encuentra la imagen de la persona en el parte superior derecha del menú de navegación. Esta acción despliega un menú de navegación. - Seleccionar la opción "Calendario".	
Resultados Esperados: El sistema muestra el calendario en el sistema.	

Tabla 3.36 Caso de Prueba: HU25_P1

Se realizaron 3 iteraciones de pruebas donde se detectaron varias no conformidades (NC) que fueron mitigadas durante el transcurso de la presente etapa. En la primera iteración de pruebas se detectaron un total de 7 NC, donde la mayoría eran por problemas de validación. Posteriormente en la segunda iteración se validó la corrección de los errores encontrados en la primera, aunque se detectaron 4 nuevas NC y por último en la tercera iteración no se encontraron NC, constituyendo esto un 100% de pruebas de aceptación exitosas, dicha iteración se realizó de manera conjunta con el cliente. En gráfica se establece una comparación de los resultados obtenidos por iteraciones.

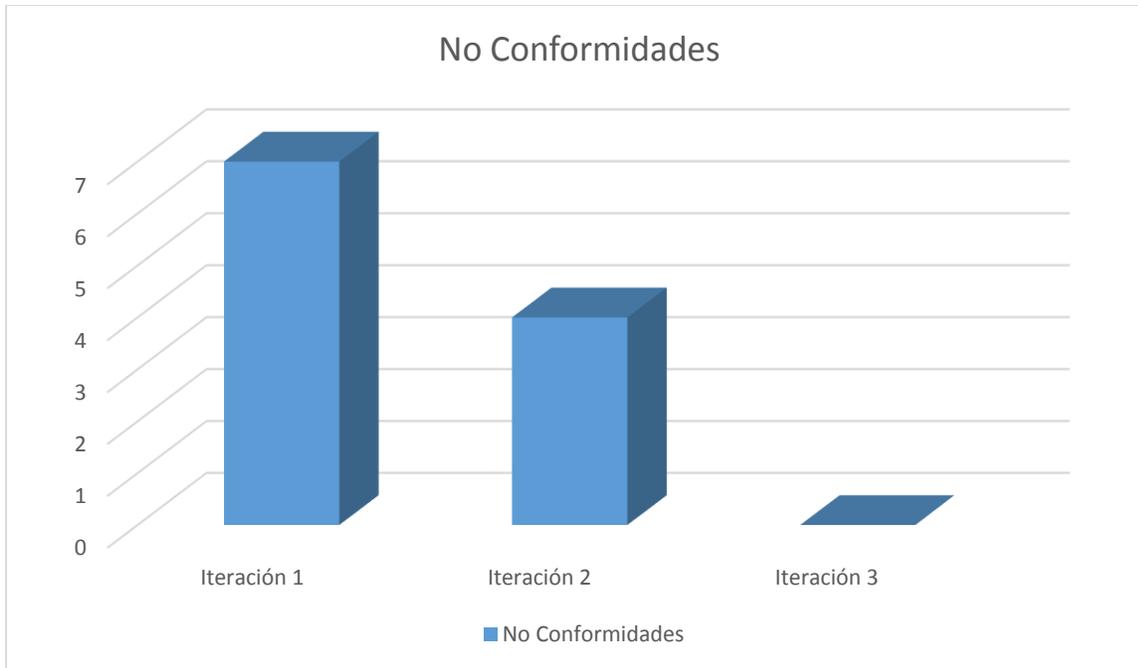


Figura 5. Gráfica de resultados por iteraciones.

3.5 Conclusiones del Capítulo

En el presente capítulo se facilitó la implementación de la propuesta de solución como resultado del estudio bibliográfico y de las soluciones similares, que permitió la selección de las herramientas, tecnologías y lenguajes de programación adecuados. Se realizó la descripción de las pruebas de aceptación a las que fue sometida la aplicación con la intención de verificar el funcionamiento de la misma, probando individualmente la implementación de cada historia de usuario en su correspondiente iteración.

Conclusiones

La evolución de la presente investigación permitió la obtención de un sistema para la planificación de las actividades curriculares y extracurriculares. Esta cumple con todos los aspectos funcionales básicos para su desarrollo, cumpliendo así con el objetivo trazado.

- La fundamentación teórica realizada en la investigación permitió justificar la selección de la metodología de desarrollo, los patrones de diseño, de arquitectura y las herramientas a utilizar para el desarrollo del sistema.
- El sistema para la planificación de las actividades proporcionó la posibilidad de gestionar actividades, gestionar calendarios públicos y privados, gestionar grupos de usuario y administrar las notificaciones.
- Se comprobó la efectividad de la solución propuesta a partir de los resultados satisfactorios obtenidos en las pruebas realizadas.

Recomendaciones

Como parte del proceso de investigación y desarrollo de la aplicación, surgieron ideas que son recomendables tener en cuenta para el futuro mejoramiento de la solución propuesta. Entre ellas se señalan:

1. Agregar como funcionalidad la posibilidad de adjuntar ficheros en la creación de actividades colaborativas.
2. Incluir un módulo que incluya como principales funcionalidades la exportación de pdf y la impresión de la información de las actividades durante un periodo de tiempo.

Bibliografía

- Aiteco. 2012.** Aiteco. *Tormenta de ideas*. [En línea] 8 de Marzo de 2012. <http://aiteco.com>.
- Allende, R. 2006.** *Desarrollo de Portales y Extranet con Plone*. 2006.
- Alvarez, M. A. 2009.** Desarrolloweb. [En línea] 2009.
- Angel, M. 2010.** Características y ventajas de las CSS. [En línea] 2010. [.http://www.desarrolloweb.com/articulos/182.php](http://www.desarrolloweb.com/articulos/182.php).
- Argota, I. 2012.** Vegamatrix. *Navegando en la dinámica Matrix del conocimiento*. [En línea] 17 de Abril de 2012. <http://vegamatrix.blogspot.com/2012/04/metodologia-agil-o-formal-en-tu-empresa.html>.
- Bakulin, P.I. 1992.** *Curso de Astronomía General*. Madrid : Editorial Rubiños, 1992.
- Barkley, E., Cross, K.P., Major, C.H. 2012.** *Collaborative learning techniques: A Handbook for college faculty*. San Francisco : Jossey-Bass, 2012.
- Beck, K. 1999.** *Extreme Programming Explained: Embrace Change*. s.l. : AddisonWesley Pub Co, 1999.
- Bibeault, Bear, Yehuda. 2013.** *JQuery in Action*. Greenwich : Manning Publications Co., 2013.
- Bravo, A. P. 2007.** *Cooperacion Tecnica Internacional*. 2007.
- Bryan, S. 2015.** Google. *Visita guiada de Google Calendar*. [En línea] 2015. <http://www.google.com/intl/es/googlecalendar/tour2.html>.
- Chaffer, Jonathan, Swedberg, Karl. 2010.** *APRENDE JQUERY 1. 3*. 2010.
- Ciberaula. 2010.** Ciberaula. *Introducción, definición y evolución de PHP*. [En línea] 25 de Febrero de 2010. http://php.ciberaula.com/articulo/introduccion_php/.
- CodeIgniter. 2010.** CodeIgniter Web Framework. [En línea] 2010. <http://www.codeigniter.com>.
- Escribano. 2002.** eXtreme Programming. [En línea] 2002. <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
- Fedderen-Beckan, T. 2008.** *Google calendar. Journal of the Medical Library Association*. s.l. : JMLA, 2008.
- Frain, B. 2012.** *Responsive Web Design with HTML5 and CSS3*. UK : Packt Publishing Ltd, 2012.
- Harris, A. 2010.** *HTML, XHTML and CSS All-In-One For Dummies*. s.l. : John Wiley & Sons, 2010.
- Issi, Lázaro. 2012.** *JavaScript*. Madrid : ANAYA MULTIMEDIA, 2012. 84-415-1384-8.
- Jacobson. 2000.** *El Proceso Unificado de Desarrollo de software*. EE.UU : s.n., 2000.
- Jeffries, R., Anderson, A., Hendrickson, C. 2001.** *Extreme Programming Installed*. s.l. : Addison-Wesley, 2001.
- Johnson, P. , Johnson, F. P. 1997.** *Joining Together: Group Theory and Group Skills*. Needham Heights, MA : Allyn & Bacon, 1997.

- Larman, C. 1999.** *UML y Patrones*. s.l. : Pearson, 1999.
- Marcías, R., Vázquez, R. 2007.** *Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP*. s.l. : Trabajo de Diploma para optar por el título de Ingeniero en ciencias informáticas, 2007.
- Mueller, Erik T. 2000.** *A calendar with common sense*. En *Proceedings of the 5th international conference on Intelligent user interfaces*. s.l. : ACM, 2000.
- Murdick, R. C. 1994.** *Sistemas de Información Administrativa*. México : Prentice-Hall Hispanoamericana, 1994.
- Negrino, Tom y Smith, Dori. 2012.** *Guía de aprendizaje JavaScript*. 3. Madrid : PEARSON EDUCACIÓN, S.A., 2012.
- Peñalver, R., Marsi, G. 2009.** *Metodología ágil para proyectos de software libre*. Ciudad de la habana : s.n., 2009.
- Pérez, J. E. 2009.** Librosweb. *CSS avanzado*. [En línea] 2009. http://www.librosweb.es/css_avanzado/.
- Peterson, C. 2014.** *Learning Responsive Web Design*. Canada : O'Reilly Media, Inc, 2014.
- Pressman, Roger S. 2005.** *Ingeniería del Software: Un Enfoque Práctico*. 2005.
- RAE. 2010.** *Diccionario RAE*. s.l. : RAE, 2010.
- Reinhold, K. 2014.** Manual de korganize. [En línea] 2014. <http://docs.kde.org/stable/es/kdepim/korganizer/index.html>.
- Rodríguez, M., Rodríguez, J. 2012.** *Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP*. 2012.
- Romer, O y Yibetza, L. 2010.** Monografías. [En línea] 9 de Marzo de 2010. <http://www.monografias.com/trabajos53/tecnologiacomunicacion/tecnologia.html>.
- Rozic, S. E. 2004.** *Bases de datos y su aplicación con SQL*. Buenos Aires : MP Ediciones SA, 2004.
- Rumbaugh, James, Jacobson. 2000.** *El Lenguaje Unificado de Modelado. Manual de referencia*. Madrid : s.n., 2000.
- Saavedra, J. 2013.** El mundo informático. [En línea] 8 de Mayo de 2013. <http://jorgesaavedra.wordpress.com/category/patrones-grasp/...>
- Salinas, J. 2000.** *El aprendizaje colaborativo con los nuevos canales de comunicación*. Madrid : s.n., 2000.
- Sommerville, I. 2005.** *Ingeniería de Software*. 2005.
- Stellman, A. , Greene, J. 2006.** *Applied Software Project Management*. s.l. : O'Reilly Media, 2006. 0596-00948-8.
- Tobías, M. 2008.** Lainformacion.com. [En línea] 8 de diciembre de 2008. <http://tobiassig08.bitacoras.com/archivos/2008/12/08/software-empresarial-y-nuevas-tecnologias-erp2>.

Toledo, Á. 2010. UPtoDOWN. [En línea]. [En línea] 2010. <http://postgresql.uptodown.com>.

Torres, B. A. 2005. *Creación de un espacio virtual para el intercambio de lenguaje científico*. 2005.

Upton, D. 2007. *CodeIgniter for Rapid PHP Application Development*. s.l. : Packt Publishing Ltd, 2007.

Vera, Fernando. 2006. [En línea] 2006. <http://trabajosfernandovera.blogspot.com/2006/07/gestin-escolarprecisando-el-concepto.html>.

Wiederhold, Gio. 1977. *Desing Database*. Nueva York : s.n., 1977.

Trabajos Citados

- Aiteco. 2012.** Aiteco. *Tormenta de ideas*. [En línea] 8 de Marzo de 2012. <http://aiteco.com>.
- Allende, R. 2006.** *Desarrollo de Portales y Extranet con Plone*. 2006.
- Alvarez, M. A. 2009.** Desarrolloweb. [En línea] 2009.
- Angel, M. 2010.** Características y ventajas de las CSS. [En línea] 2010. [.http://www.desarrolloweb.com/articulos/182.php](http://www.desarrolloweb.com/articulos/182.php).
- Argota, I. 2012.** Vegamatrix. *Navegando en la dinámica Matrix del conocimiento*. [En línea] 17 de Abril de 2012. <http://vegamatrix.blogspot.com/2012/04/metodologia-agil-o-formal-en-tu-empresa.html>.
- Bakulin, P.I. 1992.** *Curso de Astronomía General*. Madrid : Editorial Rubiños, 1992.
- Barkley, E., Cross, K.P., Major, C.H. 2012.** *Collaborative learning techniques: A Handbook for college faculty*. San Francisco : Jossey-Bass, 2012.
- Beck, K. 1999.** *Extreme Programming Explained: Embrace Change*. s.l. : AddisonWesley Pub Co, 1999.
- Bibeault, Bear, Yehuda. 2013.** *JQuery in Action*. Greenwich : Manning Publications Co., 2013.
- Bravo, A. P. 2007.** *Cooperacion Tecnica Internacional*. 2007.
- Bryan, S. 2015.** Google. *Visita guiada de Google Calendar*. [En línea] 2015. <http://www.google.com/intl/es/googlecalendar/tour2.html>.
- Chaffer, Jonathan, Swedberg, Karl. 2010.** *APRENDE JQUERY 1. 3*. 2010.
- Ciberaula. 2010.** Ciberaula. *Introducción, definición y evolución de PHP*. [En línea] 25 de Febrero de 2010. http://php.ciberaula.com/articulo/introduccion_php/.
- CodeIgniter. 2010.** CodeIgniter Web Framework. [En línea] 2010. <http://www.codeigniter.com>.
- Escribano. 2002.** eXtreme Programming. [En línea] 2002. <http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>.
- Fedderen-Beckan, T. 2008.** *Google calendar. Journal of the Medical Library Association*. s.l. : JMLA, 2008.
- Frain, B. 2012.** *Responsive Web Design with HTML5 and CSS3*. UK : Packt Publishing Ltd, 2012.
- Harris, A. 2010.** *HTML, XHTML and CSS All-In-One For Dummies*. s.l. : John Wiley & Sons, 2010.
- Issi, Lázaro. 2012.** *JavaScript*. Madrid : ANAYA MULTIMEDIA, 2012. 84-415-1384-8.
- Jacobson. 2000.** *El Proceso Unificado de Desarrollo de software*. EE.UU : s.n., 2000.
- Jeffries, R., Anderson, A., Hendrickson, C. 2001.** *Extreme Programming Installed*. s.l. : Addison-Wesley, 2001.
- Johnson, P. , Johnson, F. P. 1997.** *Joining Together: Group Theory and Group Skills*. Needham Heights, MA : Allyn & Bacon, 1997.

- Larman, C. 1999.** *UML y Patrones*. s.l. : Pearson, 1999.
- Marcías, R., Vázquez, R. 2007.** *Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP*. s.l. : Trabajo de Diploma para optar por el título de Ingeniero en ciencias informáticas, 2007.
- Mueller, Erik T. 2000.** *A calendar with common sense*. En *Proceedings of the 5th international conference on Intelligent user interfaces*. s.l. : ACM, 2000.
- Murdick, R. C. 1994.** *Sistemas de Información Administrativa*. México : Prentice-Hall Hispanoamericana, 1994.
- Negrino, Tom y Smith, Dori. 2012.** *Guía de aprendizaje JavaScript*. 3. Madrid : PEARSON EDUCACIÓN, S.A., 2012.
- Peñalver, R., Marsi, G. 2009.** *Metodología ágil para proyectos de software libre*. Ciudad de la habana : s.n., 2009.
- Pérez, J. E. 2009.** Librosweb. *CSS avanzado*. [En línea] 2009. http://www.librosweb.es/css_avanzado/.
- Peterson, C. 2014.** *Learning Responsive Web Design*. Canada : O'Reilly Media, Inc, 2014.
- Pressman, Roger S. 2005.** *Ingeniería del Software: Un Enfoque Práctico*. 2005.
- RAE. 2010.** *Diccionario RAE*. s.l. : RAE, 2010.
- Reinhold, K. 2014.** Manual de korganize. [En línea] 2014. <http://docs.kde.org/stable/es/kdepim/korganizer/index.html>.
- Rodríguez, M., Rodríguez, J. 2012.** *Estudio de Sistemas de Gestión de Contenidos basados en lenguaje PHP*. 2012.
- Romer, O y Yibetza, L. 2010.** Monografías. [En línea] 9 de Marzo de 2010. <http://www.monografias.com/trabajos53/tecnologiacomunicacion/tecnologia.html>.
- Rozic, S. E. 2004.** *Bases de datos y su aplicación con SQL*. Buenos Aires : MP Ediciones SA, 2004.
- Rumbaugh, James, Jacobson. 2000.** *El Lenguaje Unificado de Modelado. Manual de referencia*. Madrid : s.n., 2000.
- Saavedra, J. 2013.** El mundo informático. [En línea] 8 de Mayo de 2013. <http://jorgesaavedra.wordpress.com/category/patrones-grasp/...>
- Salinas, J. 2000.** *El aprendizaje colaborativo con los nuevos canales de comunicación*. Madrid : s.n., 2000.
- Sommerville, I. 2005.** *Ingeniería de Software*. 2005.
- Stellman, A. , Greene, J. 2006.** *Applied Software Project Management*. s.l. : O'Reilly Media, 2006. 0596-00948-8.
- Tobías, M. 2008.** Lainformacion.com. [En línea] 8 de diciembre de 2008. <http://tobiassig08.bitacoras.com/archivos/2008/12/08/software-empresarial-y-nuevas-tecnologias-erp2>.

Toledo, Á. 2010. UPtoDOWN. [En línea]. [En línea] 2010. <http://postgresql.uptodown.com>.

Torres, B. A. 2005. *Creación de un espacio virtual para el intercambio de lenguaje científico*. 2005.

Upton, D. 2007. *CodeIgniter for Rapid PHP Application Development*. s.l. : Packt Publishing Ltd, 2007.

Vera, Fernando. 2006. [En línea] 2006. <http://trabajosfernandovera.blogspot.com/2006/07/gestin-escolarprecisando-el-concepto.html>.

Wiederhold, Gio. 1977. *Desing Database*. Nueva York : s.n., 1977.

Anexos

Anexo 1: Historias de usuario

Historia de Usuario	
Número: 1	Nombre: Autenticar persona
Usuario: Administrador, Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El usuario introduce sus datos en el formulario, el sistema verifica que los datos entrados sean correctos. Datos requeridos: <ul style="list-style-type: none">• Usuario• Contraseña	
Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje de error en la autenticación.	

Anexo 1.1 HU Autenticar persona.

Historia de Usuario	
Número: 2	Nombre: Insertar persona
Usuario: Administrador, Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir insertar una persona, siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none">• Identificador Usuario• Nombre• Apellidos• Foto• Contraseña• Correo	

Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.

Anexo 1.2 HU Insertar persona.

Historia de Usuario	
Número: 3	Nombre: Modificar persona
Usuario: Administrador, Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir modificar una persona, siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none">• Nombre• Apellidos• Foto• Contraseña• Correo	
Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.	

Anexo 1.3 HU Modificar persona.

Historia de Usuario	
Número: 4	Nombre: Eliminar persona
Usuario: Administrador, Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir eliminar una persona	
Observaciones:	

Anexo 1.4 HU Eliminar persona.

Historia de Usuario	
Número: 5	Nombre: Mostrar persona
Usuario: Administrador, Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar los datos de una persona <ul style="list-style-type: none"> • Identificador Usuario • Nombre • Apellidos • Foto • Correo 	
Observaciones:	

Anexo 1.5 HU Mostrar persona.

Historia de Usuario	
Número: 6	Nombre: Insertar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir insertar una tarea verificando que la tarea cumpla con los datos establecidos, siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea • Notas • Lugar 	
Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.	

Anexo 1.6 HU Insertar tarea.

Historia de Usuario	
Número: 7	Nombre: Modificar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir modificar una tarea ya existente, verificando que la tarea cumpla con los datos establecidos para la modificación, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea • Notas • Lugar 	
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>	

Anexo 1.7 HU Modificar tarea.

Historia de Usuario	
Número: 8	Nombre: Eliminar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir eliminar una tarea.</p>	
<p>Observaciones:</p>	

Anexo 1.8 HU Eliminar tarea.

Historia de Usuario	
Número: 9	Nombre: Mostrar tarea
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar la información de una tarea. <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea • Notas • Lugar 	
Observaciones:	

Anexo 1.9 HU Mostrar tarea.

Historia de Usuario	
Número: 10	Nombre: Insertar actividad colaborativa
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El sistema debe permitir insertar una actividad colaborativa. Además debe permitir enviar una notificación de la actividad creada a los grupos de personas que se les fue asignada esta actividad, siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la actividad colaborativa • Notas • Grupos 	

<ul style="list-style-type: none"> • Lugar
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>

Anexo 1.10 HU Insertar actividad colaborativa.

Historia de Usuario	
Número: 11	Nombre: Modificar actividad colaborativa
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Alta
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir modificar una actividad colaborativa. Además debe permitir enviar una notificación de la actividad modificada a los grupos de personas que se les fue asignada esta actividad, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la actividad colaborativa • Notas • Grupos • Lugar 	
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>	

Anexo 1.11 HU Modificar actividad colaborativa.

Historia de Usuario	
Número: 12	Nombre: Eliminar actividad colaborativa
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1

Descripción: El sistema debe permitir eliminar una actividad colaborativa.
Observaciones:

Anexo 1.12 HU Eliminar actividad colaborativa.

Historia de Usuario	
Número: 13	Nombre: Mostrar actividad colaborativa
Usuario: Usuario	
Prioridad: Alta	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar los datos una actividad colaborativa. Además debe permitir mostrar los grupos asignados a esta actividad <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la actividad colaborativa • Notas • Grupos • Lugar 	
Observaciones:	

Anexo 1.13 HU Mostrar actividad colaborativa.

Historia de Usuario	
Número: 14	Nombre: Insertar grupo
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
Descripción:	

<p>El sistema debe permitir insertar un grupo, además debe permitir enviar una notificación del grupo creado a las posibles personas que pueden pertenecer a dicho grupo, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Propietario • Nombre del Grupo • Miembros del Grupo • Foto
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>

Anexo 1.14 HU Insertar grupo.

Historia de Usuario	
Número: 15	Nombre: Modificar grupo
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1
<p>Descripción: El sistema debe permitir modificar un grupo, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Propietario • Nombre del Grupo • Miembros del Grupo • Foto 	
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>	

Anexo 1.15 HU Modificar grupo.

Historia de Usuario	
Número: 16	Nombre: Eliminar grupo
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Media
Puntos estimados: 1	Iteración asignada: 1

Descripción: El sistema debe permitir eliminar un grupo
Observaciones:

Anexo 1.16 HU Eliminar grupo.

Historia de Usuario	
Número: 17	Nombre: Mostrar grupo
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Media
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar un grupo. <ul style="list-style-type: none"> • Propietario • Nombre del Grupo • Miembros del Grupo • Foto 	
Observaciones:	

Anexo 1.17 HU Mostrar grupo.

Historia de Usuario	
Número: 18	Nombre: Insertar notificación
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El sistema debe permitir insertar una notificación, , siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none"> • Persona Origen • Persona Destino • Tipo de Notificación 	

<ul style="list-style-type: none"> • Fecha • Hora
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>

Anexo 1.18 HU Insertar notificación.

Historia de Usuario	
Número: 19	Nombre: Modificar notificación
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1
<p>Descripción: El sistema debe permitir modificar una notificación, siempre especificando los datos requeridos por la aplicación:</p> <ul style="list-style-type: none"> • Persona Origen • Persona Destino • Tipo de Notificación • Fecha • Hora 	
<p>Observaciones: Si los datos entrados son incorrectos, están en blanco o no coinciden con lo esperado, la aplicación debe mostrar un mensaje correspondiente a cada error.</p>	

Anexo 1.19 HU Modificar notificación.

Historia de Usuario	
Número: 20	Nombre: Eliminar notificación
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1

Descripción: El sistema debe permitir eliminar una notificación
Observaciones:

Anexo 1.20 HU Eliminar notificación.

Historia de Usuario	
Número: 21	Nombre: Mostrar notificación
Usuario: Usuario	
Prioridad: Media	Nivel de complejidad: Alta
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar una notificación <ul style="list-style-type: none"> • Persona Origen • Persona Destino • Tipo de Notificación • Fecha • Hora 	
Observaciones:	

Anexo 1.21 HU Mostrar notificación.

Historia de Usuario	
Número: 22	Nombre: Insertar calendario
Usuario: Administrador, Usuario	
Prioridad: Baja	Nivel de complejidad: Baja
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir insertar un calendario, siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none"> • Fecha de creación • Tipo de calendario 	

Observaciones: Cada usuario contará con un solo sistema de calendario

Anexo 1.22 HU Insertar calendario.

Historia de Usuario	
Número: 23	Nombre: Modificar calendario
Usuario: Administrador, Usuario	
Prioridad: Baja	Nivel de complejidad: Baja
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir modificar un calendario, siempre especificando los datos requeridos por la aplicación: <ul style="list-style-type: none"> • Fecha de creación • Tipo de calendario 	
Observaciones:	

Anexo 1.23 HU Modificar calendario.

Historia de Usuario	
Número: 24	Nombre: Eliminar calendario
Usuario: Administrador, Usuario	
Prioridad: Baja	Nivel de complejidad: Baja
Puntos estimados: 1	Iteración asignada: 1
Descripción: El sistema debe permitir eliminar un calendario,	
Observaciones:	

Anexo 1.24 HU Eliminar calendario.

Historia de Usuario	
Número: 25	Nombre: Mostrar calendario
Usuario: Administrador, Usuario	
Prioridad: Baja	Nivel de complejidad: Baja
Puntos estimados: 1.5	Iteración asignada: 1
Descripción: El sistema debe permitir mostrar un calendario: <ul style="list-style-type: none">• Fecha de creación• Tipo de calendario	
Observaciones:	

Anexo 1.25 HU Mostrar calendario.

Anexo 2: Tareas de ingeniería

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 1
Nombre tarea: Autenticar persona	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: El sistema debe crear un formulario con los datos de inicio de sesión (usuario y contraseña).	

Anexo 2.1 Tarea 1 Autenticar usuario.

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 1
Nombre tarea: Validar datos	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten la captura de los datos, se verifican que sean correctos con la base de la aplicación. En caso de que sean correctos se redirecciona el usuario a la página correspondiente según su privilegio	

Anexo 2.2 Tarea 2 Validar datos.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 2
Nombre tarea: Insertar persona en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten la captura del dato (Identificador usuario, nombre, apellidos, foto, contraseña y correo) de la persona a insertar. Luego se realiza la validación de los datos, de forma tal que el mismo sea insertado correctamente.	

Anexo 2.3 Tarea 3 Insertar persona en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 3
Nombre tarea: Modificar persona en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de modificar persona en la aplicación, teniendo en cuenta que los parámetros de entrada estén correctos.	

Anexo 2.4 Tarea 4 Modificar persona en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 4
Nombre tarea: Eliminar persona en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de eliminar una persona en la aplicación.	

Anexo 2.5 Tarea 5 Eliminar persona en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 5
Nombre tarea: Mostrar persona en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de mostrar una persona en la aplicación. Los datos incluyen <ul style="list-style-type: none"> • Identificador Usuario • Nombre • Apellidos • Foto • Contraseña • Correo 	

Anexo 2.6 Tarea 6 Mostrar persona en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 6
Nombre tarea: Insertar tarea	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten insertar una tarea dentro del sistema: <ul style="list-style-type: none"> • Fecha • Hora • Nombre de la tarea • Notas • Lugar de la tarea. Luego se realiza la validación de los datos, de forma tal que la tarea quede insertada correctamente.	

Anexo 2.7 Tarea 7 Insertar tarea.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 7
Nombre tarea: Modificar tarea	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de modificar tarea en la aplicación, teniendo en cuenta que los parámetros de entrada estén correctos.	

Anexo 2.8 Tarea 8 Modificar tarea.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 8
Nombre tarea: Eliminar tarea	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de eliminar una tarea en la aplicación	

Anexo 2.9 Tarea 9 Eliminar tarea.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 9
Nombre tarea: Mostrar tarea	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de mostrar una tarea en la aplicación, la información es obtenida de la tabla de tareas, mostrando las tareas relacionadas con un usuario determinado.	

Anexo 2.10 Tarea 10 Mostrar tarea.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 10
Nombre tarea: Insertar actividad colaborativa	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de insertar una actividad colaborativa, la información es insertada en diferentes tablas de la base de datos para asegurar que todas las personas involucradas reciban la información de la tarea.	

Anexo 2.11 Tarea 11 Insertar actividad colaborativa.

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 10
Nombre tarea: Enviar notificación a personas relacionadas con la actividad	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de enviar una notificación a las personas que están relacionadas con la actividad, buscando en las tablas de grupo-persona.	

Anexo 2.12 Tarea 12 Enviar notificación a personas relacionadas con la actividad.

Tareas de Ingeniería

Número de tarea: 1	Número de HU: 11
Nombre tarea: Modificar actividad colaborativa	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: esta funcionalidad será la encargada de modificar una actividad colaborativa.	

Anexo 2.13 Tarea 13 Modificar actividad colaborativa.

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 11
Nombre tarea: Enviar notificación a personas relacionadas con la actividad modificada	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de enviar una notificación a las personas que están relacionadas con la actividad a modificar, buscando en las tablas de persona.	

Anexo 2.14 Tarea 14 Enviar notificación a personas relacionadas con la actividad modificada.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 12
Nombre tarea: Eliminar actividad colaborativa	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: esta funcionalidad será la encargada de eliminar una actividad colaborativa.	

Anexo 2.15 Tarea 15 Eliminar actividad colaborativa.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 13
Nombre tarea: Mostrar actividad colaborativa	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	

Descripción: esta funcionalidad será la encargada de mostrar una actividad colaborativa. Esta actividad aparece en los calendarios de las personas que se les fue compartida.

Anexo 2.16 Tarea 16 Mostrar actividad colaborativa.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 14
Nombre tarea: Insertar grupo en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten la captura del dato (Propietario, Nombre del grupo, Miembros del grupo y foto) del grupo a insertar. Luego se realiza la validación de los datos, de forma tal que el mismo sea insertado correctamente.	

Anexo 2.17 Tarea 17 Insertar grupo en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 15
Nombre tarea: Modificar grupo en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de modificar grupo en la aplicación, teniendo en cuenta que los parámetros de entrada estén correctos. Los datos son capturados y enviados a la base de datos de tal forma que no exista problema en la presentación de los mismos.	

Anexo 2.18 Tarea 18 Modificar grupo en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 16
Nombre tarea: Eliminar grupo en la aplicación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de eliminar un grupo en la aplicación.	

Anexo 2.19 Tarea 19 Eliminar grupo en la aplicación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 17
Nombre tarea: Mostrar grupo	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de mostrar una persona en la aplicación. Los datos incluyen: <ul style="list-style-type: none"> • Propietario • Nombre del Grupo • Miembros del Grupo • Foto 	

Anexo 2.20 Tarea 20 Mostrar grupo.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 18
Nombre tarea: Insertar notificación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten la captura del dato (Persona origen, persona destino, tipo de notificación, fecha y hora) de la notificación a insertar.	

Anexo 2.21 Tarea 21 Insertar notificación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 19
Nombre tarea: Modificar notificación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de modificar notificación en la aplicación, teniendo en cuenta que los parámetros de entrada estén correctos. Los datos son capturados y enviados a la base de datos de tal forma que no exista problema en la presentación de los mismos.	

Anexo 2.22 Tarea 22 Modificar notificación.

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 19
Nombre tarea: Enviar notificación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de enviar una notificación a los usuarios de los grupos	

Anexo 2.23 Tarea 23 Enviar notificación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 20
Nombre tarea: Eliminar notificación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de eliminar una notificación en la aplicación.	

Anexo 2.24 Tarea 24 Eliminar notificación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 21
Nombre tarea: Mostrar notificación	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de mostrar una notificación en la aplicación, la información es obtenida de la tabla de notificación, mostrando las notificaciones relacionadas con un usuario determinado.	

Anexo 2.25 Tarea 25 Mostrar notificación.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 22

Nombre tarea: Insertar calendario
Tipo de tarea: Desarrollo
Programador responsable: Pedro Alejandro Hernández Jiménez
Descripción: Implementar la funcionalidad de insertar calendario en la aplicación.

Anexo 2.26 Tarea 26 Insertar calendario.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 23
Nombre tarea: Modificar calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de insertar calendario en la aplicación, teniendo en cuenta que los parámetros de entrada estén correctos.	

Anexo 2.27 Tarea 27 Modificar calendario.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 24
Nombre tarea: Eliminar calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Implementar la funcionalidad de eliminar calendario en la aplicación.	

Anexo 2.28 Tarea 28 Eliminar calendario.

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 24
Nombre tarea: Eliminar notificaciones del calendario	

Tipo de tarea: Desarrollo
Programador responsable: Pedro Alejandro Hernández Jiménez
Descripción: Se implementan las funcionalidades que permiten eliminar todas las notificaciones del calendario a eliminar.

Anexo 2.29 Tarea 29 Eliminar notificaciones del calendario.

Tareas de Ingeniería	
Número de tarea: 1	Número de HU: 25
Nombre tarea: Mostrar información del calendario	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades que permiten mostrar la información básica de un calendario	

Anexo 2.30 Tarea 30 Mostrar calendario

Tareas de Ingeniería	
Número de tarea: 2	Número de HU: 25
Nombre tarea: Mostrar tareas del calendario (mes)	
Tipo de tarea: Desarrollo	
Programador responsable: Pedro Alejandro Hernández Jiménez	
Descripción: Se implementan las funcionalidades para mostrar todas las tareas del mes de un calendario tanto público como privado.	

Anexo 2.31 Tarea 31 Mostrar tareas del calendario en el mes

Tareas de Ingeniería	
Número de tarea: 3	Número de HU: 25
Nombre tarea: Mostrar tareas del calendario (total)	
Tipo de tarea: Desarrollo	

Programador responsable: Pedro Alejandro Hernández Jiménez

Descripción: Se implementan las funcionalidades para mostrar la cantidad de tareas planificadas en un calendario tanto público como privado.

Anexo 2.32 Tarea 32 Mostrar total de tareas del calendario