INSTITUTO SUPERIOR MINERO
METALURGICO DE MOA
DR. ANTONIO NUNEZ JIMENEZ

Ingeniería Informática
Facultad: Geología-Minas

Trabajo de Diploma

Para Optar por el Título de

Ingeniero Informático

Título: Sistema de Gestión de Mantenimientos para el Departamento de Inversiones del ISMMM

Autor: Luis Rianne Guerrero López

Tutor(es): Ing. José Rolando Pérez Sandó

Ing. Adriannys Céspedes Ramírez

Moa, 2014

"Año 56 de la Revolución"

Declaración de Autoría.

Moa, Holguín, Junio del 2013

“Año del 55 Aniversario del Triunfo de la Revolución”

Yo, Luis Rianne Guerrero López, estudiante del Instituto Superior Minero Metalúrgico de Moa (ISMM), Dr. Antonio Núñez Jiménez, declaro que soy único autor de la presente investigación titulada: “Sistema de Gestión de Mantenimientos para el Departamento de Inversiones del ISMMM” y autorizo a hacer uso de la misma en su beneficio al ISMMM.

Para que así conste firman la presente a los ___ días del mes de ___del 2014.

_____

Firma del autor

Luis Rianne Guerrero López

_____ _____

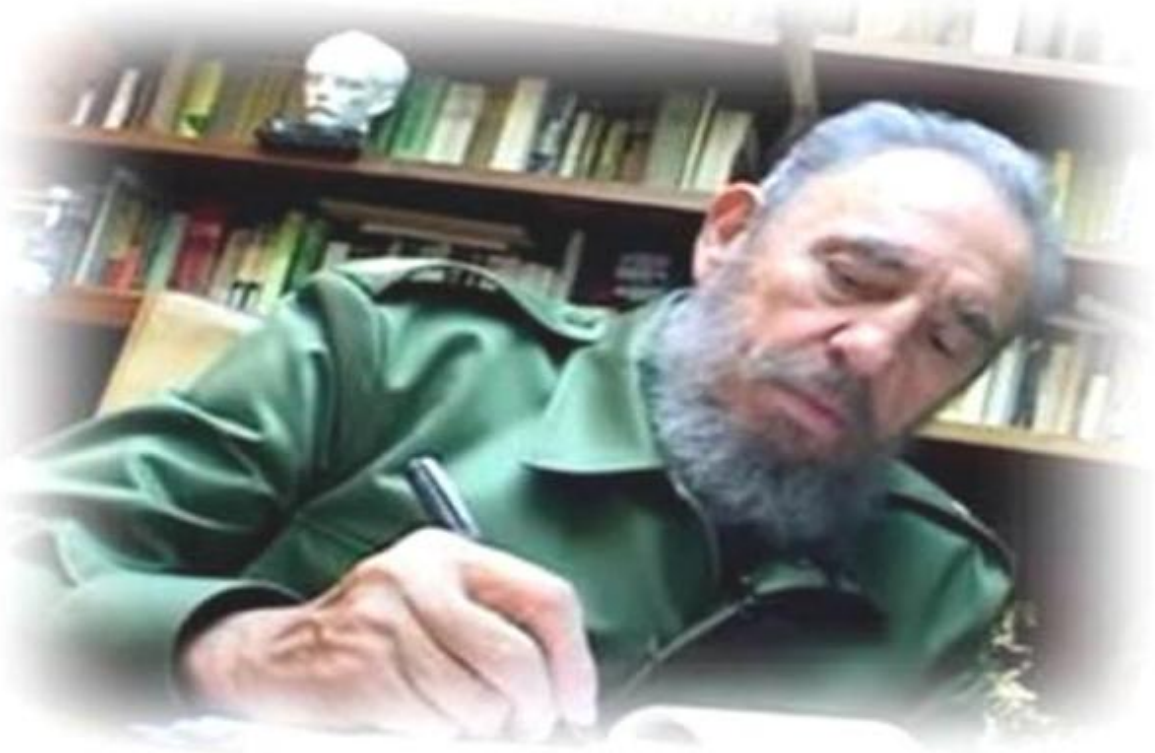
Firma del Tutor Firma del Tutor

Ing. José Rolando Pérez Sandó Ing. Adriannys Céspedes Ramírez

Pensamiento

"Podemos llegar a hacer mucho y podemos llegar muy lejos, porque tenemos lo que no tienen otros: La cantidad de talento acumulado de nuestra sociedad, la cantidad de inteligencias desarrolladas…"

Fidel Castro Ruz



Agradecimientos

Agradecerle a la VIDA por todas las oportunidades que me ha dado para ser un mejor hombre y prepararme para el futuro.

Al Instituto Superior Minero Metalúrgico de Moa y a los profesores de la carrera por convertirme en un profesional.

A mi familia por apoyarme, orientarme y estar siempre a mi lado en todos los momentos.

A mis compañeros de aula que en muchas ocasiones me ayudaron de forma incondicional y de buena fe.

A todas las personas que he conocido, porque me enseñaron a insertarme en la sociedad y superarme como ser humano.

A mis tutores y consultante por darme su ayuda en esta investigación.

A todos aquellas personas que me critican y hablan de mí, mis mayores agradecimientos porque todos los días me tienen presente

Gracias.

Dedicatoria

Dedicado a mi familia, en especial a mis padres, a mi abuela por estar conmigo todos estos años de sacrificio y a mi novia. A todos los que han sabido guiarme por el camino correcto y han aportado su granito de arena para la realización de este trabajo.

A mis amigos por estar siempre ahí, en las buenas y en las malas y aceptarme como soy, en especial a Juan Carlos.

Para todas esas personas que confiaron en mí y me apoyaron siempre,

Muchísimas gracias.

RESUMEN

El Departamento de Inversiones tiene entre sus principales objetivos garantizar la calidad de los procesos que se realizan en su entorno, siendo el encargado de gestionar las labores de mantenimiento del centro en busca de mejores condiciones para desarrollar principalmente las actividades docentes. Actualmente existen deficiencias relacionadas principalmente con la disponibilidad de la información relevante y la gestión de los recursos humanos.

Esta investigación tiene como objetivo favorecer el proceso de gestión de mantenimientos del Departamento de Inversiones del ISMMM, construyéndose una aplicación web orientada a la disponibilidad de la información relevante para los responsables de las áreas, permitiendo una planificación adecuada de los recursos humanos.

La implantación de esta herramienta informática facilitará la consulta y el procesamiento de la información relacionada con la gestión del mantenimiento gestionado por el Departamento de Inversiones del ISMMM. Además involucrará de forma activa a los responsables de las diferentes áreas administrativas, dando un seguimiento a las actividades realizadas por las brigadas de mantenimiento y controlando el costo en recursos materiales asociados a las mismas.

ABSTRACT

The Department of Investments has among its main objectives to guarantee the quality of the processes that are carried out in its environment. It is the responsible of negotiating the works of maintenance at the institution, looking forward for better conditions to develop the educational activities. Currently deficiencies exist mainly related with the readiness of the outstanding information and the administration of the human resources

The main goal of this investigation is to favor the process of maintenances management of the Department of Investments of the ISMMM. In order to accomplish this task a web oriented application will be built, looking forward to the readiness of the outstanding information for the administrative areas coordinators, allowing an appropriate planning of the human resources.

The deployment of this software will facilitate the consultation and the prosecution of the information related with the administration of the maintenance negotiated by the Department of Investments of the ISMMM. It will also involve in an active way to those responsible for the different administrative areas, giving a pursuit to the activities carried out by the maintenance brigades, and controlling the cost associated to the material resources of the activities.

Índice

INTRODUCCIÓN **1**

CAPÍTULO 1 .FUNDAMENTACIÓN TEÓRICA **1**

1.1 ESTADO DEL ARTE 1
1.1.1 Sistemas automatizados existentes vinculados al campo de acción internacionalmente *1*
1.1.2 Sistemas automatizados existentes vinculados al campo de acción en Cuba *2*
1.1.3 Análisis de Antecedentes *3*
1.2 TENDENCIAS Y TECNOLOGÍAS ACTUALES 4
1.2.1 Aplicación Web *4*
1.2.2 Ventajas de una web *4*
1.2.3 Servidor Web *6*
1.2.4 Sistemas gestores de bases de datos (SGBD) *7*
1.4 LENGUAJE DE PROGRAMACIÓN 8
1.5 FRAMEWORK DE DESARROLLO WEB 8
1.5.2 Sencha Ext JS *9*
1.6 ENTORNO DE PROGRAMACIÓN 10
1.7 METODOLOGÍA PROPUESTA PARA EL DESARROLLO DE LA APLICACIÓN 10
1.7.1 Programación Extrema (Extreme Programming, XP) *10*
Valores que promueve XP *11*
El ciclo de vida ideal de XP *11*
Prácticas en las que se fundamenta XP *12*
Fases de la Metodología XP *15*
1.8 PATRONES ARQUITECTÓNICOS 17
1.8.1 Modelo – Vista – Controlador *17*
1.9 HERRAMIENTAS CASE 20

CAPÍTULO 2: PLANEACIÓN Y DISEÑO **23**

2.1 INTRODUCCIÓN 23
2.2 LISTA DE RESERVA DEL PRODUCTO 23
2.4 HISTORIAS DE USUARIOS 28
2.4.1 Descripción de las historias de usuario *28*
2.5 PLANIFICACIÓN DE ENTREGAS 31
2.5.1 Estimación del esfuerzo por historias de usuario *32*
2.5.2 Planificación de iteraciones *33*
2.5.3 Plan de iteraciones *35*
2.6 DISEÑO 37
2.6.1 Tarjetas CRC *38*

CAPÍTULO 3 DESARROLLO Y PRUEBAS 41

INTRODUCCIÓN 41
3.3 DESARROLLO DE LAS ITERACIONES 43
3.3.1 Tareas por historias de usuario *43*
3.3.2 Tareas de ingeniería *46*
3.4 PRUEBAS 47
3.4.1 Pruebas de aceptación *48*

CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD 51

4.1 EVALUACIÓN COSTO – BENEFICIO 51
4.2 EFECTOS ECONÓMICOS 52
4.2.1 Efectos directos: *52*
4.2.2 Efectos indirectos: *52*
4.2.3 Efectos externos: *53*
4.2.4 Intangibles *53*
4.3 FICHA DE COSTO DE UN PRODUCTO INFORMÁTICO. 54

CONCLUSIONES GENERALES 58

RECOMENDACIONES 59

REFERENCIAS BIBLIOGRÁFICAS 60

BIBLIOGRAFÍA 62

GLOSARIO DE TÉRMINOS 64

ANEXO.1: HISTORIAS DE USUARIOS LXIX

ANEXO.2: TARJETAS CRC LXXIII

ANEXO.3: TAREAS DE INGENIERÍA LXXV

ANEXO.4: PRUEBAS DE ACEPTACIÓN LXXXV

Introducción

En el mundo de hoy no se conciben entidades que no utilicen las Tecnologías de la Información y las Comunicaciones. Estas son capaces de aportar soluciones eficientes y eficaces en cualquier esfera empresarial; permitiendo actuar con rapidez, obtener el máximo rendimiento del personal y tomar decisiones a la administración. Cuba, insertándose en este marco, ha diseñado e iniciado la aplicación de estrategias que permiten convertir los conocimientos y las tecnologías informáticas en instrumentos a disposición del avance y las profundas transformaciones revolucionarias.

El Instituto Superior Minero Metalúrgico de Moa (ISMMM) también se ha sumergido en el desarrollo de estos avances tecnológicos, priorizando el funcionamiento de sus departamentos docentes y administrativos. Dentro del instituto existe el Departamento de Inversiones, que tiene entre sus principales objetivos garantizar la calidad de los procesos que se realizan en su entorno, siendo el encargado de manejar la política de inversiones del centro, en busca de mejores condiciones para desarrollar principalmente las actividades docentes. Además dentro de este departamento existen varias brigadas destinadas al Mantenimiento, estas se clasifican principalmente en Brigada de Construcción y Brigada de Equipos brindándosele especial atención al área de beca. Las actividades de estas brigadas pueden estar referidas a un Mantenimiento Planificado o Mantenimiento Imprevisto.

En este departamento los planes de trabajo de cada brigada se confeccionan partiendo de un plan general, incorporándose aquellas actividades que surgen a partir de los reportes de averías que llegan desde las áreas administrativas. Los trabajos de las brigadas son controlados mediante Órdenes de Trabajo. En ocasiones se hace necesaria la solicitud de recursos a otras entidades del municipio, lo que debe quedar registrado en la orden de trabajo. Los recursos humanos están destinados a las principales actividades en las áreas priorizadas, así como los especialistas emiten sus reportes pertinentes teniendo en cuenta cuanto durará el trabajo y su costo, así los jefes de cada área plasman su conformidad con el trabajo que se realiza.

La dirección de inversiones lleva un control de los recursos que se utilizan en las reparaciones y otras actividades de mantenimiento. Este gasto de recursos se asocia al centro de costo del área administrativa donde se realizó el trabajo, registrándose luego en el Departamento de Contabilidad del ISMMM.

Actualmente la gestión del mantenimiento se ve afectada por una serie de situaciones que no permiten la correcta socialización de las actividades que se realizan. Uno de los principales problemas consiste en la disponibilidad de las informaciones relevantes a los distintos niveles, provocando desconocimiento de las que se encuentran en ejecución y su impacto en el centro. También estas informaciones no son accesibles para todas las áreas del instituto, causando problemas de coordinación al realizarse las actividades. Además el control de los procesos en ejecución y los encargados de ejecutarlos presenta deficiencias, dificultándose el control sistemático sobre las actividades y las causas que provocan demoras o interrupciones por un tiempo prolongado.

Existen dificultades con el manejo de los recursos humanos para organizar la realización de las actividades de mayor importancia. Imposibilitando la correcta atención de los jefes de las distintas áreas, provocando demoras de tiempo para realizar las actividades, aumentándose el costo de las mismas e incidiendo negativamente en su correcta ejecución.

Por lo expuesto anteriormente se plantea como **problema científico**: ¿Cómo favorecer el Proceso de Gestión de Mantenimientos del Departamento de Inversiones del ISMMM?

Para darle solución al problema planteado se identifica como **objeto de estudio** el proceso de gestión de mantenimientos del Departamento de Inversiones del ISMMM, siendo el **campo de acción** abarcado la informatización del proceso de gestión de mantenimientos del Departamento de Inversiones del ISMMM.

Como **objetivo general** de esta investigación se plantea: Desarrollar un sistema informático que favorezca el proceso de gestión de mantenimientos del Departamento de Inversiones del ISMMM.

Como **objetivos específicos** en esta investigación tenemos:

1. Definir el marco teórico-conceptual de la investigación.
2. Realizar las actividades del ciclo de desarrollo del software.
3. Realizar el estudio de factibilidad.

Para darle cumplimiento al objetivo planteado se trazaron las siguientes tareas:

1. Realizar un análisis crítico del proceso de gestión de mantenimiento en el Departamento de Inversiones del ISMMM.
2. Estudiar las soluciones informáticas vinculadas al campo de acción.
3. Determinar las necesidades de funcionamiento de la aplicación a desarrollar.
4. Valorar las tendencias actuales de las herramientas y tecnologías.
5. Elaborar la documentación correspondiente a la metodología de desarrollo seleccionada.
6. Implementar la aplicación.
7. Realizar las pruebas funcionales al sistema construido
8. Determinar la factibilidad y sostenibilidad del sistema.

Como **idea a defender** se plantea: El desarrollo de un sistema informático donde la información relevante se encuentre disponible para los responsables de las áreas, permitiendo una planificación adecuada de los recursos humanos, contribuirá significativamente a la gestión de mantenimientos del Departamento de Inversiones del ISMMM

En la investigación se llevaron a cabo diferentes **Métodos de Investigación**, de los **Métodos Empíricos** fueron utilizados el **método de la entrevista**: se realizan múltiples entrevistas al cliente para esclarecer los procesos de negocio y obtener un mayor entendimiento de sus necesidades y expectativas.

Los **Métodos Teóricos** utilizados fueron el **histórico - lógico:** se utiliza este método para efectuar un análisis de las actividades del Departamento de Inversiones, brindando un resumen del surgimiento y evolución de las mismas, para adquirir más conocimiento del tema abordado y el **método de la modelación**: se utiliza este método para modelar todos los procesos del negocio a desarrollar con el fin de conseguir un entendimiento del mismo

El trabajo está estructurado por un resumen, introducción, cuatro capítulos, conclusiones generales, recomendaciones, bibliografía y los anexos.

En el Capítulo uno **Fundamentación Teórica,** se hace una descripción de los principales conceptos utilizados en el problema, se realiza un análisis crítico del objeto de estudio, también se efectúa un estudio acerca de los diferentes sistemas existentes afines al campo de acción, además se seleccionan las herramientas y la metodología a utilizar.

En el Capítulo dos **Planificación y Diseño,** se hace uso de la metodología seleccionada en el capítulo primero para el desarrollo del proyecto, emprendiendo en detalles cada una de sus fases.

En el Capítulo tres **Implementación y Prueba,** se presentan los principales métodos y definiciones dentro de la implementación de los flujos de trabajo, además se describen las pruebas realizadas y los resultados que estas arrojaron.

En el Capítulo 4 **Estudio de Factibilidad y Sostenibilidad,** se realiza un estudio de los esfuerzos requeridos para la realización del sistema, se hace referencia a los beneficios tangibles e intangibles, y se valora la sostenibilidad del producto.

Capítulo 1 .Fundamentación Teórica

En este capítulo se realizará un estudio de diferentes sistemas existentes, relacionados con la gestión académica tanto a nivel nacional como internacional. También se abordarán conceptos fundamentales sobre el desarrollo de aplicaciones Web, las características generales de los sistemas de bases de datos y de los lenguajes de programación Web, también se analizarán las diferentes metodologías existentes para el desarrollo de aplicaciones, y se seleccionará la más conveniente a utilizar.

1.1 Estado del arte

En este epígrafe se realizó un estudio de sistemas homólogos al que se desea obtener, en el ámbito internacional, nacional y en la universidad. También se analizó en qué medida los sistemas encontrados dan solución a la situación problémica planteada.

1.1.1 Sistemas automatizados existentes vinculados al campo de acción internacionalmente

El software para la planificación y la gestión de mantenimiento (PGM|Win)

PGM|Win es una herramienta informática diseñada para dar soporte en la cada vez más complicada tarea de gestionar el mantenimiento de planta. Las nuevas tecnologías así como los actuales sistemas de calidad, medioambiente y prevención de riesgos, hacen impensable la gestión del mantenimiento sin el empleo de aplicaciones de este tipo.

Es sin duda un paquete de software para la gestión del mantenimiento industrial, diseñado por gente de mantenimiento para gente de mantenimiento, y enfocado hacia la pequeña y mediana empresa. PGM|Win ha sido desarrollado sin olvidar los beneficios que ha de reportar a los procesos de negocio de la empresa. No es software empaquetado, se trata de un sistema modular, abierto y personalizado según las necesidades del cliente. No impone una metodología de trabajo rígida, es el software quien se adapta a la forma y recursos del cliente. [1]

Software Evaluador de Proyectos de Inversión (S.E.P.I.)

La evaluación de proyectos es una técnica que ayuda a tomar decisiones, a la par q se constituye como un factor de decisión en la obtención de recurso financiero para ejecutarlos. Con la finalidad de optimizar los tiempos y costos asociados a esta tarea se presenta el “Software Evaluador de Proyectos de Inversión” (S.E.P.I.) cuyo objetivo es la aplicación de las Tecnologías de la Información en las unidades del negocio reduciendo la incertidumbre del riesgo sobre la inversión, evaluando oportunidades del negocio de una manera rápida confiable partiendo del uso de información básica sobre el proyecto en cuestión.

El S.E.P.I. permite registrar unidades de negocios y asociarles proyectos de inversión. Cada proyecto de inversión podrá a su vez registrar los insumos necesarios para poder medir y estimar el grado en que se alcanzarán los objetivos económicos planteados dentro de la realización del proyecto, ejecutando proyecciones de rentabilidad que servirán de base para determinar la viabilidad económica del proyecto. Así mismo el trabajo contribuye a la implementación de una metodología de desarrollo iterativa e incremental, que será el punto de partida para incursionar en el desarrollo y documentación de metodologías ágiles en el desarrollo de software del Cuerpo Académico de Tecnologías de la Información y Comunicación en colaboración con el Cuerpo Académico de Desarrollo de Negocios. [2]

1.1.2 Sistemas automatizados existentes vinculados al campo de acción en Cuba

La gestión del mantenimiento en entidades turísticas (CEIM)

Aborda la proyección actual que debe tenerse en cuenta en la implementación de la Gestión de Mantenimiento en empresas del turismo. Recalca en la necesidad de la informatización, enunciando sus ventajas. Durante los últimos años se ha constatado en el sector hotelero una necesidad por simplificar y optimizar la gestión del mantenimiento.

Por su flexibilidad y funcionalidad el sistema propuesto se adecua a cualquier tamaño de hotel asegurándole la configuración de un sistema que se ajuste a sus necesidades. De fácil manejo por parte del personal de mantenimiento, este programa le permitirá de

una forma sencilla el acceso a todas las opciones del sistema mediante los diferentes menús y submenús presentados.

Este sistema informativo pretende hacer más ágil y organizada la gestión de mantenimiento. Dicho sistema pone de manifiesto la potencia, eficacia y capacidad de capturar, utilizar y almacenar la información de todas las intervenciones que en cuanto a mantenimiento se refiera. Queda demostrado con este nuevo sistema informativo que la informatización es un medio importante para mejorar la gestión del mantenimiento. [3]

Software Offimantpro

Es el programa líder en la ciudad de Camagüey, creado con la finalidad de ayudar a las empresas en el control computarizado de la gestión de mantenimiento técnico para los equipos e instalaciones que lo requieren. Es un paquete que tiene un sentido lógico, pues incluye, además del desarrollo de las aplicaciones, un conjunto de consultorías informáticas, hasta la asistencia técnica para solucionar deficiencias. [4]

1.1.3 Análisis de Antecedentes

Los sistemas mencionados anteriormente presentan limitaciones que impiden su puesta en práctica en el entorno del ISMMM. Una limitación significativa la constituye el hecho de ser aplicaciones comercializadas bajo licencia propietaria (PGM|Win, S.E.P.I., Offimantpro), por lo que no se tiene acceso a su código fuente, impidiendo la reutilización de funcionalidades y componentes insertados en estas aplicaciones. Además, para la implantación de dichas aplicaciones se requeriría realizar una inversión que permita afrontar los gastos de instalación, consultoría y mantenimiento. En el caso particular del CEMI, es una herramienta diseñada para la industria hotelera, con un enfoque particularizado en este campo de acción.

Por lo anteriormente expuesto se define la necesidad de desarrollar una aplicación basada en las necesidades del cliente, incorporándole los ajustes necesarios del entorno universitario.

1.2 Tendencias y tecnologías actuales

1.2.1 Aplicación Web

En la ingeniería de software se denomina **aplicación web** a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (HTML, JavaScript, Java, etc.) en la que se confía la ejecución al navegador.

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa, un motor capaz de usar alguna tecnología web dinámica constituye la capa intermedia. Por último, una base de datos constituye la tercera y última capa.

El navegador web manda peticiones a la capa intermedia que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario. [5]

1.2.2 Ventajas de una web

- **Compatibilidad multiplataforma:** Las aplicaciones Web tienen un camino mucho más sencillo para la compatibilidad multiplataforma que las aplicaciones de software descargables. Varias tecnologías incluyendo Java, Flash, ASP y Ajax permiten un desarrollo efectivo de programas soportando todos los sistemas operativos principales.
- **Actualización:** Las aplicaciones basadas en Web están siempre actualizadas con el último lanzamiento sin requerir que el usuario tome acciones pro-activas, y sin necesitar llamar la atención del usuario o interferir con sus hábitos de trabajo, pues no se hace necesario iniciar nuevas descargas y/o procedimientos de instalación (algunas veces imposible cuando usted está trabajando dentro de grandes organizaciones).

- **Inmediatez de acceso:** Las aplicaciones basadas en Web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online y están listas para trabajar sin importar cuál es su configuración o su hardware.
- **Menos requerimientos de memoria:** Las aplicaciones basadas en Web tienen menos demandas de memoria RAM de parte del usuario final que los programas instalados localmente. Al residir y correr en los servidores del proveedor, esas aplicaciones basadas en Web usan en muchos casos la memoria de las computadoras donde ellas corren, dejando más espacio para correr múltiples aplicaciones sin incurrir en frustrantes deterioros en el rendimiento.
- **Menos Bugs:** Las aplicaciones basadas en Web son menos propensas a colgarse y crear problemas técnicos debido a software o conflictos de hardware con otras aplicaciones existentes, protocolos o software personal interno. Con aplicaciones basadas en Web, todos utilizan la misma versión, y todos los bugs pueden ser corregidos tan pronto como son descubiertos.
- **Precio:** Las aplicaciones basadas en Web no requieren la infraestructura de distribución, soporte técnico y marketing requerido por el software descargable tradicional. Esto permite que las aplicaciones online cuesten una fracción de sus contrapartes descargables y no totalmente gratuitas, mientras que ofrecen componentes adicionales y servicios Premium como una opción.
- **Múltiples usuarios concurrentes:** Las aplicaciones basadas en Web pueden ser utilizadas por múltiples usuarios al mismo tiempo. No hay más necesidad de compartir pantallas o enviar instantáneas cuando múltiples usuarios pueden ver e incluso editar el mismo documento de manera conjunta.
- **Los datos son más seguros:** Si bien la ruptura de discos no va a desaparecer, es probable que los usuarios escuchen mucho menos del tema. A medida que las compañías se hagan cargo del almacenamiento de los datos del usuario, granjas de almacenamiento de datos redundantes, altamente fiables, los usuarios van a tener mucho menos riesgo de perder sus datos debido a una ruptura de disco impredecible o a un virus de la computadora. Las compañías que proveen aplicaciones basadas en Web van a brindar amplios servicios de

resguardo de datos ya sea como una parte integral del servicio básico o como una opción paga. [6]

1.2.3 Servidor Web

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. [7]

Apache

Apache, sustancialmente, es un proyecto nacido para crear un servidor de Web estable, fiable y veloz para plataformas Unix. Apache nace, por una parte, de un código ya existente y de una serie de parches (patch) para mejorar su fiabilidad y sus características; de ahí su nombre. [8]

Características: [9]

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.

- Es una tecnología gratuita con un código fuente disponible. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esta característica le ofrece al software un grado de transparencia tal que es posible determinar en todo momento qué es lo que se está instalando, sin secretos ni puertas traseras.

- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que se instalen cuando se necesiten. Otra cosa importante es que cualquiera que posea alguna experiencia en la programación de C o Perl puede escribir un módulo para realizar una función determinada.

- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.

1.2.4 Sistemas gestores de bases de datos (SGBD)

Los sistemas de gestión de base de datos (SGBD); en inglés: Database Management System, abreviado DBMS son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. Entre los SGBD disponibles en el mercado se encuentran: PostgreSQL, MySQL, Firebird, SQLite, Oracle, Microsoft SQL Server, entre otros.

MySQL

Es una base de datos relacional de libre distribución con potencia, usada principalmente como soporte de portales web que necesitan manejar datos dinámicamente, es una alternativa a los grandes servidores de bases de datos basados en Microsoft SQL Server y Oracle. MySQL se destaca especialmente por su velocidad al efectuar lecturas y búsquedas en la base de datos.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por

relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido. [5]

MySQL cuenta con muchas **ventajas**, entre las que se encuentran las siguientes:

- Alto rendimiento
- Bajo coste
- Facilidad de configuración y aprendizaje
- Portabilidad

Requisitos de hardware

MySQL usa menos recursos del hardware. Para instalar MySQL, se necesita 32 RAM de Mb y casi cerca de 60Mb de espacio en disco duro. MySQL no requiere los recursos del CPU adicionales.

1.4 Lenguaje de programación

En el dominio de la red, los lenguajes del lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP, entre otros.

PHP (Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con numerosas características de PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil. [11]

1.5 Framework de desarrollo Web

¿Qué es un Framework?

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto. [12]

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

1.5.2 Sencha Ext JS

Sencha Ext JS (anterior ExtJS) es una biblioteca de JavaScript para el desarrollo de aplicaciones Web interactivas que además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. Creado por Jack Slocum, Brian Moeskau, Aaron Conran y Rich Waters.

Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open Source (GPL) y comerciales.

ExtJS ha marcado la diferencia en la preferencia de muchos desarrolladores. Y no solo para aplicaciones Web sino que ya es útil para productos de alta demanda en el mercado como iPhone e IPAD.

Ventajas

- Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos.
- Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, IE, Safari, Opera etc.).
- El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.
- Relación entre Cliente-Servidor balanceado: Se distribuye la carga de procesamiento entre, permitiendo que el servidor pueda atender más clientes al mismo tiempo.

- Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con la posibilidad de elegir que datos desea trasmitir al servidor y viceversa (Criterio este que puede variar con el uso de aplicaciones de pre-carga).
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

Desventajas

- Necesidad de una plataforma: Pues dependemos del paquete ExtJS para obtener los resultados deseados.
- Algunos señalan que no cuenta con una licencia LGPL. [12]

1.6 Entorno de programación

PhpDesigner 2008 v6.0.0: Es un completo entorno de desarrollo y programación especialmente diseñado para los gurús de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL.

Ofrece toda una serie de asistentes y diálogos integrados que facilitan en todo momento las tareas, además de acceso directo a librerías de código o scripts de uso habitual, utilidades diversas y toda suerte de herramientas, todo ello en una interfaz de diseño sencillo y elegante que se puede personalizar con nada menos que dieciocho temas distintos. Cuenta con cliente de FTP y navegador de ficheros integrado, utilidades de corrección y autocompletado, búsqueda integrada en Google y soporte para proyectos, además de usar un práctico esquema de color para la sintaxis del código fuente que facilita enormemente la programación. [13]

1.7 Metodología propuesta para el desarrollo de la aplicación

1.7.1 Programación Extrema (Extreme Programming, XP)

XP es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores, y propicia un buen clima de trabajo.

Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, que describe la filosofía de XP en el "Manifiesto Ágil", sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. [14]

Valores que promueve XP

Cuatro valores que promueven la metodología XP:

Simplicidad: XP propone el principio de hacer las cosas más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana.

Comunicación: Algunos problemas en los proyectos tienen su origen en que alguien no dijo algo importante en algún momento. XP hace imposible la falta de comunicación.

Retroalimentación: Retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo eficientemente.

Coraje: El coraje (valor) existe en el contexto de los otros 3 valores. [14]

El ciclo de vida ideal de XP

El ciclo de vida ideal de XP consta de seis fases:

- Exploración.
- Planificación de la Entrega (Reléase)
- Iteraciones.
- Producción.
- Mantenimiento.
- Muerte del Proyecto.

Prácticas en las que se fundamenta XP

Planificación incremental

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

El objetivo de XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. A estas versiones se les denomina release.

Un release cuenta con un cierto número de historias. La historia es la unidad de funcionalidad en un proyecto XP, y corresponde a la mínima funcionalidad posible que tiene valor desde el punto de vista del negocio. Durante cada iteración se cierran varias historias, lo que hace que toda iteración añada un valor tangible para el cliente.

Gran parte de la eficacia de este modelo de planificación se deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento. Dentro de esta división, el representante del cliente tiene las siguientes responsabilidades:

- Decidir qué se implementa en cada release o iteración.
- Fijar las fechas de fin de un release, recortando unas características o añadiendo otras.
- Priorizar el orden de implementación, en función del valor de negocio.
- Las responsabilidades del equipo de desarrollo son las siguientes:
- Estimar cuánto tiempo llevará una historia de usuario, esto es fundamental para el cliente, y puede llevarle a reconsiderar qué historias se deben incluir en una iteración.
- Proporcionar información sobre el coste de utilizar distintas opciones tecnológicas.

- Organizar el equipo.
- Estimar el riesgo de cada historia.
- Decidir el orden de desarrollo de historias dentro de la iteración.

Testing

La ejecución automatizada de tests es un elemento clave de la metodología XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor calificada para decidir cuál es la funcionalidad más valiosa.

Un efecto lateral importante de los tests, es que dan una gran seguridad a los desarrolladores; es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporciona una red de seguridad. La existencia de tests hace al código muy maleable.

Diseño simple

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

XP define un “diseño tan simple como sea posible” aquel que pasa todos los tests, no contiene código duplicado, deja clara la intención de los programadores (enfatiza el qué, no el cómo) en cada línea de código y contiene el menor número posible de clases y métodos.

Propiedad colectiva del código

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

Integración continua.

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

Clientes en el equipo

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo toma decisiones de negocio críticas. Esto no debería ocurrir, pero en el momento cumbre, con frecuencia no se obtiene retroalimentación del cliente con la fluidez necesaria.

La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para resolver dudas y para decidir qué se hace en cada momento, en función de los intereses del negocio.

Entregas pequeñas

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

Semana de 40 horas

La programación extrema lleva un modo de trabajo en el que el equipo siempre está al 100%. Una semana de 40 horas en las que se dedica la mayor parte del tiempo a tareas que suponen un avance puede dar mucho de sí, y hace innecesario recurrir a sobreesfuerzos, excepto en casos extremos. Además, el esfuerzo continuado pronto lleva a un rendimiento menor y a un deterioro de la moral de todo el equipo. [14]

Fases de la Metodología XP

Fase I: Planificación

- Se escriben historias de usuario, cuya idea principal es describir un caso de uso en dos o tres líneas con terminología del cliente (de hecho, se supone que deben ser escritos por el mismo), de tal manera que se creen *test* de aceptación para historias de usuarios (*user storie)* y permita hacer una estimación de tiempo de desarrollo del mismo.
- Se crea un plan de lanzamiento (*release planning*), que debe servir para crear un calendario que todos puedan cumplir y en cuyo desarrollo hayan participado todas las personas involucradas en el proyecto. Se usa como base las historias de usuario, participando el cliente en la elección de las que se desarrollarán, y según las estimaciones de tiempo de los mismos se crearán las iteraciones del proyecto.
- El desarrollo se divide en iteraciones, cada una de las cuales comienzan con un plan de iteración, para el que se eligen las historias de usuario a desarrollar y las tareas de desarrollo.
- Se cambia el proceso cuanto sea necesario, para adaptarlo al proyecto.

Fase II: Diseño

- Se eligen los diseños funcionales más simples.
- Se elige una metáfora del sistema para que el nombrado de clases, siga una misma línea, facilitando la reutilización y la comprensión del código.
- Se escriben tarjetas de clase-responsabilidades-colaboración (CRC) para cada objeto, que permitan abstraerse al pensamiento estructurado y que el equipo de desarrollo completo participe en el diseño.

Fase III: Codificación

- El cliente está siempre disponible, de ser posible, cara a cara. La idea es que forme parte del equipo de desarrollo, y esté presente en todas las fases de XP. La idea es usar el tiempo del cliente para estas tareas en lugar de crear una detallada especificación de requisitos, y evitar la entrega de un producto insuficiente, que le hará perder tiempo.

- El código se ajustará a unos estándares de codificación, asegurando la consistencia y facilitando la comprensión y refactorización del código.
- Las pruebas unitarias se codifican antes que el código en sí, haciendo que la codificación de este último sea más rápida, y que cuando se afronte la misma se tenga más claro, qué objetivos tiene que cumplir lo que se va a codificar.
- La programación del código se realiza en parejas, para aumentar la calidad del mismo. En cada momento, sólo habrá una pareja de programadores que integre código.
- Se integra código y se lanza dicha integración de manera frecuente, evitando divergencias en el desarrollo y permitiendo que todo el mundo trabaje con la última versión del desarrollo. De esta manera, se evitará pasar grandes períodos de tiempo integrando el código al final del desarrollo, ya que las incompatibilidades serán detectadas enseguida.
- Se usa la propiedad colectiva del código, lo que se traduce en que cualquier programador puede cambiar cualquier parte del código. El objetivo es fomentar la contribución de ideas por parte de todo el equipo de desarrollo.
- Se deja la optimización para el final.
- No se hacen horas extra de trabajo.

Fase IV: Pruebas

- Todo el código debe tener pruebas unitarias, y debe pasarlas antes de ser lanzado.
- Cuando se encuentra un error de codificación o bug, se desarrollan pruebas para evitar volver a caer en el mismo.
- Se realizan pruebas de aceptación frecuentemente, publicando los resultados de las mismas. Estas pruebas son generadas a partir de las historias de usuarios (HU) elegidas para la iteración, y son "pruebas de caja negra", en las que el cliente verifica el correcto funcionamiento de lo que se está probando. Cuando se pasa la prueba de aceptación, se considera que la correspondiente historia de usuario se ha completado. [14]

1.8 Patrones Arquitectónicos

Los patrones arquitectónicos, o patrones de arquitectura, son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen una escalera más grande.

Aunque un patrón arquitectónico comunica una imagen de un sistema, no es una arquitectura como tal. Un patrón arquitectónico es más un concepto que captura elementos esenciales de una arquitectura de software. Muchas arquitecturas diferentes pueden implementar el mismo patrón y por lo tanto compartir las mismas características. Además, los patrones son a menudo definidos como una cosa "estrictamente descrita y comúnmente disponible". Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interaccionar. Cuando esto es descrito estrictamente y comúnmente disponible, es un patrón.

Uno de los aspectos más importantes de los patrones arquitectónicos es que encarnan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad. A primeros de la fase de diseño, un arquitecto de software escoge qué patrones arquitectónicos mejor ofrecen las calidades deseadas para el sistema. [15]

1.8.1 Modelo – Vista – Controlador

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la

Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

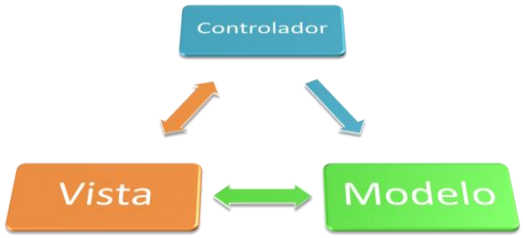


Figura 1.1 Patrón arquitectónico de Modelo-Vista-Controlador

Descripción

- Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. [16]

Ventajas de utilizar MVC

Una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilenguaje, distintos diseños de presentación, etc. sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos. Al existir la separación de vistas, controladores y modelos es más sencillo realizar labores de mejora como:

- Agregar nuevas vistas.
- Agregar nuevas formas de recolectar las órdenes del usuario (interpretar sus modelos mentales).
- Modificar los objetos de negocios bien sea para mejorar el performance o para migrar a otra tecnología.
- Las labores de mantenimiento también se simplifican y se reduce el tiempo necesario para ellas. Las correcciones solo se deben hacer en un solo lugar y no en varios como sucedería si tuviésemos una mezcla de presentación e implementación de la lógica del negocio.
- Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice. Adicionalmente el patrón MVC propende a la especialización de cada rol del equipo, por tanto en cada liberación de una nueva versión se verán los resultados. [16]

1.9 Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadoras), son aplicaciones informáticas que tienen como objetivo fundamental solucionar y afrontar los problemas de mala calidad de software y documentación inadecuada. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo de un determinado proyecto o software, ya que brindan la posibilidad de realizar cálculos de costos, generan código fuente automáticamente de un diseño previamente dado, poseen compilación automática, ayudan con la documentación y juegan un papel importante en la detección de errores.

Embarcadero ER/Studio: Es una herramienta de modelado de datos, se usa para el diseño y la construcción lógica y física de bases de datos. Su ambiente es de gran alcance y multinivel. Simple y fácil al usuario, ayuda a las organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa. (Bureaudeprensa.com)

Ventajas:

- Si se está comenzando un nuevo diseño o está manteniendo una base de datos existente, ER/Studio se combina con las características para ayudarle a conseguir el trabajo hecho con eficacia.
- La creación de diagramas es clara y rápida.
- Tiene la posibilidad de realizar diagramas con desempeño rápido. [17]

Conclusiones

En este capítulo se estableció el estado del arte, determinándose la necesidad de la presente propuesta de solución. Para el desarrollo de la aplicación informática se realizó una selección de herramientas y tecnologías, determinándose como metodología de desarrollo para guiar el ciclo de vida a la metodología ágil XP. En estos momentos es la metodología ágil con más documentación, permite una buena comunicación entre el equipo de desarrollo y el cliente. También es muy fácil de implementar y aprender.

Para la implementación del sistema se utiliza como entorno de programación el PhpDesigner. Este fue seleccionado por las ventajas que ofrece a la hora de escribir algoritmos complejos donde es necesario correr paso a paso. Además de que es un entorno para el lenguaje de programación PHP. Este lenguaje fue seleccionado como el ideal a utilizar para la construcción del sistema ya que ofrece una amplia gama de ventajas con respecto a otros. Es un lenguaje de programación libre y multiplataforma, siendo una alternativa de fácil acceso para todos. Como servidor Web se utilizó el Apache que viene incluido en el paquete de tecnología Xampp, el cual es de plataforma libre.

Como framework JavaScript se resuelve utilizar Sencha Ext JS, ya que debido a sus características es uno de los principales framework de su tipo. Además no hay necesidad de configurar el código para que funcione en cualquier navegador. También se seleccionó ya que permite crear aplicaciones complejas utilizando componentes predefinidos.

Como Sistema Gestor de Base de Datos se determina el uso de MySQL 5.0, por su rapidez cuando se trabaja con una base de datos pequeña o mediana, además de estar totalmente integrado con los entornos de desarrollo seleccionados.

Capítulo 2: Planeación y Diseño

2.1 Introducción

En el presente capítulo se expone la propuesta de solución, para ello se definen las funcionalidades y características con las que contará el sistema y también se dará a conocer las historias de usuarios y prototipos de interfaz de usuarios para el mejor entendimiento de la dinámica del sistema.

2.2 Lista de reserva del producto

En la lista de reserva del producto se pueden observar las funcionalidades y características que el sistema deberá contemplar una vez se haya concluido. También incluye la prioridad que tendrá cada requisito a la hora de desarrollarlos, esta prioridad es asignada por el cliente con el fin de obtener en cada entrega mayor valor del negocio. Esta lista se crea con la intención de tener documentadas todas las condiciones que deberá cumplir y las características a tener el sistema una vez esté realizado.

Tabla 2.1: Funcionalidades del sistema.

Código	Descripción de las funcionalidades del sistema	Prioridad
RF1	Insertar trabajadores.	alta
RF2	Modificar trabajadores.	media
RF3	Buscar trabajadores.	media
RF4	Eliminar trabajadores.	baja
RF5	Insertar brigada.	alta
RF6	Modificar brigada.	media
RF7	Buscar brigada.	media
RF8	Eliminar brigada	baja
RF9	Modificar Jefe de área.	media
RF10	Autenticar usuario.	alta
RF11	Insertar usuario.	alta

RF12	Modificar usuario.	media
RF13	Buscar usuario.	media
RF14	Eliminar usuario.	baja
RF15	Insertar área de administración.	alta
RF16	Modificar área de administración.	media
RF17	Buscar área de administración.	media
RF18	Eliminar área de administración.	baja
RF19	Insertar material.	alta
RF20	Modificar material.	media
RF21	Buscar material.	media
RF22	Eliminar material.	baja
RF23	Insertar tipo de material.	alta
RF24	Modificar tipo de material.	media
RF25	Buscar tipo de material.	media
RF26	Eliminar tipo de material.	baja
RF27	Insertar Orden de Trabajo.	alta
RF28	Modificar Orden de Trabajo.	media
RF29	Buscar Orden de Trabajo.	media
RF30	Eliminar Orden de Trabajo.	baja
RF31	Insertar Actividad General.	alta
RF32	Modificar Actividad General.	media
RF33	Buscar Actividad General.	media
RF34	Eliminar Actividad General.	baja
RF35	Insertar Reporte.	alta
RF36	Modificar Reporte.	media
RF37	Buscar Reporte.	media
RF38	Eliminar Reporte.	baja
RF39	Mostrar listado de trabajadores.	media
RF40	Mostrar listado de brigada	media

RF41	Mostrar listado de usuario.	media
RF42	Mostrar listado de Jefe de área.	media
RF43	Mostrar listado de Área de administración.	media
RF44	Mostrar listado de Orden de Trabajo.	media
RF45	Mostrar listado de materiales.	media
RF46	Mostrar listado de tipo de material.	media
RF47	Mostrar listado de Actividad General.	media
RF48	Mostrar listado de Reporte.	media
RF49	Mostrar los trabajadores de una brigada	media
RF50	Agregar un trabajador a una brigada	alta
RF51	Agregar Materiales a una Orden de Trabajo	alta
RF52	Asignar actividad(es) a una brigada	alta
RF53	Mostrar el plan de Actividades Generales en un período	alta
RF54	Calcular el gasto de materiales en un período	alta
RF55	Mostrar el listado de Actividades Imprevistas en un período	alta
RF56	Mostrar el listado de Actividades por estado	alta
RF57	Mostrar el plan de Actividades de una brigada	alta
Características del Sistema		
	Usabilidad	
RNF1	Facilidad de uso por parte de los usuarios: el sistema debe presentar una interfaz amigable que permita la fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Debe, además, ser una interfaz de manejo cómodo que posibilite a los usuarios sin experiencia una rápida adaptación.	
RNF2	Especificación de la terminología utilizada: el sistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por parte del cliente de la herramienta de trabajo.	

RNF3	Emplear perfiles de usuario: diferenciar las interfaces y opciones para los usuarios que accedan al sistema según los diferentes roles que estos tengan.
RNF4	Menús: el sistema debe presentar una serie de menús que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.
RNF5	Interfaces intuitivas: Potencialidades de capacitación orientadas a interfaces intuitivas, lo que enaltece la posibilidad de que el usuario aprenda mediante el uso y explotación de la herramienta.
	Fiabilidad
RNF6	Seguridad de las bases de datos: la seguridad de la base de datos está a nivel de roles, con el fin de mantener la integridad de los datos en función del acceso de cada uno de ellos, trayendo consigo además la protección de la información.
RNF7	Servicios web restringidos: los servicios web que brinde el sistema deben estar restringidos a grupos de usuarios definidos y aprobados previamente.
RNF8	Políticas de seguridad por usuario y rol: el sistema debe contar con un grupo de políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado.
	Eficiencia
RNF9	Cantidad de conexiones: el sistema debe soportar una conexión simultánea de más de 3 0 usuarios.
	Restricciones de diseño
RNF10	Servidor de base de datos con MySQL 5.3 o superior.
RNF11	Servidor de aplicaciones: Apache 2.2 o superior.

RNF12	Navegador web: Internet Explorer 7 o superior, Mozilla Firefox 2.3 o superior.
	Requisitos para la documentación de usuarios en línea y ayuda del sistema
RNF13	Documentación actualizada del grupo de desarrollo: se precisa que la documentación del sistema esté actualizada en todos los aspectos, fases de trabajo y ciclos de desarrollo del mismo, permitiendo con ello un respaldo tanto ingenieril como legal del desarrollo de dicho sistema.
	Componentes Comprados
RNF14	Este acápite no procede, para el desarrollo del sistema no fue necesario comprar ningún componente.
	Interfaz
RNF15	Interfaz web: la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.
	Interfaces Hardware
RNF16	Comunicación entre el cliente y el servidor de aplicaciones: se realiza empleando el protocolo HTTP.
	Interfaces de Comunicación
RNF17	Este acápite no procede, el sistema no presenta comunicaciones a otros sistemas o dispositivos como las redes de área locales.
	Requisitos de Licencia

RNF18	No hay ninguna restricción de uso para el sistema.
	Estándares Aplicables
RNF19	Referirse al epígrafe X: Estándar de código, en el mismo se especifica los requisitos de estándares aplicables

2.4 Historias de Usuarios

Teniendo como entrada principal la lista de reserva del producto se describieron las historias de usuario (HU) del Departamento de Inversiones en el área de Mantenimiento. Estas HU son la técnica utilizada en XP para especificar los requisitos del software, cada una de ellas debe estar lo bastante delimitada para que los programadores puedan implementarla en un corto plazo de tiempo.

2.4.1 Descripción de las historias de usuario

Tabla 2.2: Historia de Usuario " Gestionar Orden de Trabajo"

Historia de Usuario	
Código: HU9.	**Nombre Historia de Usuario:** Gestionar Orden de Trabajo
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ítems RF27, RF28, RF29, RF30, RF45	
Programador: Luis Guerrero López.	**Iteración Asignada:** Primera
Prioridad: Alta	**Puntos Estimados:**1
Riesgo en Desarrollo: Alto	**Puntos Reales:** 1
Descripción: La historia de usuario permite Gestionar una Orden de Trabajo. Se podrá insertar, modificar, buscar, eliminar y mostrar el listado de una Orden de Trabajo en el sistema.	
Observaciones: Roles. Los roles relacionado con esta función. Confirmado por el cliente.	

Prototipo de interface:

Tabla 2.3: Historia de Usuario " Gestionar Actividad General"

Historia de Usuario	
Código: HU10.	**Nombre Historia de Usuario:** Gestionar Actividad General
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ítems RF31, RF32, RF33, RF34, RF47	
Programador: Luis Guerrero López.	**Iteración Asignada:** Primera
Prioridad: Alta	**Puntos Estimados:**1
Riesgo en Desarrollo: Alta	**Puntos Reales:** 1
Descripción: La historia de usuario permite Gestionar una Gestionar Actividad General. Se podrá insertar, modificar, buscar , eliminar y mostrar el listado de una Gestionar	

Actividad General en el sistema
Observaciones: Roles. Los roles relacionado con esta función. Confirmado por el cliente.
Prototipo de interface:

Tabla 2.4: Historia de Usuario " Mostrar los trabajadores de una brigada”

Historia de Usuario	
Código: HU7.	**Nombre Historia de Usuario:** Mostrar los trabajadores de una brigada
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ítems RF49	
Programador: Luis Guerrero López.	**Iteración Asignada:** Primera

Prioridad: Media	**Puntos Estimados:** 2
Riesgo en Desarrollo: Medio	**Puntos Reales:** 2

Descripción:

La historia de usuario permite Mostrar los trabajadores de una brigada. Se podrá mostrar los trabajadores de una brigada en el sistema.

Observaciones:

Roles. Los roles relacionado con esta función.

Confirmado por el cliente.

Prototipo de interface:

2.5 Planificación de entregas

En esta fase se establece la prioridad de cada HU, y a continuación, se realiza una estimación del esfuerzo necesario de cada una de ellas por parte de los programadores. Se toman acuerdos sobre el contenido de la primera entrega y se

determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de dos a tres meses.

Las estimaciones asociadas a la implementación de las historias se establecen empleando como medida el punto de estimación. Un punto de estimación equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo, trabajan el tiempo planeado sin ningún tipo de interrupción, este punto de estimación que se utiliza para representar la semana ideal, es de 5 días. Las historias generalmente tienen un valor de 1 a 3 puntos. Además, se mantiene un registro de la velocidad de desarrollo, establecida por puntos de iteración, basado fundamentalmente en la suma de los puntos de estimación correspondientes a las HU, que fueron terminadas en la última iteración.

2.5.1 Estimación del esfuerzo por historias de usuario

Para el buen desarrollo de la aplicación se realizó una estimación para cada una de las historias de usuario identificadas, y se obtienen los resultados que se muestran a continuación:

Tabla 2.5 Estimación de esfuerzos por Historia de Usuario

Historias de usuarios	**Puntos de estimación**
Autenticar usuarios	1
Gestionar trabajadores	1
Gestionar brigada	1
Modificar Jefe de área	1
Gestionar área de administración	1
Gestionar materiales	1
Gestionar tipo de material	2

Gestionar Orden de Trabajo	1
Gestionar Actividad General	1
Gestionar reporte	1
Mostrar los trabajadores de una brigada	2
Agregar un trabajador a una brigada	1
Agregar Materiales a una Orden de Trabajo	1
Asignar actividad(es) a una brigada	1
Mostrar el plan de Actividades Generales en un período	1
Calcular el gasto de materiales en un período	1
Mostrar el listado de Actividades Imprevistas en un período	1
Mostrar el listado de Actividades por estado	1
Mostrar el plan de Actividades de una brigada	1

2.5.2 Planificación de iteraciones

A partir de las HU antes expuestas y la estimación del esfuerzo propuesto para la realización de las mismas, se procede a realizar la planificación de la etapa de implementación del sistema, apoyándose en el tiempo e intentando concentrar las funcionalidades relacionadas en una misma iteración. En este plan se establece cuántas iteraciones serán necesarias realizar sobre el sistema para su terminación. El plan de iteraciones puede contener indicaciones sobre cuáles HU

se incluirán en un *release*, lo cual debe ser consistente con el contenido de una o dos iteraciones.

En relación con lo antes tratado se decide realizar el sistema en tres iteraciones, las cuales se explican de forma detallada a continuación:

Primera iteración:
Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo de la aplicación. Al concluir dicha iteración se contará con todas las funcionalidades 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 39, 40, 41, 42, 43, 49 y 50 las cuales hacen alusión a la creación de la brigada como sus trabajadores u su jefe respectivo, se gestionarán los usuarios y sus permisos, y también se definirán las áreas con sus responsables. Además se tendrá la primera versión de prueba, que incorporan todas las funcionalidades antes vistas, la cual se presentará al cliente con el objetivo de obtener una retroalimentación del mismo para posteriores iteraciones del producto.

Segunda Iteración:
Esta iteración tiene como finalidad desarrollar las funcionalidades 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 51 y 52. Las mismas brindan que se gestionen en la Orden de Trabajo y en las Actividades Generales, gestionando los materiales y los tipos de materiales creando su Reporte, y se asignarán actividades a una brigada.

Tercera Iteración:
En la tercera y última iteración se llevaran a cabo las funcionalidades 53, 54, 55, 56 Y 57. Con estas se mostrarán el plan de Actividades generales de un período, el listado de Actividades Imprevistas en un período, el listado de Actividades por estado, el plan de Actividades de una brigada, así como calcular el gasto de materiales en un período.

2.5.3 Plan de iteraciones

Después de realizar un análisis de las historias de usuarios y de priorizar cada una de estas se realizó el siguiente plan de iteración. El mismo tiene como entrada la relación de historias de usuario previamente definida.

Tabla 2.6: Plan de duración de iteraciones

Iteraciones	Descripción de la Iteración	Orden de la HU a implementar	Duración de cada HU(días)	Duración total(días)
Primera	En esta iteración se van a implementar las HU que sean relacionadas con la creación de la brigada como sus trabajadores y su jefe respectivo, se gestionarán los usuarios y sus permisos y también se definirán las áreas con sus responsables.	Gestionar trabajadores	4	21 días (3 semanas)
		Gestionar brigada	4	
		Modificar Jefe de Brigada	1	
		Autenticar usuario	2	
		Gestionar usuario	4	
		Gestionar área de administración	4	
		Mostrar los trabajadores de una brigada	1	
		Agregar un trabajador a una brigada	1	

Segunda	En esta iteración se van a implementar las HU que se gestionen en la Orden de Trabajo y en las Actividades Generales, añadiendo los materiales y los tipos de materiales, así como los reportes pertinentes	Gestionar Orden de Trabajo	4	21 días (3 semanas)
		Gestionar Actividad General	4	
		Gestionar Reporte	3	
		Gestionar Material	2	
		Gestionar tipo de material	2	
		Agregar Materiales a una Orden de Trabajo	3	
		Asignar actividad(es) a una brigada	3	
Tercera	En esta iteración se van a implementar las HU se mostrarán el plan de Actividades generales de un período, el listado de Actividades Imprevistas en	Mostrar el plan de Actividades Generales en un período	4	21 días (3 semanas)
		Calcular el gasto de materiales en un período	5	
		Mostrar el listado de Actividades Imprevistas en un período	4	
		Mostrar el listado de Actividades por estado	4	
		Mostrar el plan de Actividades de una	4	

	un período, el listado de Actividades por estado, el plan de Actividades de una brigada, así como calcular el gasto de materiales en un período.	brigada		
Total			63 días	9 semanas (2 meses y una semana)

2.6 Diseño

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como tarjetas CRC, las cuales se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño.

2.6.1 Tarjetas CRC

Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU.

Tabla 2.7: Tarjeta CRC: Orden de Trabajo

Nombre de la Clase: Orden de Trabajo	
Tipo de clase	pública
Descripción	Guarda las Órdenes de Trabajo al sistema
Atributos	
Id_orden	
Id_actividad	
no_orden	
Fecha_expendida	
Fecha_registro	
Taller de servicio	
Solicitante	
Descripción_trabajo	
Responsabilidades Colaboradores	
Insertar Orden de Trabajo	Actividad General

Mostrar Orden de Trabajo	
Modificar Orden de Trabajo	
Eliminar Orden de Trabajo	
Mostrar listado de Orden de Trabajo	

Tabla 2.8: Tarjeta CRC: Actividad General

Nombre de la Clase: Actividad General	
Tipo de clase	pública
Descripción	Guarda las Actividades Generales al sistema
Atributos	
id_actividad	
id_area	
id_brigada	
id_estado	
fecha_inicio_est	
fecha_fin_est	
fecha_inicio_real	
fecha_fin_real	
tiempo_realiza	

Responsabilidades Colaboradores	
Insertar Actividad General	Área de Administración
Mostrar Actividad General	
Modificar Actividad General	Estado
Eliminar Actividad General	
Mostrar listado de Actividad General	

Conclusiones

Como resultado de este capítulo se abordó la fase de planeación y diseño donde se delinearon las historias de usuarios (HU) con la participación del cliente, se llevó a efecto la planificación de iteraciones de cada HU a partir de la estimación del esfuerzo necesario de las mismas. Se presentaron además las principales clases que se utilizarán en el desarrollo de la aplicación a través de las llamadas tarjetas CRC culminando así esta fase y se determina que el equipo de trabajo está listo para pasar a la siguiente etapa de desarrollo.

Capítulo 3 Desarrollo y Pruebas

En este capítulo se siguen las fases de Desarrollo y Pruebas conformadas en la Metodología XP. Se muestra un modelo de datos correspondiente a la aplicación concluyente, se realiza a través del desarrollo de las iteraciones un desglose de las tareas a realizar para cada una de las HU expuestas en el capítulo anterior. Por último se describen las pruebas realizadas, mostrando las diferentes respuestas al aplicarse cada una de las funcionalidades.

Introducción

En este capítulo se adentra en las últimas fases de la metodología XP Desarrollo y Pruebas de la aplicación. En las cuales se encuentran como encargado el programador en la parte de desarrollo y el cliente en la realización de las pruebas de aceptación.

A continuación mostramos el modelo de datos utilizado en el sistema:

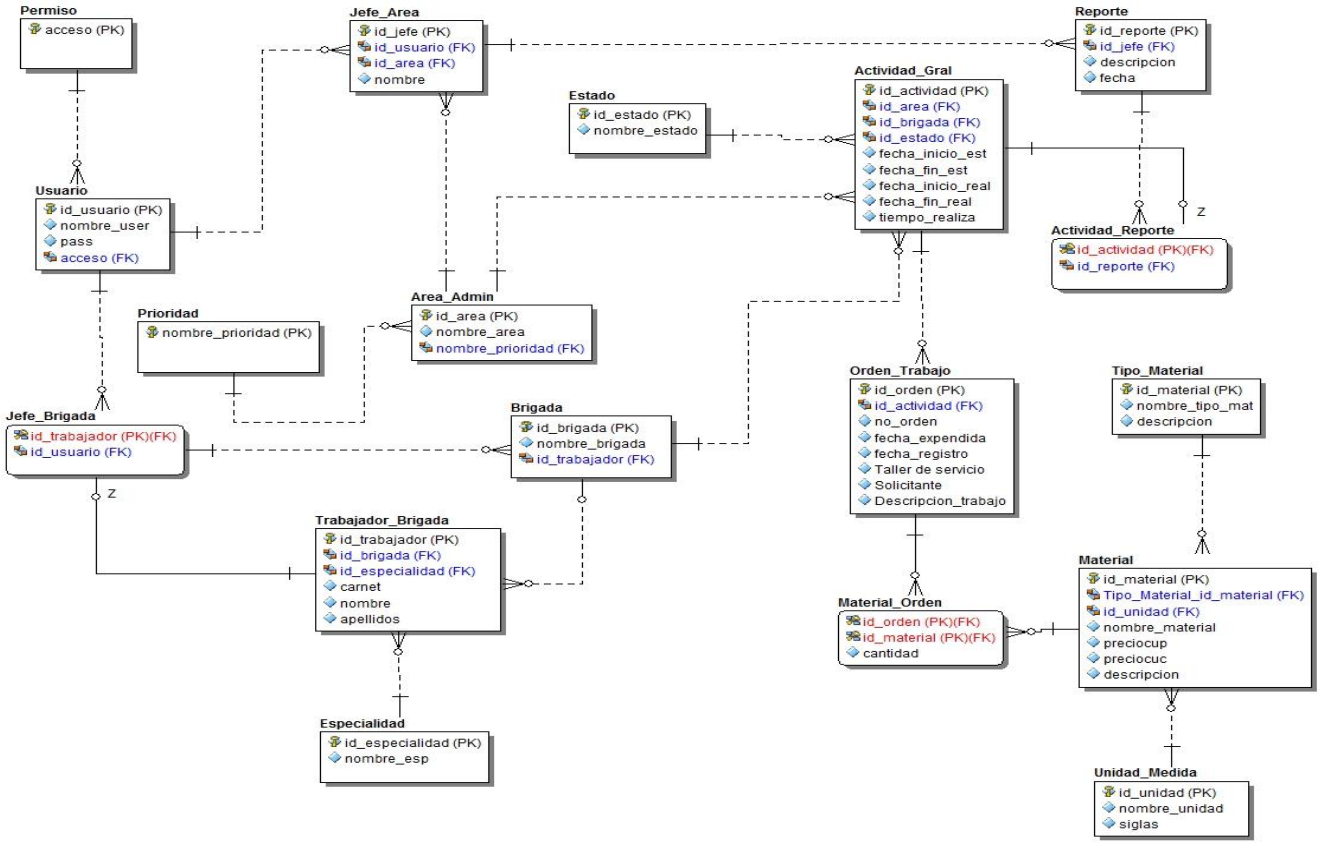


Figura 3.1 Modelo de datos

3.3 Desarrollo de las iteraciones

En el capítulo anterior se detallaron todas las historias de usuarios de cada una de las iteraciones en la que se va a desarrollar el sistema, y para la realización de cada una de las HU se realizará una revisión al plan de iteraciones, para ver si pueden ocurrir cambios.

3.3.1 Tareas por historias de usuario

Dentro del contenido de este plan, las HU se descomponen en tareas de programación o ingeniería, y a su vez estas son asignadas al equipo de desarrollo para su implementación. Estas se representan mediante las tarjetas de tareas.

Tabla 3.1: Distribución de tareas por historias de usuario

Historia de usuario	Tareas
Gestionar trabajador	Insertar trabajador Mostrar trabajador Modificar trabajador Eliminar trabajador Mostrar listado de trabajadores
Gestionar brigada	Insertar brigada Mostrar brigada Modificar brigada
	Eliminar brigada Mostrar listado de brigada
Modificar Jefe de área	Modificar Jefe de área

	Mostrar listado de Jefe de área
Autenticar usuario	Autenticar usuario
Gestionar Usuario	Insertar Usuario Mostrar Usuario Modificar Usuario Eliminar Usuario Mostrar listado de Usuario
Gestionar área de administración	Insertar área de administración Mostrar área de administración Modificar área de administración Eliminar área de administración Mostrar listado de área de administración
Gestionar materiales	Insertar materiales Mostrar materiales Modificar materiales Eliminar materiales Mostrar listado de materiales
Gestionar tipo de material	Insertar tipo de material Mostrar tipo de material

	Modificar tipo de material Eliminar tipo de material Mostrar listado de tipo de material
Gestionar Orden de Trabajo	Insertar Orden de Trabajo Mostrar Orden de Trabajo Modificar Orden de Trabajo Eliminar Orden de Trabajo Mostrar listado de Orden de Trabajo
Gestionar Actividad General	Insertar Actividad General Mostrar Actividad General Modificar Actividad General Eliminar Actividad General Mostrar listado Actividad General
Gestionar reporte	Insertar reporte Mostrar reporte Modificar reporte Eliminar reporte Mostrar listado de reporte
Mostrar los trabajadores de una brigada	Mostrar los trabajadores de una brigada
Agregar un trabajador a una brigada	Agregar un trabajador a una brigada

Agregar Materiales a una Orden de Trabajo	Agregar Materiales a una Orden de Trabajo
Asignar actividad(es) a una brigada	Asignar actividad(es) a una brigada
Mostrar el plan de Actividades Generales en un período	Mostrar el plan de Actividades Generales en un período
Calcular el gasto de materiales en un período	Calcular el gasto de materiales en un período
Mostrar el listado de Actividades Imprevistas en un período	Mostrar el listado de Actividades Imprevistas en un período
Mostrar el listado de Actividades por estado	Mostrar el listado de Actividades por estado
Mostrar el plan de Actividades de una brigada	Mostrar el plan de Actividades de una brigada

3.3.2 Tareas de ingeniería

En XP generalmente cada historia se divide en tareas de ingeniería o tareas de programación. Las tareas no tienen que ser entendidas necesariamente por el cliente, pues las mismas, sólo son utilizadas por los miembros del equipo de desarrollo, por lo que pueden ser escritas en lenguaje técnico. Estas se crean para obtener una mejor planificación y cumplir con las funcionalidades básicas que luego conformaran las funcionalidades generales de cada historia de usuario.

Tabla 3.2: Tarea de programación Insertar Orden de Trabajo

Tarea de Programación	
No. Tarea: 27	No. Historia: 9
Nombre de la tarea: insertar Orden de Trabajo.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 21/03/2014	Fecha fin: 21/03/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite insertar la Orden de Trabajo que va a guardar en el sistema.	

Tabla 3.3: Tarea de programación Insertar Actividad General

Tarea de Programación	
No. Tarea: 31	No. Historia: 10
Nombre de la tarea: Insertar Actividad General.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/03/2014	Fecha fin: 26/03/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite insertar la Actividad General que va a guardar en el sistema.	

3.4 Pruebas

En la metodología XP las pruebas juegan un papel fundamental, pues estas permiten las pruebas de aceptación. El desarrollo constante de las pruebas da

lugar a que se desarrolle un software con mayor calidad dando una mayor seguridad de lo que se está haciendo.

En la metodología escogida hay dos tipos de prueba; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración verificando así que se cumplió la funcionalidad requerida por el cliente.

3.4.1 Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que la HU ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los arquetipos anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación.

La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación.

Tabla 3.4: Prueba para comprobar la entrada de datos de la Orden de Trabajo

Prueba de aceptación
Historia de usuario # 9: Gestionar Orden de Trabajo
Nombre: Prueba para comprobar la entrada de datos de la Orden de Trabajo.
Descripción: Validación de entrada de los datos de la Orden de Trabajo.
Condiciones de ejecución: El usuario debe introducir los datos de una Orden de Trabajo.
Entrada/Pasos de ejecución: El usuario escribe los datos de la Orden de Trabajo.
Resultado esperado: ➢ Si se introducen los datos correctamente se insertará el Orden de Trabajo. Se emite un mensaje de error en caso de que: ➢ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

Tabla 3.5: Prueba para comprobar la entrada de datos de la Actividad General

Prueba de aceptación
Historia de usuario # 10: Gestionar Actividad General
Nombre: Prueba para comprobar la entrada de datos de la Actividad General.
Descripción: Validación de entrada de los datos de la Actividad General.

Condiciones de ejecución: El usuario debe introducir los datos de una Actividad General.
Entrada/Pasos de ejecución: El usuario escribe los datos de la Actividad General.
Resultado esperado: ➢ Si se introducen los datos correctamente se insertará el Actividad General. Se emite un mensaje de error en caso de que: ➢ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

Conclusiones

Con el desarrollo de este capítulo se llevó a cabo la fase de desarrollo y prueba donde se presenta el modelo de datos de la aplicación a obtener, logrando una visión detallada de sus atributos y las relaciones entre sus clases, además, las principales clases mediante el empleo de las tarjetas CRC. Se realiza el desarrollo de las iteraciones a partir de la distribución de tareas de ingeniería por HU, y se les practica las pruebas de aceptación a las funcionalidades del sistema.

Capítulo 4 Estudio de Factibilidad

En la actualidad para realizar de manera satisfactoria el desarrollo de cualquier proyecto se hace necesario el estudio de factibilidad para tener en cuenta una valoración de los costos a incurrir determinando si será factible o no desarrollar el mismo.

En general los productos informáticos no están exentos de posibles riesgos en la concepción del proyecto, por lo que es válido minimizar de forma razonable recursos humanos, materiales y financieros, de ahí que es de vital importancia estimar la relación costo – beneficio, así como el esfuerzo, capital humano y el tiempo de desarrollo que se demanda en la ejecución de los mismos.

En este capítulo se expone el estudio de factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo. Se presentarán los efectos económicos, los beneficios y costos intangibles.

4.1 Evaluación Costo – Beneficio

Para el estudio de la factibilidad de este proyecto fue utilizada la metodología Costo Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores: [5]

Costo: que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados.

Eficiencia: se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo). [6]

Esta es una de las partes más importantes en la elaboración de cualquier proyecto ya que haciendo un correcto estudio de factibilidad se puede ahorrar meses e

incluso años de trabajo, hasta evitar poner en duda la reputación profesional si se realiza un sistema mal planificado desde una etapa temprana.

4.2 Efectos Económicos

- Efectos directos
- Efectos indirectos
- Efectos externos
- Intangibles

4.2.1 Efectos directos:

Positivos:

- Disminución de la acumulación de materiales impresos relacionados con los reportes, actividades, órdenes de trabajo.
- Ahorro de tiempo en la búsqueda de información de reportes, actividades, órdenes de trabajo.
- Seguridad al guardar la información.
- Se cuenta con una herramienta capaz de mantener la seguridad e integridad de los datos que se procesan.

Negativos:

- Para el uso de esta aplicación implementada en plataforma Web se necesitará que la misma sea ejecutada con un navegador compatible, recomendamos para mejor visibilidad el Mozilla Firefox en su versión más actual, porque es con el que se trabajó en la elaboración del producto por lo que el diseño está adaptado a este tipo de navegador.

4.2.2 Efectos indirectos:

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

4.2.3 Efectos externos:

- Se obtendrá un producto disponible que le facilitará gran parte del trabajo a los usuarios encargados de la gestión de la información del Departamento de Inversiones en el área de Mantenimiento del ISMMM.

4.2.4 Intangibles

En la estimación económica siempre hay elementos como perjuicio o beneficio, pero al momento de darle valor en unidades monetarias esto resulta difícil o prácticamente imposible. [6]

A fin de medir con precisión los efectos, deberán considerarse dos situaciones: la situación sin proyecto y la situación con proyecto.

Situación sin proyecto (Solución manual):

Son complejas las actividades de llenar una Orden de Trabajo para la actividad que se quiera realizar, mostrar el listado de todos los trabajadores, usuarios, Jefes de Brigadas y de áreas, de las brigadas, de los reportes, de los materiales, etc., ya que no se cuenta con la infraestructura informática para que el encargado de este trabajo acceda a los datos y los llene él mismo o que se tengan registrados todos costos de las operaciones realizadas, trayendo consigo no tener una vía rápida de operación y análisis de la información.

Situación con proyecto (Solución automatizada):

Mediante este sistema es posible gestionar y controlar todo el flujo de información concerniente al Departamento de Inversiones en el área de Mantenimiento. Queda garantizada la centralización de los datos teniendo además una accesibilidad fácil y rápida de todas las partes involucradas en el proceso.

Costos:

- Resistencia al cambio

Beneficios:

- Mayor comodidad y organización de información para los usuarios.
- Conectividad desde cualquier ordenador que esté enlazado a la red del centro.
- Mayor rapidez a la hora de realizar el proceso.
- Mejora en la accesibilidad y visibilidad de la información.

4.3 Ficha de costo de un producto informático.

Para determinar el costo económico del proyecto se utilizará el Procedimiento para elaborar una ficha de costo de un producto informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costos en Moneda Libremente Convertible:

- **Costos Directos**
  1. Compra de equipos de cómputo: No procede.
  2. Alquiler de equipos de cómputo: No procede.
  3. Compra de licencia de Software: No procede.
  4. Depreciación de equipos: $64.93 mensual
  5. Materiales directos: No procede.

  Total: $64.93 CUC

Total General: $ 64.94 CUC.

- **Costos indirectos**
  1. Formación del personal que elabora el proyecto: No procede.
  2. Gastos en llamadas telefónicas: No procede.
  3. Gastos para el mantenimiento del centro: No procede.
  4. Know How: No procede.
  5. Gastos en representación: No procede.

  Total: $0.00.
- **Gastos de distribución y venta.**
  1. Participación en ferias o exposiciones: No procede.
  2. Gastos en transportación: No procede.
  3. Compra de materiales de propagandas: No procede.

Total: $0.00.

Costos en Moneda Nacional:

- **Costos Directos.**
  1. Salario del personal que laborará en el proyecto: $100.00 mensual.
  2. El 12% del total de gastos por salarios se dedica a la seguridad social: No procede.
  3. El 0.09% del salario total, por concepto de vacaciones a acumular: No procede.
  4. Gasto por consumo de energía eléctrica: 68.64 kW mensual (1 kW = $ 0.67) 68.64 * 0.67 = $ 45.99.
  5. Gastos en llamadas telefónicas: No procede.
  6. Gastos administrativos: No procede.

- **Costos indirectos**
  1. Know How: $ 0.0.

     Total: $ 145.99.

Total General: $ 145.99

El análisis de costo-beneficio se basa en un principio muy simple:

Compara los beneficios y los costos de un proyecto particular y si los primeros exceden a los segundos entrega un elemento de juicio inicial, que indica su aceptabilidad. Mientras que el análisis costo-efectividad sigue la misma lógica, compara los costos con las potencialidades de alcanzar más eficientemente los objetivos no expresables en moneda; si no en productos. Para esta técnica es imprescindible definir una variable directa que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo, el tiempo en minutos empleado para realizar las actividades de capacitación en la empresa y la variable seria, complejidad de las pruebas que se desarrollan durante este proceso.

Valores de la variable (Solución manual)

1. Gestionar información de las actividades. (10 min)
2. Gestionar información de las órdenes de trabajo. (11 min)
3. Buscar información de los trabajadores. (8 min)
4. Crear reportes. (15 min)

Valores de la variable (Solución con el sistema)

1. Gestionar información de las actividades. (2 min)
2. Gestionar información de las órdenes de trabajo. (2 min)
3. Buscar información de los trabajadores. (0.4 min)
4. Crear Reportes. (1 min).

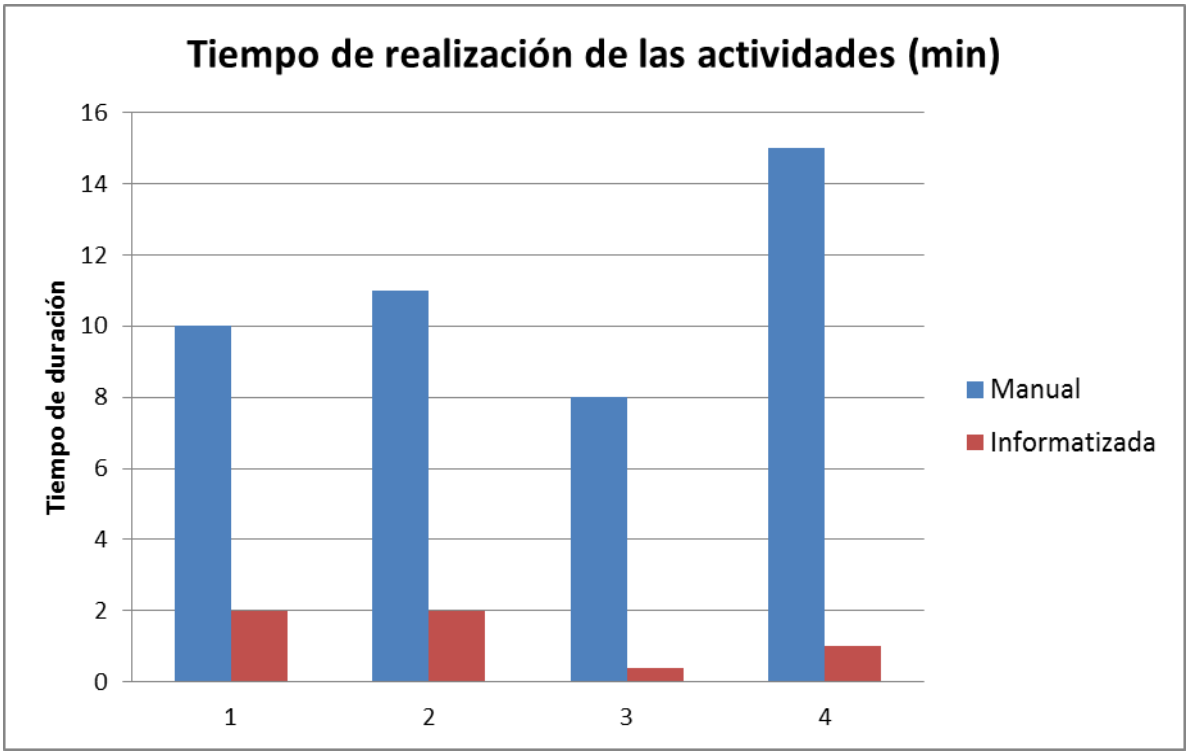


Figura 4.1 Gráfica de la solución sin el producto y solución con el producto.

Teniendo en cuenta los resultados reflejados en la gráfica queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de variables) y el tiempo que demora la solución del mismo de forma manual y automatizada.

Conclusiones

En este capítulo se realizó el estudio de factibilidad del proyecto mediante la Metodología Costo Efectividad (Beneficios). Se analizaron todos los factores directos, indirectos, externos e intangibles., se analizaron los efectos económicos, los beneficios y costos, además se calculó el costo de ejecución del proyecto mediante la Ficha de Costo, la cual arrojó como resultado $ 64.94 CUC y $ 145.99 MN mensual. Se analizó además el tiempo empleado para realizar las operaciones sin y con el proyecto, demostrándose así la factibilidad del producto.

Conclusiones Generales

Al concluir la investigación se han cumplido el objetivo y las tareas propuestas:

Se elaboró el marco teórico metodológico que fundamenta la investigación, esto permitió que quedara identificada la situación problémica existente y se establecieran las bases para el posterior diseño e implementación de la aplicación.

Se estableció el estado del arte de la investigación, resaltándose que las herramientas existentes no responden a las necesidades reflejadas por el entorno, por lo que se decide el desarrollo de una aplicación informática orientada a la situación actual.

Se realizó un levantamiento en el que se identificaron los principales elementos del proceso de gestión de mantenimientos del Departamento Inversiones del ISMMM, determinándose las principales funcionalidades a implementar en la propuesta de solución.

Con el desarrollo de la aplicación web para la gestión de la información en el Departamento de Inversiones en el área de Mantenimiento del ISMMM se dio cumplimiento al objetivo general de la presente investigación, pues se obtuvo como resultado una herramienta informática donde la información relevante se encuentra disponible para los responsables de las áreas, permitiendo una planificación adecuada de los recursos humanos.

Se efectuó un estudio de la factibilidad técnica, económica, y operativa del software en cuestión; se calculó el costo de ejecución del proyecto mediante la Ficha de Costo, la cual arrojó como resultado $ 64.94 CUC y $ 323.88 MN por el período en que se realizó la aplicación, además se analizó el tiempo empleado para realizar las operaciones, demostrándose así la factibilidad del producto.

Recomendaciones

El autor de la presente investigación ha considerado pertinente ofrecer las siguientes recomendaciones:

- Realizar la implantación del software en el ISMMM.
- Realizar encuestas de satisfacción a los usuarios finales que permitan realizarle nuevas versiones a la aplicación web teniendo en cuenta las recomendaciones hechas por ellos.
- Incorporar al sistema construido funcionalidades que permitan involucrar el proceso de inversiones gestionado por el Departamento de Inversiones, lográndose una herramienta más completa.

Referencias bibliográficas

1. *Software para la planificación y gestión del mantenimiento (PGM | Win).* [En línea] www.seprisa.net marzo de 2014.
2. *Software Evaluador de Proyectos de Inversión (S.E.P.I.)*
3. *La gestión del mantenimiento en entidades turísticas (CEIM).* [En línea] http://www.ispjae.edu.cu marzo de 2014
4. *Software Offimantpro.* [En línea] http://www.cadenagramonte.cu/articulos marzo de 2014.
5. **Lecusay, Pablo.** *Sistema informático para favorecer el proceso de selección de personal en el instituto superior minero metalúrgico de moa.* Moa : ISMMM, 2012.
6. **Silva, Ing. Exneider proenza.** *Sistema de Costos de Calidad para la Empresa .* 2010.
7. Vegas Jesús El Servidor Web [Libro]. - [s.l.]: Universidad de Valladolid, 2002.
8. Htmlpoint.com [En línea] // ¿Qué es Apache? - 2008. - febrero de 2014. - http://www.htmlpoint.com/faq/apache/01.htm.
9. Ciberaula.com Ciberaula.com [En línea] // Una Introducción a APACHE. - 2005.- febrero de 2014. - http://linux.ciberaula.com/articulo/linux_apache_intro.
10. HERNÁN RUIZ, Marcelo. Programación Web Avanzada [Libro]. La Habana, Cuba. Editorial Félix Varela, 2006.
11. **Meriño, Pricido Fidel Rusiaux.** *Herramienta Informática para la gestión de información sobre Incidentes y Accidentes Ambientales en la Empresa Che Guevara.* Moa : ISMMM, 2013.
12. http://es.wikipedia.org/wiki/Framework.htm marzo de 2014

13. HERNÁN RUIZ, Marcelo. Programación Web Avanzada [Libro]. La Habana, Cuba. Editorial Félix Varela, 2006.

14. **Jeffries, R.** What is Extreme Programming? [Online] [Cited: febrero de 2014.] http://www.xpprogramming.com/xpmag/whatisxp.htm.

15. **Pressman, Roger S.** *Ingeniería de Softwar, un enfoque práctico.* s.l. : McGraw-Hill, 2011.

16. **Gabriel Duarte.** Preguntale al Gabo. [Online] [Cited: febrero de 2014.] http://gabrielduarte77.wordpress.com/category/yii-framework.

17. EmbarcaderoER/Studio. [Online] [Cited: febrero de 2014.] http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=115375597.

Bibliografía

1. [BECK, 1999]. Kent Beck. *Extreme Programming Explained: Embrace Change*, Pearson Education 1999. Addison-Wesley, 2000.

2. Rodríguez Pérez, Agustín. *SIGEP: Sistema para La Gestión del Presupuesto en el ISMMM*. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, junio 2012.

3. Hernández Luque, Eylin. *Aplicación Web para la gestión de información de instrumentos meteorológicos del Instituto de Meteorología.* Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de las Ciencias Informáticas, Ciudad de la Habana, mayo 2007.

4. Calvo Burke, Aurelio. Sistema de Gestión de Información sobre Derrames de Hidrocarburos en el puerto de Moa. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, julio 2013.

5. [HERNANDEZ, 2006]. Edelsys Hernández Meléndrez. *Como escribir una tesis.* Escuela Nacional de Salud Pública, 2006.

6. Schenone Marcelo Hernán *Diseño de una Metodología Ágil de Desarrollo de Software.* Tesis de Grado en Ingeniería en Informática. Universidad de Buenos Aires, argentino, abril 2004.

7. [FREEMAN, 2005].Elisabeth Freeman. *A Brain-Friendly Guide to HTML & CSS*. Head First Html With Css And Xhtml. O´Reilly, 2005.

8. [MARTEENS, 1999]. Ian Marteens. *La cara oculta de C++ Builder*. Madrid, 1999.

9. Manual de SQL.WebExperto.COM, 2001.

10. [CANOS; LETELIER; PENADES, 2005]. H. Canós, José, Letelier, Patricio, Penadés, M. Carmen. Metodologías Ágiles en el desarrollo de software. Universidad Politécnica de Valencia, 2005.

Bibliografía

11. Peñaloza B, Leisewitz T, Bastías G, Zárate V, Depaux R, Villarroel L, et al. Metodología para la evaluación de la relación costo-efectividad en centros de atención primaria de Chile. Revista Panam. Salud Pública. 2010.
12. [LETELIER; PENADES, 2005]. Letelier, Patricio; Penadés, Mª Carmen. Metodologías ágiles para el desarrollo de software: Extreme Programming (XP). Universidad Politécnica de Valencia, 2005.
13. [BEIGHLEY; MORRISON, 2005]. Beighley, Lynn, Morrison, Michael. Head First PHP & MySQL. O'Reilly, 2005.
14. GDP. *Manual de PHP*. Grupo de Documentación de PHP, 2002.
15. [FERNANDEZ, 2002]. Gerardo Fernández Escribano. *Introducción a Extreme Programming*. Ingeniería de Software II. 9-12-2002.
16. [WELLING; THOMSON, 2004]. Welling Luke, Thomson, Laura. Desarrollo Web con PHP y MySQL. Anaya Multimedia, 2004.
17. Salazar Pupo, Ricardo. Sistema de gestión de capacitación ECRIN. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, julio 2013.
18. Suárez Falcón, Yuniel. Sistema de gestión de información para la estimación y mitigación de riesgos de desastres en la Provincia Holguín. Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Minero Metalúrgico de Moa "Dr. Antonio Núñez Jiménez", Moa, julio 2013.

Glosario de Términos

Actualización: Cambios que se aplican a un programa, para corregir errores, agregarle funcionalidades, etc.

Análisis costo beneficio: El coste-beneficio es una lógica o razonamiento basado en el principio de obtener los mayores y mejores resultados al menor esfuerzo invertido, tanto por eficiencia técnica como por motivación humana. Se supone que todos los hechos y actos pueden evaluarse bajo esta lógica, aquellos dónde los beneficios superan el coste son exitosos, caso contrario fracasan.

Aplicación: Programa preparado para una utilización específica. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

Aplicación web: Una aplicación Web es un sistema informático que los usuarios usan accediendo a un servidor Web a través de los protocolos de Internet. Las aplicaciones Web son populares por su practicidad del navegador Web como cliente ligero.

API: Una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

ASP: Active Server Pages, también conocido como ASP clásico, es una tecnología de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

Ajax: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Beca: Subvención para realizar estudios o investigaciones.

Glosario de Términos

Becado: Persona que disfruta de una beca.

Beneficio: Ganancia económica que se obtiene de un negocio, inversión u otra actividad mercantil.

Bugs: Errores que se presentan en la aplicación que atentan contra su correcto funcionamiento.

Cliente: Persona, organización o grupo de personas que solicita la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

Costo: En economía el coste o costo es el valor monetario de los consumos de factores que supone el ejercicio de una actividad económica destinada a la producción de un bien o servicio.

Compatibilidad: Condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente tanto directamente o indirectamente mediante un algoritmo.

Concurrencia: Coincidencia, concurso simultáneo de varias circunstancias. Es la simultaneidad en la ejecución de múltiples tareas interactivas. Estas tareas pueden ser un conjunto de procesos o hilos de ejecución creados por un único programa.

Gestión de información: es un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención - por creación o captura, hasta su disposición final - archivada o eliminada. Los procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados.

Herramientas: Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

Historia de usuario: es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario.

HTML: Hyper Text Markup Language, o simplemente HTML, es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web. Si se hace la traducción de su nombre del inglés al castellano, sería “Lenguaje de

Marca de Hipertextos", ya que es justamente un lenguaje que se basa en las marcas para crear los hipertextos.

IDE: Integrated Development Enviroment / Entorno de Desarrollo Integrado. Entorno de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI.

Informatización: Aplicar los conjuntos de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores.

Interfaz: Conjunto de representaciones de operaciones públicas.

Iteraciones: En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio.

Múltiples iteraciones contribuyen a crear un producto completamente integrado.

JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Java: Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

Lógica del negocio: Parte de un sistema que se encarga de las tareas relacionadas con los procesos de un negocio, tales como ventas, control de inventario, contabilidad, etc.

Multiplataforma: Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas o sistemas operativos.

Metodología Ágil: Constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de proyectos que las metodologías convencionales debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración.

Metodología de desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Metodologías tradicionales: Metodologías basadas en procesos.

Navegador web: Un navegador o navegador web (del inglés, web browser) es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que podamos ser capaces de leerla, (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local).

PHP: PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

Proceso: Conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial. *INF*. Programa en ejecución.

Proyecto: Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

Programación extrema: Es una metodología ágil centrada en potenciarlas relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Requisitos: Son las funciones, servicios y restricciones operativas del sistema.

Release: Versión candidata definitiva de un producto de software y se refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

Script: En informática un guión, archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guión.

Sistema: Delimita el mundo sobre el cual se está construyendo el modelo

SQLite: es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 kB) biblioteca en C. SQLite es un proyecto de dominio público

Servidor Web: Un programa que corre sobre el servidor que escucha las peticiones HTTP (Hyper Tex Markup Language) que le llegan y las satisface. Dependiendo del tipo de petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición

Software: Se refiere a los programas y datos almacenados en un ordenador. Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Testing: Proceso de pruebas usado para identificar posibles fallos de implementación, calidad, o usabilidad de un programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Usuario: Persona que interactúa con el sistema

Anexo.1: Historias de Usuarios

Tabla 2.9: Historia de Usuario " Gestionar Materiales”

Historia de Usuario	
Código: HU12.	**Nombre Historia de Usuario:** Gestionar Materiales
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ítems RF19, RF20, RF21, RF22, RF45	
Programador: Luis Guerrero López.	**Iteración Asignada:** Primera
Prioridad: Alta	**Puntos Estimados:**1
Riesgo en Desarrollo: Alto	**Puntos Reales:** 1
Descripción: La historia de usuario permite Gestionar Materiales. Se podrá insertar, modificar, buscar, eliminar y mostrar el listado de los materiales en el sistema.	
Observaciones: Roles. Los roles relacionado con esta función. Confirmado por el cliente.	
Prototipo de interface:	

Tabla 2.10: Historia de Usuario " Gestionar Tipo de Material"

Historia de Usuario	
Código: HU13.	**Nombre Historia de Usuario:** Gestionar Tipo de Material
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ítems RF23, RF24, RF25, RF26, RF46	
Programador: Luis Guerrero López.	**Iteración Asignada:** Primera
Prioridad: Alta	**Puntos Estimados:**1
Riesgo en Desarrollo: Alto	**Puntos Reales:** 1
Descripción: La historia de usuario permite Gestionar Orden de Trabajo. Se podrá insertar, modificar, buscar, eliminar y mostrar el listado de las órdenes de trabajo en el sistema.	

Observaciones: Roles. Los roles relacionado con esta función. Confirmado por el cliente.
Prototipo de interface:

Tabla 2.11: Historia de Usuario " Agregar Materiales a una Orden de Trabajo”

	Historia de Usuario
Código: HU14.	**Nombre Historia de Usuario:** Agregar Materiales a una Orden de Trabajo
Modificación de Historia de Usuario Número: Ninguna.	
Referencia: Ítems RF51	

Programador: Luis Guerrero López.	**Iteración Asignada:** Primera
Prioridad: Media	**Puntos Estimados:**2
Riesgo en Desarrollo: Medio	**Puntos Reales:** 2

Descripción:

La historia de usuario permite Agregar Materiales a una Orden de Trabajo. Se podrá agregar materiales a una Orden de Trabajo en el sistema.

Observaciones:

Roles. Los roles relacionado con esta función.

Confirmado por el cliente.

Prototipo de interface:

NOTA: Las otras historias de usuario se encuentran en el expediente de proyecto.

Anexo.2: Tarjetas CRC

Tabla 2.12: Tarjeta CRC: Área de administración

Nombre de la Clase: Área de administración	
Tipo de clase	pública
Descripción	Guarda las áreas de administración en el sistema
Atributos	
id_area	
nombre_area	
nombre_prioridad	
Responsabilidades Colaboradores	
Insertar Área de administración	Prioridad
Mostrar Área de administración	
Modificar Área de administración	
Eliminar Área de administración	
Mostrar listado de Área de administración	

Tabla 2.13: Tarjeta CRC: Materiales

Nombre de la Clase: Materiales	
Tipo de clase	pública
Descripción	Guarda los materiales en el sistema
Atributos	
id_material	
Tipo_Material_id_material	
id_unidad	
nombre_material	
preciocup	
preciocuc	
descripcion	
Responsabilidades Colaboradores	
Insertar Materiales	Tipo de material
Mostrar Materiales	Unidad de medida
Modificar Materiales	
Eliminar Materiales	
Mostrar listado de Materiales	

Tabla 2.14: Tarjeta CRC: Tipo de material

Nombre de la Clase: Tipo de material	
Tipo de clase	pública
Descripción	Guarda los tipos de materiales en el sistema
Atributos	
id_material	
nombre_tipo_mat	
descripcion	
Responsabilidades Colaboradores	
Insertar Tipo de material	Material
Mostrar Tipo de material	
Modificar Tipo de material	
Eliminar Tipo de material	
Mostrar listado de Tipo de material	

NOTA: Las otras tarjetas CRC están en el expediente de proyecto.

Anexo.3: Tareas de Ingeniería

Tabla 3.6: Tarea de programación Insertar Materiales

Tarea de Programación	
No. Tarea: 19	No. Historia: 12
Nombre de la tarea: insertar materiales.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 06/02/2014	Fecha fin: 07/02/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite insertar materiales en la base de datos.	

Tabla 3.7: Tarea de programación Modificar Materiales

Tarea de Programación	
No. Tarea: 21	No. Historia: 12
Nombre de la tarea: modificar materiales.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 10/02/2014	Fecha fin: 10/02/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite modificar materiales en la base de datos.	

Tabla 3.8: Tarea de programación Eliminar Materiales

Tarea de Programación	
No. Tarea: 22	No. Historia: 12
Nombre de la tarea: eliminar materiales.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 11/02/2014	Fecha fin: 11/02/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite eliminar materiales de la base de datos.	

Tabla 3.9: Tarea de programación Insertar Tipo de Material

Tarea de Programación	
No. Tarea: 23	No. Historia: 13
Nombre de la tarea: insertar tipo de material	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 11/02/2014	Fecha fin: 12/02/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite insertar tipo de material en la base de datos.	

Tabla 3.10: Tarea de programación Modificar Actividad General

Tarea de Programación	
No. Tarea: 33	No. Historia: 10
Nombre de la tarea: modificar actividad general.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 29/03/2014	Fecha fin: 29/03/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite modificar actividad general en la base de datos.	

Tabla 3.11: Tarea de programación Eliminar Actividad General

Tarea de Programación	
No. Tarea: 34	No. Historia: 10
Nombre de la tarea: eliminar actividad general.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 30/03/2014	Fecha fin: 30/03/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite eliminar actividad general en la base de datos.	

Tabla 3.12: Tarea de programación Mostrar listado de Actividad General

Tarea de Programación	
No. Tarea: 45	No. Historia: 10
Nombre de la tarea: Mostrar listado de Actividad General	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 09/04/2014	Fecha fin: 10/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar listado de Actividad General en la base de datos.	

Tabla 3.13: Tarea de programación Mostrar listado de Reporte

Tarea de Programación	
No. Tarea: 46	No. Historia: 11
Nombre de la tarea: Mostrar listado de Reporte	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 11/04/2014	Fecha fin: 11/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar listado de Reporte en la base de datos.	

Tabla 3.14: Tarea de programación Mostrar listado de Material

Tarea de Programación	
No. Tarea: 47	No. Historia: 12
Nombre de la tarea: Mostrar listado de Material	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 12/04/2014	Fecha fin: 12/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar listado de Material en la base de datos	

Tabla 3.15: Tarea de programación Mostrar listado de Tipo de Material

Tarea de Programación	
No. Tarea: 48	No. Historia: 13
Nombre de la tarea: Mostrar listado de Tipo de Material.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 13/04/2014	Fecha fin: 13/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar listado de Tipo de Material en la base de datos.	

Tabla 3.16: Tarea de programación Mostrar listado de los trabajadores en una brigada

Tarea de Programación	
No. Tarea: 49	No. Historia: 27
Nombre de la tarea: Mostrar listado de los trabajadores en una brigada.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 14/04/2014	Fecha fin: 14/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar listado de los trabajadores en una brigada en la base de datos.	

Tabla 3.17: Tarea de programación Agregar un trabajador a una brigada

Tarea de Programación	
No. Tarea: 50	No. Historia: 8
Nombre de la tarea: Agregar un trabajador a una brigada	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 15/04/2014	Fecha fin: 15/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Agregar un trabajador a una brigada en la base de datos.	

Tabla 3.18: Tarea de programación Agregar Materiales a una Orden de Trabajo

Tarea de Programación	
No. Tarea: 51	No. Historia: 14
Nombre de la tarea: Agregar Materiales a una Orden de Trabajo	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/04/2014	Fecha fin: 16/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Agregar Materiales a una Orden de Trabajo en la base de datos.	

Tabla 3.19: Tarea de programación Asignar actividad(es) a una brigada

Tarea de Programación	
No. Tarea: 52	No. Historia: 15
Nombre de la tarea: Asignar actividad(es) a una brigada	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 17/04/2014	Fecha fin: 19/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Asignar actividad(es) a una brigada en la base de datos.	

Tabla 3.20: Tarea de programación Mostrar el plan de Actividades Generales en un período

Tarea de Programación	
No. Tarea: 53	No. Historia: 16
Nombre de la tarea: Mostrar el plan de Actividades Generales en un período	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/04/2014	Fecha fin: 25/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar el plan de Actividades Generales en un período en la base de datos	

Tabla 3.21: Tarea de programación Calcular el gasto de materiales en un período

Tarea de Programación	
No. Tarea: 54	No. Historia: 17
Nombre de la tarea: Calcular el gasto de materiales en un período	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/04/2014	Fecha fin: 26/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Calcular el gasto de materiales en un período en la base de datos.	

Tabla 3.22: Tarea de programación Mostrar el listado de Actividades Imprevistas en un período

Tarea de Programación	
No. Tarea: 55	No. Historia: 18
Nombre de la tarea: Mostrar el listado de Actividades Imprevistas en un período	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 27/04/2014	Fecha fin: 27/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar el listado de Actividades Imprevistas en un período en la base de datos.	

Tabla 3.23: Tarea de programación Mostrar el listado de Actividades por estado

Tarea de Programación	
No. Tarea: 56	No. Historia: 19
Nombre de la tarea: Mostrar el listado de Actividades por estado	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 28/04/2014	Fecha fin: 28/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar el listado de Actividades por estado en la base de datos.	

Tabla 3.24: Tarea de programación Mostrar el plan de Actividades de una brigada

Tarea de Programación	
No. Tarea: 57	No. Historia: 20
Nombre de la tarea: Mostrar el plan de Actividades de una brigada	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 29/04/2014	Fecha fin: 30/04/2014
Programador responsable: Luis Rianne Guerrero López	
Descripción: Permite Mostrar el plan de Actividades de una brigada.	

NOTA: Las otras tareas de ingeniería están en el expediente de proyecto.

Anexo.4: Pruebas de aceptación

Tabla 3.25: Prueba para comprobar la entrada de datos de los Materiales

Prueba de aceptación
Historia de usuario # 12: Gestionar Materiales
Nombre: Prueba para comprobar la entrada de datos de los Materiales.
Descripción: Validación de entrada de los datos de los Materiales.
Condiciones de ejecución: El usuario debe introducir los datos de los Materiales.
Entrada/Pasos de ejecución: El usuario escribe los datos de los Materiales.
Resultado esperado: ➢ Si se introducen los datos correctamente se insertará los Materiales. Se emite un mensaje de error en caso de que: ➢ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

Tabla 3.26: Prueba para comprobar la entrada de datos mostrados en el plan de Actividades Generales en un período

Prueba de aceptación
Historia de usuario # 16: Mostrar el plan de Actividades Generales en un período
Nombre: Prueba para comprobar la entrada de datos del plan de Actividades Generales en un período.
Descripción: Validación de entrada de los datos del plan de Actividades Generales

en un período.
Condiciones de ejecución: El usuario debe introducir los datos del plan de Actividades Generales en un período.
Entrada/Pasos de ejecución: El usuario escribe los datos del plan de Actividades Generales en un período.
Resultado esperado: ➢ Si se introducen los datos correctamente se insertará el plan de Actividades Generales en un período. Se emite un mensaje de error en caso de que: ➢ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

Tabla 3.27: Prueba para comprobar la entrada de datos calculados del gasto de los materiales en un período

Prueba de aceptación
Historia de usuario # 17: Calcular el gasto de materiales en un período
Nombre: Prueba para comprobar la entrada de datos de los materiales en un período.
Descripción: Validación de entrada de los datos de los materiales en un período.
Condiciones de ejecución: El usuario debe introducir los datos de los materiales en un período.
Entrada/Pasos de ejecución: El usuario escribe los datos de los materiales en un

período.
Resultado esperado: ➢ Si se introducen los datos correctamente se insertará los materiales en un período. Se emite un mensaje de error en caso de que: ➢ Se dejen campos obligatorios vacíos.
Evaluación de la prueba: Aceptada.

NOTA: Las otras pruebas de aceptación están en el expediente de proyecto.

Anexo.3