

Universidad de las Ciencias

Informáticas



*“Herramienta para el Control
Integrado de Proyectos Informáticos”*

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

AUTORES:

José Pablo Julian Espada

Migdiell Reyes Pereda

TUTORES:

Ing. Denys Buedo Hidalgo

Ing. Ariagnis Yero Guevara

Ciudad de La Habana, 2009



Declaración de la Autoría

Declaro que somos los únicos autores de este trabajo y autorizamos al Proyecto TIP de la Facultad 2 y a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Autores

Migdiell Reyes Pereda

José Pablo Julian Espada

Tutores

Ing. Ariagnis Yero Guevara

Ing. Denys Buedo Hidalgo

Dedicatoria Jose Pablo

A ti, que fuiste faro y guía de este largo camino, que me diste tu mano amiga como ayuda y como bendición un beso. A ti que sufres con mi llanto y ríes mis alegrías, que me has enseñado que la vida es de un duro bregar, que los obstáculos son parte ella, y que cada victoria no es más que el inicio de la próxima contienda. De mis virtudes, hoy, yo tendré la serenidad, tu, puedes tener el orgullo. A ti, porque esté es nuestro sueño, mi querido, mi viejo, mi amigo.

A mi madre, por la fe inquebrantable que ha inculcado en mí, a ti que eres la tierna palabra de aliento, el beso más puro, la sonrisa más sincera, por enseñarme de niño lo que vale el cariño, por arrullarme en tus brazos y regalarme en pedazos uno a uno tu corazón entero, a ti mami!, este sueño.

A “tata”, que siempre está en mi corazón.

A mi tía, Fefi, que en sus abrazos siempre encontramos el calor de una madre.

A mis hermanos: Danielito, que encuentre en esta realidad su propio sueño que perseguir, mi Yayi, que siempre será mi hermanita linda, la que cuando menciono el orgullo se me sale del pecho, y Pablito, que en sus travesuras de niño pregunta, exclama y extraña a quien siempre será, su “Tote”.

A Lianet, por convertirse en mi segunda hermana, por su amor, cariño y apoyo, por brindarme la paz de saber que mi papá estaba en sus brazos.

A toda mi familia, mis vecinos y todos los que simplemente alguna vez han preguntado, “cuando te vas”, “cuando regresas”, o sencillamente, “esta todo bien, necesitas ayuda”.

A mis amigos, porque compartir este sueño con ustedes fue empezar a hacerlo realidad, y porque nunca será largo el camino que vaya hasta nuestro encuentro.

A los que, una que otra noche, soñaron mis anhelos, aquí se los dedico, convertidos en realidad,

Jose.

Dedicatoria Migdiell

A mis padres, a quienes debo cada centímetro de piel, de los que hoy soy fruto; a su sacrificio y abnegación. A ellos que fueron capaces de estar siempre a mi lado aun cuando la distancia era grande y permitieron que hoy mi sueño se haya materializado.

A mi hermano, mi amigo, mi ejemplo, que desde pequeño cargo conmigo mostrándome siempre el mejor de los caminos a seguir. A él, que saco la cara por mí cuando fue necesario.

A mis abuelos, mis tías, tíos y primos que de tanto apoyo han sido. A ellos, que me han inspirado a seguir aun cuando las cosas no van bien.

A mis profesores, sacrificados forjadores de futuro, la mayor de las bibliotecas, con quienes he aprendido no solo lo académico, sino a mirar la vida desde la perspectiva de ser mejor es aprender más.

A mis amigos, esas personas que siempre están cuando las necesitas, los del consejo oportuno, los de la compañía imprescindible.

A todos los que de una forma u otra me ayudaron en algún momento: va dedicado este trabajo de diploma;

Migdiell.

Agradecimientos

A nuestros padres, por permitirnos la vida y guiarnos hasta aquí; por hacernos hoy, hombres de bien.

A nuestros maestros y profesores, que con tanto sacrificio nos han dado un poco más que conocimientos.

A la UCI, por ser a lo largo de cinco años más que una escuela, haber sido nuestra casa, nuestro barrio, nuestro pueblo. Por dotarnos de tantos conocimientos y experiencias inolvidables.

A todos nuestros amigos y hermanos de la vida, que con solo una mirada lograron descubrir nuestros estados de ánimos y con su apoyo nos incitaron a continuar adelante, a los que siempre estarán en nuestros pensamientos.

Y porque esa es la vida, a cada paso un escalón, y con cada escalón la posibilidad de cumplir un sueño, para todos los que han forjado parte de este sueño que hoy hacemos realidad, con toda la gratitud que puedan abrazar dos corazones agradecidos,

Jose y Migdiell.

Frase

**“La sabiduría suprema es tener sueños bastante grandes
para no perderlos de vista mientras se persiguen”**

Napoleón Bonaparte



Índice

Declaración de la Autoría.....	ii
Dedicatoria Jose Pablo	iii
Dedicatoria Migdiell.....	iv
Agradecimientos	v
Frase	vi
Índice.....	vii
Resumen	xii
Introducción	- 1 -
Capítulo I. Fundamentación Teórica.	- 4 -
Introducción.....	- 4 -
1.1 Gestión de Proyectos Informáticos	- 4 -
1.1.1 Funcionalidades Para la Gestión de Proyectos Informáticos.	- 4 -
Planificación	- 4 -
Gestión Documental	- 4 -
Control de Versiones	- 5 -
Gestión de Incidencias.....	- 5 -
1.2 Herramientas para la Gestión de Proyectos.....	- 5 -
1.2.1 Trac.....	- 6 -
1.2.2 DotProject.....	- 6 -
1.2.3 WR TimeTracker.....	- 8 -
1.3 Técnicas, Metodologías y Herramientas en las que se Apoya la Solución.....	- 8 -
1.3.1 Arquitectura de Software.....	- 8 -



1.3.1.1 Arquitectura Cliente/Servidor	- 8 -
Características de la arquitectura Cliente/Servidor	- 9 -
Patrón de Arquitectura Modelo Vista Controlador	- 10 -
1.3.2 Servidor Web	- 11 -
Servidor Web Apache:	- 12 -
1.3.3 Lenguajes de Programación para la Web.	- 12 -
Lenguaje de Programación PHP	- 13 -
Lenguaje de Programación Java Script	- 14 -
1.4 Sistemas de Gestión de Bases de Datos.....	- 15 -
1.4.1 Sistema Gestor de Bases de Datos MySQL.....	- 15 -
1.5 Editor de Código PHP Zend Studio.	- 16 -
1.6 Metodologías de Desarrollo.....	- 18 -
1.6.1 Metodología de Desarrollo XP.	- 18 -
Conclusiones:.....	- 20 -
Capítulo II. Características del sistema.	- 22 -
Introducción.....	- 22 -
2.1 Flujo de los procesos del negocio.....	- 22 -
2.1.1 Reporte de Tareas y Tiempos.....	- 23 -
2.1.2 Control y reporte de errores	- 23 -
2.1.3 Estado del proyecto. Cronograma.....	- 23 -
2.1.4 Gestión, Control y Distribución de Recursos.	- 24 -
2.2 Análisis Crítico del Funcionamiento de los Sistemas Actuales.....	- 24 -
2.3 Características de la Propuesta.....	- 25 -



2.3.1 Descripción del Módulo: Reporte de Tareas y Tiempos.	25 -
2.3.2 Descripción del Módulo: Reporte y Control de Errores.....	26 -
2.3.3 Descripción del Módulo: Estado y Cronograma del Proyecto.	26 -
2.3.4 Descripción del Módulo: Distribución y Control de Recursos en el Proyecto. .-	26 -
2.4 Análisis comparativo de otras soluciones existentes con la propuesta.	27 -
Conclusiones.....	27 -
Capítulo III. Exploración, Planificación y Diseño.	28 -
Introducción.....	28 -
3.1 Historias de Usuario	28 -
3.2 Planificación.	32 -
3.2.1. Estimación de Esfuerzo por Historias de Usuario.....	32 -
3.2.2 Plan de Iteraciones.	33 -
3.2.2.1 Iteración 1	33 -
3.2.2.2 Iteración 2.....	33 -
3.2.2.3 Iteración 3.....	34 -
3.2.3 Plan de duración de las Iteraciones	34 -
3.2.4 Plan de Entregas	34 -
3.3 Diseño.....	35 -
3.3.1 Tarjeta CRC.....	35 -
3.3.2 Diseño de base de datos.....	39 -
Conclusiones.....	39 -
Capítulo IV. Implementación y Prueba.....	40 -



Introducción.....	- 40 -
4.1 Iteración 1.	- 40 -
4.2. Iteración 2.	- 46 -
4.3 Iteración 3.	- 49 -
4.4 Tareas Generales de la implementación.	- 55 -
4.5 Módulos y Librerías Utilizadas en la Implementación.	- 57 -
4.5.2 Módulo Jpgraph.	- 58 -
4.5.3 LDAP.	- 59 -
4.6 Pruebas.....	- 60 -
4.6.1 Pruebas Unitarias.	- 60 -
4.6.2 Pruebas de Aceptación.	- 60 -
Conclusiones.....	- 62 -
Capitulo V. Estudio de Factibilidad.....	- 63 -
Introducción.....	- 63 -
5.1 Modelo matemático COCOMO II.....	- 63 -
5.2 Características del Proyecto.....	- 64 -
5.2.1 Entradas externas.....	- 64 -
5.2.2 Salidas Externas.....	- 65 -
5.2.3 Consultas Externas.....	- 65 -
5.2.4 Archivos Lógicos Internos.....	- 66 -
5.2.4 Archivos de Interfaz Externos.....	- 66 -
5.2.5 Puntos de Función Desajustados.....	- 67 -



5.3 Cálculo de Instrucciones Fuentes, Esfuerzo, Tiempo de Desarrollo, Cantidad de Hombres y Costo.....	- 67 -
5.3.1 Cálculo de Instrucciones Fuentes.	- 67 -
5.4 Cálculo de Esfuerzo Nominal	- 68 -
5.4.1 Ajuste del Esfuerzo Nominal	- 70 -
5.4.1.1 Calculo del Tiempo de Desarrollo del Software.....	- 71 -
5.4.2 Cálculo del Costo Total del Proyecto	- 71 -
5.5 Resultados.....	- 72 -
Conclusiones.....	- 73 -
Conclusiones	- 74 -
Recomendaciones	- 75 -
Bibliografía:.....	- 76 -
Anexos.....	- 77 -
Anexo 1 Pruebas de Aceptación	- 77 -
Anexo 2 Diseño de Base de Datos.....	- 95 -

Resumen

En el presente trabajo de diploma se documenta la implementación de una herramienta para el Control Integrado de Proyectos (CIPI). Dicha herramienta se propone como un sistema de gestión de proyectos informáticos. Surge a partir de la necesidad existente en el proyecto Tele Identificador Personal (TIP) de una aplicación que agrupe las funcionalidades necesarias para la gestión y reporte de tareas y tiempos, el control y reporte de errores, la creación de un cronograma del proyecto y facilite además la asignación y gestión de los recursos del proyecto. Aunque existen aplicaciones con funcionalidades similares a las expuestas, es muy difícil encontrar alguna que agrupe todas las necesidades del proyecto en una sola, de forma que TIP para realizar las actividades de gestión de proyectos utiliza tres herramientas distintas, Trac, WR Time Tracker y DotProject, mientras que la asignación y gestión de recursos del proyecto se hace de forma manual.

Se realizó un estudio que permitió la posterior selección de la metodología y herramientas a utilizar en la confección de la aplicación.

La aplicación CIPI en su primera versión permite agrupar las funcionalidades mínimas necesarias para la gestión de proyectos en un solo sistema, contar con una base de datos centralizada de los registros de las actividades llevadas a cabo por el proyecto TIP, así como de los integrantes del equipo de desarrollo y el rol que juegan dentro del proyecto.

Introducción

El avance de las Tecnologías de la Informática y las Comunicaciones (TIC) ha propiciado la aparición de herramientas informáticas que facilitan la dirección de proyectos debido a que simplifican en gran medida las actividades de gestión, planificación, control y seguimiento de proyectos. Un grupo de funcionalidades específicas son agrupadas en aplicaciones, ya sean de tipo web¹ o desktop², dando la posibilidad tanto a líderes de proyectos como al equipo de desarrolladores³, de llevar a cabo un trabajo fácilmente organizado; haciendo de la planificación, revisión y el control de los recursos utilizados en el proyecto los elementos más importantes durante el desarrollo del mismo.

La construcción de software⁴ está considerada uno de los problemas técnicos cruciales de nuestros tiempos; desde la aparición del software a mediados del siglo XX, este ha ido ganando complejidad. En un período relativamente corto de tiempo el software pasó de ser un elemento secundario, a constituir un eslabón fundamental en la vida económica, social y política de nuestro planeta. El mundo del software pasó de pequeños programitas de cómputo realizados por unos pocos programadores y en ocasiones por uno solo, a gigantescas herramientas capaces de controlar todos los procesos de una empresa y que para su desarrollo exigen enormes y multidisciplinarios equipos de trabajo.

Los nuevos equipos de trabajo, con grandes cantidades de personas involucradas en el desarrollo de un mismo software, trajeron consigo la necesidad de lograr y mantener una organización y control sobre los recursos materiales y/o humanos, sobre los objetivos que debían proponerse para cada integrante del equipo de desarrollo y las tareas específicas que lo guiarían hasta él. El poder manejar en cada momento de la etapa de desarrollo el estado del proyecto, redefinir tareas y responsabilidades, examinar las ya logradas y resaltar errores cometidos, obtener un cronograma eficiente, revisar su cumplimiento y comunicar los cambios, forman parte esencial en la obtención de una mayor calidad en el producto final que se desea, lo cual conduce a la utilización de herramientas que faciliten el proceso.

¹ Nombre que se le da a las aplicaciones que son utilizadas a través de internet, estas requieren de una red donde exista al menos una computadora haciendo función de servidor de aplicaciones y otras de cliente.

² Hace alusión a los programas que se ejecutan en una computadora, sin necesidad de una red de computadoras para su ejecución.

³ Conjunto de programadores, encargados de llevar a cabo la realización de proyectos.

⁴ Programa de computación. Traducido literalmente como *parte blanda* de la computadora.

Actualmente existe gran cantidad y variedad de herramientas con estas características. Se pueden citar algunos ejemplos como es el caso del DotProject, el Trac y el TimeTracker. El DotProject fue creado en el año 2000 por dotmarketing.org, con el fin de construir una herramienta para la gestión de proyecto, capaz de controlar el estado del proyecto y generar un cronograma del mismo. El Trac es un sistema web inspirado en CVSTrac, desarrollado y mantenido por Edgwall Software que se utiliza para el seguimiento de los errores, mientras que el TimeTracker permite la gestión de tareas y tiempos. Los ejemplos antes mencionados constituyen algunas de las herramientas más utilizadas en el desarrollo de productos informáticos en la UCI, y en especial en la Facultad 2, pero ninguna de estas cumplen por si sola todos los requisitos para el control total de las tareas y recursos de un grupo de trabajo. Se necesita entonces de la integración de todas las funcionalidades en una misma aplicación, lo que permitirá agilizar el proceso y poder generar y obtener la mayor información relacionada con el proyecto.

A partir de la situación anteriormente expresada surge el siguiente problema: ¿cómo lograr la integración de las funcionalidades necesarias para la gestión de proyectos informáticos?

El **objeto de estudio** se enmarca en la gestión de las actividades, recursos y personal involucrado en proyectos productivos. Delimitando como **campo de acción** la gestión de las actividades, recursos y personal involucrado en el proyecto TIP⁵ en la Facultad 2 de la Universidad de las Ciencias Informáticas (UCI).

Con el propósito de solucionar el problema planteado anteriormente se plantea como **objetivo general** de este trabajo:

Desarrollar una herramienta web que permita integrar las funcionalidades necesarias para la gestión de proyectos informáticos.

Consecuentemente se definieron los siguientes **objetivos específicos**:

Lograr que el sistema permita:

1. Realizar reporte de tareas y tiempos del proyecto.
2. Realizar el control y reporte de errores.

⁵ Tele Identificador Personal.



3. Controlar el estado del proyecto en todo momento y generar un cronograma del mismo.
4. Controlar la distribución y disponibilidad de los recursos dentro del proyecto.

Dentro de las **tareas** propuestas para la realización de este trabajo se encuentran:

- ✓ Estudio de la herramienta WR TimeTracker, que se utiliza para realizar reporte de tareas y tiempos del proyecto.
- ✓ Estudio de la herramienta DotProject., utilizada para el control del estado del proyecto en todo momento y generar un cronograma del mismo.
- ✓ Estudio de la herramienta Trac, que se utilizan para llevar a cabo el reporte y control de errores dentro del proyecto.
- ✓ Estudio de los módulos existentes para la autenticación, en especial la librería LDAP y el modulo Autenticator v2.01.
- ✓ Estudio de el módulo Jpgraph, utilizado para la generación de diagramas sobre web.
- ✓ Estudio de lenguajes de programación web, en especial PHP 5 y Java Script.
- ✓ Estudio de patrones de arquitectura, en especial el patrón Modelo Vista Controlador.
- ✓ Estudio de los servidores web más utilizados, en especial el servidor web Apache.
- ✓ Estudio de los Sistemas Gestores de Bases de Datos más utilizados en la actualidad, en especial MySQL.
- ✓ Estudio de las metodologías de desarrollo de software, en especial XP.

Capítulo I. Fundamentación Teórica.

Introducción

En el presente capítulo se realiza un análisis de los temas vinculados con el área en que el futuro software se debe desempeñar, puntualizando conceptos claves antes de comenzar con el desarrollo del mismo. Incluye además un breve estudio de los sistemas similares existentes, así como un análisis de las metodologías y herramientas a utilizar para el desarrollo del sistema propuesto.

1.1 Gestión de Proyectos Informáticos

En el proceso de realizar cualquier gestión, sin importar su naturaleza, existen funcionalidades básicas que no se deben pasar por alto. En el siguiente subepígrafe se describen las funcionalidades necesarias para el caso de la gestión de proyectos informáticos.

1.1.1 Funcionalidades Para la Gestión de Proyectos Informáticos.

Para gestionar adecuadamente un proyecto de desarrollo de software es recomendable disponer de las siguientes funcionalidades:

Planificación

La planificación permite organizar el proyecto en función de hitos⁶, tareas y subtareas, con asignación y control de tiempos, recursos materiales y humanos. Idealmente un sistema de planificación debe permitir también hacer el seguimiento y reajustar la planificación en función de la evolución del proyecto.

Para lograr una planificación eficiente le es recomendable disponer de herramientas que permitan llevar el control de tiempos estimados y empleados para cada tarea; así como promover el registro del tiempo dedicado a cada tarea por las personas que trabajan en el proyecto.

Gestión Documental

La gestión documental permite almacenar y mantener los documentos obtenidos o generados durante el desarrollo del proyecto y acceder a ellos cómodamente. Cada hito, tarea o subtarea puede implicar la obtención o generación de documentación

⁶ Objetivos, metas lo que se pretende obtener luego de concluir una labor.

(actas de reuniones, documentos de diseño, etc.); idealmente el sistema de gestión de proyectos debe permitir que se almacene la documentación en el propio sistema o fuera de este.

Control de Versiones

El control de versiones permite el desarrollo concurrente y mantener la historia del código fuente y parte de la documentación producida en el proyecto. Al tratarse de proyectos informáticos lo normal es que se trabaje con código fuente y con documentos que van evolucionando a lo largo del desarrollo y que deben ser modificados por múltiples personas, por lo que resulta casi imprescindible disponer de un sistema de control de versiones que permita mantener la historia de los ficheros generados y que más de una persona trabaje concurrentemente sobre el mismo código.

Gestión de Incidencias

La gestión de incidencias permite el seguimiento de los errores detectados y sus correcciones, tanto aquellos reportados por los responsables de la prueba del software como por los desarrolladores o los usuarios finales. Un sistema de gestión de incidencias también se puede utilizar como sistema de seguimiento de tareas de corta duración asociadas a fases del proyecto, a errores detectados o a cambios relacionados con solicitudes de mejora realizadas por el cliente.

1.2 Herramientas para la Gestión de Proyectos.

Existen multitud de paquetes de software especializados que proporcionan las funcionalidades necesarias para la gestión de proyectos, ya sea por separado o integrando múltiples funcionalidades en un único producto.

En el estudio realizado en la búsqueda de sistemas informáticos similares en el ámbito nacional no se encontró ningún ejemplo, siendo este el primero de su tipo en el país. Sin embargo en el ámbito internacional se encontraron aplicaciones informáticas similares a la que se propone. Estos sistemas, aunque tienen características que los dotan de potencialidades para la gestión de proyectos, no se adecuan del todo a las necesidades planteadas por el proyecto TIP, ya que no presentan facilidades de trabajo con dominios en el manejo de usuarios y no logran integrar todas las funcionalidades exigidas.

A continuación se muestran algunos ejemplos de estas herramientas, las cuales han sido utilizadas por proyectos como TIP en la Universidad de las Ciencias Informáticas. Debido a que la documentación existente no es muy amplia, el análisis que se realiza no puede ser más abarcador, por lo cual se exponen sólo los aspectos fundamentales de cada una de las aplicaciones.

1.2.1 Trac

Trac es un sistema web libre para la gestión de proyectos y seguimiento de errores. Está inspirado en CVSTrac, y su nombre original era SVNTrac, debido a su fuerte dependencia de Subversion. Está desarrollado y mantenido por Edgewall Software y escrito en Python. Hasta mediados de 2005 estaba disponible bajo GNU (*General Public License*, del inglés), pero desde su versión 0.9, se distribuye de acuerdo a una modificación de la licencia BSD (ambas son licencias libres). Este sistema permite enlazar información entre una base de datos de errores de software, un sistema de control de versiones y el contenido de una wiki. (1)

1.2.2 DotProject

DotProject fue creado por dotmarketing.org en el año 2000, con el fin de construir una herramienta para la Gestión de Proyectos. Está construida por aplicaciones de Código abierto y es mantenida por un pequeño pero dedicado grupo de voluntarios. Es una aplicación basada en web, multiusuario, soporta varios lenguajes y es Software libre.

Está programada en PHP, y utiliza MySQL como base de datos (aunque otros motores como Postgres también pueden ser utilizados). La plataforma recomendada para utilizar DotProject se denomina LAMP (Linux + Apache + MySQL + PHP). De todas formas, existen binarios para instalar DotProject bajo otros sistemas operativos tales como Microsoft Windows (NT, 2000, XP) y Mac.

El grupo que desarrolla DotProject basa su espíritu de trabajo en los siguientes puntos:

- Proveer a los usuarios de funcionalidad orientada a la Gestión de Proyectos.
- Construir una herramienta con una interfaz de usuario simple, clara y consistente.
- Ser de código abierto, libre acceso y utilización.

El software es libre para todo el que desee descargarlo. Además existe soporte gratuito provisto por voluntarios en los foros de la comunidad y vía la documentación

online. También es posible contratar soporte del equipo de desarrollo a través del llamado “Foro de Soporte Prioritario”.

Como ya se ha mencionado anteriormente, DotProject es una herramienta orientada a la Gestión de Proyectos. Para eso se orienta a la administración de recursos para desarrollar un producto cuya producción requiera de un conjunto de actividades o tareas.

La aplicación consta de un conjunto de entidades ordenadas jerárquicamente las cuales permiten brindar la funcionalidad del producto.

A continuación se mencionan las entidades más importantes de DotProject:

- Compañías: entidades que agrupan proyectos, actividades y usuarios.
- Departamentos: áreas dentro de las compañías, que permiten agrupar usuarios en dicho nivel.
- Usuarios/Contactos: los usuarios de DotProject son capaces de loguearse y trabajar dentro del esquema de permisos que posea el rol que tienen asignado. Los contactos son usuarios especiales que asignados a un determinado proyecto pueden recibir por ejemplo: correo, actualizaciones y noticias pero no necesariamente deben tener acceso al sistema DotProject. Los usuarios y contactos pertenecen a una compañía.
- Proyecto: entidad que contiene el grupo de tareas necesarias para desarrollar un determinado producto.
- Actividades: tareas asignadas dentro de un proyecto. Son los componentes sobre los cuales se controla la duración, dependencias, recursos asignados y progreso. Las actividades deben de pertenecer a un único proyecto.
- Diagramas de Gantt: permite ver en forma gráfica las actividades ordenadas jerárquicamente, mostrando las dependencias y solapamientos de las mismas.
- Tickets: permite administrar todos los problemas relacionados a un proyecto.
- Archivos: permite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos.
- Foros: permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto del foro.
- Administración del Sistema: contiene las actividades relacionadas a la administración de usuarios, roles y configuración del sistema.
- Recursos: permite asignar recursos materiales (oficinas, equipamiento, etc.) a un proyecto. (2)

1.2.3 WR TimeTracker.

WR Time Tracker es un sistema de fuente abierta para el monitoreo de horario de trabajo basado en la red. Es simple y muy fácil de utilizar. Permite crear entradas al sistema de usuario y organizarlos en equipos, crear y modificar proyectos y actividades, ingresar horario para el trabajo, así como la generación y envío de informes y facturas para ser enviadas a correo electrónico. El sistema puede operar con FreeBSD⁷, Linux o Windows y está disponible en 7 idiomas: alemán, inglés, francés, holandés, noruego, portugués, rumano, ruso y español.. El hosting gratuito de este servicio está disponible para el público. (3)

1.3 Técnicas, Metodologías y Herramientas en las que se Apoya la Solución

Para la elaboración de la aplicación que se propone como solución al problema planteado es necesario un conjunto de técnicas y herramientas que permitan agilizar el proceso, así como una metodología de desarrollo de software que sirva de guía en el transcurso del mismo.

A continuación se describen las técnicas, metodologías y herramientas seleccionadas.

1.3.1 Arquitectura de Software

La arquitectura de software proporciona las bases para el desarrollo e implementación de sistemas informáticos. En el caso de las aplicaciones web desarrolladas en la actualidad se debe tener en cuenta la arquitectura Cliente/Servidor y el patrón de arquitectura Modelo Vista Controlador (MVC), ambos elementos se describen en el transcurso del presente epígrafe.

1.3.1.1 Arquitectura Cliente/Servidor

La arquitectura Cliente/Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos. En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, inicia un proceso de diálogo: produce una demanda de información o solicita recursos. La computadora que responde a la demanda del cliente, se conoce como servidor.

⁷ Sistema Operativo.

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura. Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet. Cliente/Servidor es el modelo de interacción más común entre aplicaciones en una red, no forma parte de los conceptos de la Internet como los protocolos IP, TCP o UDP, sin embargo todos los servicios estándares de alto nivel propuestos en Internet funcionan según este modelo. Se puede decir que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información; estableciendo así un enlace de comunicación transparente entre los elementos que conforman la estructura. No existe una definición específica adoptada universalmente de la Arquitectura Cliente/Servidor, las empresas de cómputo enfocan el concepto basándose en la funcionalidad que representa según los servicios que ellas mismas ofrecen. (4)

Características de la arquitectura Cliente/Servidor

Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Los sistemas desarrollados con arquitectura Cliente/Servidor poseen características que los distinguen de otras formas de software distribuido:

- Servicio. El servidor es un proveedor de servicios; el cliente es un consumidor de servicios.
- Recursos compartidos. Un servidor puede atender a muchos clientes al mismo tiempo y regular su acceso a recursos compartidos.

- Mezcla e igualdad. El software es independiente del hardware o de las plataformas de software del sistema operativo; se puede tener las mismas o diferentes plataformas de cliente y servidor.
- Intercambio basado en mensajes. Los sistemas interactúan a través de un mecanismo de transmisión de mensajes: la entrega de solicitudes y respuestas del servicio.
- Encapsulamiento de servicios. Los servidores pueden ser sustituidos sin afectar a los clientes, siempre y cuando la interfaz para recibir peticiones y ofrecer servicios no cambie.
- Facilidad de escalabilidad. Los sistemas Cliente/Servidor pueden escalarse horizontal o verticalmente. Es decir, se pueden adicionar o eliminar clientes (con apenas un ligero impacto en el desempeño del sistema); o bien, se puede cambiar a un servidor más grande o a servidores múltiples.
- Integridad. El código y los datos del servidor se conservan centralmente; esto implica menor costo de mantenimiento y protección de la integridad de los datos compartidos. Además, los clientes mantienen su individualidad e independencia. (5)

Patrón de Arquitectura Modelo Vista Controlador

En la medida que el tamaño de los sistemas crece, los algoritmos y las estructuras dejan de convertirse en el mayor de los problemas. El nuevo reto lo constituye diseñar y especificar la estructura global del sistema. Para esto en la actualidad se plantea el seguimiento de patrones de arquitectura para la construcción del software. Para el desarrollo de CIPI el patrón de arquitectura que se seguirá es el Modelo Vista Controlador (MVC).

El Modelo Vista Controlador es aplicable al desarrollo de cualquier aplicación independientemente del lenguaje de programación elegido. MVC consiste en dividir el código de una aplicación en capas, específicamente 3:

- Modelo
- Vista
- Controlador

Cada una de estas capas puede ser sustituida en cualquier momento sin afectar a las otras, puede tener diferentes vistas para un mismo modelo. Esta división del código hace más fácil la portabilidad y la adaptación a los requerimientos del usuario.

- Modelo es todo acceso a datos y las funciones que llevan lo que llaman "lógica de negocio", o sea datos y reglas de negocio. Lleva un registro de las vistas y controladores del sistema. Cada acceso a datos se pone en su función individual porque, de esta forma, si se cambia de gestor de bases de datos este cambio sólo afecta a estas funciones, no al resto de la aplicación. Tener el modelo bien delimitado permite la existencia de varias aplicaciones que compartan el mismo.
- Vista en una aplicación Web, es el HTML y lo necesario para convertir datos en HTML. O sea, muestra la información del modelo al usuario. Tienen un registro de su controlador asociado (normalmente porque además lo instancia). Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo. Tener la vista separada del controlador permite cambiar la aplicación para que genere, en lugar de HTML, algo distinto (por ejemplo, WML), sin tener que tocar más que una parte completamente delimitada del código.
- Controlador es lo que une la vista y el modelo. Por ejemplo, son las funciones que toman los valores de un formulario, consultan la base de datos (a través el modelo) y producen valores, que la vista tomará y convertirá en HTML. En resumen gestiona las entradas del usuario. Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. De este modo, el código que "hace algo" está perfectamente separado del código dedicado a crear HTML.

1.3.2 Servidor Web

En la Web, un servidor es un ordenador que usa el protocolo http para enviar páginas Web al ordenador de un usuario cuando el usuario las solicita. Un servidor Web no es más que un programa que se ejecuta de forma continua en un ordenador (también se utiliza el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador de Internet) y que contesta a estas peticiones de forma adecuada, sirviendo una página Web que será mostrada en el navegador o mostrando el mensaje correspondiente si se detectó algún error. Dichas características se adaptan perfectamente al tipo de aplicación que se desea construir, un servidor Web que procese, ejecute y de respuesta a las peticiones de los usuarios que lo soliciten.

Uno de los servidores Web más populares del mercado, y el más utilizado actualmente, es Apache, de código abierto y gratuito, disponible para Windows y GNU/Linux.

Servidor Web Apache:

Es el principal servidor Web del mundo del software libre. Después de la segunda mitad de la década del 90 ha tomado un gran auge en las aplicaciones Web, fundamentalmente en el soporte de aplicaciones programadas en PHP. Entre sus principales características están la flexibilidad para configurar los mensajes de error, contiene sus propias bases de datos de autenticación y negociado de contenido.

Principales ventajas:

- **Fiabilidad:** Alrededor del 90% de los servidores con más alta disponibilidad funcionan bajo servidores Apache.
- **Software Libre:** Apache es totalmente gratuito, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.
- **Extensibilidad:** Se pueden añadir módulos para ampliar las ya de por si amplias capacidades de Apache.

También dispone de una amplia variedad de módulos, que permiten generar contenido dinámico (con PHP, Java, Perl, Python,...), monitorizar el rendimiento del servidor, atender peticiones encriptadas por SSL, crear servidores virtuales por IP o por nombre (varias direcciones Web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos están disponibles junto con su código fuente, por lo cual pueden ser incluso modificados por cualquier persona con conocimientos de programación. (6)

Teniendo en cuenta las características antes mencionadas se escoge el servidor Web Apache como servidor Web para la aplicación que será desarrollada.

1.3.3 Lenguajes de Programación para la Web.

Entre los distintos lenguajes de programación para la Web que existen en la actualidad, se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente / Servidor, característica de los sistemas Web. El primer grupo está formado por los lenguajes que se ejecutan en el servidor (en inglés, *Server Side Languages*), dentro de los que se encuentran PERL, ASP, PHP y Java.

Estos lenguajes se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos y el tratamiento de la información. El segundo grupo lo integran los lenguajes que se ejecutan en el cliente (en inglés, *Client Side Languages*), ejemplos de ellos son: Java Script y el Visual Basic Script, ambos encargados de aportar dinamismo a la aplicación en los navegadores.

El estudio realizado permitió seleccionar para la implementación de la propuesta a PHP como lenguaje del lado del servidor y Java Script como lenguaje del lado del cliente. Las características de ambos lenguajes se expresan a continuación.

Lenguaje de Programación PHP

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él

PHP es un acrónimo recurrente que significa "*Hypertext Pre-processor*", es un lenguaje de programación interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. (7)

Fue creado por Rasmus Lerdorf a finales de 1994, aunque no hubo una versión utilizable por otros usuarios hasta principios de 1995. Esta primera versión se llamó Personal Home Page Tools. Actualmente PHP se encuentra en su versión 5, la cual utiliza el motor Zend. (8)

Debido a que es un lenguaje de fácil uso y que posee similitud con los lenguajes más comunes de programación estructurada, permite a la mayoría de los programadores experimentados crear aplicaciones complejas con una curva de aprendizaje muy suave, así como involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones y prácticas.

Su interpretación y ejecución se da en el servidor y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página Web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, Microsoft SQL Server, entre otros; lo cual permite la creación de aplicaciones web muy robustas.

Entre las ventajas que brinda su uso se pueden citar:

- Lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Permite leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Permite crear formularios para la Web.
- Posee una biblioteca nativa de funciones sumamente amplia.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos.

Lenguaje de Programación Java Script

Los lenguajes de programación del lado del cliente, como su nombre lo indica, son los que se ejecutan en el cliente o navegador, son los encomendados para darle dinamismo a la página sin necesidad de enviar información al servidor para realizar las operaciones requeridas. Son lenguajes interpretados que pueden acceder a la información HTML que se muestra en el navegador, modificarla o actualizarla según las necesidades del programador.

Java Script es uno de los lenguajes del lado del cliente más utilizados por ser compatibles con la mayoría de los navegadores modernos. Es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts así como la de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Permite además el acceso a los elementos que componen la página Web,

brindándole la posibilidad al programador modificar el contenido de la página dinámicamente. (9)

1.4 Sistemas de Gestión de Bases de Datos

Un Sistema Gestor de Base de Datos (SGBD) se puede definir como un conjunto de programas que permiten crear y mantener una Base de Datos (BD), asegurando la integridad, confidencialidad y seguridad de la misma. Por tanto es necesario que el sistema permita:

- Definir una BD: especificar tipos, estructuras y restricciones de datos.
- Construir la BD: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la BD: realizar consultas, actualizar o eliminar la información almacenada y generar informes.

Los beneficios que se pueden obtener de las funcionalidades brindadas por el SGBD facilitan el trabajo de los desarrolladores e influyen en la calidad y seguridad de la BD, por lo que existen algunas características deseables en un SGBD, ejemplo de ellas son las que se mencionan a continuación:

- Control de la redundancia.
- Restricción de los accesos no autorizados.
- Cumplimiento de las restricciones de integridad.
- Facilidad de manipulación de la información.
- Control centralizado.

Entre los SGBD comúnmente utilizados en el mundo se encuentran Oracle, MySQL, Microsoft SQL Server, PostgreSQL e Interbase. Estos SGBD presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

En el siguiente epígrafe se describen las características del sistema gestor de base de datos seleccionado para la implementación del sistema.

1.4.1 Sistema Gestor de Bases de Datos MySQL

MySQL se ha convertido en uno de los SGBD de código abierto más populares y aceptados en la actualidad debido a su gran rendimiento, alta fiabilidad y facilidad de

uso. Constituye una elección para la nueva generación de aplicaciones construidas sobre la pila de LAMP (Linux, Apache, MySQL, PHP / Perl / Python). Se ejecuta en más de 20 plataformas, incluyendo Linux, Windows, OS/X, HP-UX, AIX, NetWare. (10)

Las características fundamentales de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos.

MySQL es ideal para las aplicaciones Web debido a que aunque en estas últimas existe baja concurrencia en la modificación de datos, el entorno es intensivo en cuanto a lectura de datos.

Teniendo en cuenta las características de este SGBD y las condiciones antes mencionadas de las aplicaciones WEB se decide utilizar como SGBD a MYSQL.

1.5 Editor de Código PHP Zend Studio.

Debido al incremento de la necesidad del uso de aplicaciones Web y el ascenso de la complejidad de estas, se hace necesaria la utilización de lenguajes de programación que las doten de nuevas potencialidades. En el caso de la aplicación que se propone el lenguaje de programación que se utilizará será PHP, y aunque este lenguaje puede ser fácilmente utilizado desde editores de texto se hace necesaria la utilización de un editor especializado en dicho lenguaje, en este caso Zend Studio.

Zend Studio es una poderosa herramienta para el trabajo con aplicaciones Web usando el lenguaje de programación PHP. Además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa en conjunto está implementado en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para



Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Lo más notable es que contiene una ayuda contextual con todas las librerías de funciones del lenguaje que asiste en todo momento, ofreciendo nombres de las funciones y parámetros que deben recibir. Aunque esta ayuda contextual no sólo se queda en las funciones definidas en el lenguaje, sino que también reporta ayudas con las funciones que se vayan creando, incluso en páginas que estén incluidas con la función `include()`. Otras ayudas que ofrece a la hora de escribir son las típicas en editores avanzados, como permitir editar varios archivos, y moverse fácilmente entre ellos, marcar a qué elementos corresponden los inicios y cierres de las etiquetas, paréntesis o llaves, moverse al principio o al final de una función, identificación automática del código. (11)

Las mejores cualidades de Zend Studio no se encuentran en la parte de edición sino en las posibilidades de gestión de proyectos y depuración. La barra de la izquierda, que permite navegar los archivos, también dispone de herramientas para gestionar los proyectos, muy útiles para mejorar la productividad en la programación. Los proyectos permiten guardar mucha más información al programa sobre los archivos, discos, servidores, etc. que se gestionen en aplicaciones PHP.

Una vez que los archivos se han añadido al proyecto se pueden guardar señales como puntos de ruptura en las depuraciones, asimismo, cuando se ejecuta Zend Studio, se vuelven a abrir los archivos que estuvieran abiertos la última vez que el programa se cerró y las herramientas de completar código mejoran sus comportamientos, asumiendo toda la información de los archivos relacionados con el proyecto.

Zend Studio implementa además unas interesantes opciones para trabajar en grupo, al integrar el sistema de trabajo conocido como Control Version System (CVS). Sin duda, más de una vez los programadores de PHP se han visto en un duro problema por no encontrar un error en algún script que está dando resultados inesperados, para facilitar el trabajo, Zend Studio dispone de una herramienta muy interesante de debug o depuración. Gracias a ella se pueden ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación y las variables del entorno como las cookies, las recibidas por formulario o en la sesión. Se pueden colocar puntos de parada de los scripts y realizar las acciones típicas de depuración.

Además de la ventana para visualizar el contenido de las variables, dispone de otras donde muestra la salida del script según se va generando, así como las alertas y

errores. Las posibilidades se completan con distintos tipos de depuración, en local, en remoto o a partir de una URL. (11)

Debido a las numerosas facilidades que ofrece este editor se decidió utilizarlo como editor de código para la implementación de la aplicación propuesta.

1.6 Metodologías de Desarrollo

Todo desarrollo de software se torna riesgoso y difícil de controlar, pero si no se utiliza una metodología para guiar el proceso, las posibilidades de obtener un producto de baja calidad, clientes y desarrolladores insatisfechos con el resultado, atrasos considerables en el cronograma e incluso gastos superiores a los estimados se incrementan considerablemente.

En la actualidad existen muchas metodologías para el desarrollo de software. Las OO (orientadas a objetos) son varias, dentro de las que se pueden mencionar se encuentran Extreme Programming (XP), METRICA 3 y Rational Unified Process (RUP).

A continuación se muestra la metodología seleccionada para guiar el desarrollo de la aplicación propuesta.

1.6.1 Metodología de Desarrollo XP.

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, equipo reducido y cuyo plazo de entrega era "ayer". La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Se basa en:

- *Pruebas Unitarias*: pruebas realizadas a los principales procesos, de tal manera que al adelantarse en algo hacia el futuro, se pueden hacer pruebas de las fallas que pudieran ocurrir. Es como si el equipo 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. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

¿Qué es lo que propone XP?

- Empieza en pequeño, o sea, de forma simple y sin mucha complejidad y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

El cliente al ser participante activo del proceso de desarrollo posee algunos derechos como:

- Decidir qué se implementa.
- Saber el estado real y el progreso del proyecto.
- Añadir, cambiar o quitar requerimientos en cualquier momento.
- Obtener lo máximo de cada semana de trabajo.
- Obtener un sistema funcionando cada 3 o 4 meses.

Derechos del Desarrollador:

- Decidir cómo se implementan los procesos.
- Crear el sistema con la mejor calidad posible.
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- Estimar el esfuerzo para implementar el sistema.
- Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

Se decidió utilizar XP debido a que se adapta en gran medida tanto al tipo de proyecto a desarrollar como a las condiciones de trabajo. A continuación se exponen varias de las razones que llevaron al uso de esta metodología:

- El proyecto es pequeño. XP está concebida para ser utilizada dentro de proyectos pequeños.
- No existe un contrato previo especificando tiempo, recursos y alcance. Para el desarrollo del sistema no se dispone de un contrato con un presupuesto ni un alcance previamente definidos.
- El cliente forma parte del equipo de desarrollo. Mediante la aplicación de XP se puede lograr una retroalimentación mayor y lograr un producto que satisfaga sus necesidades.
- El riesgo de desarrollo es elevado debido al corto tiempo de entrega planteado. XP está diseñada a mitigar los riesgos en proyectos con estas características.
- Poca disponibilidad de personal. El sistema debe ser realizado por dos personas solamente, no siendo posible la existencia de muchos roles ni la especialización en un rol específico por parte de los miembros. Uno de los principios básicos de XP es la programación en equipos pequeños (2 a 12 personas) con pocos roles, pudiendo los miembros del equipo intercambiar responsabilidades en un momento determinado.

Conclusiones:

En este capítulo se presentaron la metodología y las tecnologías utilizadas para la realización de la Herramienta para el Control Integrado de Proyectos Informáticos (CIPI). Como lenguaje de programación del lado del servidor se empleará PHP, utilizando el Zend Studio como herramienta de desarrollo, MySQL como Sistema Gestor de Base de Datos y Apache como servidor de aplicaciones web, además del lado del cliente se utilizará Java Script para lograr la interactividad con el usuario en el navegador.



Todas estas herramientas ofrecen gran número de potencialidades además de formar parte del grupo de software de código abierto, solución por la que aboga la universidad.

Capítulo II. Características del sistema.

Introducción

En el presente capítulo se describen los flujos de procesos del negocio que se implementará. Se establece una comparación entre la solución propuesta y otros sistemas existentes en la actualidad y se realiza además un análisis de las características del sistema a desarrollar.

2.1 Flujo de los procesos del negocio.

El análisis del flujo de procesos permite reconocer cómo funciona realmente el negocio para producir uno o varios resultados. El resultado puede ser un producto, un servicio, una información o combinaciones de ellos. Analizar el flujo de procesos permite revelar problemas potenciales.

Con el objetivo de aumentar su productividad y lograr el nivel de calidad requerido para sus soluciones informáticas, el proyecto TIP lleva a cabo un control de tareas, tiempos y errores en el que interactúan todos los miembros del equipo de desarrollo.

Actualmente este proceso está regido por una serie de acciones que permiten controlar distintas variables sobre el trabajo del equipo de desarrollo y ofrecen un medio factible con el que interactúan todos los miembros del equipo. En la siguiente gráfica se muestra la estructura existente con relación a la interacción entre los usuarios y los tres sistemas utilizados en el proyecto ENUM. En los epígrafes siguientes se describirán los flujos y acciones llevados a cabo durante todo el proceso de control del proyecto.

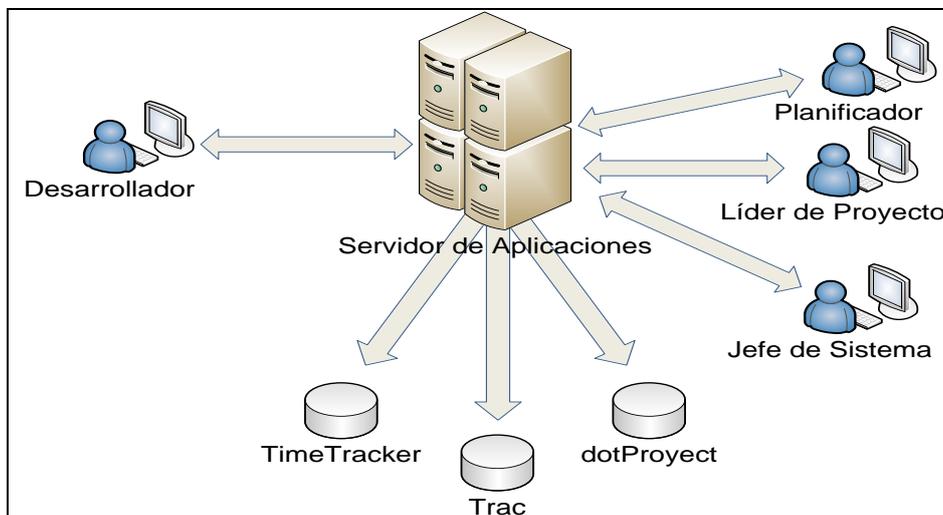


Figura 1 Representación del Campo de Acción

2.1.1 Reporte de Tareas y Tiempos

Durante todo el proceso de desarrollo del software, se generan tareas que son distribuidas entre los miembros del grupo de trabajo en dependencia de sus roles dentro del equipo y de la disponibilidad de cada cual. Estas tareas son definidas por los jefes de sistema y publicadas utilizando la herramienta web TimeTracker, a la cual ya se ha hecho alusión en el capítulo 1.

Los miembros del equipo, acceden diariamente a las tareas que les corresponden realizar y registran el tiempo que dedican a la terminación de las mismas, de esta forma los jefes de módulo, apoyados en la herramienta antes mencionada, realizan los reportes necesarios sobre las tareas realizadas, incumplidas o sin culminación, así como el tiempo que se ha requerido para el cumplimiento de cada una de ellas.

2.1.2 Control y reporte de errores

Durante la fase de prueba el proceso de detección e informe de los errores encontrados es fundamental para garantizar la calidad del producto final que se desea obtener. En esta fase, los probadores junto al líder de proyecto luego de detectar los errores publican, en esta ocasión usando el sistema de tickets que proporciona la herramienta Trac que ya se ha mencionado, los datos de los mismos y los notifican al responsable de darle tratamiento para erradicarlos.

Cada miembro del equipo puede visualizar a través de los tickets los errores que le corresponden y su descripción, luego informar sobre el curso que ha tomado la corrección del error a los probadores y al líder de proyecto. El líder de proyecto puede mediante el Trac, obtener un reporte de todos los errores que se han ido descubriendo y el estado actual en el que se encuentran.

2.1.3 Estado del proyecto. Cronograma.

En este proceso, los planificadores del proyecto en conjunto con el líder, realizan un cronograma inicial, mediante la herramienta DotProject incluyen las tareas definidas además de su tiempo de realización (fecha inicio, fecha fin) y los responsables de las mismas. La herramienta, con toda la información incluida, genera un diagrama de Gantt inicial que sirve de punto de partida. Durante toda la etapa de desarrollo los

planificadores actualizan el estado de las tareas en valores porcentuales y a su vez pueden hacer reportes del estado de cumplimiento en el que se encuentra el proyecto.

2.1.4 Gestión, Control y Distribución de Recursos.

A pesar de contar con las tres herramientas mostradas en la figura 1, TIP carece de una aplicación que facilite tanto al líder de proyecto como al planificador la gestión, el control y la distribución de recursos. Estos recursos pueden ser de tipo material (puestos de trabajo existentes en el proyecto) o humanos (son los miembros del equipo de desarrollo del proyecto). En el caso de la gestión de recursos humanos los tres sistemas deben ser actualizados de forma independiente, lo que no permite establecer un control centralizado al aparecer en ocasiones los usuarios con nombres y contraseñas distintos. En cuanto a los recursos materiales no se registra en ninguna de las aplicaciones la existencia ni el estado de los mismos, esta tarea tiene que ser llevada íntegramente por el planificador del proyecto en formato duro. Para la distribución de los recursos (entiéndase la asignación de puestos de trabajo a los desarrolladores en las tres secciones existentes en el día: mañana, tarde y noche) el planificador tiene que elaborarla de forma manual y luego enviarla a todos los miembros del equipo de desarrollo por la vía del correo electrónico.

2.2 Análisis Crítico del Funcionamiento de los Sistemas Actuales.

Existen, en la actualidad, diversos sistemas para el control de proyectos, cada uno con sus características propias, algunos comerciales y otros tantos de código abierto.

Debido a la distribución de las funcionalidades en tres herramientas, la información que se almacena sobre los estados en que se encuentra el proyecto TIP está separada, de ahí que no se cuente con un único reporte general y el líder de proyecto tenga que acudir a pequeños reportes, por separado, generados por cada uno de los sistemas. Para darle solución a este problema se brindará la posibilidad de obtener reportes generales sobre los estados en que se encuentre el proyecto, además de los reportes específicos de cada uno de los procesos.

Los sistemas mencionados no trabajan con las facilidades del uso de dominios en cuanto al manejo de los datos de los usuarios que utilizan los sistemas. Como parte del sistema propuesto se trabajará con los usuarios pertenecientes al dominio uci.cu, permitiendo que sólo usuarios registrados en este dominio sean los que accedan a la aplicación siempre y cuando estos hayan sido incluidos previamente como usuarios del sistema por el líder de proyecto.

En el siguiente epígrafe se describirán las características del sistema que se propone.

2.3 Características de la Propuesta.

En el presente trabajo se propone la implementación de un sistema sobre plataforma web que incluya funcionalidades y características similares a los sistemas existentes y otras funcionalidades adicionales para mejorar las prestaciones, integrándolas en una misma aplicación. Estas funcionalidades se ajustan a las necesidades actuales del proyecto TIP de la Facultad 2 de la Universidad de las Ciencias Informáticas. Estará programado en PHP, utilizando MySQL como gestor de Bases de Datos y Apache como servidor web. Será independiente tanto del sistema operativo donde se ejecute como del lenguaje y plataforma utilizados para desarrollar los proyectos.

El sistema ha desarrollar constará de cuatro módulos fundamentales: reporte de tareas y tiempos, reporte y control de errores, estado y cronograma del proyecto, distribución y control de recursos en el proyecto.

A continuación se muestra un gráfico que describe la arquitectura del sistema propuesto y seguidamente la descripción de los módulos a desarrollar.

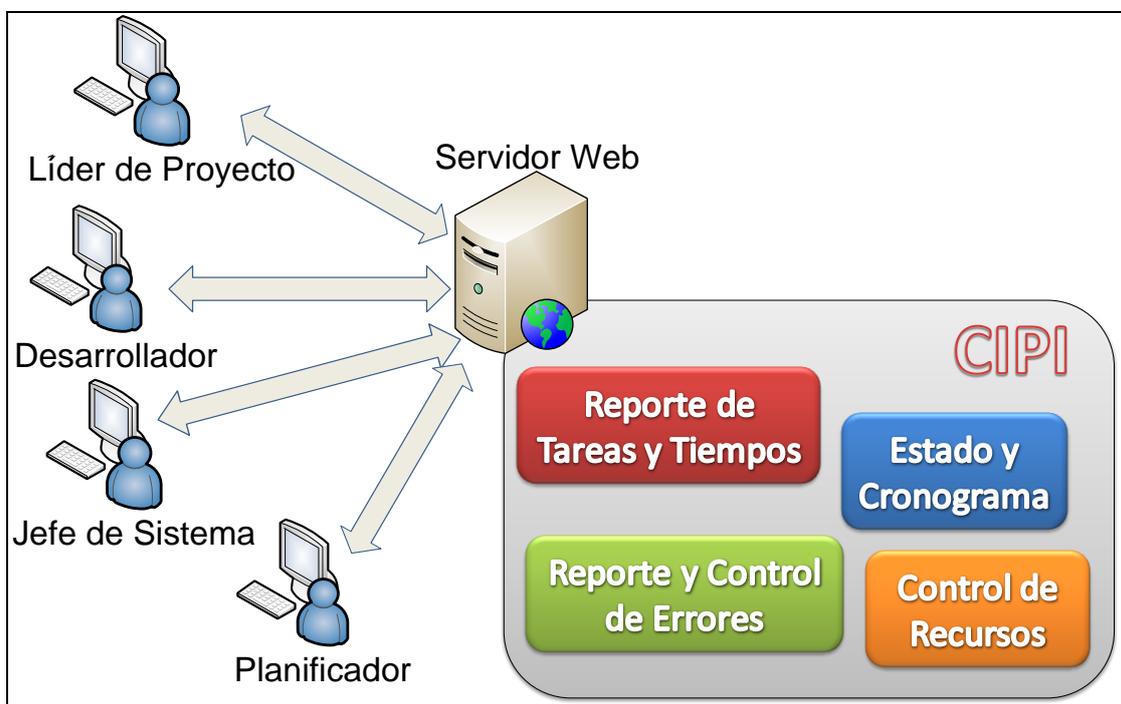


Figura 2. Arquitectura del sistema propuesto.

2.3.1 Descripción del Módulo: Reporte de Tareas y Tiempos.

Este módulo garantizará las funciones correspondientes a los flujos y procesos tratados en el apartado 2.1.1 del presente trabajo. Garantizará la creación de nuevas tareas por parte de los jefes de sistema y el líder del proyecto. Además se implementarán en él las funcionalidades para el control del tiempo empleado en estas tareas por los desarrolladores, avalando que en cada caso se tome como hora de inicio y fin la hora registrada por el servidor de aplicaciones y no la de los puestos de trabajo.

2.3.2 Descripción del Módulo: Reporte y Control de Errores.

Este módulo permitirá llevar a cabo los procesos descritos en el apartado 2.1.2. Mediante él, los probadores y el líder de proyecto podrán publicar los errores detectados durante la etapa de prueba y a su vez realizar reportes de todos los errores tratados por los miembros del equipo de desarrollo, los cuales a través de los tickets accederán a los que les corresponde tratar.

2.3.3 Descripción del Módulo: Estado y Cronograma del Proyecto.

Este módulo permitirá generar el diagrama de Gantt inicial del proyecto, y con el objetivo de que el líder de proyecto y los planificadores puedan conocer en que estado se encuentra permitirá su actualización constante.

2.3.4 Descripción del Módulo: Distribución y Control de Recursos en el Proyecto.

Este módulo permitirá al líder de proyecto:

El control de recursos materiales: el líder podrá almacenar y gestionar los recursos materiales existentes en el proyecto, teniendo como prioridad el estado en el que se encuentran estos recursos, propiciando que su explotación sea siempre la adecuada.

Gestionar los recursos humanos: el líder podrá crear o eliminar usuarios y asignar roles dentro del proyecto.

Realizar la distribución de recursos: el líder podrá realizar la distribución de los recursos materiales en el proyecto, asignándole a cada individuo los medios con los que va a trabajar, lo que permitirá que sea óptima la explotación de los mismos.

2.4 Análisis comparativo de otras soluciones existentes con la propuesta.

Durante el desarrollo de los estudios realizados, previos a la implementación del sistema, se analizaron algunas de las herramientas más utilizadas para la gestión de proyectos informáticos, de estas se tomaron ideas con el objetivo de incorporarlas a la herramienta propuesta. Se definieron además un conjunto de funcionalidades que no se encontraban presente en los sistemas estudiados y se hacía necesario incorporarlas al nuevo sistema.

En general las aplicaciones estudiadas incluyen funcionalidades para la gestión de proyectos, y aunque son implementadas de forma diferente, coinciden en que la planificación de tiempos y tareas a desarrollar, la organización del trabajo por parte del equipo, el seguimiento y corrección de errores, junto al almacenamiento y control de versiones son características indispensables para llevar a cabo la adecuada gestión de proyectos.

La herramienta web propuesta incluye parte de estas funcionalidades además de incorporar el control de recursos, ya sean humanos o materiales, con el fin de convertir la organización y la eficiencia en el arma fundamental para el éxito y la rentabilidad del proyecto, además, dadas las características de los proyectos en la UCI y la necesidad de mantener registrados usuarios pertenecientes al dominio uci.cu la aplicación permitirá la autenticación de estos utilizando el usuario asignado por el dominio antes mencionado, aspectos que no fueron implementados en ninguna de las aplicaciones estudiadas.

Conclusiones.

En el presente capítulo se realizó un análisis sobre los procesos llevados a cabo dentro del entorno de desarrollo y que se encuentran vinculados al funcionamiento del negocio, enfatizando en cuáles de ellos serán objeto de automatización, además de una descripción detallada sobre las características de la solución propuesta.

Capítulo III. Exploración, Planificación y Diseño.

Introducción

En el presente capítulo se hace alusión a las fases de exploración y planificación propias de la metodología de desarrollo utilizada para guiar la elaboración del sistema que se propone. Se exponen además los artefactos generados durante el transcurso de las mismas.

3.1 Historias de Usuario

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario (*UH User History en inglés*) con el cliente. Las UH tienen la misma finalidad que los casos de uso pero con algunas diferencias en cuanto al modo de descripción: constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una UH, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Las UH son clasificadas de la siguiente forma:

Teniendo en cuenta la **Escala Nominal de Prioridad en el Negocio:**

Alta: Se le otorga a las Historias de Usuario que resultan funcionalidades fundamentales en el desarrollo del proyecto, a las que el cliente define como principales para el control integral de proyectos.

Media: Se le otorga a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.

Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

Teniendo en cuenta la **Escala Nominal de Riesgo en Desarrollo:**

Alta: Cuando para la implementación de la UH se considera la posible existencia de errores que lleven a inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la UH que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

A continuación se muestran las UH que serán utilizadas en el desarrollo de la aplicación propuesta y que reflejan la información brindada por el cliente.

Tabla 3.1.1 UH. Gestión de Tareas.

Historias de Usuario	
Número: 1	Nombre: Gestión de Tareas
Usuario: Planificador, Líder de Proyecto	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 2	Iteración Asignada: 1
Descripción: Se crean y planifican las tareas necesarias para el desarrollo de un proyecto creado con anterioridad, en general estas contarán con la fecha de inicio y de fin propuestas para el desarrollo de la tarea.	
Observaciones:	

Tabla 3.1.2 UH. Reporte de Tiempos.

Historias de Usuario	
Número: 2	Nombre: Reporte de Tiempos
Usuario: Desarrolladores	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Medio

Puntos Estimados: 1	Iteración Asignada: 1
Descripción: Se reportan las fechas y horas de inicio y fin del trabajo sobre una tarea, dando esto la posibilidad de calcular el tiempo real de desarrollo de la tarea.	
Observaciones:	

Tabla 3.1.3 UH. Reporte de Errores.

Historias de Usuario	
Número: 3	Nombre: Reporte de Errores
Usuario: Desarrolladores, Líder de Proyecto, Planificador, Probador	
Prioridad en el Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: 1
Descripción: Luego de publicar el código generado por el desarrollo de una tarea se hará una revisión crítica, como resultado se obtendrán los errores que serán reportados al o los desarrolladores responsables de la tarea, estos darán tratamiento a los errores y reportarán la corrección del error	
Observaciones: Se utiliza el sistema de tickets para llevar a cabo los reportes de errores.	

Tabla 3.1.4 UH. Cronograma y Estado del Proyecto.

Historias de Usuario	
Número: 4	Nombre: Creación de Cronograma
Usuario: Planificador, Líder de Proyecto	
Prioridad en el Negocio: Media	Riesgo en Desarrollo: Medio
Puntos Estimados: 2	Iteración Asignada: 2

Descripción: Se generará el cronograma del proyecto partiendo de la fecha de inicio y fin de las tareas y el porcentaje de realización en el que se encuentran; este porcentaje sólo podrá ser actualizado por el planificador, el líder y el colíder del proyecto. El porcentaje o estado del proyecto se calculará a partir del estado de las tareas que le pertenecen.

Observaciones: La planificación sólo será llevada a cabo por el planificador.

Tabla 3.1.5 UH. Control de Recursos Materiales.

Historias de Usuario	
Número: 5	Nombre: Control de Recursos Materiales.
Usuario: Planificador, Líder de Proyecto	
Prioridad en el Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: 3
Descripción: Se registra los datos generales de los recursos materiales (puesto de trabajo), así como su disponibilidad.	
Observaciones: No se registrara en la base de datos el inmobiliario destinado al proyecto.	

Tabla 3.1.6 UH. Gestión de Recursos Humanos

Historias de Usuario	
Número: 6	Nombre: Gestión de Recursos Humanos.
Usuario: Líder de Proyecto	
Prioridad en el Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: 3

Descripción: Gestiona, de acuerdo a las necesidades del proyecto, el personal para la creación, asignación y relevo de roles dentro del equipo de desarrolladores.
Observaciones:

Tabla 3.1.7 UH. Distribución de Recursos.

Historias de Usuario	
Número: 7	Nombre: Distribución de Recursos.
Usuario: Planificador	
Prioridad en el Negocio: Baja	Riesgo en Desarrollo: Medio
Puntos Estimados: 1	Iteración Asignada: 3
Descripción: Distribuye los recursos materiales haciendo asignación de tiempo de máquina al equipo de desarrollo del proyecto.	
Observaciones:	

3.2 Planificación.

En el desarrollo de la fase de planificación se lleva a cabo la estimación del esfuerzo que costará implementar cada UH. Se utilizará como unidad de medida el punto de estimación, tomando como que una semana de trabajo ideal (cuando no existen interrupciones y se lleva a cabo un trabajo fluido) es equivalente a un punto. Esta estimación tendrá en cuenta el esfuerzo necesario para la implementación de cada UH.

3.2.1. Estimación de Esfuerzo por Historias de Usuario

Para la construcción del sistema que se propone se llevó a cabo la estimación que se muestra en la siguiente tabla:

Tabla 3.2.1 Estimación de esfuerzo por UH.

No. UH	Historia de Usuario (UH)	Puntos Estimados
--------	--------------------------	------------------

1	Gestión de Tareas.	2
2	Reporte de Tiempos.	1
3	Reporte de Errores.	1
4	Creación de Cronograma y Estado del Proyecto.	2
5	Control de Recursos Materiales.	1
6	Gestión de Recursos Humanos.	1
7	Distribución de Recursos.	1

3.2.2 Plan de Iteraciones.

Teniendo las historias de usuario del sistema definidas y la estimación del esfuerzo para implementar cada una de ellas se pasa a realizar la planificación de la etapa de implementación del proyecto. Atendiendo a lo mencionado con anterioridad se decide realizar tres iteraciones durante el desarrollo del sistema. Se llevará a cabo la creación de una interfaz web amigable de acorde a las preferencias del cliente y de fácil navegabilidad en cada una de las iteraciones. A continuación se describen las iteraciones.

3.2.2.1 Iteración 1

En esta iteración se implementarán las historias de usuario de nivel de prioridad Alta, creando en ella la línea base de la arquitectura del sistema. Además se obtendrá la versión 0.1 en la que el cliente podrá probar algunas de las funcionalidades solicitadas como Gestión de Tareas, Reporte de Tiempos y Reporte de Errores, descritas en las UH 1, 2 y 3.

3.2.2.2 Iteración 2

En la segunda iteración se llevará a cabo la implementación de la historia de usuario cuatro, de prioridad media, en la que se dispondrá como resultado final una funcionalidad que permita al cliente la creación del cronograma compuesto por las tareas generales, así como del estado actualizado del proyecto en el momento que se

le pida a la aplicación. Al concluir esta iteración se le dará al cliente la versión de prueba 0.2 de la propuesta.

3.2.2.3 Iteración 3

Esta iteración es la dedicada a la implementación de las historias de usuario de nivel de prioridad Baja. Se desarrollará en el transcurso de la misma la versión 1.0 de CIPI, adicionando las funcionalidades descritas en las UH 5, 6 y 7.

3.2.3 Plan de duración de las Iteraciones

Con la utilización de la metodología de desarrollo XP se hace necesario la creación de un plan de duración de iteraciones como parte del ciclo de vida del proyecto. En dicho plan de duración se contará con cada una de las iteraciones a realizar, las historias de usuarios que se implementarán en la iteración y la duración en semanas.

Tabla 3.2.3 Plan de Duración de las Iteraciones.

Iteración	Historias de Usuario a Implementar	Duración en Semanas
Iteración 1	1- Gestión de Tareas. 2- Reporte de Tiempos. 3- Reporte de Errores.	4 Semanas.
Iteración 2	1- Creación de Cronograma y Estado del Proyecto.	2 Semanas.
Iteración 3	1- Control de Recursos Materiales. 2- Gestión de Recursos Humanos. 3- Distribución de Recursos.	3 Semanas.

3.2.4 Plan de Entregas

A continuación se muestra una tabla con el plan de entrega propuesto para cada una de las versiones de CIPI.

Tabla 3.2.4 Plan de Entregas.

Sistema	Fin Iteración 1	Fin Iteración 2	Fin Iteración 3
	3ra Semana Abril	1ra Semana Mayo	4ta Semana Mayo
CIPI	0.1	0.2	1.0

3.3 Diseño

Durante el diseño de la solución, y de cada historia de usuario que lo requiera, la máxima simplicidad posible es la clave para el éxito de XP. Se debe tener en cuenta que un diseño complejo siempre tarda más en desarrollarse que uno simple, y que siempre es más fácil añadir complejidad a un diseño simple que quitarla de uno complejo, siempre se deben hacer las cosas lo más sencillas posible, evitando añadir una funcionalidad no contemplada a una iteración.

3.3.1 Tarjeta CRC

Para el diseño del sistema Xtreme Programing sugiere llevar a cabo reuniones entre todos los desarrolladores implicados, donde se toman las decisiones mediante el uso de tarjetas CRC (class, responsibilities and collaboration, en inglés). Cada tarjeta representa un objeto del sistema, en la que su nombre aparece escrito en la parte superior, sus responsabilidades en la parte izquierda y los objetos con los que colabora en la parte derecha.

A continuación se muestran las tarjetas CRC correspondientes a las clases utilizadas en la implementación del sistema.

Tabla 3.3.1 Tarjeta CRC correspondiente a la clase CControladora

Nombre de la clase: CControladora	Superclase:	Subclase:
Responsabilidad Principal: Gestión de las tareas del proyecto		
Responsabilidades		Colaboraciones

Gestión de tareas	CTareas, Modelo
Asignación de tareas	CTareas, Modelo
Reportes de tareas	CTareas, CUsuario, CProyectos, Modelo
Registro de tiempos de trabajo	CTareas, CUsuario, Modelo, CHora
Gestión de proyectos	CProyectos, CTareas, CUsuario, Modelo
Gestión de tickets	CTareas, CTicket, CUsuario Modelo
Notificación de tickets	CTicket, CUsuario, Modelo
Reporte de tickets	CTicket, CProyectos, CUsuario, Modelo
Creación del cronograma	CProyectos, CTareas, Modelo
Gestión de usuario	CUsuario, Modelo
Gestión de puestos de trabajo	CRecursos, Modelo
Asignación de recursos	CRecursos, CUsuario, Modelo
Reporte de horario de trabajo	CRecursos, CUsuario, Modelo

Tabla 3.3.2 Tarjeta CRC correspondiente a la clase CTareas

Nombre de la clase: CTareas	Superclase:	Subclase:
Responsabilidad Principal: Gestión de tareas		
Responsabilidades		Colaboraciones
Creación de tareas		Modelo
Editar tareas		Modelo
Eliminar tareas		Modelo
Obtener datos de las tareas		Modelo

Tabla 3.3.3 Tarjeta CRC correspondiente a la clase CProyectos

Nombre de la clase: CProyectos	Superclase:	Subclase:
Responsabilidad Principal: Gestión de proyectos		
Responsabilidades		Colaboraciones
Creación de proyectos		Modelo
Editar proyectos		Modelo
Eliminar proyectos		Modelo
Obtener datos de los proyectos		Modelo
Cálculo de la duración del proyecto		Modelo, CTareas

Tabla 3.3.4 Tarjeta CRC correspondiente a la clase CUsuario

Nombre de la clase: CUsuario	Superclase:	Subclase:
Responsabilidad Principal: Gestión de usuarios		
Responsabilidades		Colaboraciones
Creación de usuarios		Modelo
Editar usuarios		Modelo
Eliminar usuarios		Modelo
Obtener datos de los usuarios		Modelo
Autenticar usuario		Modelo, LDAP
Registro de tiempo de trabajo		Modelo, CTareas, CProyectos, CHora

Tabla 3.3.5 Tarjeta CRC correspondiente a la clase CTicket

Nombre de la clase: CTicket	Superclase:	Subclase:
Responsabilidad Principal: Gestión de ticket		
Responsabilidades		Colaboraciones
Creación de ticket		Modelo
Editar ticket		Modelo
Eliminar ticket		Modelo
Obtener datos de los ticket		Modelo
Notificación de ticket		Modelo, CUsuario

Tabla 3.3.6 Tarjeta CRC correspondiente a la clase CRecursos

Nombre de la clase: CRecursos	Superclase:	Subclase:
Responsabilidad Principal: Gestión de recursos		
Responsabilidades		Colaboraciones
Creación de recursos		Modelo
Editar recursos		Modelo
Eliminar recursos		Modelo
Obtener datos de los recursos		Modelo
Realizar planificación		Modelo, CUsuario
Obtener horario de trabajo		Modelo, CUsuario

Tabla 3.3.7 Tarjeta CRC correspondiente a la clase CHora

Nombre de la clase: Chora	Superclase:	Subclase:
Responsabilidad Principal: Trabajo con horas y fechas		
Responsabilidades		Colaboraciones
Calcular diferencia en fechas y horas		

3.3.2 Diseño de base de datos.

Para la facilitar la implementación del sistema se realizó, además, el diseño de una base de datos en la cual se registrarán todos lo datos necesarios para el correcto funcionamiento de la aplicación. [*Anexo 2 Diseño de Base de Datos*]

Conclusiones

En este capítulo se exponen los artefactos obtenidos en cada una de las fases de Exploración, Planificación y Diseño correspondientes a la metodología de desarrollo XP. Estos artefactos son de vital importancia debido a que son usados posteriormente e la fase de Implementación, por tanto si se logra un diseño que responda a las necesidades del cliente, la probabilidad de que se encuentren errores en la implementación del sistema disminuirá considerablemente.

Capítulo IV. Implementación y Prueba.

Introducción

La implementación según plantea la metodología de desarrollo XP, debe realizarse de forma iterativa teniendo como fruto de cada iteración un producto funcional que pueda ser probado y mostrado al cliente. El equipo de desarrollo se retroalimentará con las sugerencias y críticas que haga el cliente sobre la versión culminada al final de cada iteración.

En el presente capítulo se detallan las iteraciones que serán llevadas a cabo durante la implementación del sistema, se describen las tareas generadas por cada una de las Historias de Usuario y las pruebas de aceptación efectuadas.

A lo largo del desarrollo de las iteraciones se realizará la implementación de las historias de usuario que componen cada una de estas. Se llevará a cabo una revisión del plan de iteraciones y se modificará de ser necesario. En correspondencia con el plan de desarrollo se descomponen las UH en tareas, asignándolas posteriormente a un equipo (o una persona) responsable de su implementación. Estas tareas serán usadas por los programadores, pueden ser descritas utilizando un lenguaje técnico y no necesariamente deben ser entendibles para el cliente.

Se llevarán a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose al finalizar un producto listo para su aplicación. A continuación se muestran cada una de las iteraciones.

4.1 Iteración 1.

En esta iteración se abordarán las UH de nivel de prioridad ALTA y se construirá la base de la arquitectura del sistema. Se obtendrá la primera versión con la que el cliente podrá probar las funcionalidades que considera principales en la aplicación.

Tabla 4.1 UH pertenecientes a la primera iteración.

Historias de Usuario	Estimación	Real
Gestión de Tareas	2	2
Reporte de Tiempos	1	1

Reporte de Errores	1	1
Total	4	4

Gestión de Tareas

Tabla 4.1.1 Tarea # 1 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea	
Número de Tarea: 1	Número de Historia: 1
Nombre de Tarea: Creación de Proyectos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se implementará una funcionalidad que permita la creación de nuevos proyectos (que pueden ser vistos como módulos de un proyecto en general), a estos se les asignará un nombre y un líder de proyecto.	

Tabla 4.1.2 Tarea # 2 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea	
Número de Tarea: 2	Número de Historia: 1
Nombre de Tarea: Edición de Proyectos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Luego de seleccionar un proyecto previamente creado se podrán modificar los campos: nombre y líder de proyecto.	

Tabla 4.1.3 Tarea # 3 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea



Número de Tarea: 3	Número de Historia: 1
Nombre de Tarea: Eliminación de Proyecto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se selecciona el proyecto y se elimina del sistema junto a las tareas que lo conforman y los tickets para el mismo.	

Tabla 4.1.4 Tarea # 4 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea	
Número de Tarea: 4	Número de Historia: 1
Nombre de Tarea: Creación de Tareas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se implementará una funcionalidad que permita la creación de nuevas Tareas, a estas se les concederá un nombre, proyectos a los que pertenece, y la estimación de duración de la misma.	

Tabla 4.1.5 Tarea # 5 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea	
Número de Tarea: 5	Número de Historia: 1
Nombre de Tarea: Edición de Tareas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Luego de seleccionar de un proyecto la tarea solicitada se podrán modificar los campos: nombre y estimación de duración.	

Tabla 4.1.6 Tarea # 6 perteneciente a la UH Gestión de Tareas de la Iteración 1.



Tarea	
Número de Tarea: 6	Número de Historia: 1
Nombre de Tarea: Eliminación de Tareas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Luego de seleccionar de un proyecto la tarea solicitada esta podrá ser eliminada del sistema, eliminado además los tickets generados para la tarea, así como los registros de trabajos realizados en ella.	

Tabla 4.1.7 Tarea # 7 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea	
Número de Tarea: 7	Número de Historia: 1
Nombre de Tarea: Implementación de la clase CProyecto	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se implementará la clase CProyecto para la gestión de proyectos. Esta clase contará con funcionalidades que permitirán crear, modificar o eliminar un proyecto. Brindará además otras facilidades como conocer datos de un proyecto, las tareas que lo conforman y los usuarios vinculados a él.	

Tabla 4.1.8 Tarea # 8 perteneciente a la UH Gestión de Tareas de la Iteración 1.

Tarea	
Número de Tarea: 8	Número de Historia: 1
Nombre de Tarea: Implementación de la clase CTarea	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	



Descripción: Se implementará la clase CTarea para la gestión de tareas, con funcionalidades que permitirán crear, modificar o eliminar tareas. Brindará además la posibilidad de consultar información de una tarea.

Reporte de Tiempos

Tabla 4.1.7 Tarea # 1 perteneciente a la UH Reporte de Tiempos de la Iteración 1.

Tarea	
Número de Tarea: 1	Número de Historia: 2
Nombre de Tarea: Registro de Tiempos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se almacenará en la base de datos el Proyecto, la Tarea, Hora de Inicio y Fin, Duración y Descripción del trabajo que realizó. Los campos mostrados utilizarán funciones JavaScript y Ajax para las validaciones.	

Tabla 4.1.8 Tarea # 2 perteneciente a la UH Reporte de Tiempos de la Iteración 1.

Tarea	
Número de Tarea: 2	Número de Historia: 2
Nombre de Tarea: Reporte de Tiempos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Permitirá al usuario generar un reporte de tiempos general, en el que constará con toda la información registrada en la base de datos, agrupada por proyectos, mostrando la fecha, el usuario, la tarea, la hora de inicio y fin del trabajo, y la correspondiente descripción. Este reporte puede ser personalizado según desee el usuario. Se brindará la posibilidad de exportar el reporte a un archivo con formato pdf.	

Reporte de Errores

Tabla 4.1.9 Tarea # 1 perteneciente a la UH Reporte de Errores de la Iteración 1.

Tarea	
Número de Tarea: 1	Número de Historia: 3
Nombre de Tarea: Creación de Tickets	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se seleccionarán el proyecto, la tarea y los usuarios que trabajen en dicha tarea y se detallará una descripción del error encontrado, el ticket contará además con un estado (No Tratado ó Tratado, inicialmente se crea como No Tratado), la fecha de creación y de modificación, luego el sistema informará en la página inicial a los usuarios correspondientes que se ha encontrado un error en la tarea que realizó.	

Tabla 4.1.110 Tarea # 2 perteneciente a la UH Reporte de Errores de la Iteración 1.

Tarea	
Número de Tarea: 2	Número de Historia: 3
Nombre de Tarea: Control de Errores	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Después de recibir la notificación en la página principal de la existencia de un error, el usuario verá un ticket representado en una tabla con la información referente al error cometido. Luego de haberlo corregido hará una descripción del tratamiento del error y cambiará el estado del mismo de No Tratado a Tratado.	

Tabla 4.1.11 Tarea # 3 perteneciente a la UH Reporte de Errores de la Iteración 1.

Tarea	
Número de Tarea: 3	Número de Historia: 3
Nombre de Tarea: Reporte de Errores	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: El sistema permitirá la creación de un reporte general de tickets, el cual agrupará todos los tickets reportados, hayan sido tratados o no, organizado por proyectos. Permitirá además personalizar el reporte en cuanto a usuarios, fecha, tareas y proyectos, seleccionando las filas que se desean.	

4.2. Iteración 2.

En la segunda iteración se llevará a cabo la implementación de la historia de usuario Cronograma y Estado de Proyecto, de nivel de prioridad media. Como resultado de esta iteración se obtendrá la versión 2.0 de CIPI, que podrá ser probada y mostrada al cliente.

Tabla 4.2 UH pertenecientes a la segunda iteración.

Historias de Usuario	Estimación	Real
Cronograma y Estado de Proyecto	2	2
Total	2	2

Tabla 4.2.1 Tareas #1 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 1	Número de Historia: 4
Nombre de Tarea: Definición de datos a mostrar en el cronograma	
Tipo de Tarea: Desarrollo	Puntos Estimados:



Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda
Descripción: Se seleccionarán los datos a mostrar en las cabeceras del cronograma utilizando el método <i>ShowHeaders</i> y <i>SetColTitles</i> de la clase <i>GanttGraph</i> perteneciente al módulo <i>Jpgraph</i> .

Tabla 4.2.2 Tareas #2 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 2	Número de Historia: 4
Nombre de Tarea: Selección de estilo del cronograma	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se seleccionará el estilo visual en que se mostrará el cronograma utilizando métodos de la clase <i>GanttGraph</i> perteneciente al módulo <i>Jpgraph</i> .	

Tabla 4.2.3 Tareas #3 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 3	Número de Historia: 4
Nombre de Tarea: Obtención de los proyectos relevantes para el cronograma	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Sólo se mostrarán en el cronograma los proyectos que tienen asignadas tareas.	

Tabla 4.2.4 Tareas #4 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 4	Número de Historia: 4
Nombre de Tarea: Obtención de tareas pertenecientes a los proyectos relevantes.	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se obtendrán las tareas de cada proyecto a mostrar en el cronograma además de los datos necesarios para cada una.	

Tabla 4.2.5 Tareas #5 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 2	Número de Historia: 4
Nombre de Tarea: Cálculo de datos de los proyectos.	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Partiendo de los datos obtenidos de las tareas relacionadas con cada uno de los proyectos relevantes para el cronograma se realizará por cada proyecto el cálculo de sus datos, definiendo como fecha de inicio y fin del proyecto la menor fecha de inicio y la mayor de fin entre las tareas. Se realizará el cálculo del porcentaje en el que se encuentra el proyecto.	

Tabla 4.2.6 Tareas #6 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 2	Número de Historia: 4
Nombre de Tarea: Construcción de las barras del cronograma	

Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Utilizando la clase <i>GanttBar</i> del módulo <i>Jpgraph</i> se crearán cada una de las barras ha mostrar en el cronograma, en ella se representarán los datos anteriormente obtenidos (fechas de inicio, fechas de fin, porcentos). Estas barras serán adicionadas al objeto <i>\$graph</i> a partir del cual se generará el cronograma final.	

Tabla 4.2.7 Tareas #7 perteneciente a la UH Creación de cronograma de la Iteración 2.

Tarea	
Número de Tarea: 7	Número de Historia: 4
Nombre de Tarea: Generar y mostrar el cronograma	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Se generará el cronograma final que será mostrado en una nueva ventana utilizando el método <i>Stroke</i> de la clase <i>GanttGraph</i> perteneciente al módulo <i>Jpgraph</i> .	

4.3 Iteración 3.

En la tercera iteración se llevará a cabo la implementación de las historias de usuario Control de Recursos Materiales, Gestión de Recursos Humanos y Distribución de Recursos, de nivel de prioridad baja. Como resultado de esta iteración se obtendrá la versión 1.0 de CIPI, que podrá ser probada, mostrada y finalmente entregada al cliente.

Tabla 4.3 UH pertenecientes a la tercera iteración.

Historias de Usuario	Estimación	Real
Control de Recursos Materiales	1	1

Gestión de Recursos Humanos	1	1
Distribución de Recursos	1	1
Total	3	3

Control de Recursos Materiales

Tabla 4.3.1 Tarea # 1 perteneciente a la UH Control de Recursos Materiales de la Iteración 3.

Tarea	
Número de Tarea: 1	Número de Historia: 5
Nombre de Tarea: Mostrar Puestos Existentes	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: El sistema mostrará una tabla con todos lo puestos que hayan sido creados con anterioridad, indicando cual es el estado actual del puesto de trabajo (Disponible ó No Disponible) y si es o no servidor de aplicaciones.	

Tabla 4.3.2 Tarea # 2 perteneciente a la UH Control de Recursos Materiales de la Iteración 3.

Tarea	
Número de Tarea: 2	Número de Historia: 5
Nombre de Tarea: Creación de Puestos de Trabajo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: El sistema permitirá crear nuevos puestos de trabajo, para esto se indicará el número del puesto, el estado y se seleccionará si es servidor de	

aplicaciones.

Tabla 4.3.3 Tarea # 3 perteneciente a la UH Control de Recursos Materiales de la Iteración 3.

Tarea	
Número de Tarea: 3	Número de Historia: 5
Nombre de Tarea: Edición de Puestos de Trabajo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: La página EditarPuesto.php permitirá seleccionar el número de un puesto de trabajo y hacer cambios en los atributos con que cuenta este (número, si es servidor de aplicaciones y estado del puesto).	

Tabla 4.3.4 Tarea # 4 perteneciente a la UH Control de Recursos Materiales de la Iteración 3.

Tarea	
Número de Tarea: 4	Número de Historia: 5
Nombre de Tarea: Eliminación de Puestos de Trabajo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: La página EliminarPuesto.php permitirá eliminar un puesto de trabajo del listado de puestos con que cuenta el proyecto, para esto se seleccionará el número del puesto que se desea eliminar.	

Gestión de Recursos Humanos

Tabla 4.3.5 Tarea # 1 perteneciente a la UH Gestión de Recursos Humanos de la Iteración 3.

Tarea	
Número de Tarea: 1	Número de Historia: 6
Nombre de Tarea: Mostrar Usuarios del Sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: El sistema permitirá mediante la página UsuarioSistema.php mostrar el listado de todos los usuarios registrados en él, además de estos se conocerá su rol y en los proyectos en los que trabaja.	

Tabla 4.3.6 Tarea # 2 perteneciente a la UH Gestión de Recursos Humanos de la Iteración 3.

Tarea	
Número de Tarea: 2	Número de Historia: 6
Nombre de Tarea: Creación de Usuarios del Sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: El líder de proyecto junto a los colíderes podrán crear nuevos usuarios del sistema mediante la página CrearUsuario.php, de estos se validará que pertenezcan al dominio uci.cu con una función ajax mientras se está introduciendo el usuario. Del usuario se registra en el sistema el nombre de usuario, proyectos en los que trabajará y el rol que tiene para la interacción con la aplicación.	

Tabla 4.3.7 Tarea # 3 perteneciente a la UH Gestión de Recursos Humanos de la Iteración 3.

Tarea



Número de Tarea: 3	Número de Historia: 6
Nombre de Tarea: Edición de Datos de Usuarios de Sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.3
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: Cuando exista un error en los datos del usuario guardados en el sistema, o se desee modificar estos por otra razón, a través de la página EditarUsuario.php se podrán hacer las modificaciones deseadas. Esta página contará con un combobox que permitirá seleccionar el usuario y un campo para reemplazar el nombre del mismo, además se le brindará la posibilidad de escoger el (los) proyectos y el rol.	

Tabla 4.3.8 Tarea # 4 perteneciente a la UH Gestión de Recursos Humanos de la Iteración 3.

Tarea	
Número de Tarea: 4	Número de Historia: 6
Nombre de Tarea: Eliminación de Usuarios del Sistema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.2
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: En la página EliminarUsuario.php el sistema dará la posibilidad de eliminar los usuarios que no pertenezcan a los proyectos. Esta contará con un combobox para hacer la selección del usuario que se desea eliminar.	

Distribución de Recursos

Tabla 4.3.9 Tarea # 1 perteneciente a la UH Distribución de Recursos de la Iteración 3.

Tarea



Número de Tarea: 1	Número de Historia: 7
Nombre de Tarea: Mostrar asignación de Recursos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.4
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: En AsignacionRecursos.php los usuarios del sistema podrán ver la planificación de los tiempos de máquina a través de una tabla que contendrá los días y los puestos de trabajo que le han sido asignados en la semana, se podrá acceder a cualquiera de las tres sesiones de trabajo (Mañana, Tarde y Noche) mediante los botones de la parte inferior de la página.	

Tabla 4.3.10 Tarea # 2 perteneciente a la UH Distribución de Recursos de la Iteración 3.

Tarea	
Número de Tarea: 2	Número de Historia: 7
Nombre de Tarea: Asignación de Recursos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.6
Responsable: José Pablo Julián Espada – Migdiell Reyes Pereda	
Descripción: El líder de proyecto, los colíderes y el planificador podrán llevar a cabo la planificación de los tiempos de máquina en la página AsignacionRecursos.php, estos tendrán acceso a un formulario que agrupa cuatro combobox: día (para hacer selección del día de la semana que será planificado), sesión (Mañana, Tarde o Noche), puesto (puestos de trabajo) y usuario. Así se irán planificando todos los tiempos de máquina de la semana, según entre un usuario se actualizará la tabla de la planificación.	

4.4 Tareas Generales de la implementación.

Para la implementación del sistema se llevan a cabo una serie de tareas que no se encuentran comprendidas en las Historias de Usuarios descritas por el cliente y que se han definido como tareas generales a realizar. Con la ejecución de las tareas generales se cumplen requisitos indispensables en el correcto funcionamiento del sistema, garantizando el control de la autenticación en la aplicación, la navegabilidad y la interacción de los usuarios con el mismo.

4.4.1 Tarea # 1 perteneciente a las tareas generales de la implementación

Tarea	
Número de Tarea: 1	Número de Historia:
Nombre de Tarea: Creación de la Base de Datos	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada	
Descripción: Utilizando MySQL como gestor, se crea la base de datos dp_cipi en la cual se almacenarán los datos persistentes generados durante la utilización de la aplicación.	

4.4.2 Tarea # 2 perteneciente a las tareas generales de la implementación

Tarea	
Número de Tarea: 2	Número de Historia:
Nombre de Tarea: Implementación de la clase abstracta Modelo	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada	
Descripción: Se creará una clase abstracta para la construcción de consultas a la base de datos. Esta clase posibilitará migrar en caso de que se necesite a otros gestores de bases de datos realizando el mínimo de cambios posibles en el código de la aplicación.	

4.4.3 Tarea # 3 perteneciente a las tareas generales de la implementación

Tarea	
Número de Tarea: 3	Número de Historia: 2
Nombre de Tarea: Implementación de la clase Controlador	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada	
Descripción: Esta clase se encargará del control general de la secuencia de actividades durante la interacción de un usuario con el sistema. Colaboran con esta las clases: CTicket, CUsuario, CProyecto, CTarea, CPuestoTrabajo; esta colaboración permite la gestión de los datos mostrados al usuario e introducidos por él. Además muestra las vistas correspondientes en cada petición.	

4.4.4 Tarea # 4 perteneciente a las tareas generales de la implementación

Tarea	
Número de Tarea: 4	Número de Historia:
Nombre de Tarea:	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: José Pablo Julián Espada	
Descripción: Para la validación de usuarios del dominio se utilizará la librería LDAP.php y del módulo Autenticator v201, con el fin de asegurarse que los usuarios que accedan al sistema pertenezcan al dominio uci.cu y estos hayan sido incluidos como usuarios del sistema con anterioridad.	

4.4.5 Tarea # 5 perteneciente a las tareas generales de la implementación

Tarea	
Número de Tarea: 5	Número de Historia:



Nombre de Tarea: Diseño de Interfaz Web.	
Tipo de Tarea: Desarrollo	Puntos Estimados:
Responsable: Migdiell Reyes Pereda	
Descripción: Se diseñará una interfaz web no funcional, haciendo utilización del lenguaje HTML, creando las vistas necesarias para que lo usuarios interactúen con el sistema pudiendo mostrar y entrar de datos.	

4.5 Módulos y Librerías Utilizadas en la Implementación.

La reutilización de código es el comportamiento y las técnicas que garantizan que una parte o la totalidad de un programa informático existente se pueda emplear en la construcción de otro programa. De esta forma se aprovecha el trabajo anterior, se economiza tiempo, y se reduce la redundancia.

La manera más fácil de reutilizar código es copiarlo total o parcialmente desde el programa antiguo al programa en desarrollo. Pero es trabajoso mantener múltiples copias del mismo código, por lo que en general se elimina la redundancia dejando el código reusable en un único lugar, y llamándolo desde los diferentes programas. Este proceso se conoce como abstracción. La abstracción puede verse claramente en las librerías, módulos y bibliotecas de software, en las que se agrupan varias operaciones comunes a cierto dominio para facilitar el desarrollo de programas nuevos.

En el epígrafe actual se describen las librerías y módulos utilizados para la implementación del sistema propuesto.

4.5.1 Módulo Autenticator v2.01.

El módulo Autenticator es un script⁸ para la Gestión de Páginas restringidas sólo a Usuarios registrados, con nivel de acceso y gestión de errores en el momento de logueo. Es ideal para zonas de Administración o zonas "sólo usuarios" que requieran autenticación para acceder a las mismas. Incluye administración de usuarios (altas/bajas/modificaciones) de ejemplo. Este gestor de usuarios está programado en lenguaje PHP, usa motor de Base de datos Mysql y se basa en el uso de sesiones; los cuales deben estar habilitados en el servidor que se instale. Desde su creación hasta

⁸ Debe su nombre a que ha sido programado en un lenguaje de tipo script, para dar funcionalidades a las páginas web.

la actualidad ha contado con varias versiones que han ido siendo mejoradas hasta la obtención de la versión actual. Autenticator está publicado bajo licencia GPL⁹.

4.5.2 Módulo Jpgraph.

La creación de gráficas a partir de datos es una habilidad que debe ser dominada por un profesional de la web. Cuando se habla de gráficas se refiere a todo tipo de imágenes que sirvan para representar datos, como gráficas de barras, de líneas de progreso, de pastel¹⁰, entre otras. Obviamente, la creación de gráficas no es un tema trivial, sino que requerirá de una gran dosis de dedicación y esfuerzo. Las gráficas, que generalmente se muestran en imágenes, pueden ser de distintos tipos y sólo el hecho de tratar de dibujar en una imagen líneas, barras o incluso gráficos de pastel en tres dimensiones, puede ser sobradamente complicado. Sin embargo, existen sistemas como JPGraph, que pueden facilitar la tarea de una manera muy interesante, pues ofrecen mecanismos para la generación de las imágenes con las gráficas, de modo que sólo hay que centrarse en cargar los datos a representar y escoger el tipo de gráfica que se desea visualizar.

JPGraph es una librería que incluye un conjunto de clases (código orientado a objetos) que sirven para crear imágenes con todo tipo de gráficas, dinámicamente desde PHP. El sistema está muy depurado¹¹ y soporta multitud de funcionalidades, por lo que seguramente se puede encontrar la solución a casi cualquier necesidad en el ámbito de creación de gráficas. Además, la mayoría de las configuraciones de las gráficas vienen con opciones por defecto, así que resulta bastante sencillo obtener resultados rápidamente.

Algunas de las características que presenta el sistema se mencionan a continuación:

- Reducido peso en bytes de las imágenes resultantes.
- Soporte a las librerías GD1¹² o GD2.
- Uso de la Interpolación matemática para obtener curvas a partir de unos pocos valores.

⁹ Licencia que permite la libre utilización del código, libre de costos y autoriza la modificación de este.

¹⁰ Gráfico que hace representación de forma circular de un todo (*entiendase como 100% de algo*) descompuesto en varias partes que lo conforman.

¹¹ Adj. Pulido, trabajado, elaborado cuidadosamente.

¹² GD es una biblioteca de código fuente abierto para la creación dinámica de imágenes.



- Diversos tipos de gráficas 2D o 3D, como de puntos, líneas, gráficos de pastel, barras, cajas, etc.
- Escalas flexibles tanto en el eje X como el Y, que se ajustan al juego de datos que se tenga que representar.
- Soporte para generar gráficas con varios juegos de valores a la vez.
- Configurable con distintos tipos de colores, leyendas, tipografías, imágenes de fondo, etc.

Esta librería será utilizada en la implementación de la UH número cuatro durante la iteración 2, para generar el diagrama de Gantt que mostrará la duración de todas las tareas pertenecientes a los proyectos y la duración general del proyecto.

4.5.3 LDAP.

LDAP (Lightweight Directory Access Protocol), (Protocolo Ligero de Acceso a Directorios) es un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. LDAP también es considerado una base de datos (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

Un directorio es un conjunto de objetos con atributos organizados en una manera lógica y jerárquica. El ejemplo más común es el directorio telefónico, que consiste en una serie de nombres (personas u organizaciones) que están ordenados alfabéticamente, con cada nombre teniendo una dirección y un número de teléfono adjuntos.

Un árbol de directorio LDAP a veces refleja varios límites políticos, geográficos y/o organizacionales, dependiendo del modelo elegido. Los despliegues actuales de LDAP tienden a usar nombres de Sistema de Nombres de Dominio (DNS por sus siglas en inglés) para estructurar los niveles más altos de la jerarquía. Conforme se desciende en el directorio pueden aparecer entradas que representan personas, unidades organizacionales, impresoras, documentos, grupos de personas o cualquier cosa que representa una entrada dada en el árbol (o múltiples entradas).

Habitualmente, almacena la información de autenticación de un dominio (usuario y contraseña) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red,

permisos, certificados, etc). De forma sintetizada se puede decir que LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red

4.6 Pruebas.

La fase de Prueba es uno de los pilares que sostienen la metodología XP. Durante esta etapa serán probados todos los componentes del producto desarrollado, tanto por el cliente como por el equipo de desarrolladores. Las pruebas realizadas en algunos casos generarán artefactos que serán guardados por el equipo de desarrollo para futuras mejoras del producto.

XP propone, debido a la cercanía de los desarrolladores con el cliente la realización de dos tipos de pruebas fundamentales: Unitarias y de Aceptación.

4.6.1 Pruebas Unitarias.

A lo largo de la implementación de un sistema, cada uno de los desarrolladores tiene que ir probando constantemente lo que va obteniendo para garantizar que las funcionalidades exigidas por el cliente estén siendo implementadas correctamente. A estas pruebas se le llama Pruebas Unitarias. Aunque estas no generan artefactos y no son directamente palpables para el cliente, son de vital importancia para el desarrollo de un proyecto.

4.6.2 Pruebas de Aceptación.

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. En estas serán probadas las funcionalidades exigidas por el cliente y descritas en las historias de usuario, además de los aspectos de seguridad requeridos. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- *Código*: servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- *UH*: tendrá el nombre de la historia de usuario a la que hace referencia la prueba a realizar.
- *Nombre*: nombre que se le da a la prueba a realizar.
- *Descripción*: se describe la funcionalidad que se desea probar.

- *Condiciones de Ejecución:* mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- *Entradas / Pasos de Ejecución:* se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- *Resultado esperado:* se hará una breve descripción del resultado que se espera obtener con la prueba realizada.
- *Evaluación de la prueba:* acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultado que a continuación se describen:
 - *Bien:* cuando el resultado de la prueba es exactamente el esperado por el usuario.
 - *Parcialmente bien:* cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
 - *Mal:* cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la UH.

Las pruebas de aceptación se llevarán a cabo siguiendo los pasos que se muestran a continuación:

1. Se redactarán los casos de prueba teniendo en cuenta el orden de las UHs y los niveles de prioridad dados a las funcionalidades.
2. Se hará la planificación con el cliente de cuándo y cuáles pruebas serán llevadas a cabo.
3. Se reunirán los miembros del proyecto seleccionados para realizar las pruebas.
4. Se completarán cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba.
5. Los artefactos generados como fruto de las pruebas de aceptación podrán ser vistos en los Anexos del presente trabajo. [Anexo 1 Pruebas de Aceptación].

Conclusiones

Las fases de Implementación y Prueba sin lugar a dudas son las que cierran con broche de oro el desarrollo de una aplicación siguiendo la metodología XP, los artefactos generados son de vital importancia para comprobar si ha sido satisfactorio el trabajo realizado a lo largo de la implementación.

En el presente capítulo se expusieron los artefactos elaborados como resultado de las actividades realizadas en las fases de Implementación y Prueba.

Capítulo V. Estudio de Factibilidad

Introducción

La rentabilidad, estimación de esfuerzo y costo de un proyecto de software son sin lugar a dudas una de las tareas de mayor importancia para la producción de software. Debido a que cada vez es mayor la necesidad de obtener datos objetivos que permitan evaluar, predecir y mejorar la calidad del software así como el tiempo y coste de desarrollo del mismo.

5.1 Modelo matemático COCOMO II

COCOMO, propuesto y desarrollado por Barry Boehm, es uno de los modelos de estimación de costos mejor documentados y utilizados. El modelo permite determinar el esfuerzo y tiempo que se requiere en un proyecto de software a partir de una medida del tamaño del mismo expresada en el número de líneas de código que se estimen generar para la creación del producto software.

Consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (*SLOC*, *Source Lines of Code*) estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo (*cost drivers*) que influyen en el esfuerzo requerido para el desarrollo del software.

Está compuesto por tres modelos que se adaptan tanto a las necesidades de los diferentes sectores descritos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

El estudio de factibilidad de este sistema se realizó mediante el modelo de diseño temprano que se utiliza en las primeras etapas del desarrollo, en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, se conoce muy poco del tamaño del producto a ser desarrollado, de la naturaleza de la plataforma, del personal a ser incorporado al proyecto o aspectos específicos del proceso a utilizar. Ese nivel de detalle en este modelo es consistente con el nivel general de información disponible y con el nivel general de estimación detallada que es necesaria en estas etapas, lo que concuerda con el uso de Puntos de

Función. Para estimar tamaño usa Puntos de Función No Ajustados como métrica de medida y un número reducido de factores de costo.

5.2 Características del Proyecto

El primer paso a llevar a cabo para la estimación del proyecto consiste en la obtención de los Puntos de Función desajustados, los cuales están dados por la suma de cada una de las entradas, las salidas y las consultas externas del sistema, así como los archivos lógicos internos y de interfaz externos. A continuación se muestran cada una de estas características aplicadas al software en cuestión.

5.2.1 Entradas externas

Se definen como un proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro. En el caso particular de la aplicación propuesta se cuenta con siete entradas externas, especificadas en la siguiente tabla:

Tabla 5.2.1 Entradas Externas.

Nombre de la Entrada Externa	Cantidad de ficheros	Cantidad de elementos	Clasificación (simple, media y compleja)
Adicionar proyecto	1	2	Simple
Adicionar Tarea	1	3	Simple
Registro de Tiempo	1	3	Simple
Adicionar Usuario	1	3	Simple
Adicionar Puesto de Trabajo	1	3	Simple
Planificación y Asignación de Recursos Materiales	1	3	Simple
Adicionar Ticket	1	1	Simple

Total		18	
--------------	--	-----------	--

5.2.2 Salidas Externas

Se definen como un proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (esto es, datos que se calculan a partir de otros datos) cruzan la frontera del sistema desde adentro hacia afuera. Las salidas externas vinculadas al proyecto se describen a continuación.

Tabla 5.2.2 Salidas Externas

Nombre de la Salida Externa	Cantidad de ficheros	Cantidad de elementos	Clasificación (simple, media y compleja)
Reporte de Tiempo	1	6	Simple
Reporte de Ticket	1	1	Simple
Mostrar Planificación			
Cronograma	1	7	Simple
Total		14	

5.2.3 Consultas Externas

Se definen como un proceso elemental con componentes de entrada y de salida donde un Actor del sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos).

Tabla 5.2.3 Consultas Externas

Nombre de la Petición	Cantidad de Ficheros	Cantidad de elementos	Clasificación (simple, media y compleja)
LDAP	1	1	Simple
Total		1	

5.2.4 Archivos Lógicos Internos

Constituyen un grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de entradas externas.

Tabla 5.2.4 Archivos Lógicos Internos

Nombre del fichero interno	Cantidad de records	Cantidad de elementos de datos	Clasificación (simple, media y compleja)
BD db_cipi	1	6	Simple
Total		6	

5.2.4 Archivos de Interfaz Externos

Son un grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es un Archivo Lógico Interno de otra aplicación. En este caso no fueron identificados Archivos de interfaz externa.

Tabla 5.2.4 Archivos de Interfaz Externos

Nombre de la	Cantidad de	Cantidad de	Clasificación
--------------	-------------	-------------	---------------

entrada externa	records	elementos de datos	(simple, media y compleja)
Total		0	

5.2.5 Puntos de Función Desajustados

La siguiente tabla está basada en las características del sistema anteriormente expuestas, el producto de la cantidad existente de cada una de ellas y el peso correspondiente a las mismas dan como resultado final, los puntos de función desajustados pertenecientes al proyecto.

Tabla 5.2.5 Puntos de Función Desajustados

Elementos	Simple		Medio		Complejo		Subtotal
	No.	Peso	No.	Peso	No.	Peso	
Entrada Externa	18	3	0	4	0	6	54
Salida Externa	14	4	0	5	0	7	56
Consultas Externas	1	3	0	4	0	6	3
Fichero Lógico Interno	6	7	0	10	0	15	42
Fichero interfaz Externa	0	5	0	7	0	10	0
Total (UFP)							155

5.3 Cálculo de Instrucciones Fuentes, Esfuerzo, Tiempo de Desarrollo, Cantidad de Hombres y Costo.

5.3.1 Cálculo de Instrucciones Fuentes.

Luego de obtener la cantidad total de puntos de función desajustados pertenecientes al proyecto, se procede a calcular la cantidad de instrucciones fuentes del mismo, para lo cual se utiliza la siguiente ecuación:

$$SLOC = UFP \times ratio$$

Donde

SLOC: Cantidad de instrucciones fuentes. (9145)

UFP: Puntos de función desajustados. (155)

Ratio: Conversión de puntos de función desajustados a líneas de código para el lenguaje PHP. (59)

Partiendo de la ecuación anterior se obtiene el siguiente resultado:

$$SLOC = 155 \times 59$$

$$SLOC = 9145 = 9.1 \text{ KSLOC}$$

Las características y valores obtenidos anteriormente han sido plasmados en la siguiente tabla:

Tabla 5.3.1 Características del Sistema

Características	Valor
Puntos de función desajustados	155
Lenguaje (PHP)	44
Instrucciones fuentes por puntos de función	6820 SLOC
Instrucciones fuentes	6.8 KSLOC

5.4 Cálculo de Esfuerzo Nominal

Posteriormente se procede al cálculo del esfuerzo nominal, ecuación que se toma como base tanto en el método de diseño preliminar, al cual se hace alusión anteriormente, como en el modelo Post arquitectura, ambos definidos por COCOMO II.

$$PM_{nominal} = A \times (Size)^B$$

$$PM_{nominal} = 2.94 \times 6.8^{0.98}$$

$$PM_{nominal} = 19.2 \text{ meses/hombre}$$

Donde

$PM_{nominal}$: Esfuerzo nominal requerido en meses – hombre

Size: Tamaño estimado del software, en Puntos de Función sin ajustar (KSLOC)

A: constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. (Valor: 2.94).

B: Constante denominada Factor escalar y su valor esta dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

$$B = 0.91 + 0.01 \times \sum(W_i)$$

$$B = 0.91 + 0.01 \times 7.74$$

$$B = 0.98$$

Donde

W_i : Variables escalares que indican las características que el proyecto presenta en lo que a su complejidad y entorno de desarrollo se refiere. La siguiente tabla muestra los valores asignados a cada una de estas variables:

Tabla 5.3.2 Factores de Escala

Nombre	Valor	Justificación
PREC	1.24	Existen varios proyectos similares a nivel internacional, pero no se encontró ninguno a nivel nacional.
FLEX	1.01	Cuenta con alta flexibilidad en cuanto a los requerimientos establecidos inicialmente.
TEAM	1.10	El equipo de desarrollo presenta una alta cohesión.

RESL	2.83	Se identificaron riesgos críticos en un porcentaje elevado.
PMAT	1.56	Se cuenta con la experiencia necesaria como para que el software cumpla con las funcionalidades requeridas
Total (SF)	7.74	

5.4.1 Ajuste del Esfuerzo Nominal

El esfuerzo calculado anteriormente es un valor nominal y debe ser ajustado en base a las características del proyecto para lo cual se tiene un conjunto de Multiplicadores de esfuerzo (ME_i) que representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software (2).

Tabla 5.3.3 Multiplicadores de Esfuerzo

Nombre	Valor	Justificación
RCPX	1.74	La complejidad del sistema es alta
RUSE	1.00	Pretende reutilizarse solo una parte del código
PDIF	1.00	Uso de memoria y almacenamiento normal, plataforma estable
PREX	1.12	Baja experiencia del personal en la utilización de lenguaje, así como de la plataforma
PERS	1.00	La capacidad del personal es alta
FCIL	1.00	La utilización de entornos de desarrollo integrados, así como de herramientas CASE simples facilitan en gran medida el trabajo.
SCED	1.00	El sistema se desarrolló en el tiempo establecido
Total (EM)	1.95	

$$PM_{ajustado} = PM_{nominal} \times \prod (ME_i)$$

$$PM_{ajustado} = 19.2 \text{ meses/hombre} \times 1.95$$

$$PM_{ajustado} = 37.44 \text{ hombres/mes}$$

5.4.1.1 Calculo del Tiempo de Desarrollo del Software

El tiempo requerido para el desarrollo del proyecto está dado por la siguiente ecuación:

$$TDEV = C \times (PM_{ajustado})^F$$

$$TDEV = 3.64 \times 37.44^{0.25}$$

$$TDEV = 9.00$$

Donde

C: constante con valor 3.64

$$F = D + 0.2 \times 0.01 \times \sum SF$$

$$F = 0.24 + 0.2 \times 0.01 \times 7.74$$

$$F = 0.25$$

Donde

D: constante cuyo valor es 0.24

SF: valor de los factores de escala

5.4.2 Cálculo del Costo Total del Proyecto

Para el cálculo del costo total correspondiente al proyecto en cuestión, COCOMO II define la siguiente ecuación:

$$C = CHM \times PM$$

$$C = 200 \times 19.2$$

$$C = 3840$$

Donde

C: costo total

CHM: costo teniendo en cuenta salario de todos los obreros, el cual es calculado por la ecuación

$$CHM = CH \times sal$$

$$CHM = 2 \times 100$$

$$CHM = 200$$

Donde

Sal: Salario medio por cada trabajador

CH : Cantidad de personas destinadas al proyecto, lo que es calculado a través de la ecuación:

$$CH = PM/TDEV$$

$$CH = 19.2/9.00$$

$$CH \approx 2$$

5.5 Resultados

La siguiente tabla muestra los resultados obtenidos luego de haber realizado todo el análisis correspondiente y haber efectuado todos los cálculos para determinar el costo y esfuerzo requeridos por el proyecto.

Tabla 5.4.1 Resultados

Cálculo de:	Valor
Esfuerzo	19.2 hombres/mes
Tiempo de desarrollo	6 meses
Cantidad de hombres	2 hombres

Salario medio	100 pesos
Costo	3840 pesos

Conclusiones

Todo producto conlleva en si un costo, este sólo es justificado cuando al final de la etapa de producción el fruto del producto supera con creces lo que haya costado producirlo. En el caso de la aplicación propuesta se puede concluir que es factible ya que sólo aparece con gasto significativo el salario de los desarrolladores.

Conclusiones

Con la presente investigación se ha obtenido la aplicación web CIPI en su primera versión, en la cual se cumplieron todos los objetivos específicos planteados inicialmente. Con el desarrollo de esta aplicación se pretende facilitar el control de los proyectos informáticos en la Universidad de las Ciencias Informáticas y en especial en el proyecto TIP de la Facultad 2. Se obtuvo una herramienta web capaz de unir las funcionalidades de gestión y reporte de tareas y tiempos, gestión, control y reporte de errores, gestión del cronograma del proyecto así como la gestión de los recursos materiales y humanos, y la distribución y planificación de dichos recursos.

El desarrollo de la aplicación ha brindado la posibilidad de:

- Conocer un nuevo lenguaje de desarrollo de aplicaciones web de altas potencialidades como el PHP.
- Conocer y emplear una metodología de desarrollo de software diferente a las estudiadas anteriormente.
- Mejorar el proceso de control de los proyectos en la Universidad, aportando una herramienta que permite agilizar el mismo.

Recomendaciones

Se recomienda:

A la dirección de la facultad:

- Desarrollar otro perfil de tesis donde a partir de esta primera versión de la aplicación CIPI, se le incluyan otras funcionalidades como el gestor documental, que permitirá gestionar a través de la aplicación toda la documentación generada por los proyectos; y el control de versiones, posibilitando el desarrollo concurrente y la mantención de la historia del código fuente generado durante el desarrollo del proyecto.

A los desarrolladores de futuras versiones:

- Construir un manual de usuario para la preparación del personal que trabajará con la aplicación.
- Aplicar patrones de diseño e implementación que mejoren la calidad y el rendimiento del sistema en sus futuras versiones.
- Investigar sobre la capacidad de la aplicación para ser desplegada en toda la universidad.
- Añadir a la aplicación funcionalidades como:
 1. Agregar al sistema de avisos o tickets una funcionalidad que permita informar al líder de proyecto de incumplimientos que puedan ser detectados por el sistema, como ausentarse a las secciones de trabajo planificadas.
 2. Generar la planificación del tiempo de máquina de manera automática.
 3. Permitir el acceso a la aplicación sólo desde el área de trabajo.

A los usuarios del sistema:

- Emitir sugerencias constructivas que mejoren la interacción de los usuarios con las interfaces del sistema.
- Proponer, de ser necesario, nuevas funcionalidades que se deban tener en cuenta para futuras versiones además de las descritas en este documento.

Bibliografía:

1. **Edgewall Software.** Welcome to the Trac Open Source Project. *Welcome to the Trac Open Source Project.* [En línea] Edgewall Software, 2005.
<http://trac.edgewall.org/>.
2. **L., T.S.U Ronald Y. Rodríguez.** Manual del DotProject, Rol Administrador. *Manual del DotProject, Rol Administrador.* [En línea]
http://www.sic.inep.gov.br/index2.php?option=com_docman&task=doc_view&gid=489&Itemid=49.
3. **WR Consulting.** About WR Time Tracker. *About WR Time Tracker.* [Online]
<http://wrconsulting.com/cms/wrtimetracker/>.
4. **Monografías.** Definición arquitectura cliente servidor. *Definición arquitectura cliente servidor.* [En línea] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml?monosearch> .
5. **ALFARO, F.M.** Cliente/Servidor. *Cliente/Servidor.* [Online]
<http://www.inei.gob.pe/web/metodologias/attach/lib616/CAP0302.HTM>.
6. **hospedaje.cinfonet.com.** Apache. *Apache.* [Online]
<http://hospedaje.cinfonet.com/?web=apache>.
7. **AUTORES, C. D.** Manual de PHP. *Manual de PHP.* [Online] 2008.
<http://es.php.net/manual/es/introduction.php>.
8. **adictos, PHP.** Historia de PHP. *Historia de PHP.* [Online]
<http://www.phpadictos.com/?secc=leer&id=62>.
9. **PÉREZ, Ivan Nieto.** Introducción a Java Script. *Introducción a Java Script.* [Online] 2006. <http://www.elcodigo.net/tutoriales/javascript/javascript1.html>.
10. Why is MySQL. *Why is MySQL.* [Online] <http://www.mysql.com/why-mysql/>.
11. **COMPTIS, Doralis de la Cruz – MONTALVO, Giorbis Miguel Lorié.** Sistema automatizado para la gestión de información de los cursos optativos. *Sistema automatizado para la gestión de información de los cursos optativos.* [Online] 2007.
http://bibliodoc.uci.cu/TD/TD_0354_07.pdf.

Anexos

Anexo 1 Pruebas de Aceptación

Prueba de Aceptación	
Código: 00_P1	UH: Autenticación de usuario.
Nombre: Autenticación válida de usuario.	
Descripción: Prueba para comprobar la funcionalidad autenticación de usuarios con datos válidos.	
Condiciones de Ejecución: Los datos de autenticación son correctos.	
Entrada / Pasos de Ejecución: Se accede al sistema. Se insertan los datos del usuario para la autenticación. Se intenta la autenticación.	
Resultado Esperado: Se autentica el usuario. Se le permite al usuario acceder al sistema con los privilegios que su rol le permite. No se producen errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: 00_P2	UH: Autenticación de usuario.
Nombre: Autenticación no válida de usuario.	
Descripción: Prueba para comprobar la funcionalidad autenticación de usuarios con	

datos inválidos.
Condiciones de Ejecución: Los datos de autenticación son incorrectos.
Entrada / Pasos de Ejecución: Se accede al sistema. Se insertan los datos del usuario para la autenticación. Se intenta la autenticación.
Resultado Esperado: Se vuelve a la pagina de logueo mostrando el error detectado No se producen errores.
Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH1_P1	UH: 1. Gestión de Tareas
Nombre: Creación de un nuevo proyecto con datos válidos.	
Descripción: Prueba para comprobar la funcionalidad de crear un nuevo proyecto.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción. Se utilizarán datos válidos para insertar el proyecto.	
Entrada / Pasos de Ejecución: Se intenta insertar un proyecto con datos validos.	
Resultado Esperado:	

El proyecto se insertar sin errores.

El sistema regresa a la vista de proyectos donde se muestra además de los ya existentes con anterioridad el nuevo proyecto creado.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH1_P2

UH: 1. Gestión de Tareas.

Nombre: Edición de los datos de un proyecto ya existente.

Descripción: Prueba para comprobar la funcionalidad de editar los datos de un proyecto ya existente en el sistema.

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción.

El proyecto debe estar registrado en el sistema.

Se utilizarán datos válidos para editar el proyecto.

Entrada / Pasos de Ejecución:

Se intenta seleccionar el proyecto que se editará.

Se intenta realizar los cambios al proyecto con datos válidos.

Resultado Esperado:

Se muestra la vista con los datos del proyecto seleccionado para editar.

Los datos del proyecto son modificados sin errores.

El sistema regresa a la vista de proyectos.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH1_P3	UH: 1. Gestión de Tareas
Nombre: Eliminación de un proyecto	
Descripción: Prueba para comprobar la funcionalidad de eliminar un proyecto.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción. El proyecto debe estar registrado en el sistema.	
Entrada / Pasos de Ejecución: Se intenta eliminar un proyecto.	
Resultado Esperado: El proyecto es eliminado sin errores. Las tareas pertenecientes al proyectos son eliminados sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: UH1_P4	UH: 1. Gestión de Tareas
Nombre: Creación de un nuevo proyecto con datos no válidos	
Descripción: Se intentará crear un proyecto ya existente o con datos inválidos.	
Condiciones de Ejecución: El proyecto ya existe o los datos no son válidos.	
Entrada / Pasos de Ejecución: Se ejecutan los pasos para crear un nuevo proyecto.	
Resultado Esperado:	

Se produce un error que es notificado al usuario

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH1_P5

UH: 1. Gestión de Tareas

Nombre: Creación de una nueva tarea con datos válidos.

Descripción: Prueba para comprobar la funcionalidad de crear una nueva tarea y vincularla a proyectos ya existentes.

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción.

Se utilizarán datos válidos para insertar la tarea.

Entrada / Pasos de Ejecución:

1. Se intenta insertar una tarea con datos validos y vincularla a los proyectos seleccionados.

Resultado Esperado:

1. La tarea se insertar sin errores.
1. El sistema regresa a la vista de tareas donde se muestra además de las ya existentes con anterioridad la nueva tarea creada.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH1_P6

UH: 1. Gestión de Tareas.

Nombre: Edición de los datos de una tarea ya existente.
Descripción: Prueba para comprobar la funcionalidad de editar los datos de una tarea ya existente en el sistema.
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción. La tarea debe estar registrada en el sistema. Se utilizarán datos válidos para editar la tarea.
Entrada / Pasos de Ejecución: Se intenta seleccionar la tarea que se editará. Se intenta realizar los cambios a la tarea con datos válidos.
Resultado Esperado: Se muestra la vista con los datos de la tarea seleccionada para editar. Los datos de la tarea son modificados sin errores. El sistema regresa a la vista de tareas.
Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH1_P7	UH: 1. Gestión de Tareas
Nombre: Eliminación de una nueva tarea	
Descripción: Prueba para comprobar la funcionalidad de eliminar una nueva tarea.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción. La tarea debe estar registrada en el sistema.	

Entrada / Pasos de Ejecución:

Se intenta eliminar una tarea.

Resultado Esperado:

La tarea es eliminada sin errores.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH1_P8	UH: 1. Gestión de Tareas
Nombre: Creación de una nueva tarea con datos no válidos	
Descripción: Se intentará crear una tarea ya existente o con datos inválidos.	
Condiciones de Ejecución: La tarea ya existe o los datos no son válidos.	
Entrada / Pasos de Ejecución:	
Se ejecutan los pasos para crear una nueva tarea.	
Resultado Esperado:	
Se produce un error que es notificado al usuario	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: UH2_P1	UH: 2. Reporte de Tiempo
Nombre: Registro de tiempo de trabajo	
Descripción: Prueba para comprobar la funcionalidad registrar el tiempo de trabajo de	

un usuario del sistema en una tarea determinada

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario.

La tarea y el proyecto deben estar registrados en el sistema.

Entrada / Pasos de Ejecución:

Se intenta seleccionar el proyecto en el que se estará trabajando.

Se intenta seleccionar la tarea en la que se estará trabajando.

Se intenta marcar la hora de inicio de trabajo en la tarea.

Se intenta marcar la hora de fin de trabajo en la tarea.

Se intenta registrar los datos en el sistema.

Resultado Esperado:

Se muestran las tareas correspondientes al proyecto seleccionado.

Se selecciona correctamente la tarea.

Se muestra la hora inicial en el momento en que es marcada.

Se muestra la hora final en el momento en que es marcada.

Se calcula la duración del trabajo y es mostrada.

Todos los datos son insertados sin errores.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH2_P2

UH: 2. Reporte de Tiempo

Nombre: Mostrar reporte de tiempos de trabajo

Descripción: Prueba para comprobar la funcionalidad realizar y mostrar reporte de

tiempos de trabajo
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario.
Entrada / Pasos de Ejecución: Se selecciona los parámetros del reporte a obtener. Se intenta obtener el reporte.
Resultado Esperado: Se muestra el reporte solicitado No se producen errores.
Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH3_P1	UH: 3. Reporte de Errores.
Nombre: Creación de un nuevo ticket.	
Descripción: Prueba para comprobar la funcionalidad de crear un nuevo ticket.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario.	
Entrada / Pasos de Ejecución: Se intenta crear un nuevo tickets.	
Resultado Esperado: El ticket es creado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: UH3_P2	UH: 3. Reporte de Errores.
Nombre: Notificación de ticket a un usuario.	
Descripción: Prueba para comprobar la funcionalidad de notificar a un usuario que tiene asignado un ticket.	
Condiciones de Ejecución: El sistema debe estar levantado. Debe haber ticket creado con anterioridad y asignado al usuario logueado.	
Entrada / Pasos de Ejecución: Se accede a la página inicial del sistema. Se accede a la vista de los tickets mediante la notificación mostrada en la página inicial.	
Resultado Esperado: Se muestra la notificación al usuario de que tiene tickets asignados. Se muestra los tickets que tiene asignado el usuario con todos sus datos. Se muestra al usuario la posibilidad de modificar el estado del ticket	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: UH3_P3	UH: 3. Reporte de Errores.
Nombre: Modificación del estado de un ticket asignado a un usuario.	
Descripción: Prueba para comprobar la funcionalidad de modificar el estado de un ticket asignado al un usuario.	
Condiciones de Ejecución: El sistema debe estar levantado.	

Debe haber ticket creado con anterioridad y asignado al usuario logueado.

Entrada / Pasos de Ejecución:

Se intenta modificar el estado del ticket y la descripción del mismo.

Resultado Esperado:

El estado del tickets es modificado.

La descripción del ticket es modificada.

Se elimina la notificación de este ticket a los usuarios

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH3_P4

UH: 3. Reporte de Errores

Nombre: Mostrar reporte de los tickets creados

Descripción: Prueba para comprobar la funcionalidad realizar y mostrar reporte de tickets

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario.

Entrada / Pasos de Ejecución:

Se selecciona los parámetros del reporte a obtener.

Se intenta obtener el reporte.

Resultado Esperado:

Se muestra el reporte solicitado

No se producen errores.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH4_P1	UH: 4. Creación de Cronograma
Nombre: Creación del cronograma del proyecto	
Descripción: Prueba para comprobar la funcionalidad crear y mostrar el cronograma del proyecto	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario. Debe haber registrado proyectos y tareas con su fecha de inicio y fin.	
Entrada / Pasos de Ejecución: Se accede a la página del cronograma.	
Resultado Esperado: Se muestra el cronograma del proyecto con todos sus datos en una nueva ventana. No se producen errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: UH5_P1	UH: 5. Control de Recursos Materiales
Nombre: Creación de un nuevo puesto de trabajo con datos válidos	
Descripción: Prueba para comprobar la funcionalidad de crear un nuevo puesto de trabajo.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción. Se utilizarán datos válidos para insertar el puesto de trabajo.	
Entrada / Pasos de Ejecución:	

Se intenta insertar un puesto de trabajo con datos validos.

Resultado Esperado:

El puesto de trabajo se insertar sin errores.

El sistema regresa a la vista de puestos de trabajo donde se muestra además de los ya existentes con anterioridad el nuevo puesto de trabajo creado.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH5_P2

UH: 5. Control de Recursos Materiales

Nombre: Edición de los datos de un puesto de trabajo ya existente.

Descripción: Prueba para comprobar la funcionalidad de editar los datos de un puesto de trabajo ya existente en el sistema.

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción.

El puesto de trabajo debe estar registrado en el sistema.

Se utilizarán datos válidos para editar el puesto de trabajo.

Entrada / Pasos de Ejecución:

Se intenta seleccionar el puesto de trabajo que se editará.

Se intenta realizar los cambios al puesto de trabajo con datos válidos.

Resultado Esperado:

Se muestra la vista con los datos del puesto de trabajo seleccionado para editar.

Los datos del puesto de trabajo son modificados sin errores.

El sistema regresa a la vista de puestos de trabajo

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación	
Código: UH5_P3	UH: 5. Control de Recursos Materiales
Nombre: Eliminación de un puesto de trabajo	
Descripción: Prueba para comprobar la funcionalidad de eliminar un puesto de trabajo.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción. El puesto de trabajo debe estar registrado en el sistema.	
Entrada / Pasos de Ejecución: Se intenta eliminar un puesto de trabajo.	
Resultado Esperado: El puesto de trabajo es eliminado sin errores.	
Evaluación de la Prueba: Prueba satisfactoria.	

Prueba de Aceptación	
Código: UH5_P4	UH: 5. Control de Recursos Materiales
Nombre: Creación de un nuevo puesto de trabajo con datos no válidos	
Descripción: Se intentará crear un puesto de trabajo ya existente o con datos inválidos.	
Condiciones de Ejecución: El puesto de trabajo ya existe o los datos no son válidos.	
Entrada / Pasos de Ejecución:	

Se ejecutan los pasos para crear un nuevo puesto de trabajo.

Resultado Esperado:

Se produce un error que es notificado al usuario

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH6_P1

UH: 6. Gestión de Recursos Humanos

Nombre: Registro de un nuevo usuario con datos válidos

Descripción: Prueba para comprobar la funcionalidad de registrar un nuevo usuario.

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción.

Se utilizarán datos válidos para registrar un nuevo usuario.

Entrada / Pasos de Ejecución:

Se intenta insertar un usuario con datos válidos

Resultado Esperado:

El usuario se registra sin errores.

El sistema regresa a la vista de usuarios del sistema donde se muestra además de los ya existentes con anterioridad el usuario registrado.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH6_P2

UH: 5. Control de Recursos Materiales

Nombre: Edición de los datos de un usuario ya registrado

<p>Descripción: Prueba para comprobar la funcionalidad de editar los datos de un usuario ya registrado</p>
<p>Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción.</p> <p>El usuario debe estar registrado en el sistema.</p> <p>Se utilizarán datos válidos para editar los datos del usuario.</p>
<p>Entrada / Pasos de Ejecución:</p> <p>Se intenta seleccionar el usuario que se editará.</p> <p>Se intenta realizar los cambios al usuario con datos válidos.</p>
<p>Resultado Esperado:</p> <p>Se muestra la vista con los datos del usuario seleccionado para editar.</p> <p>Los datos del usuario son modificados sin errores.</p> <p>El sistema regresa a la vista de usuarios del sistema.</p>
<p>Evaluación de la Prueba: Prueba satisfactoria.</p>

Prueba de Aceptación	
Código: UH6_P3	UH: 6. Gestión de Recursos Humanos
Nombre: Eliminación de un usuario del sistema.	
Descripción: Prueba para comprobar la funcionalidad de eliminar usuario del sistema.	
Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario con privilegios para realizar ese tipo de acción.	
El usuario debe estar registrado en el sistema.	
Entrada / Pasos de Ejecución:	

Se intenta eliminar un usuario.

Resultado Esperado:

El usuario es eliminado sin errores.

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH6_P4

UH: 6. Gestión de Recursos Humanos

Nombre: Registro de un nuevo usuario con datos no válidos

Descripción: Se intentará registrar un usuario ya existente o con datos inválidos.

Condiciones de Ejecución: El usuario ya se encuentra registrado en el sistema o los datos no son válidos.

Entrada / Pasos de Ejecución:

Se ejecutan los pasos para registrar un nuevo usuario.

Resultado Esperado:

Se produce un error que es notificado al usuario

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH7_P1

UH: 7. Distribución de Recursos.

Nombre: Planificación de horarios de trabajo.

Descripción: Prueba para comprobar la funcionalidad de planificar los horarios de trabajo y con ella la asignación de recursos.

Condiciones de Ejecución: El sistema debe estar levantado y logueado un usuario

con privilegios para realizar ese tipo de acción.

Debe existir al menos un puesto de trabajo registrado en el sistema.

Debe existir al menos un usuario registrado en el sistema.

Entrada / Pasos de Ejecución:

Se intenta realizar la asignación de horario y puesto de trabajo a un usuario del sistema.

Resultado Esperado:

La asignación es realizada sin errores sin errores.

Se muestra la sección de trabajo en la que se realizó la asignación

Evaluación de la Prueba: Prueba satisfactoria.

Prueba de Aceptación

Código: UH7_P2

UH: 7. Distribución de Recursos.

Nombre: Mostrar horarios de trabajo.

Descripción: Prueba para comprobar la funcionalidad de mostrar los horarios de trabajos planificados para el proyecto.

Condiciones de Ejecución: El sistema debe estar.

Entrada / Pasos de Ejecución:

Se intenta ver la planificación de horarios de trabajo en una sección determinada.

Resultado Esperado:

Se muestra la planificación de horarios de trabajo en la sección seleccionada.

Evaluación de la Prueba: Prueba satisfactoria.

Anexo 2 Diseño de Base de Datos

