

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO
SISTEMA WEB DE GESTIÓN DE PROYECTOS
CASO: CONSTRUCTORA Y MANTENIMIENTO
ELÉCTRICO ATWOOD SRL.

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

Postulante: Univ. Johnny Chavez Quispe

Tutor Metodológico: Lic. Freddy Miguel Toledo Paz

Asesor: M.Sc. Aldo Ramiro Valdez Alvarado

LA PAZ - BOLIVIA
2014



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON EstrictAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

DEDICATORIA: A Dios por guiarme y cuidarme en todo momento, brindándome su amor, bendición y fuerza en mi vida ayudándome a sobrellevar todas las adversidades

A mis padres Leandro y Andrea por todo el amor, paciencia y el apoyo incondicional en todo momento, quienes me encaminaron a lograr esta meta profesional.

A mi hermano Willy por compartir, tantas alegrías y tristezas.

AGRADECIMIENTOS

Quiero expresar mis sinceros agradecimientos:

A Dios por haberme dado la oportunidad de vivir todo lo hermoso del mundo.

A mi familia por estar siempre a mi lado brindándome su apoyo y confianza para seguir adelante, que sirvió para que hoy pueda concluir mis estudios.

A mi docente en calidad de Tutor metodológico: Lic. Freddy Miguel Toledo Paz, por su orientación, profesionalismo y experiencia para la elaboración de este proyecto.

A mi docente en calidad de revisor: M.Sc. Aldo Ramiro Valdez Alvarado, por el seguimiento a este trabajo, quien con su constante orientación profesional, sus consejos, observaciones, correcciones, experiencia y paciencia fueron un aporte invaluable en la elaboración de este proyecto de grado.

A la Empresa “ADWOOT SRL” por haberme permitido desarrollar mi proyecto en sus instalaciones, con la colaboración del personal para la implementación del sistema.

A la Universidad Mayor de San Andrés, a la Facultad de Ciencias Puras y a la Carrera de Informática por abrirme las puertas brindarme una formación académica.

A los docentes de la carrera de informática que me brindaron su enseñanza, orientación y tiempo, que sirvieron para que hoy pueda concluir mis estudios académicos.

A todos mis compañeros y amigos con quienes compartí momentos inolvidables, quienes me brindaron su amistad sincera y colaboración en distintas situaciones.

Muchas Gracias a todos...!

RESUMEN

Hoy en día las redes informáticas es uno de los medios más eficientes para mantener la información actualizada ya que pueden compartir información entre varias computadoras y usuarios simultáneamente. Debido a esto en la actualidad las empresas públicas y privadas pretenden mejorar sus niveles de eficiencia basándose en la gestión de proyectos y con la ayuda de los medios tecnológicos, lograr la disminución de costos y la excelencia operacional.

El presente proyecto es un sistema web con lo cual se busca mejorar el funcionamiento de la empresa “Constructora y Mantenimiento Eléctrico ATWOOD SRL”, en el proceso de gestión de proyectos, además de captar nuevos clientes para poder contar con un mercado de construcciones más amplio. El sistema brinda una serie de funcionalidades que fueron plasmadas en historias de usuarios, estas funcionalidades buscan satisfacer los requerimientos de la empresa y los funcionarios internos.

El proyecto fue desarrollado utilizando la metodología ágil XP (programación extrema) en sus distintas fases como son: planificación, diseño, desarrollo y pruebas; en la planificación se desarrollaron varias historias de usuario, las cuales fueron desarrolladas e implementadas en las diferentes iteraciones del proyecto. El diseño de los procesos e interfaz fue realizado bajo la metodología de modelado Webml que cuenta con diversos esquemas para la representación grafica de estos procesos.

Para la implementación se utilizó como gestor de base de datos MySQL, además, como lenguaje de programación se utilizó las tecnologías de: Php, Ajax, jquery los cuales incluyen herramientas de desarrollo Cliente Servidor. La evaluación de la calidad del software se detalla cualitativamente la calidad mediante la aplicación de métricas, tomando como factor de calidad el estándar ISO 9126.

Al finalizar el proyecto se implemento un sistema web que permita tener un control de los proyectos que realiza la empresa, cumpliendo con los objetivos propuestos.

ÍNDICE

CAPÍTULO 1

MARCO INTRODUCTORIO

1.1	INTRODUCCIÓN	1
1.2	ANTECEDENTES.....	2
1.3	OBJETO DE ESTUDIO.....	3
1.4	PLANTEAMIENTO DEL PROBLEMA.....	3
1.5	JUSTIFICACIÓN	4
1.5.1	Justificación Social	4
1.5.2	Justificación Técnica.....	5
1.5.3	Justificación Económica	6
1.6	OBJETIVOS	7
1.6.1	Objetivo General.....	7
1.6.2	Objetivos Específicos.....	7
1.7	LIMITES Y ALCANCES	7
1.7.1	Limites	8
1.7.2	Alcances	8
1.8	APORTES.....	9
1.9	METODOLOGÍA	9
1.10	IMPORTANCIA DE ESTUDIO	11

CAPÍTULO 2

MARCO TEÓRICO

2.1	INGENIERÍA DE SOFTWARE.....	12
2.1.1	Definición: Ingeniería	12
2.1.2	Definición: Software.....	12
2.2	METODOLOGÍAS TRADICIONALES	13
2.3	METODOLOGÍAS ÁGILES	14
2.4	MANIFIESTO ÁGIL	15
2.5	METODOLOGÍAS ÁGILES vs METODOLOGÍAS TRADICIONALES.....	17
2.6	EL PORQUE UTILIZAR METODOLOGÍAS ÁGILES.....	18
2.6.1	Metodologías Ágiles de desarrollo de software	19
2.7	PROGRAMACIÓN EXTREMA (XP)	20

2.7.1	Metodología Programación Extrema (XP).....	21
2.7.2	Ciclo de Vida de XP	21
2.7.3	Fases de la metodología XP	22
2.7.3.1	Fase 1: Planificación	22
2.7.3.2	Fase 2: Diseño.....	26
2.7.3.3	Fase 3: Desarrollo	28
2.7.3.4	Fase 4: Pruebas	29
2.8	INGENIERÍA WEB	31
2.8.1	Modelos de Procesos WEB	32
2.9	WEBML	32
2.9.1	Modelo de Estructura.....	33
2.9.2	Modelo de Hipertexto	34
2.9.3	Modelo de Composición	36
2.9.4	Modelo de Navegación	38
2.9.5	Modelo de Presentación.....	39

CAPÍTULO 3

MARCO APLICATIVO

3.1	INTRODUCCIÓN	42
3.2	PLANIFICACIÓN	43
3.2.1	Obtención de requerimientos.....	43
3.2.2	Historias de usuario	43
3.2.3	Plan de entrega (Release planning).....	48
3.2.3.1	Primera iteración	49
3.2.3.2	Segunda iteración.....	53
3.2.3.3	Tercera iteración	55
3.3	DISEÑO.....	60
3.3.1	Modelo de Entidad Relación	60
3.3.2	Modelo de Componentes	63
3.3.3	Tarjetas CRC (clase - responsabilidad - colaboración)	64
3.3.4	Modelo de Navegación	66
3.4	DESARROLLO	70
3.5	PRUEBAS	74

3.5.1	Pruebas de aceptación	74
3.5.2	Pruebas de Stress	78

CAPÍTULO 4

CALIDAD Y SEGURIDAD

4.1	QUE ES CALIDAD DEL SOFTWARE.....	82
4.1.1	Funcionalidad	82
4.1.2	Confiabilidad	86
4.1.3	Mantenibilidad.....	88
4.1.4	Portabilidad	89
4.1.5	Usabilidad	90
4.1.6	Resultado Final.....	91
4.2	SEGURIDAD	92
4.2.1	Seguridad del software.....	92
4.2.2	Seguridad de la base de datos.....	93
4.2.3	Seguridad del hardware.....	94
4.3	ANÁLISIS DE COSTO BENEFICIO	94
4.3.1	Análisis Utilizando COCOMO II.....	94
4.3.2	Análisis del Costo.....	97
4.3.3	Costo de Implementación del Sistema.....	100
4.3.4	Costo de Elaboración del Proyecto.....	101
4.3.5	Costo Total	101
4.3.6	CALCULO DE BENEFICIOS CON EL VAN y el TIR.....	102
4.3.6.1	Calculo del VAN.....	102
4.3.6.2	Tasa Interna de Retorno (TIR).....	104
4.3.7	COSTO / BENEFICIO	105

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1	CONCLUSIONES	106
5.2	RECOMENDACIONES	107
	BIBLIOGRAFÍA.....	109
	REFERENCIAS WEB.....	109
	ANEXOS	

ÍNDICE DE FIGURAS

Figura 2. 1	Evolución de los largos ciclos de desarrollo en cascada (a), a ciclos más cortos (b), y a la mezcla que hace XP (c).....	20
Figura 2. 2	Metodología XP	22
Figura 2. 3	Formato de la Tarjeta de la Historia de Usuario	23
Figura 2. 4	Modelos Webml.....	33
Figura 2. 5	Notaciones básicas	34
Figura 2. 6	Unidades de contenido básico.....	36
Figura 2. 7	Notación gráfica de unidades y su posible ejecución en HTML	37
Figura 2. 8	Diagrama de navegación con enlaces contextuales y su posible ejecución en HTML	38
Figura 2. 9	Ejemplo de WebML	39
Figura 3. 1	Modelo Entidad Relación (E - R)	60
Figura 3. 2	Modelo de Estructura	61
Figura 3. 3	Base de Datos.....	62
Figura 3. 4	Modelo de componentes.....	63
Figura 3. 5	Diagrama navegacional: Usuario Administrador.....	67
Figura 3. 6	Modelado Web: Pagina Principal	68
Figura 3. 7	Diagrama Navegacional: Usuario Encargado de proyecto.....	69
Figura 3. 8	Modelado Web: Proyectos.....	69
Figura 3. 9	Inicio del Sistema.....	71
Figura 3. 10	Apertura de Proyectos	71
Figura 3. 11	Registro Seguimiento de Proyectos.....	72
Figura 3. 12	Registro Reprogramación de Proyectos	72
Figura 3. 13	Registro de Personal.....	73
Figura 3. 14	Registro de agencias	73
Figura 3. 15	Configuración del JMeter	78

ÍNDICE DE TABLAS

Tabla 2. 1	Tabla comparativa	18
Tabla 2. 2	Divergencias entre las metodologías ágiles	20
Tabla 3. 1	Fases y procesos de la Metodología XP y WebML	42
Tabla 3. 2	Lista de Requerimientos	43
Tabla 3. 3	Historia de usuario: Registro de apertura de proyectos	44
Tabla 3. 4	Historia de usuario: Registrar seguimiento de proyectos	45
Tabla 3. 5	Historia de usuario: Registrar reprogramación de proyectos.....	46
Tabla 3. 6	Historia de usuario: Registro de personal	46
Tabla 3. 7	Historia de usuario: Registro de agencias.....	47
Tabla 3. 8	Historia de usuario: Reportes	48
Tabla 3. 9	Resumen de las historias de usuario	48
Tabla 3. 10	Tarea: diseño del formulario de registro de maquinarias	50
Tabla 3. 11	Tarea: modificación de datos de los proyectos	50
Tabla 3. 12	Tarea: Eliminación del proyecto	50
Tabla 3. 13	Tarea: Listado de proyectos	51
Tabla 3. 14	Tarea: Registro de seguimiento de proyectos	51
Tabla 3. 15	Tarea: modificación de datos del registro de seguimiento de proyectos....	52
Tabla 3. 16	Tarea: Eliminación del registro de seguimiento de proyectos	52
Tabla 3. 17	Tarea: Listado de proyectos	53
Tabla 3. 18	Tarea: Registro de reprogramación de proyectos.....	53
Tabla 3. 19	Tarea: diseño del formulario de registro de personal	54
Tabla 3. 20	Tarea: modificación de datos del personal.....	54
Tabla 3. 21	Tarea: Eliminación del registro de personal	55
Tabla 3. 22	Tarea: Listado del registro de personal.....	55
Tabla 3. 23	Tarea: diseño del formulario de registro de agencia	56
Tabla 3. 24	Tarea: actualizar agencia.....	56
Tabla 3. 25	Tarea: Eliminación de agencia	57
Tabla 3. 26	Tarea: Listado de agencias	57
Tabla 3. 27	Tarea: reporte de proyectos.....	58
Tabla 3. 28	Tarea: reporte de agencias	58
Tabla 3. 29	Tarea: reporte de seguimiento de proyectos	58
Tabla 3. 30	Tarea: reporte de seguimiento de proyectos	59
Tabla 3. 31	Cronograma de actividades: iteraciones del proyecto	59
Tabla 3. 32	Prueba de aceptación: Registro y modificación de proyectos.....	75
Tabla 3. 33	Prueba de aceptación: Registro y modificación seguimiento proyectos	75
Tabla 3. 34	Prueba de aceptación: Registro reprogramación de proyectos	76
Tabla 3. 35	Prueba de aceptación: Registro de Personal	76
Tabla 3. 36	Prueba de aceptación: Registro de Agencia.....	77
Tabla 3. 37	Prueba de aceptación: Reportes.....	77
Tabla 3. 38	Informe Agregado para la prueba de 350 usuarios.....	80
Tabla 3. 39	Informe Agregado para la prueba de 325 usuarios.....	81

Tabla 4. 1	Calculo de los Parámetros de Medición	84
Tabla 4. 2	Escala de evaluación para los valores del Ajuste de Complejidad	84
Tabla 4. 3	Valor de Ajuste de Complejidad	85
Tabla 4. 4	Escala de Punto Función.....	86
Tabla 4. 5	Escala Ponderación para responder a la encuesta	90
Tabla 4. 6	Resultados de la encuesta	90
Tabla 4. 7	Conversión de puntos de Función	98
Tabla 4. 8	Relación de valores en el modelo COCOMO	99
Tabla 4. 9	Costo de elaboración del Proyecto	101
Tabla 4. 10	Costo Total del Proyecto.....	101
Tabla 4. 11	Calculo del VAN.....	103
Tabla 4. 12	Interpretación del VAN	103
Tabla 4. 13	Tasa Interna de Retorno	104

CAPÍTULO 1

MARCO INTRODUCTORIO

1.1 INTRODUCCIÓN

Hoy en día las organizaciones son cada vez más consistentes de la importancia de adecuarse a los cambios tecnológicos de manejo de información con el objeto de realizar una buena toma de decisiones, y alcanzar sus objetivos.

Por este motivo se van automatizando las distintas necesidades que se presentan en las distintas entidades públicas o privadas con la capacidad de proyectarse al futuro.

La evolución del Internet como red de comunicación global y el surgimiento y desarrollo Web como servicio imprescindible para compartir información, creo un excelente espacio para la interacción del hombre con la información hipertextual, a la vez que sentó las bases para el desarrollo de una herramienta integradora de los servicios existentes en Internet.

Es así que las empresas de construcción hoy en día cobran importancia de involucrarse en la medida que la tecnología lo permita, este es el caso de la empresa “Constructora y Mantenimiento Eléctrico ATWOOD SRL” es vital hacer un seguimiento y monitoreo de estos proyectos para que el flujo de información de los proyectos sea de manera oportuna, inmediata y eficaz.

Es por ello que el presente Proyecto de Grado pretende generar un núcleo de información de todos los proyectos que se generan en la empresa, para que todas las personas involucradas en el trabajo puedan acceder de manera inmediata a la información, satisfaciendo las necesidades en lo referente al seguimiento y control a los proyectos de construcción que se van generando.

Por el momento todo este trabajo se lo desarrolla de manera manual, si se pudiera informatizar estos procesos se llegaría a informar y atender más apropiadamente las necesidades de las personas involucradas en los proyectos.

1.2 ANTECEDENTES

En la carrera de Informática de la Universidad Mayor de San Andrés (U.M.S.A.) se verifico la existencia de proyectos de grado relacionados con el sistema de control y administración, dichos proyectos están orientados a la automatización de los procesos manuales.

Se verifico los siguientes proyectos de grado:

- Sistema de información y Control de Almacenes, realizado por (Univ. Alberto Agustín Lozano) en el año 2006, Caso: Superintendencia de Electricidad, cuyo objetivo es desarrollar e implementar un sistema de tal manera que permita mejorar el movimiento de los materiales, suministros desde su ingreso hasta su utilización.
- Sistema de Control y Administración “SARIRI”, realizado por (Univ. Juan Carlos Paredes Mamani) el año 2005, Caso: Bolivian Bus empresa de transporte de pasajeros internacional, cuyo objetivo es diseñar, desarrollar e implementar un sistema de información computarizado que interactúe en un entorno de red y centralice la información en general de la empresa. Aplicando metodologías de análisis, diseño de sistemas de información, automatizando el seguimiento de los buses para llevar un mejor control de los horarios de los mismos para ofrecer un mejor servicio a los clientes de la empresa.
- Sistema de Información Vía Web de Seguimiento y Control a los Proyectos de Inversión, realizado por (Univ. Erik Márquez Mayta) en el año 2013, Caso: Gobierno autónomo Municipal de Palos Blancos, cuyo objetivo es desarrollar e implementar un Sistema que permita hacer un seguimiento y control de todos los

proyectos que se van desarrollando en el municipio de Palos Blancos y tener un óptimo control logístico del municipio.

1.3 OBJETO DE ESTUDIO

Empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL, es una sociedad constituida el 17 de Diciembre 2001. Sus socios son los señores Marcelo Huanca, Constructor Civil. En estos 13 años de trayectoria en el área de la vivienda ha logrado gran eficiencia constructiva entregando la mejor relación precio/calidad a sus clientes.

En Mayo de 2010 la empresa certificó en el sistema de gestión de la calidad ISO 9001-2008. Esta certificación le fue otorgada por la asociación española de normalización y certificación AENOR.

La dirección hace especial énfasis en el compromiso permanente de todos sus trabajadores con la calidad y mejora en el desarrollo de sus actividades cotidianas, incentivando un proceso continuo de retroalimentación con sus clientes, su personal y sus proveedores.

La Dirección en vías de implantar un Sistema Web de Gestión de Proyectos con el propósito de administrar la información de los diferentes proyectos en ejecución y por ejecutarse para su mejor toma de decisiones. El sistema va a permitir que la información este centralizado para su mejor administración y transparencia en los procesos.

1.4 PLANTEAMIENTO DEL PROBLEMA

Luego de realizar una investigación sobre los procesos que realiza la empresa en relación a la gran cantidad de información que maneja, se pudo identificar las siguientes dificultades.

- Retardo en la elaboración personalizada de formularios para cada uno de los proyectos, esto causa falta de repositorio que especifique las características de

los tipos de proyectos y su efecto es la demora en la inicialización de los proyectos.

- Retardo de 5 a 10 minutos de la secretaria o jefe de área en la búsqueda del kardex del estado de un proyecto, porque los registros están elaborados manualmente y su efecto es en el enfado y demora en la atención de esta información al responsable de presupuesto.
- Perdida de 5 a 10 minutos de la secretaria en la búsqueda de información acerca de los materiales que serán parte de un proyecto, porque los materiales con que cuentan están registrados y archivados en Kardex y su efecto es el retardo de registrar y hacer actualizaciones de los materiales existentes.
- Falta de control en la asignación de recursos a los distintos proyectos, porque tienen un deficiente forma de seguimiento al cronograma de los empleados por tal razón se tarda en la entrega de los informes de avance de los proyectos.
- Carencia de presentación e información de los proyectos que realiza la empresa, por falta de un Sistema de Web de Gestión de Proyectos, por tal razón los administradores de la empresa desconocen acerca de la elaboración y ejecución de proyectos que realiza la empresa.
- Inadecuada forma de seguridad en el manejo de la información, porque la empresa tiene carencias de políticas y controles de seguridad en informes y reportes de la empresa, por tal razón la mala toma de decisiones.

Por los grandes beneficios que nos brindan las herramientas informáticas la empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL no se puede dar el lujo de prescindir de ellas, por ello se plantea la siguiente pregunta.

¿Cómo se puede mejorar el seguimiento sobre el manejo y ejecución de proyectos en la Constructora y Mantenimiento Eléctrico ATWOOD SRL. ?.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Social

Este proyecto se justifica socialmente porque brindara información segura, oportuna y actualizada de los proyectos que se realizan en la empresa, mejorando la calidad de todos los proyectos. Ofreciendo proyectos de construcción de alta calidad y mejor atención al cliente.

El Sistema Web de Gestión de Proyectos es una herramienta fundamental de apoyo a la Gerencia General, cuya función principal está encargada de la planificación, coordinación, dirección y ejecución de los proyectos.

El implementar el sistema web permite mejorar las tareas que se realizan en Gerencia, Jefatura de Personal y clientes porque proveerá información rápida y actualizada al personal, ayudando a los criterios de decisión, ofreciendo un mejor entorno de trabajo y mejorando la calidad de servicio.

Brindando beneficios, así como el tiempo para realizar consultas y reportes requeridos, logrando una forma de trabajo ordenado, capacitando al personal para que este se relacione con la tecnología actual satisfaciendo así las necesidades de información.

El buen funcionamiento de los procesos internos en la Empresa “Constructora y Mantenimiento Eléctrico ADWOOD SRL.”, repercute en la sociedad, quien asimila el buen servicio ofrecido, esto hace que la sociedad tenga una muy buena imagen de la empresa.

1.5.2 Justificación Técnica

Se justifica técnicamente, porque la empresa “Constructora y Mantenimiento Eléctrico ATWOOD SRL.” cuenta con las herramientas tecnológicas suficientes como recursos de Software y Hardware para implementar el Sistema Web de Gestión de Proyectos.

La modernización en el manejo de la información es indispensable para poder mejorar las actividades que se realizan. Los componentes técnicos como tecnológicos que se

están empleando en el presente desarrollo de software como por ejemplo computadoras, Internet y lenguajes de programación impulsaran al mejor tratamiento de información, tan bien a la innovación tecnológica que nos permitirá diferenciarnos de las otras empresas.

Se empleará la tecnología Cliente/Servidor para compartir o distribuir la información de la Base de Datos que esté disponible en cualquier computador conectado a la red de Intranet, utilizando medidas de seguridad para que esta información pueda ser accedida solo por usuarios autorizados, además de usar una interfaz adecuada y fácil de operar por cualquier usuario.

1.5.3 Justificación Económica

El Sistema Web de Gestión de Proyectos necesita un servidor dedicado con un grado empresarial con la siguiente descripción:

- **2.4GHZ** Dual Xeon E5620 Quad
- **12 GB** DDR3
- **1 TB** Disco Sata
- **5 IP** Dedicadas
- 100 MBit/s
- Supermicro/Dell Hardware

El costo de este equipo es de 1000 \$ que es un equivalente en Bs. 7070.

También una conexión de internet ADSL con una Banda Ancha Empresarial el costo es de Bs. 780 mensual. La inversión es de un total de Bs. 7850 esto beneficiara a largo plazo porque reducirá el costo de hospedaje del sistema web.

Con la implementación del sistema se obtendrá información confiable, rápida y actualizada para el control de los proyectos, anulando la perdida de dinero, al realizar el

registro de los proyectos, lo cual reducirá el tiempo en la búsqueda de la información requerida y aumentara la seguridad de información de la empresa.

Se justifica económicamente el proyecto porque proporcionara beneficios a corto y largo plazo que se observara a través del proceso de control, teniendo como resultado una reducción de tiempo y esfuerzo, permitiendo generar y consultar información en tiempo real obteniendo información confiable e inmediata, lo que implica un ahorro económico, teniendo en cuenta la disminución en costos.

1.6 OBJETIVOS

1.6.1 Objetivo General

Implementar un sistema Web para la gestión y el seguimiento de los proyectos para la empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL.

1.6.2 Objetivos Específicos

- Generar formularios para el registro de los proyectos de construcción en base al manual de formulación, evaluación y monitoreo de proyectos de construcción.
- Generar informes diarios estadísticos para el monitoreo del avance y evaluación de cada proyecto en base al método de evaluación y control.
- Implementar un sistema web que presente y publique la información acerca de los distintos materiales que existe en la empresa y poder actualizarlos.
- Establecer un cronograma que permita controlar la asignación de recursos.
- Desarrollar una página web que presente y publique información acerca de los distintos proyectos que tiene la empresa.
- Implementar métodos criptográficos para la seguridad y resguardo de la información de los proyectos.
- Generación de reportes en línea permitiendo la descarga de los proyectos.

1.7 LIMITES Y ALCANCES

1.7.1 Limites

El sistema geográficamente estará en el departamento de La Paz para la empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL, ubicado en El Alto Zona Villa Exaltación, Calle 8, N° 345.

El sistema estará limitado al manejo de información de las aéreas administrativas, con la finalidad de aumentar el prestigio y la confianza de la empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL, en sus procesos de administración y control de proyectos.

- La actualización de datos solo será realizado por el personal autorizado.
- El sistema obviara el registro de comprobantes de pagos.
- El responsable será el encargado de supervisar que los proyectos realizados estén documentados, estos datos estarán reflejados en el sistema.
- El sistema será inaccesible para las oficinas, o unidades que no tuvieran acceso a internet.

1.7.2 Alcances

El presente Proyecto de Grado, mejorara los servicios de información ofrecidos a los clientes, aumentando así la eficiencia y eficacia en la gestión de proyectos y garantizando la calidad en el desarrollo de proyectos cumpliendo la finalidad de cada una de estas actividades realizadas por parte de la empresa. Involucrando los siguientes puntos que se detallan a continuación:

- Modulo que realice el registro de proyectos.
- Modulo de registro de personal.
- Modulo de registro de Agencias.
- Modulo de generación de reportes.
- Modulo de reprogramación.

- Modulo de Seguimiento de proyectos
- Modulo de Lista de proyectos

1.8 APORTES

Para lograr los diferentes alcances que se proponen se utilizaran técnicas y metodologías, mencionadas a continuación:

- La apertura de proyectos se realizara mediante la obtención de una cuenta de acceso al Sistema.
- El seguimiento de los diferentes programas y proyectos se ejecuta a través de la información obtenida de los proyectos en ejecución.
- El control de ejecución de proyectos se determina mediante el grado de cumplimiento de los objetivos en un determinado presupuesto y tiempo.
- La utilización del uso de una contraseña para el ingreso al sistema son utilizados por la personas autorizadas de los proyectos.
- Se aplicara la metodología ágil XP (Extreme Programming) y una metodología de modelado de Aplicaciones Web denominado WebML (Web Modeling Language).

1.9 METODOLOGÍA

Utilizamos el Método científico, que es un método de estudio sistemático de la naturaleza que incluye las técnicas de observación, reglas para el razonamiento y la predicción, ideas sobre la experimentación planificada y los modos de comunicar los resultados experimentales y teóricos.

Aunque el método científico es uno, existen diversas formas de identificar su práctica o aplicación en la investigación.

Los aspectos de la investigación que tomamos en cuenta son:

La investigación exploratoria: que es considerada como el primer acercamiento científico a un problema. Se utiliza cuando éste aún no ha sido abordado o no ha sido suficientemente estudiado y las condiciones existentes no son aún determinantes; para luego pasar a la Investigación Descriptiva: que se efectúa cuando se desea describir, en todos sus componentes principales, una realidad.

El desarrollo del presente trabajo utilizara la metodología de Programación Extrema (XP) en sus cuatro fases (planificación, diseño, desarrollo y pruebas) que es una metodología de desarrollo de software que se basa en la simplicidad, la comunicación y la retroalimentación.

Para el desarrollo del proyecto se llevaran a cabo los siguientes pasos metodológicos y técnicos a medida que se avance con el proyecto:

- Se realizara la investigación acerca de los sistemas de seguimiento de proyectos existentes y sistemas de logística.
- Una vez analizada y depurada la información realizaremos un estudio profundo de los sistemas de seguimiento de control de proyectos aplicando las herramientas para el manejo y control de la información.
- Finalmente de acuerdo a las necesidades de la empresa se realizara la creación del Sistema Web de Gestión de Proyectos.

Para el modelado utilizaremos el Diseño Conceptual de Aplicaciones Web: WebML. Provee gráficos, formalismos, especificaciones, y diseño de procesos apoyados por herramientas gráficas basadas en el diseño de la interfaz.

Define cinco tipos de modelos: estructura, derivación, composición, navegación y presentación. En el modelo de estructura se definen las entidades o contenedores de datos y sus relaciones. En el modelo de derivación se definen diferentes vistas y agrupaciones de los mismos datos. En el modelo de composición se especifican las páginas que componen el hipertexto, así como el contenido de éstas.

El modelo de navegación especifica los links entre páginas y entre unidades de una misma página.

Finalmente en el modelo de presentación se describe la apariencia gráfica de las páginas. Para la implementación de este proyecto se utilizará las siguientes herramientas tecnológicas: Plataforma: Windows, Lenguaje de Programación: PHP, Sistema Gestor de Base de Datos MySQL o PostgreSQL además de otras herramientas necesarias para el diseño y implementación del sistema web.

1.10 IMPORTANCIA DE ESTUDIO

El aporte del trabajo a pesar que está dirigido a una Empresa Privada es de suma importancia para el control y seguimiento de los proyectos, para que su acceso a ellos sea rápido.

Proporcionar una herramienta que coadyuve y brinde información veraz, oportuna y confiable de todos los proyectos que se realizan en la empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL.

Indicar de forma clara, precisa y eficiente la situación de cada empleado y su eficiencia en la empresa.

CAPÍTULO 2

MARCO TEÓRICO

2.1 INGENIERÍA DE SOFTWARE

La ingeniería de software es una disciplina o área de la informática o ciencias de la computación, que ofrece método y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy es cada vez más frecuente la consideración de la Ingeniería de software como una nueva área de la ingeniería, y el Ingeniero de Software comienza a ser una profesión implantada en el mundo laboral internacional, con derechos, deberes y responsabilidades que cumplir, junto a una y reconocida consideración social en el mundo empresarial y por suerte para esas personas con brillante futuro. [Sommerville, 2004]

2.1.1 Definición: Ingeniería

La ingeniería es el estudio de la aplicación de las distintas ramas de la tecnología. El profesional en este ámbito recibe el nombre de ingeniero. La ingeniería también supone la aplicación de la inventiva y del ingenio para desarrollar una cierta actividad. Esto, por supuesto, no implica que no se utilice el método científico para llevar a cabo los planes. [Bauer, 1972]

2.1.2 Definición: Software

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un computacional. [Std. 729, IEEE]. El software no son solo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.

Un sistema de software consiste de diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica cómo utilizar el sistema y sitios web que permitan a los usuarios descargar la información de productos recientes. [Sommerville, 2004]

El software de computadora es el producto que los ingenieros de software construyen y después mantienen en el largo plazo.

El software se forma con:

- Las instrucciones (programas de computadora) que al ejecutar se proporcionan las características, funciones y el grado de desempeño deseados;
- Las estructuras de datos que permiten que los programas manipulen información de manera adecuada;
- Los documentos que describen la operación y uso de los programas. [Pressman, 2005]

2.2 METODOLOGÍAS TRADICIONALES

Las metodologías tradicionales en el desarrollo de software son basadas en las metodologías existentes sobre el desarrollo de proyectos en otras áreas.

Básicamente es dividir el proceso de desarrollo de etapas, de una manera secuencial, son más un resultado de una urgencia por dotar al desarrollo de software de orden para poder completar los objetivos deseados.

Tenemos entre las metodologías tradicionales, el RUP que es la que integra el modelado UML, estos métodos ofrecen una documentación muy completa, exhaustiva y un plan de proyecto cuidadosamente definido, el desarrollo se basa en un modelo de procesos estrictamente definido. [Kendall, 2006]

2.3 METODOLOGÍAS ÁGILES

A principios de la década del 90, surgió un enfoque que fue bastante revolucionario para su momento ya que iba en contra de toda creencia de que procesos altamente definidos se iba a lograr obtener software en tiempo, costo y con la requerida calidad. El enfoque fue planteado por primera vez por Martin y se dio a conocer en la comunidad de Ingeniería de Software con el nombre de RAD o Rapid Application Development. RAD consistía en un entorno de desarrollo altamente productivo, en el que participan grupos pequeños de programadores utilizando herramientas que generan código en forma automática tomando como entradas sintaxis de alto nivel. En general se considera que este fue uno de los primeros hitos en pos de la agilidad en los procesos de desarrollo.

La historia de las Metodologías Ágiles y su apreciación como tales en la comunidad de la ingeniería de software tiene en sus inicios en la creación de una de las metodologías utilizada como arquetipo: XP-eXtreme Programming, que nace de la mente de Kent Beck, tomando ideas recopiladas junto a Ward Cunningham.

Durante 1006, Beck es llamado por Chryler como asesor del proyecto Chryler Comprehensive Compensation (C3) payrollsystem. Dada la poca calidad del sistema que se estaba desarrollando, Beck decide tirar todo el código y empezar de cero utilizando las prácticas que él había ido definiendo a lo largo del tiempo. El sistema que administra la liquidación de aproximadamente 10.000 empleados, y consiste de 2.000 clases y 30.000 métodos, es puesto en operación en mayo de 1997 casi respetando el calendario propuesto. Como consecuencia del éxito de dicho proyecto, Kent Beck dio origen a XP iniciando el movimiento de metodologías ágiles al que se anexarían otras metodologías surgidas mucho antes que el propio Beck fuera convocado por Chryler. (Beck K., 1999)

Es así como que este tipo de metodologías fueron inicialmente llamadas “metodologías livianas”, sin embargo, aun no contaban con una aprobación pues se le consideraba por muchos desarrolladores como meramente intuitiva. Luego, con el pasar de los años, en

febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace formalmente el término “ágil” aplicado al desarrollo de software. En esta misma reunión un grupo de 17 expertos de la industria del software, incluyendo uno de los creadores o impulsores de metodologías de software con el objetivo de esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. (Cockburn, 2000)

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”.

2.4 MANIFIESTO ÁGIL

El manifiesto Ágil comienza enumerando los principales valores del desarrollo ágil. Según el Manifiesto se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto de software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir el entorno y esperar que el equipo se adapte automáticamente. (Abrahamsson, 2002)
- Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante “. Estos documentos deben

ser cortos y centrarse en lo fundamental.

- La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto a metas a seguir y organización del mismo. Los principios son:

- I.** La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II.** Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III.** Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV.** La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V.** Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el proyecto.
- VI.** El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- VII.** El software que funciona es la medida principal de progreso.

- VIII.** Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX.** La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X.** La simplicidad es esencial.
- XI.** Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII.** En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

2.5 METODOLOGÍAS ÁGILES vs METODOLOGÍAS TRADICIONALES

La tabla recoge esquemáticamente las principales diferencias de las metodologías ágiles con respecto a las tradicionales (“no ágiles”). Estas diferencias que afectan no solo al proceso en sí, sino también al contexto del equipo así como su organización, (véase tabla 2.1).

Metodologías ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Específicamente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.

Pocos artefactos.	Más artefactos.
Pocos roles.	Más roles.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Tabla 2. 1 Tabla comparativa
Fuente: Universidad de Trujillo, 2007

2.6 EL PORQUE UTILIZAR METODOLOGÍAS ÁGILES

Tomando las ideas de la tabla anterior podemos decir que las metodologías tradicionales presentan los siguientes problemas a la hora de abordar proyectos:

- Existen unas costosas fases previas de especificación de requisitos, análisis y diseño. La corrección durante el desarrollo de errores introducidos en estas fases será costosa, es decir, se pierde flexibilidad ante los cambios.
- El proceso de desarrollo será encorsetado por documentos firmados.
- El desarrollo es más lento. Es fácil para los desarrolladores entender un sistema complejo en su globalidad.

Las metodologías ágiles de desarrollo están especial mente indicadas en proyectos con requisitos poco definidos o cambiantes. Estas metodologías se aplican bien en equipos pequeños que resuelven problemas concretos, lo que no está reñido con su aplicación en el desarrollo de grandes sistemas, ya que una correcta modulación de los mismos es fundamental para su exitosa implantación. Dividir el trabajo en módulos abordables minimiza los fallos y el costo. (Beck K, 1999)

- Las metodologías ágiles presentan diversas ventajas, entre las q podemos destacar:
- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.

- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo
- Importancia de la simplicidad, eliminando el trabajo innecesario.
- Atención continúa a la excelencia técnica y al buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

2.6.1 Metodologías Ágiles de desarrollo de software

Se apreciara las convergencias y divergencias en la definición de las metodologías ágiles más importantes (véase tabla 2.2).

Metodología	Acrónimo	Creación	Tipo de modelo	Característica
Adaptive Software Development	ASD	Highmuth 2000	Practicas + ciclo de vida	Inspirado en sistemas adaptativos complejos
Agile Modeling	AM	Ambler 2002	Metodología basada en la practica	Suministra modelo ágil a otros métodos.
Cristal Methods	CM	Cockbum 1998	Familia de metodologías	Metodología ágil con énfasis en modelo de ciclos.
Agile RUP	dX	Booch Martin, Newkirk 1998	Framework/Disciplina	XP dado vuelta con artefactos RUP
Evolutionary Project	EVO	Gilb 1976	Framework	Primer método

Management			adaptativo	ágil existente.
eXtremeProgaming	XP	Beck 1999	Disciplina en prácticas de ingeniería	Metodo ágil radical

Tabla 2. 2 Divergencias entre las metodologías ágiles
Fuente: universidad de Trujillo, 2007

2.7 PROGRAMACIÓN EXTREMA (XP)

La programación extrema se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ligeras, que son aquéllas en la que se da prioridad a las tareas que dan resultados directos y que reducen la burocracia que hay alrededor tanto como sea posible (pero no más). La programación extrema, dentro de las metodologías ágiles, se puede clasificar dentro de las evolutivas como se muestra en la Figura 2.1.

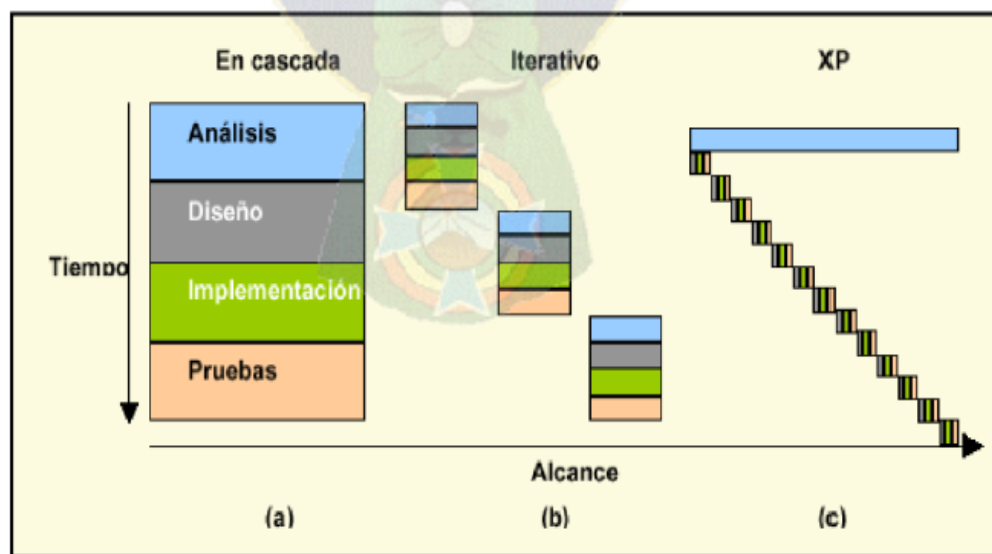


Figura 2. 1 Evolución de los largos ciclos de desarrollo en cascada (a), a ciclos más cortos (b), y a la mezcla que hace XP (c)

Fuente: [Fernan, 2002]

2.7.1 Metodología Programación Extrema (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP 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. XP es la primera metodología ágil y la que le dio conciencia al movimiento actual de metodologías ágiles. Podríamos decir que XP nace (oficialmente) hace algunos años en un proyecto desarrollado por Kent Beck cuando buscaba una nueva aproximación al problema del desarrollo de software que hiciera las cosas más simples de lo que nos tenían acostumbrados los métodos existentes.

Kent definió cuatro grandes tareas a realizar en el desarrollo de todo proyecto: planificación, diseño, desarrollo y pruebas; teniendo siempre presente las cuatro características básicas que debe reunir un programador XP: simplicidad en el desarrollo, comunicación entre las partes implicadas, realimentación para poder reutilizar y coraje.

2.7.2 Ciclo de Vida de XP

El ciclo de vida de XP se enfatiza en el carácter interactivo e incremental del desarrollo, Según Julio Ariel Hurtado una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas que en el caso de XP corresponden a un conjunto de historias de usuarios.

Las iteraciones son relativamente cortas ya que se piensa que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se va a obtener y esto va a representar una mejor calidad del producto a largo plazo. Existe una fase de análisis inicial orientada a programar las iteraciones de desarrollo y cada iteración incluye

diseño, codificación y pruebas, fases superpuestas de tal manera que no se separen en el tiempo.

2.7.3 Fases de la metodología XP

XP clasificada como un tipo de metodología ágil, según Pressman (2010) engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro fases estructurales: Planificación, Diseño, Desarrollo y Pruebas, las cuales puede observarse en la figura 2.2.

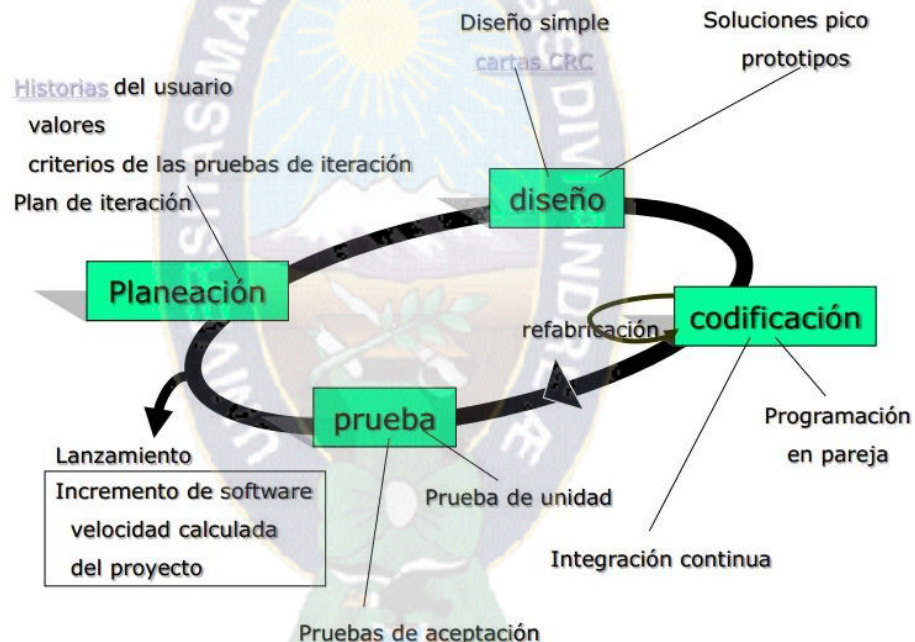


Figura 2. 2 Metodología XP
Fuente. [Pressman, 2010]

2.7.3.1 Fase 1: Planificación

XP plantea la planificación como un permanente diálogo entre la parte empresarial y técnica del proyecto, en la que los primeros decidirán el alcance, la prioridad, la composición de las versiones y la fecha de las mismas.

En cuanto a los técnicos, son los responsables de estimar la duración requerida para implementar las funcionalidades deseadas por el cliente, de informar sobre las

consecuencias de determinadas decisiones, de organizar la cultura de trabajo y, finalmente, de realizar la planificación detallada dentro de cada versión. XP no es sólo un método centrado en el código, sino que sobre todo es un método de gestión de proyectos software. [Beck, 2000]

a. Se redactan las historias de usuarios

Las historias de usuario tienen el mismo propósito que los casos de uso, pero no son lo mismo. Las escriben los propios clientes, tal y como ven ellos las necesidades del sistema.

Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Las historias de usuario son similares al empleo de escenarios, con la excepción de que no se limitan a la descripción de la interfaz de usuario.

El formato de las Historias de usuario consta de 8 partes (Figura 2.3), donde la descripción el objetivo y el nombre son el fundamento para obtener un buen desarrollo.

Historia de Usuario	N°	1	8
Nombre:	2	Prioridad:	3
Historia:	4		
Objetivo:	5		
Estimación:	6	Dependencia:	7

Figura 2. 3 Formato de la Tarjeta de la Historia de Usuario
Fuente. [Beck, 1999].

1. Número de Historia de Usuario.
2. Nombre de la Historia de Usuario.
3. Prioridad en la Planificación de entregas.
4. Descripción de la Historia de Usuario.
5. Objetivo de la Historia de Usuario.
6. Estimación del tiempo de desarrollo.
7. Dependencia del Programador.
8. Nombre del Sistema.

b. Se crea un plan de entregas

Las historias de usuario servirán para crear el plan estimado de entrega. Se convocará una reunión para crear el plan de entregas. El plan de entregas se usará para crear los planes de iteración para cada iteración. Es en este momento cuando los técnicos tomarán las decisiones técnicas y los comerciales las decisiones comerciales. En esta reunión estarán presentes tanto desarrolladores como los usuarios.

Con cada historia de usuario previamente evaluada en tiempo de desarrollo ideal, el cliente las agrupará en orden de importancia. Una semana ideal es cuánto tiempo costaría implementar dicha historia si no tenemos nada más que hacer, incluyendo la parte de test correspondiente.

De esta forma se puede trazar el plan de entregas en función de estos dos parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente. Las iteraciones individuales son planificadas en detalle justo antes de que comience cada iteración.

c. Se controla la velocidad del proyecto

La velocidad del proyecto es una medida de cuán rápido se está desarrollando. La velocidad de proyecto se usa para determinar cuántas historias de usuario pueden ser implementadas antes de una fecha dada (tiempo), o cuánto tiempo es necesario para llevar cabo un conjunto de historias (alcance). Cuando se realiza una planificación por alcance se divide el número total de semanas entre la velocidad de proyecto para determinar cuántas iteraciones estarán disponibles.

d. Se divide el proyecto en iteraciones

Cada iteración corresponde a un periodo de tiempo de desarrollo del proyecto de entre una y tres semanas. De esta forma, un proyecto, se divide en una docena de iteraciones, más o menos. Al principio de cada iteración se debería convocar una reunión para trazar el plan de iteración correspondiente.

Está prohibido intentar adelantarse e implementar cualquier cosa que no esté planeada para la iteración en curso. Habrá suficiente tiempo para añadir la funcionalidad extra cuando sea realmente importante según el plan de entregas.

e. Al comienzo de cada iteración se traza el plan de iteración

El plan de iteración consiste en seleccionar las historias de usuario que, según el plan de entregas, corresponderían a esta iteración. También se eligen qué pruebas de aceptación fallidas se corregirán. Cada historia de usuario se transformará en tareas de desarrollo. Cada tarea de desarrollo corresponderá a un periodo ideal de uno a tres días de desarrollo.

f. Corregir la propia metodología XP cuando falla

Cuando comencemos con un proyecto, seguiremos la metodología XP, pero debemos cambiar aquello que no funcione. Además los cambios que se realicen

deberán ser comunicados al resto del equipo, todo el mundo debe estar al corriente de los cambios. Esto no significa que cambiemos lo que no nos guste, sino que cambiemos aquello que no funciona con nuestro problema en particular.

2.7.3.2 Fase 2: Diseño

La metodología XP hace especial énfasis en los diseños simples y claros. XP establece unas recomendaciones o premisas a la hora de abordar esta etapa.

a. Simplicidad

Siempre costará menos tiempo de implementar un diseño sencillo que uno complejo. Por lo que, trataremos siempre de realizar las cosas de la manera más sencilla posible.

Si alguna parte de la implementación resulta especialmente compleja, deberías replantearla (divide y vencerás).

Así, cualquier cambio y modificación será mucho más sencillo. En ocasiones, realizar un diseño sencillo puede resultar una tarea especialmente difícil.

b. Elegir una metáfora para el sistema

Una metáfora para el sistema es una historia que todo el mundo puede contar acerca de cómo el sistema funciona [Beck, 1999]. La tarea de elegir una metáfora para el sistema nos permitirá mantener la coherencia de nombres de todo aquello que se va a implementar.

El nombre de los objetos o partes de nuestro sistema es muy importante. La tarea de “poner nombre”, sencillo a simple vista, no lo es tanto. Debemos elegir un sistema de nombres que permita que cualquiera que lo vea adivine la relación entre el objeto y aquello que representa.

c. Usar tarjetas CRC

Para poder diseñar el sistema como un equipo deberemos cumplir con tres principios: Cargo o Clase, Responsabilidad y Colaboración (CRC). Las tarjetas CRC permitirán desprendernos del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Una tarjeta CRC representa un objeto.

El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

d. No se añadirá funcionalidad en las primeras etapas

Debemos evitar caer en la tentación de ir añadiendo funcionalidades según se nos vayan ocurriendo, aun incluso que sepamos exactamente cómo implementarlas. Es decir, debemos centrarnos en la tarea que se ha fijado para hoy, y hacerla lo mejor posible.

Programaremos lo que se ha fijado, y no perderemos el tiempo en desarrollar código que no sabemos si será utilizado.

e. Re aprovechar cuando sea posible

Cuando eliminamos redundancia, eliminamos funcionalidad inútil, y rejuvenecemos antiguos diseños, estamos reciclando código. El reciclaje, dentro del ciclo de vida de un proyecto, ahorra tiempo e incrementa la calidad.

El reciclaje implicará mantener el código limpio y fácil de comprender, modificar y ampliar. Esto puede resultar un poco costoso al principio, pero resulta fundamental a la hora de realizar diseños futuros.

2.7.3.3 Fase 3: Desarrollo

Esta etapa debe reunir las siguientes características o cualidades: Análisis, test, codificación y diseño.

a. El cliente está siempre disponible

Una de las pocas condiciones que impone la metodología XP es tener al usuario siempre disponible. No sólo para ayudar al equipo de desarrollo, sino formando parte de él.

Todas las fases que se realizan en un proyecto XP requieren de comunicación con el usuario, preferiblemente cara a cara, en persona, sin intermediarios. Durante la reunión del plan de entregas, el usuario propondrá qué historia de usuario se incluye en cada plan. También se negociarán los plazos de entrega. El usuario o cliente tomará las decisiones que le afecten para alcanzar los objetivos de su negocio.

b. Se debe escribir código de acuerdo a los estándares

El código ha de ser desarrollado siguiendo los estándares de desarrollo para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo. Es decisiva, para poder plantear con éxito la propiedad colectiva del código. Ésta sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo.

c. Integrar frecuentemente

Los programadores deberán actualizar sus módulos con las versiones más recientes del trabajo realizado tan pronto como les sea posible.

De esta manera, todo el mundo trabajará siempre con la última versión. Dicha actualización es responsabilidad de cada pareja de programadores.

Esta integración se llevará a cabo cuando el éxito en las pruebas para su test correspondiente sea del 100 %, o cuando se trate de una parte que constituye un todo funcional, en cuanto esté acabada.

Esta frecuencia con la que se inserta el nuevo código nos permitirá una rápida detección de los problemas de compatibilidad.

d. Dejar las optimizaciones para el final

No optimizaremos el código hasta el final. Nunca trataremos de averiguar cuáles serán los posibles cuellos de botella del programa. Haz el trabajo, hazlo bien, y entonces hazlo rápido.

e. No trabajar más de 40 horas semanales

Trabajar horas extras absorbe el espíritu y la motivación del equipo. Aquellos proyectos que requieren horas extras para acabarse a tiempo pueden convertirse en un problema. En lugar de esto, utilizaremos las conocidas reuniones de plan de entregas para cambiar los objetivos del proyecto. También es una mala idea incorporar nueva gente al proyecto, una vez que este ya ha comenzado. Se trabajará un máximo de 40 horas semanales. Nadie es capaz de trabajar 60 horas a la semana y hacerlo con calidad.

2.7.3.4 Fase 4: Pruebas

Las unidades de test o pruebas constituyen uno de los pilares básicos de la Extreme Programming (XP).

Uno de los errores que se suele cometer es pensar que podemos dejar la construcción de los test para los últimos meses en la realización de un proyecto. Descubrir todos los errores que pueden aparecer lleva tiempo, y más si dejamos la depuración de todos para el final.

Las unidades de test están directamente relacionadas con el concepto de posesión del código. En cierta manera, una parte del código no será reemplazado si no supera los test que existen para ese código.

a. Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test - driven programming”). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o re codificar parte del mismo.

b. Detección y corrección de errores

Cuando se encuentra un error (“bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir.

Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.

c. Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra” (“Black box system tests”). Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución.

Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

2.8 INGENIERÍA WEB

La Ingeniería Web incluye nuevos enfoques disciplinarios, metodologías, herramientas, técnicas, guías y patrones donde tuvieron en cuenta científicos de este nuevo medio para cubrir los requisitos únicos de las aplicaciones web.

Internet y la World Wide Web están cambiando nuestras vidas.

Cada día es más común que tareas tales como leer el periódico la realicemos conectados con nuestro ordenador a Internet.

Es así que, durante la última década hemos asistido al crecimiento vertiginoso del desarrollo y uso de aplicaciones y sistemas Web cada vez más complejos y sofisticados.

Una definición de ingeniería Web es: “proceso utilizado para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad”.

“La ingeniería Web está relacionada con el establecimiento y utilización de principios científicos, de ingeniería y gestión, con enfoques sistemáticos y disciplinados del éxito y desarrollo, empleo y mantenimiento de sistemas y aplicaciones basadas en la Web de alta calidad”.

Es importante porque las aplicaciones web se integran cada vez más en las estrategias de negocios de las organizaciones grandes y pequeñas. Es necesario que estas aplicaciones sean prácticas, confiables y adaptables.

2.8.1 Modelos de Procesos WEB

Los modelos de procesos Web adaptan la filosofía de desarrollo ágil. El desarrollo ágil enfatiza un enfoque de desarrollo riguroso que incorpora rápidos ciclos de desarrollo. Dichas actividades del marco de trabajo se deben definir dentro de un proceso que:

- Adopte el cambio.
- Aliente la creatividad y la independencia del equipo de desarrollo y fortalezca la interacción con el usuario.
- Construya sistemas que utilicen pequeños equipos de desarrollo.
- Subraye el desarrollo evolutivo o incremental mediante el uso de cortos ciclos de desarrollo.

2.9 WEBML

WebML (Web Modeling Language) es un lenguaje de modelado gráfico utilizado para apoyar el diseño de sitios Web complejos que usan datos intensivamente.

Provee especificaciones gráficas formales para un proceso de diseño completo que puede ser asistido por herramientas de diseño visuales.

A diferencia de los otros métodos, WebML no hace énfasis en ser un método de desarrollo de hipermedia, sino un método para diseñar sitios web.

El proceso de desarrollo que propone se compone de la especificación de requisitos, el diseño del modelo de datos Entidad Relación, diseño del hipertexto, diseño de la arquitectura e implementación, cerrando el ciclo con una etapa de evaluación y pruebas.

Como el proceso es iterativo, cada una de las etapas del proceso puede retroalimentar a las anteriores. La última etapa es la instalación del sistema, que pasa de estar en un estado de desarrollo a uno de producción.

WebML está orientado para aplicaciones que tienen una alta interacción con datos y cuenta con los siguientes modelos (Figura 2.4): Estructura, Composición, Navegación y Presentación.

- Estructura: la organización de datos.
- Composición: definición de las páginas del sitio como un conjunto de subpáginas y las unidades primarias de publicación.
- Navegación: definición de los enlaces entre páginas y entre las unidades.
- Presentación: el posicionamiento de las unidades en la página y la definición de la apariencia grafica.

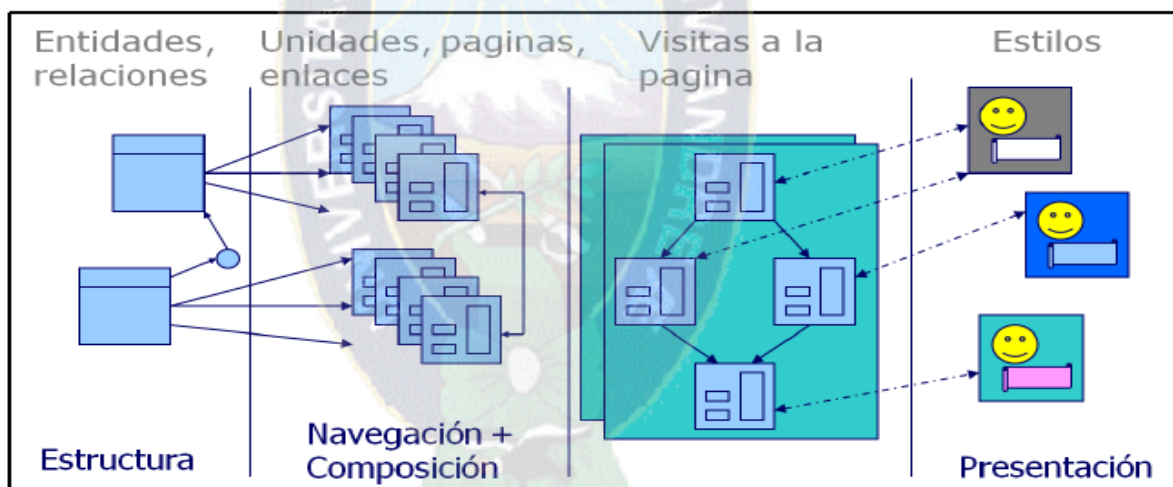


Figura 2. 4 Modelos Webml
Fuente: [Politécnico Milano, 2003]

2.9.1 Modelo de Estructura

El modelo de estructura o de datos representa las diferentes tablas de datos y sus relaciones que son necesarias para una aplicación Web concreta.

La notación que sigue WebML es diagramas de Entidad-Relación (E-R) que muestran todas las tablas, los diferentes campos de cada tabla, y las relaciones entre ellas. Entidad Relación como por ejemplo se ilustra la Figura 2.5.

Cuenta con las siguientes notaciones básicas:

- Las entidades son las cajas, etiquetadas con el nombre de la entidad en la mitad superior de la caja.
- Los atributos se enumeran en la mitad inferior de la caja de la entidad (opcional).
- Las relaciones están representadas por una línea continua entre dos cajas de la entidad.
- Jerarquías ISA se indican mediante una flecha que apunta desde el sub-entidad (concepto más específico) de la super-entidad (concepto general).



Figura 2. 5 Notaciones básicas
Fuente: [Politécnico Milano, 2003]

2.9.2 Modelo de Hipertexto

El modelo de hipertexto es llamado siteview, que describe las vistas del modelo estructural que estarán publicadas en el sitio web. Cada hipertexto describe una vista del sitio.

a. Elementos para el modelado

Para el modelo de hipertexto, Webml cuenta con diferentes elementos de modelado (Figura 2.6) como son: Unidad de Datos (Data-unit): Representa una entidad del modelo estructural y sus atributos visibles, muestra información sobre un solo objeto, son definidas para seleccionar una mezcla de información. Para definir una unidad de datos se requiere la indicación del concepto al cual se refiere la unidad y la selección de los atributos de la unidad.

Unidad Multidatos (Multidata-unit): Presenta varias instancias de una entidad simultáneamente, de acuerdo a algún criterio de selección, muestra información sobre un conjunto de objetos, presenta múltiples instancias de una entidad o componente. Una unidad multidatos tiene dos partes: el contenedor que incluye las instancias que se desean mostrar y la unidad de datos usada para la presentación de cada instancia.

Unidad Índice (Indexunit): Representa a un conjunto de referencias a entidades, donde cada referencia presenta una descripción de la entidad a la que apunta.

Presenta múltiples instancias de una unidad o componente como una lista, esta unidad tiene dos partes principales: el contenedor que incluye las instancias que se desean mostrar (las instancias deben ser una entidad, una relación o un componente) y los atributos usados como clave del índice.

Unidad de desplazamiento (Scrollerunit): Representa la visualización secuencial de un conjunto de entidades y está asociado a una unidad de datos; es más conocida como visita guiada, provee comandos para desplazarse a través de los objetos en un contenedor.

Esta unidad es normalmente usada junto con una unidad de datos, la cual representa el elemento actual visualizado del contenedor.

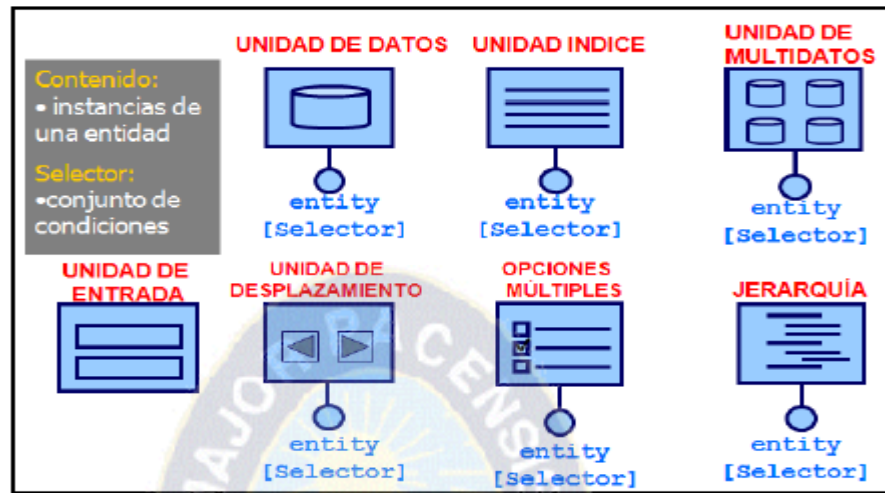


Figura 2. 6 Unidades de contenido básico
Fuente: [Politécnico Milano, 2003]

La vista del sitio se divide en dos modelos, el de composición y el de navegación.

2.9.3 Modelo de Composición

El modelo de composición define qué páginas estarán contenidas en el sitio web, y qué contenido tiene cada página. En el modelo de composición se construyen nodos de hipertexto con enlaces estructurales o internos, que no son convertidos en hiperenlaces (Figura 2.7).

El propósito del diagrama de composición es definir los nodos que forman parte del hipertexto contenido en el sitio Web, es decir, se especifican las páginas y las unidades (elementos atómicos de información que deben aparecer en el sitio Web) que componen el sitio Web. Elementos de contenido de cada página

- Unidades de datos. Dan la información de un solo objeto de datos. Ejemplo: de una actividad
- Unidad Multi-datos. Dan información de varios objetos de datos
- Unidad Índice. Da la lista de varios objetos, pero sin mostrar la información de dichos objetos.

- Unidad Scroller. Da un conjunto de elementos que se puede avanzar entre ellos con scroll.
- Unidad de Filtro. Da una página de búsqueda donde se pueden poner diferentes parámetros para buscar según unos criterios.

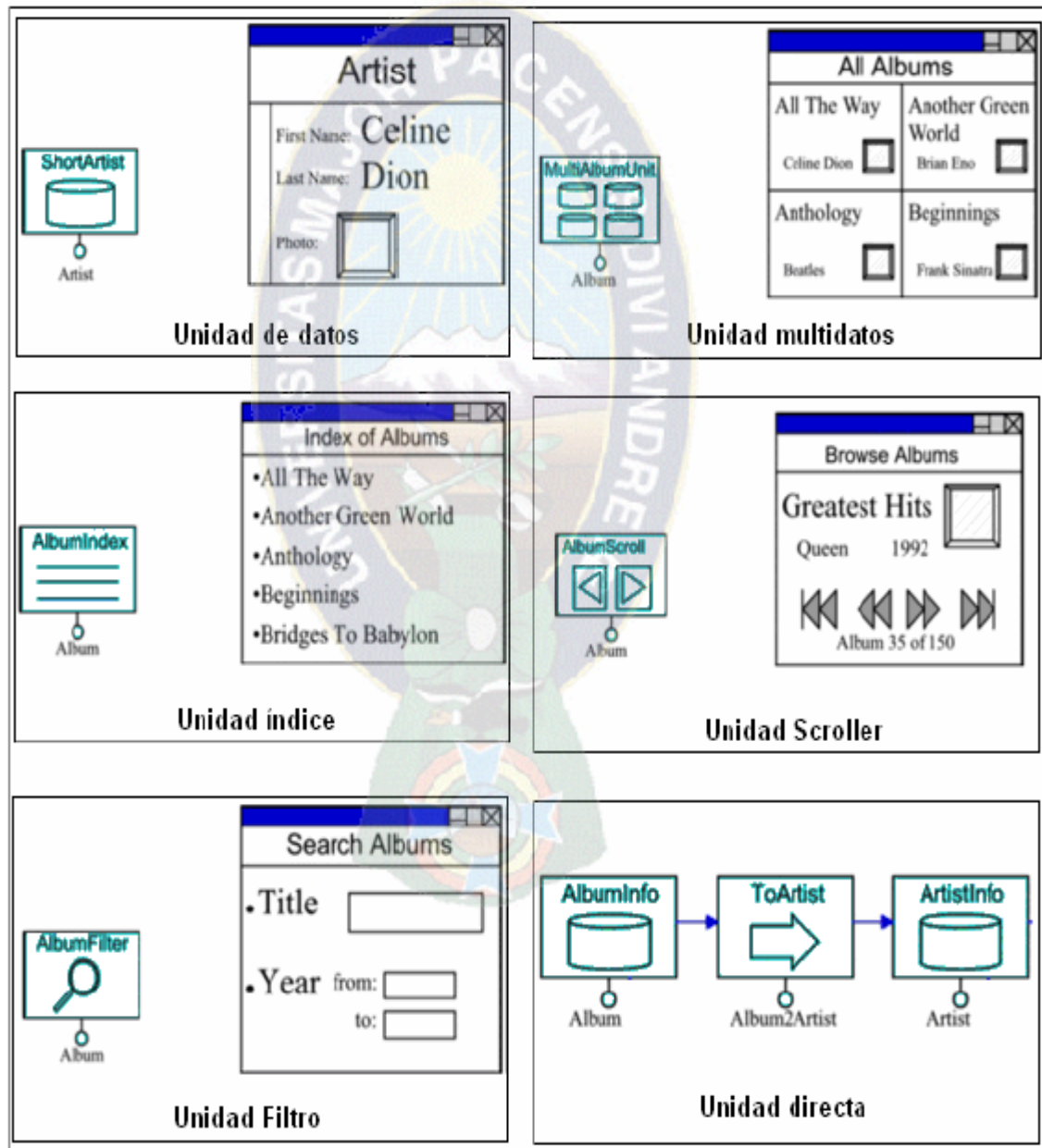


Figura 2. 7 Notación gráfica de unidades y su posible ejecución en HTML
Fuente: [Ceri, 2000]

2.9.4 Modelo de Navegación

El modelo de navegación representa los enlaces entre las diferentes páginas y sus elementos de contenido, tiene el propósito de especificar las relaciones. Pueden definirse las relaciones entre las unidades dentro de una sola página, o de una unidad en diferentes páginas, además de especificar la forma en la cual las unidades y las páginas son conectadas para formar un hipertexto así como se ilustra en la Figura 2.8, para esto WebML provee la noción de enlaces, de los cuales hay dos tipos:

Enlaces contextuales: conectan unidades de una forma coherente a la semántica expresada por el diagrama de estructura de la aplicación. Un enlace contextual lleva información (de contexto) de la unidad de origen a la unidad destino, esta información es usada para determinar el objeto o conjunto de objetos a ser mostrados en la unidad destino.

Enlaces no contextuales: conectan páginas libremente, independientemente del contexto.



Figura 2. 8 Diagrama de navegación con enlaces contextuales y su posible ejecución en HTML

Fuente: [Webml training3]

Se integran los modelos de composición y de navegación, porque las características de uno puede afectar al otro modelo (por ejemplo, para mostrar un juego de unidades relacionadas en una página, deben establecerse los enlaces apropiados).

Colectivamente, ellos forman al modelo del hipertexto que completamente define el volumen de información y topología de conexiones de un sitio de WebML.

2.9.5 Modelo de Presentación

El modelo de presentación en WebML se efectúa a través de la aplicación de hojas de estilo XSL a documentos XML que representan una instancia del modelo navegacional, y el resultado de la transformación es un documento HTML. Se incorpora el estilo de presentación a cada página.

El ejemplo de la librería se muestra con la notación de WebML en la Figura 2.9.

El modelo que se muestra es un siteview con los modelos de composición y navegación.

El modelo de composición está formado por cada nodo individual y por cada página.

La página principal muestra un enlace a una página que es un índice de editoriales, y contiene también dos formularios de búsqueda, en WebML una unidad de datos es una selección de atributos de una entidad de datos del modelo E-R; por ejemplo, la unidad de datos biografía se deriva de la entidad autor y sólo selecciona el campo biografía.

El modelo de navegación está integrado por los enlaces que cruzan la frontera de las páginas y que salen de las unidades que se presentan de forma individual.

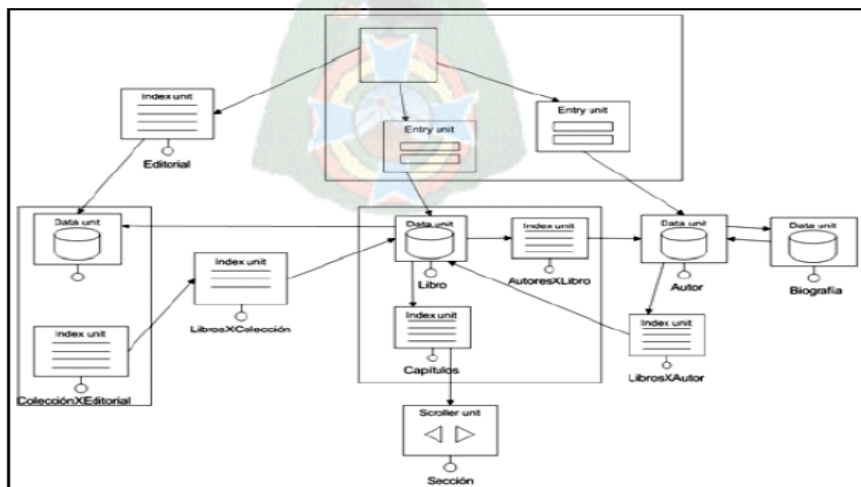


Figura 2.9 Ejemplo de WebML
Fuente: [Solís, 2008]

2.10 TECNOLOGÍAS DE SOFTWARE

Las herramientas necesarias para el desarrollo e implementación de cada iteración y entrega del sistema son las siguientes:

- Lenguaje de programación PHP.
- Servidor web apache.
- Administrador de base de datos My SQL.
- Sistema operativo Windows XP o Linux en sus diferentes versiones.

PHP es un lenguaje de programación generalmente usado en la programación de sitios web dinámicos y actualmente es casi el lenguaje de desarrollo de sitios web más usado en todo el mundo. Fue originalmente **creado por Rasmus Lerdorf** para presentar su portafolio de trabajo y otras cosas en el año de 1994. Originalmente fue desarrollado en perl. PHP al principio significaba **Personal Home Page** pero con el tiempo como ya es desarrollado por otro grupo se llama **PHP Hypertext preprocesor**.

PHP permite la conexión a todo tipo de servidores de base de datos como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Características del lenguaje PHP

- PHP es un **lenguaje interpretado**, solo se necesita un navegador web para ejecutarlo.
- Es un **lenguaje del lado del servidor**, por lo que los script se ejecutan remotamente y el resultado aparece en la máquina cliente (local).
- Tiene soporte para muchos tipos de bases de datos, entre las principales están MySQL, PostgreSQL, SQLite, entre otras.
- La sintaxis es parecida a la del lenguaje C (Que también tiene un parecido a perl).
- Es embebido en código HTML.
- Soporte de orientación a objetos.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (Linux), Windows y Mac OS X, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache.

El **servidor HTTP Apache** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de datos. Esto puedes ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder y procesar datos guardados en un computador.

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas.

MySQL es muy utilizado en aplicaciones Web como en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación Web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente.

Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras.

CAPÍTULO 3

MARCO APLICATIVO

3.1 INTRODUCCIÓN

Para el desarrollo de cualquier sistema de información es necesario contar con un modelo de desarrollo de software, el cual indique todas las etapas necesarias que se deben seguir para poder lograr desarrollar un producto de calidad.

En este capítulo se hará uso de la metodología XP Programación Extrema para las fases de la planificación, diseño, desarrollo y pruebas, además de utilizar la metodología WEBML para el diseño navegacional y el diseño de la interfaz, ya que debido a que la segunda fase de la metodología xp esta orientado al diseño y es por eso que se justifica su uso.

En este capítulo se muestra la aplicación de cada una de las fases y procesos de la metodología ágil XP (programación extrema) además de los procesos de modelado con WebML las cuales mostramos en la siguiente tabla:

FACES DE LA METODOLOGÍA XP	Procesos XP	Procesos WebML
Fase: PLANIFICACIÓN	<ul style="list-style-type: none">• Historias de Usuarios• Plan de Entregas• Iteraciones	
Fase: DISEÑO	<ul style="list-style-type: none">• Tarjetas CRC	<ul style="list-style-type: none">• Modelo de estructura• Modelo de componentes• Modelo de Navegación
Fase: DESARROLLO	<ul style="list-style-type: none">• Interfaz grafico	
Fase: PRUEBAS	<ul style="list-style-type: none">• Pruebas de Aceptación• Pruebas de Stress	<ul style="list-style-type: none">• Modelo de Presentación

Tabla 3. 1 Fases y procesos de la Metodología XP y WebML
Fuente: Elaboración Propia

3.2 PLANIFICACIÓN

En esta fase se llevaron a cabo reuniones en las cuales el cliente describió cada uno de los requerimientos y funcionalidades que necesitan que el sistema pueda atender.

3.2.1 Obtención de requerimientos

La obtención de los requerimientos es una parte importante en el desarrollo de software. A continuación se muestran en la siguiente Tabla 3.2 cada uno de los requerimientos que fueron descritos por el cliente y los futuros usuarios del sistema:

REFERENCIAS	REQUERIMIENTOS	PRIORIDAD
R1	Registro de apertura de proyectos.	Alta
R2	Registrar seguimiento de proyectos	Alta
R3	Registro de reprogramación de proyectos	Alta
R4	Registro de personal	Medio
R5	Registro de Agencias	Alta
R6	Reportes	Alta

Tabla 3. 2 Lista de Requerimientos

Fuente: Elaboración Propia

3.2.2 Historias de usuario

A partir del conjunto de requerimientos, se pudo construir junto al cliente un conjunto de historias de usuario, las cuales cuentan con prioridades, riesgos e iteraciones que son descritos a continuación:

- **Prioridad:** de acuerdo a conversaciones con el cliente se tiene tres grados de prioridad para el desarrollo e implementación de las historias de usuario, alta, media y baja.
- **Riesgos en desarrollo:** es el riesgo que existe al desarrollar de forma inadecuada la solución de las historias de usuario, se tienen tres grados de riesgo los cuales son alto, medio y bajo.

- **Iteración asignada:** es el numero de iteración en el cual se espera poder implementar la historia de usuario, el tiempo promedio de entrega en cada iteración es aproximadamente de 5 semanas; se pretende desarrollar e implementar todas las historias de usuario en 3 iteraciones.
- **Puntos estimados:** es el tiempo promedio en semanas de desarrollo los cuales se miden en la escala de 1 a 5 semanas de desarrollo aproximadamente.

Luego de establecer los puntos de medida de las historias de usuario a continuación se describirán cada una de estas, junto una descripción de las mismas y algunas observaciones necesarias para su desarrollo.

Registro de apertura de proyectos: El sistema debe registrar la información necesaria de los proyectos que realiza la empresa, lo cual es realizado por el administrador del sistema.

Historia de Usuario	
Numero: 1	Usuario: Administrador
Nombre historia: Registro de apertura de proyectos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Johnny Chavez Quispe	
Descripción: El registró de proyectos es indispensable para el sistema, se lo realizara de acuerdo a las características que la empresa requiere tomando los datos necesarios de los proyectos los cuales son: código del proyecto, nombre del proyecto, objetivo del proyecto, responsable del proyecto, fecha de inicio, fecha de fin, presupuesto.	
Observaciones: Los datos del proyecto deben ser proporcionados por las áreas técnicas de la empresa, los requisitos marcados con (*) son obligatorios además que esta operación solo lo puede realizar el administrador del sistema.	

Tabla 3.3 Historia de usuario: Registro de apertura de proyectos
Fuente: Elaboración Propia

Registrar seguimiento de proyectos: El sistema debe registrar la información del avance de cada proyecto que realiza la empresa.

Historia de Usuario	
Numero:2	Usuario: Encargado
Nombre historia: Registrar seguimiento de proyectos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Johnny Chavez Quispe	
Descripción: El registro del seguimiento de los proyectos es indispensable para el sistema, se lo realizara cada día para que la empresa este informado del avance de los proyectos, tomando los datos necesarios del seguimiento del proyecto los cuales son: código del proyecto, objetivo del proyecto, fecha de inicio, seguimiento hasta la fecha, estado del avance del proyecto, presupuesto establecido, total invertido saldo, observación para la conclusión del proyecto.	
Observaciones: Los datos del seguimiento del proyecto deben ser proporcionados por los encargados de cada proyecto, los requisitos marcados con (*) son obligatorios además que esta operación solo lo puede realizar el encargado de cada proyecto.	

Tabla 3. 4 Historia de usuario: Registrar seguimiento de proyectos
Fuente: Elaboración Propia

Registro de reprogramación de proyectos: El sistema debe registrar la reprogramación de los proyectos si así la empresa lo necesitaría.

Historia de Usuario	
Numero: 3	Usuario: Administrador
Nombre historia: Registrar reprogramación de proyectos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Johnny Chavez Quispe	

<p>Descripción: El registro de reprogramación de los proyectos es indispensable para el sistema, se lo realizara en consenso con el gerente general de la empresa, tomando los datos necesarios para la reprogramación del proyecto los cuales son: código del proyecto, fecha de inicio, fecha de fin, justificación de la reprogramación, presupuesto, nuevo presupuesto, estado del proyecto, observación del proyecto.</p>
<p>Observaciones: Los datos de la reprogramación del proyecto deben ser proporcionados por los encargados de cada proyecto y el gerente de la empresa, los requisitos marcados con (*) son obligatorios además que esta operación solo lo puede ser realizado por el administrador del sistema.</p>

Tabla 3. 5 Historia de usuario: Registrar reprogramación de proyectos
Fuente: Elaboración Propia

Registro de personal: El sistema debe registrar a todo el personal que constituye la empresa con sus diferentes cargos y ocupaciones, que puedan ser tomados en cuenta en los diferentes proyectos que la empresa realiza.

Historia de Usuario	
Numero: 4	Usuario: Administrador
Nombre historia: Registro de personal	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Johnny Chavez Quispe	
<p>Descripción: El registro de personal es indispensable para el sistema, se lo realizara mediante la unidad de recursos humanos, tomando los datos necesarios para el registro de personal los cuales son: código de la persona, cedula, nit, profesión, cargo, nombres, Ap. Paterno, Ap. Materno, celular, teléfono, nacimiento, sexo, civil, N° de hijos, país.</p>	
<p>Observaciones: Los datos del registro de personal deben ser proporcionados por el área de recursos humanos, los requisitos marcados con (*) son obligatorios además que esta operación puede ser realizado por el administrador del sistema y el encargado del área recursos humanos.</p>	

Tabla 3. 6 Historia de usuario: Registro de personal
Fuente: Elaboración Propia

Registro de Agencias: El sistema deberá registrar las agencias con las que cuenta la empresa y las nuevas, teniendo en cuenta los datos que sean necesarios para el manejo adecuado de los proyectos que se realizan en la empresa.

Historia de Usuario	
Numero: 5	Usuario: Administrador
Nombre historia: Registro de Agencias	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Johnny Chavez Quispe	
Descripción: El registro de agencias es un modulo que se crea de acuerdo a las agencias con las que cuenta la empresa y las agencias que la empresa pretende implementar teniendo un seguimiento adecuado para cada agencia asignando los datos reales para cada agencia los cuales son: código de agencia, dirección, ciudad.	
Observaciones: Los datos para el registro de agencias son de acuerdo a la ubicación de las agencias de la empresa ya establecidos, teniendo en cuenta que todos los datos son obligatorios y son realizados por el administrador del sistema.	

Tabla 3. 7 Historia de usuario: Registro de agencias
Fuente: Elaboración Propia

Reportes: El sistema realiza los reportes de los módulos, así brindando los reportes de manera general, por fecha, por código de acuerdo al reporte que se requiera además dando la opción a imprimir y generar un documento en formato pdf del reporte.

Historia de Usuario	
Numero: 6	Usuario: Administrador
Nombre historia: Reportes	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2.5	Iteración asignada: 3
Programador responsable: Johnny Chavez Quispe	

Descripción: Es un modulo indispensable para tener un control adecuado de los módulos implementados en el sistema, ya que a través de los reportes podemos mostrar los resultados de las operaciones que realiza la empresa a través del sistema.

Observaciones: Los reportes se los realiza de acuerdo al requerimiento, tomando en cuenta las características y opciones que te da cada reporte según el modulo.

Tabla 3. 8 Historia de usuario: Reportes

Fuente: Elaboración Propia

La siguiente Tabla 3.9 muestra un resumen de las historias de usuario y su implementación.

No	Nombre:	Prioridad	Riesgo	Puntos estimados	Iteración
1	Registro de apertura de proyectos.	Alta	Alto	2	1
2	Registrar seguimiento de proyectos	Alta	Alto	2	1
3	Registro de reprogramación de proyectos	Alta	Alto	3	2
4	Registro de personal	Medio	Medio	2	2
5	Registro de agencias	Alta	Alto	2	3
6	Reportes	Alta	Alto	2,5	3

Tabla 3. 9 Resumen de las historias de usuario

Fuente: Elaboración Propia

3.2.3 Plan de entrega (Release planning)

Después de que ya definimos las historias de usuario, se creó junto al cliente un plan de publicaciones, en donde se detalla la implementación de las historias de usuario para cada versión del sistema y una aproximación del tiempo de desarrollo requerido para su implementación, además de las tarjetas de tareas (task cards) en la que se tendrá una aproximación del tiempo de desarrollo de cada una de las tareas necesarias para la implementación de las principales historias de usuario en las distintas iteraciones.

Al igual que las historias de usuario, las tareas cuentan con una estructura similar, tienen un nombre, un número de tarea, puntos estimados y el número de historia de usuario a la cual pertenecen, cuentan además con:

- **Tipo de tarea:** existen varios tipos de tarea entre los cuales están el de desarrollo, corrección, mejora, o también alguna otra que será especificada.
- **Fecha de inicio y finalización:** son las fechas en que iniciaran y finalizaran las actividades de cada una de las tareas.

3.2.3.1 Primera iteración

En esta iteración se pretende implementar las historias de usuario con mayor grado de prioridad para el cliente, ya que los mismos son necesarios para que el cliente de la empresa pueda ya familiarizarse con la interfaz del sistema destinada a la atención al cliente y con los productos que la empresa dispone. Las historias de usuario que se implementaron en esta iteración son las siguientes:

- H1: Registro de apertura de proyectos.
- H2: Registro de seguimiento de proyectos.

A continuación mostramos el conjunto de tareas necesarias para la implementación de la historia de usuario N°1 Registro de apertura de proyectos.

Diseño del formulario de registro de apertura de proyectos: esta tarea permitirá almacenar los datos de los proyectos con los que cuenta la empresa, ya que es indispensable para la realizar las diferentes operaciones con estos proyectos.

Tarea	
Numero tarea: 1	Numero historia: 1
Nombre tarea: Registro de apertura de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 2

Fecha de inicio: 01-01-2014	Fecha fin: 10-01-2014
Descripción: Se creara un formulario que permita introducir el código del proyecto, nombre del proyecto, objetivo del proyecto, responsable del proyecto, fecha de inicio, fecha de fin, presupuesto y verificar si la operación se realizo con éxito.	

Tabla 3. 10 Tarea: diseño del formulario de registro de proyectos
Fuente: Elaboración Propia

Diseño del formulario de modificación de Proyectos: Esta tarea nos permitirá realizar las modificaciones necesarias de acuerdo a los datos actuales según lo que se requiera cambiar.

Tarea	
Numero tarea: 2	Numero historia: 1
Nombre tarea: Modificar apertura de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 01-01-2014	Fecha fin: 10-01-2014
Descripción: Este formulario nos permite realizar cambios de los proyectos según el requerimiento.	

Tabla 3. 11 Tarea: modificación de datos de los proyectos
Fuente: Elaboración Propia

Diseño del formulario de eliminación del proyecto: Esta tarea nos permitirá realizar la eliminación de los proyectos que sean innecesarios para el sistema.

Tarea	
Numero tarea: 3	Numero historia: 1
Nombre tarea: Eliminación del proyecto	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 13-01-2014	Fecha fin: 17-01-2014
Descripción: Este formulario nos permite realizar la eliminación del proyecto ya que este es innecesario para el sistema.	

Tabla 3. 12 Tarea: Eliminación del proyecto
Fuente: Elaboración Propia

Listado de proyectos: Con esta tarea se podrá visualizar un listado de los proyectos según se han registrado, además se los mostrara alfabéticamente.

Tarea	
Numero tarea: 4	Numero historia: 1
Nombre tarea: Listado de Proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 20-01-2014	Fecha fin: 24-01-2014
Descripción: Se mostrara un listado de todos los proyectos ordenados alfabéticamente según como fueron registrados los datos.	

Tabla 3. 13 Tarea: Listado de proyectos
Fuente: Elaboración Propia

A continuación mostramos el conjunto de tareas necesarias para la implementación de la historia de usuario N°2 Registro de seguimiento de proyectos.

Diseño del formulario de Registro de seguimiento de proyectos: Esta tarea permitirá almacenar los datos de los proyectos en ejecución con los que la empresa controla cada proyecto.

Tarea	
Numero tarea: 1	Numero historia: 2
Nombre tarea: Registro de seguimiento de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 27-01-2014	Fecha fin: 31-01-2014
Descripción: Se creara el formulario que nos permite registrar los datos de los proyectos e introducir código del proyecto, objetivo del proyecto, fecha de inicio, seguimiento hasta la fecha, estado del avance del proyecto, presupuesto establecido, total invertido saldo, observación para la conclusión del proyecto.	

Tabla 3. 14 Tarea: Registro de seguimiento de proyectos
Fuente: Elaboración Propia

Diseño del formulario de modificación de seguimiento de proyectos: Esta tarea nos permitirá realizar las modificaciones necesarias de acuerdo a los datos actuales según lo que se requiera cambiar.

Tarea	
Numero tarea: 2	Numero historia: 2
Nombre tarea: Modificar seguimiento de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 03-02-2014	Fecha fin: 07-02-2014
Descripción: Este formulario nos permite realizar cambios en el registro de seguimiento de proyectos según el requerimiento.	

Tabla 3. 15 Tarea: modificación de datos del registro de seguimiento de proyectos
Fuente: Elaboración Propia

Diseño del formulario de eliminación del registro de seguimiento de proyectos: Esta tarea nos permitirá realizar la eliminación de los registros de seguimiento proyectos que sean innecesarios para el sistema.

Tarea	
Numero tarea: 3	Numero historia: 2
Nombre tarea: Eliminación del registro de seguimiento de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 10-02-2014	Fecha fin: 14-02-2014
Descripción: Este formulario nos permite realizar la eliminación del registro de seguimiento de proyecto ya que este sea innecesario para el sistema.	

Tabla 3. 16 Tarea: Eliminación del registro de seguimiento de proyectos
Fuente: Elaboración Propia

Listado de los registros de seguimiento de proyectos: Con esta tarea se podrá visualizar un listado de los registros de los seguimientos de cada proyecto según se han registrado, además se los mostrara alfabéticamente.

Tarea	
Numero tarea: 4	Numero historia:2
Nombre tarea: Listado de los registros de seguimientos de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 17-02-2014	Fecha fin: 21-02-2014
Descripción: Se mostrara un listado de todos los registros de seguimientos de un determinado proyecto ordenados alfabéticamente según como fueron registrados los datos.	

Tabla 3. 17 Tarea: Listado de proyectos

Fuente: Elaboración Propia

3.2.3.2 Segunda iteración

En esta iteración ya se implementaran las funcionalidades dedicadas a la reprogramación de proyectos y además al registro del personal según sean necesarios para manejo adecuado de la información. Las historias de usuario que se implementaron en esta iteración son la las siguientes:

- H3 Registro de reprogramación de proyectos
- H4 Registro de personal

A continuación mostraremos el conjunto de tareas necesarias para la implementación de la historia de usuario N°3 Registro de reprogramación de proyectos:

Diseño del formulario de registro de reprogramación de proyectos: Esta tarea nos permitirá reprogramar los diferentes proyectos que realiza la empresa.

Tarea	
Numero tarea: 1	Numero historia:3
Nombre tarea: Registro de reprogramación de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 24-02-2014	Fecha fin: 28-02-2014
Descripción: Se creara el formulario que nos permite reprogramar los distintos proyectos que tiene que realiza la empresa introduciendo primeramente los datos de: código del proyecto, fecha de inicio, fecha de fin, reprogramación del proyecto, presupuesto, nuevo presupuesto, estado del proyecto, observación del proyecto.	

Tabla 3. 18 Tarea: Registro de reprogramación de proyectos

Fuente: Elaboración Propia

A continuación mostraremos el conjunto de tareas necesarias para la implementación de la historia de usuario No 4 Registro de personal.

Diseño del formulario de registro de personal: esta tarea nos permitirá crear y registrar los datos del personal de la empresa.

Tarea	
Numero tarea: 1	Numero historia: 4
Nombre tarea: Registro de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 03-03-2014	Fecha fin: 07-03-2014
Descripción: Se creará un formulario que permita introducir datos necesarios del personal de la empresa, como ser el código de la persona, cedula, nit, profesión, cargo, nombres, Ap. Paterno, Ap. Materno, celular, teléfono, nacimiento, sexo, civil, N° de hijos, país y verificar si la operación se realizó con éxito.	

Tabla 3. 19 Tarea: diseño del formulario de registro de personal

Fuente: Elaboración Propia

Diseño del formulario de modificación del registro de personal: Esta tarea nos permitirá realizar las modificaciones necesarias de acuerdo a los datos actuales según lo que se requiera cambiar.

Tarea	
Numero tarea: 2	Numero historia: 4
Nombre tarea: Modificar registro de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 10-03-2014	Fecha fin: 14-03-2014
Descripción: Este formulario nos permite realizar cambios en los registros de personal según el requerimiento de la unidad de recursos humanos.	

Tabla 3. 20 Tarea: modificación de datos del personal

Fuente: Elaboración Propia

Diseño del formulario de eliminación del registro de personal: Esta tarea nos permitirá realizar la eliminación del personal que sean innecesarios para la empresa y para el sistema.

Tarea	
Numero tarea: 3	Numero historia: 4
Nombre tarea: Eliminación del registro de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 17-03-2014	Fecha fin: 21-03-2014
Descripción: Este formulario nos permite realizar la eliminación del personal ya que este es innecesario para la empresa.	

Tabla 3. 21 Tarea: Eliminación del registro de personal
Fuente: Elaboración Propia

Listado de los registros del personal: Con esta tarea se podrá visualizar un listado del personal según se han registrado, además se los mostrara alfabéticamente.

Tarea	
Numero tarea: 4	Numero historia: 1
Nombre tarea: Listado del personal	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 24-01-2014	Fecha fin: 28-01-2014
Descripción: Se mostrara un listado de todo el personal de la empresa ordenados alfabéticamente según como fueron registrados los datos y según el requerimiento de la empresa.	

Tabla 3. 22 Tarea: Listado del registro de personal
Fuente: Elaboración Propia

3.2.3.3 Tercera iteración

En esta última iteración se pretende implementar las últimas funcionalidades necesarias para cubrir todos los requerimientos expuestos por el cliente en las historias de usuario.

Las historias de usuario que se implementaron en esta iteración son las siguientes:

- H5 Registro de Agencias
- H6 Reportes

A continuación mostramos el conjunto de tareas necesarias para la implementación de la historia de usuario N°5 Registro de Agencias.

Diseño de la interfaz de registro de agencias: Esta tarea permitirá registrar las agencias ya establecidas y las nuevas agencias que la empresa pretenda abrir en el futuro.

Tarea	
Numero tarea: 1	Numero historia: 5
Nombre tarea: Creación de agencia	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 31-03-2014	Fecha fin: 04-04-2014
Descripción: Se creara el formulario para crear las agencias que tiene la empresa registrando los datos de: código de agencia, dirección, ciudad.	

Tabla 3. 23 Tarea: diseño del formulario de registro de agencia
Fuente: Elaboración Propia

Diseño del formulario de actualizar agencia: Esta tarea nos permitirá actualizar la información de los datos de las agencias creadas de acuerdo a los cambios necesarios.

Tarea	
Numero tarea: 2	Numero historia: 5
Nombre tarea: Actualizar agencia	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 07-04-2014	Fecha fin: 11-04-2014
Descripción: Se creara el formulario de actualización para realizar los cambios necesarios en las agencias.	

Tabla 3. 24 Tarea: actualizar agencia
Fuente: Elaboración Propia

Diseño del formulario de eliminación de agencia: Esta tarea nos permitirá eliminar o dar de baja a las agencias que sean innecesarios para la empresa.

Tarea	
Numero tarea: 3	Numero historia: 5
Nombre tarea: Eliminación de agencia	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 14-04-2014	Fecha fin: 18-04-2014
Descripción: Se creara el formulario para eliminar la agencia que ya no funcione.	

Tabla 3. 25 Tarea: Eliminación de agencia
Fuente: Elaboración Propia

Listado de agencias: Con esta tarea se podrá visualizar un listado de las agencias según se han registrado.

Tarea	
Numero tarea: 4	Numero historia: 5
Nombre tarea: Listado de agencias	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 21-04-2014	Fecha fin: 25-04-2014
Descripción: Se mostrara un listado de todo el personal de la empresa ordenados alfabéticamente según como fueron registrados los datos y según el requerimiento de la empresa.	

Tabla 3. 26 Tarea: Listado de agencias
Fuente: Elaboración Propia

A continuación mostramos el conjunto de tareas necesarias para la implementación de la historia de usuario N°6 Reportes.

Diseño de la interfaz de reportes de proyectos: Esta tarea permitirá seleccionar los reportes de los proyectos por agencias, podemos sacar los reportes por: Listado general, por fecha, por código, por agencia.

Tarea	
Numero tarea: 1	Numero historia: 6
Nombre tarea: Reporte de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 28-04-2014	Fecha fin: 02-05-2014
Descripción: Se mostrara un listado de todo el personal de la empresa ordenados alfabéticamente según como fueron registrados los datos y según el requerimiento.	

Tabla 3. 27 Tarea: reporte de proyectos

Fuente: Elaboración Propia

Diseño de la interfaz de reportes de agencias: Esta tarea permitirá seleccionar la agencia para ver los proyectos en ejecución y los proyectos disponibles en dicha agencia para luego con ello tener un control de los proyectos, además de tener las opciones de imprimir o crear un documento en formato pdf.

Tarea	
Numero tarea: 2	Numero historia: 6
Nombre tarea: Reportes de agencias	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 05-05-2014	Fecha fin: 09-05-2014
Descripción: Se creara la interfaz para realizar el reporte de las agencias para ver los proyectos disponibles de la agencia con la opción de imprimirlo o crear un documento pdf.	

Tabla 3. 28 Tarea: reporte de agencias

Fuente: Elaboración Propia

Diseño de la interfaz de reportes de seguimiento de proyectos: Esta tarea permitirá los reportes de los registros de los seguimientos de cada proyecto se realizara por listado general, por fecha y por código de proyecto esto según el requerimiento establecido.

Tarea	
Numero tarea: 3	Numero historia: 6
Nombre tarea: Reportes de seguimiento de proyectos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 05-05-2014	Fecha fin: 09-05-2014
Descripción: Se creara la interfaz para realizar el reporte de los seguimientos de los proyectos para ver el avance del proyecto con la opción de imprimirlo.	

Tabla 3. 29 Tarea: reporte de seguimiento de proyectos

Fuente: Elaboración Propia

Diseño de la interfaz de reportes del personal: Esta tarea permitirá seleccionar el reporte del personal de cada proyecto, que se realizaron o se realizan en cada proyecto a través de un listado general, por fecha.

Tarea	
Numero tarea: 4	Numero historia: 6
Nombre tarea: Reportes de personal	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 12-05-2014	Fecha fin: 16-05-2014
Descripción: Se creara la interfaz para realizar el reporte del personal de cada proyecto con la opción de imprimirlo o crear un documento pdf.	

Tabla 3. 30 Tarea: reporte de seguimiento de proyectos
Fuente: Elaboración Propia

Luego de especificar el conjunto de tareas necesarias en las historias de usuario podemos construir la tabla 3.31 con el cronograma de actividades a desarrollarse durante el ciclo de vida del proyecto.

N	Fases de desarrollo	Dici		Enero				Febrero				Marzo				Abril				Mayo			
		Se		Semanas				Semanas				Semanas				Semanas				Semanas			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Planificación																						
2	Diseño																						
3	Desarrollo																						
4	Pruebas																						
				Planificación Global				1ª iteración				2ª iteración				3ª iteración							

Tabla 3. 31 Cronograma de actividades: iteraciones del proyecto
Fuente: Elaboración propia

3.3 DISEÑO

En esta fase se presentan diseños simples como sugiere la metodología XP, ya que para lograr una mejor comprensión de la funcionalidad del sistema, el diseño debe ser sencillo entendible para que los costos de tiempo, esfuerzo y desarrollo sean bajos. El diseño de los modelos está basado en el lenguaje de modelado WebML ya que la parte más importante de un S.C.M. es la integración de los módulos utilizando la tecnología.

3.3.1 Modelo de Entidad Relación

Para el diseño de los datos empleamos la herramienta case como: Enterprise Architect para Webml, la cual cuenta con un entorno de desarrollo grafico basado en diagramas, además de otras herramientas case que ayuden al diseño de los modelos y diagramas que necesitemos emplear. La siguiente Figura 3.1 muestra todas las entidades, las jerarquías utilizadas en la base de datos del sistema, además muestra las relaciones y la cardinalidad que existe entre cada una de estas entidades.

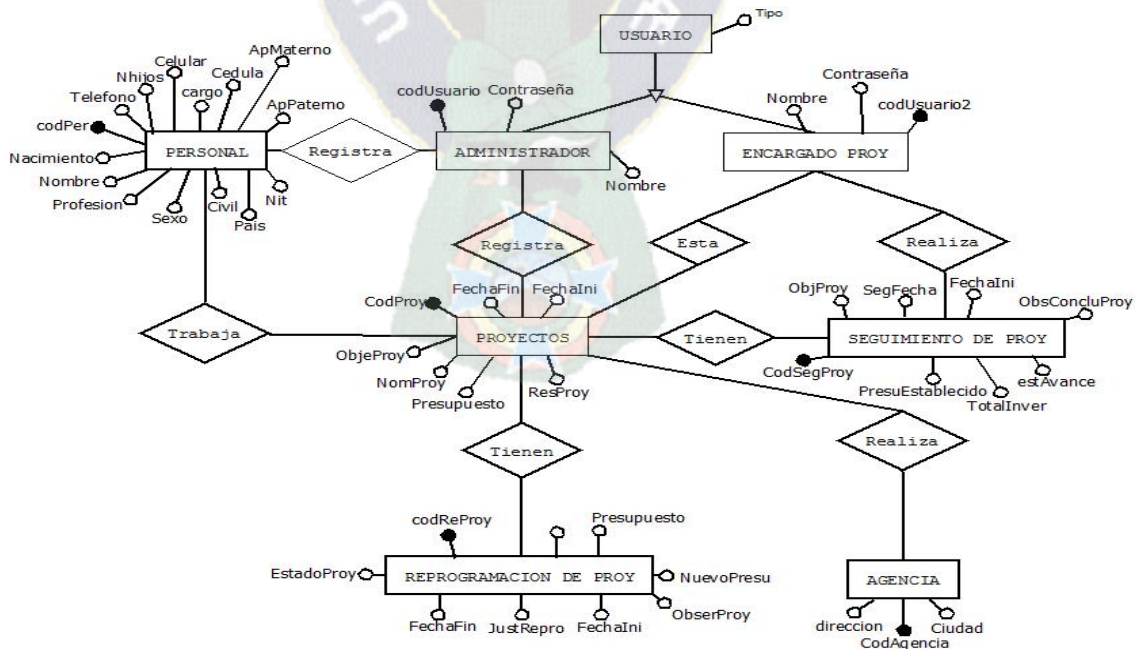


Figura 3.1 Modelo Entidad Relación (E - R)

Fuente: Elaboración propia

El modelo de estructura representa las diferentes tablas de datos y las relaciones que serán necesarias para la aplicación Web, así como se muestra en la Figura 3.2.

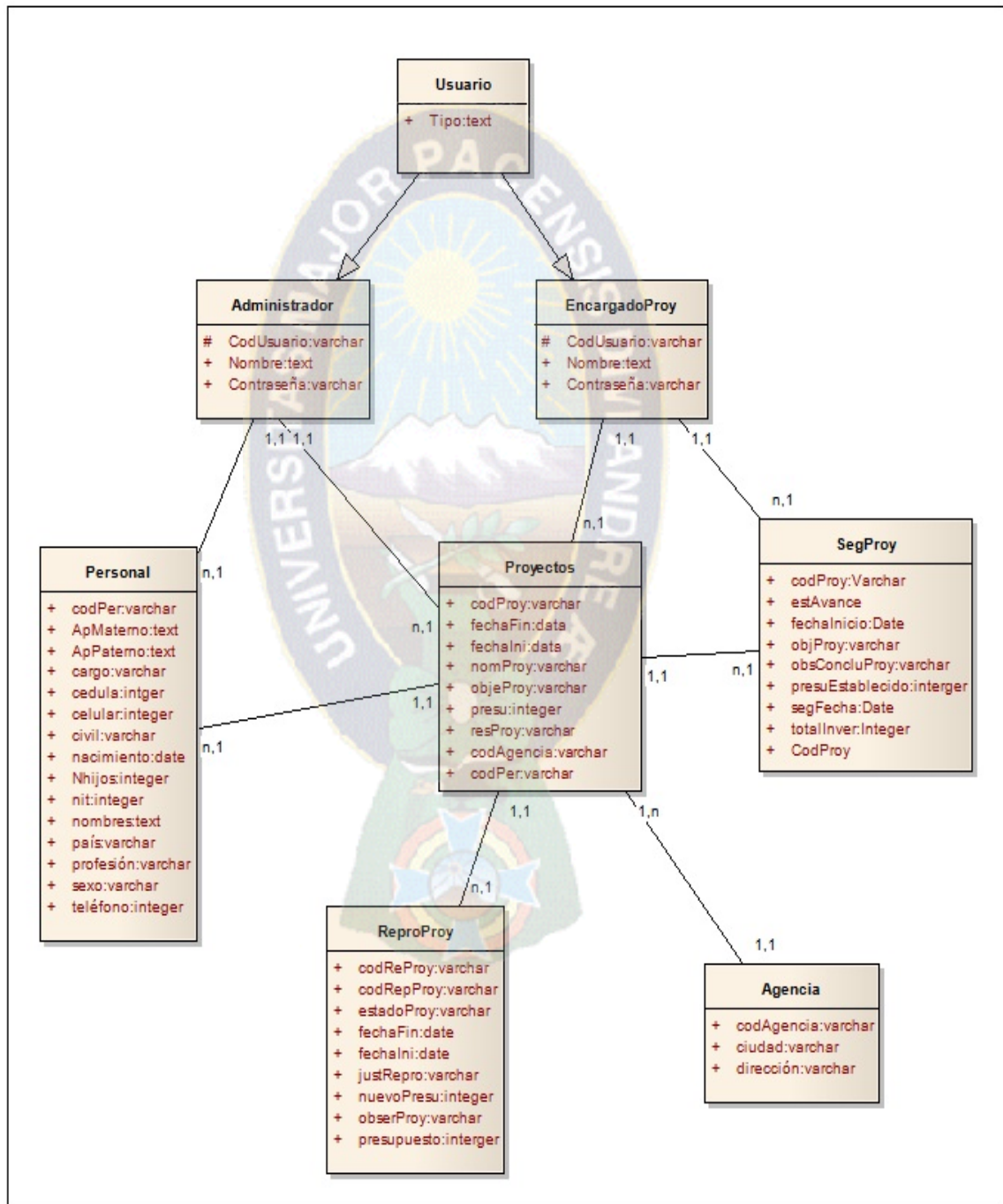


Figura 3. 2 Modelo de Estructura
Fuente: Elaboración propia

Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno *web*, el diseñador debe decidir además qué información debe ser almacenada.(SCHWABE, 1998).

Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicación web de alguna organización dada, esta sea independiente de tipo comercial, técnico, científico u otro. Incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema, proteger y hacer volcados de datos.

A continuación se muestra en la Figura 3.3, la captura de la pantalla de la base de datos que está asociado al sistema, que cuenta con todas las tablas, con cada uno de los atributos derivados de las entidades y las relaciones.

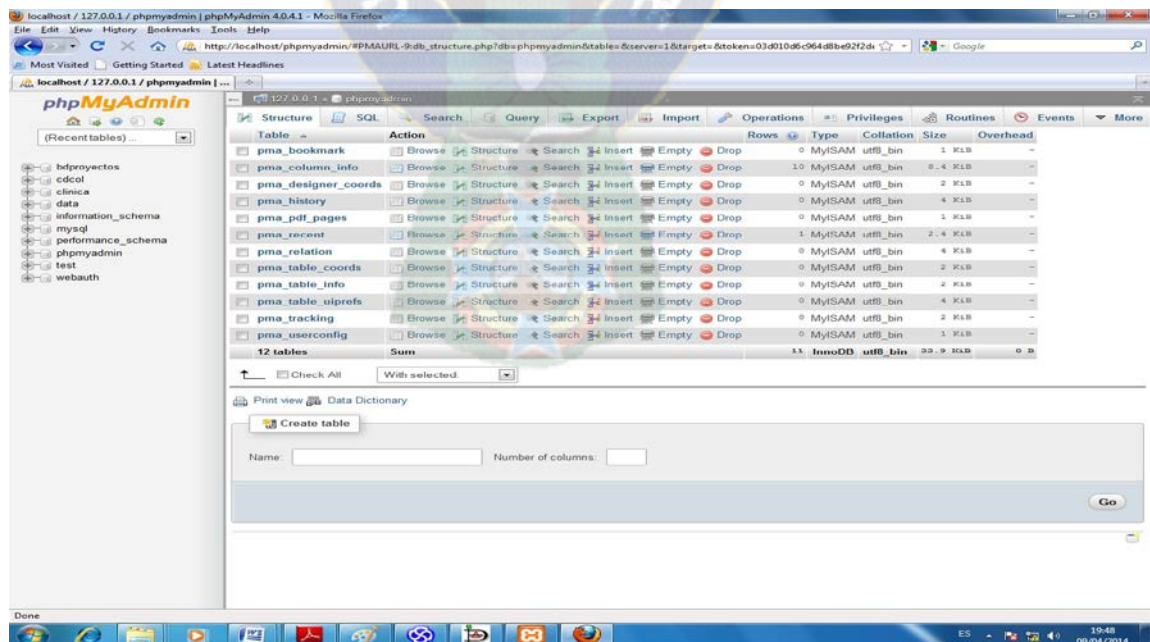


Figura 3.3 Base de Datos
Fuente: Elaboración propia

A partir del Modelo entidad relación (E - R), se pudo elaborar el modelo físico de la base de datos, que en WebML tiene el nombre de modelado de componentes.

3.3.2 Modelo de Componentes

A continuación la Figura 3.4 ilustra el Modelo de componentes que está asociado al sistema web de gestión de proyectos.



Figura 3.4 Modelo de componentes
Fuente: Elaboración propia

3.3.3 Tarjetas CRC (clase - responsabilidad - colaboración)

Para el diseño de las tarjetas CRC se necesitan identificar que se debe hacer y sobre quien se deben realizar las acciones.

A continuación se muestra el diseño de las tarjetas CRC para las principales historias de usuario:

- Historia de usuario: Registro apertura de proyectos

Apertura de Proyectos	
<ul style="list-style-type: none">• Nuevo Proyecto• Actualizar Proyecto• Eliminar Proyecto• Lista de Proyectos	<p>Administrador</p> <p>Registro de Proyectos.</p> <p>Base de datos Proyectos.</p>

- Historia de usuario: Registro de seguimiento de proyectos

Seguimiento de proyectos	
<ul style="list-style-type: none">• Nuevo seguimiento de Proyectos.• Modificar Seguimiento Proyecto• Eliminar Seguimiento de Proyectos• Lista de Seguidimientos de proyectos	<p>Administrador</p> <p>Encargado del proyecto</p> <p>Registro de Seguimiento de Proyectos.</p> <p>Base de datos Seguimiento de Proyectos.</p>

- Historia de usuario: Registro de Reprogramación de proyectos

Reprogramación de proyectos	
<ul style="list-style-type: none"> • Nueva reprogramación de Proyectos. • Modificar Reprogramación de Proyecto • Eliminar Reprogramación de Proyectos • Lista de Reprogramación de proyectos 	<p>Administrador</p> <p>Jefe de reprogramación</p> <p>Registro reprogramación de Proyectos.</p> <p>Base de datos</p> <p>Reprogramación de Proyectos.</p>

- Historia de usuario: Registro de Personal

Personal	
<ul style="list-style-type: none"> • Nuevo Personal. • Modificar Personal • Eliminar Personal • Lista de Personal 	<p>Administrador</p> <p>Jefe de recursos humanos</p> <p>Registro de Personal.</p> <p>Base de datos Personal.</p>

- Historia de usuario: Registro de Agencias

Agencias	
<ul style="list-style-type: none"> • Nuevo Agencia. • Modificar Agencia • Eliminar Agencia • Lista de Agencias 	Administrador Registro de Agencia. Base de datos de Agencia.

- Historia de usuario: Reportes

Reportes	
<ul style="list-style-type: none"> • Reporte proyectos. • Reporte seguimiento de Proyectos • Reporte Reprogramación de proyectos • Reporte Personal • Reporte Agencias 	Administrador Búsqueda y listado de Reportes. Salida del sistema

3.3.4 Modelo de Navegación

Esta parte del diseño es muy importante ya que el sistema web es la interfaz más directa del sistema hacia los usuarios de la empresa, además de que se presenta la imagen de la empresa con los clientes.

A continuación se muestra en la Figura 3.5 que corresponde al diagrama navegacional del usuario como administrador del sistema, el cual tendrá acceso a todos los módulos del sistema.

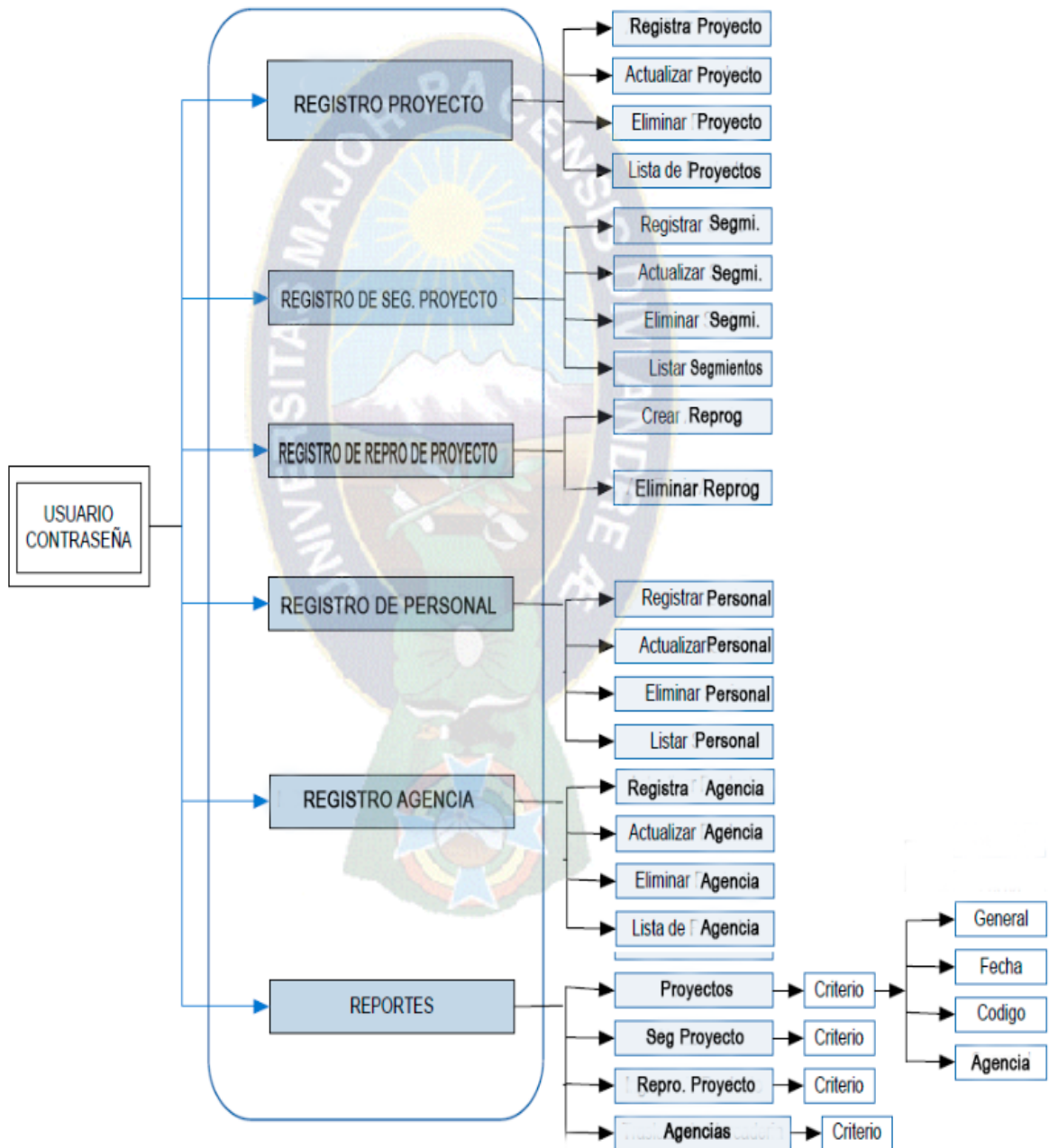


Figura 3. 5 Diagrama navegacional: Usuario Administrador
Fuente: Elaboración propia

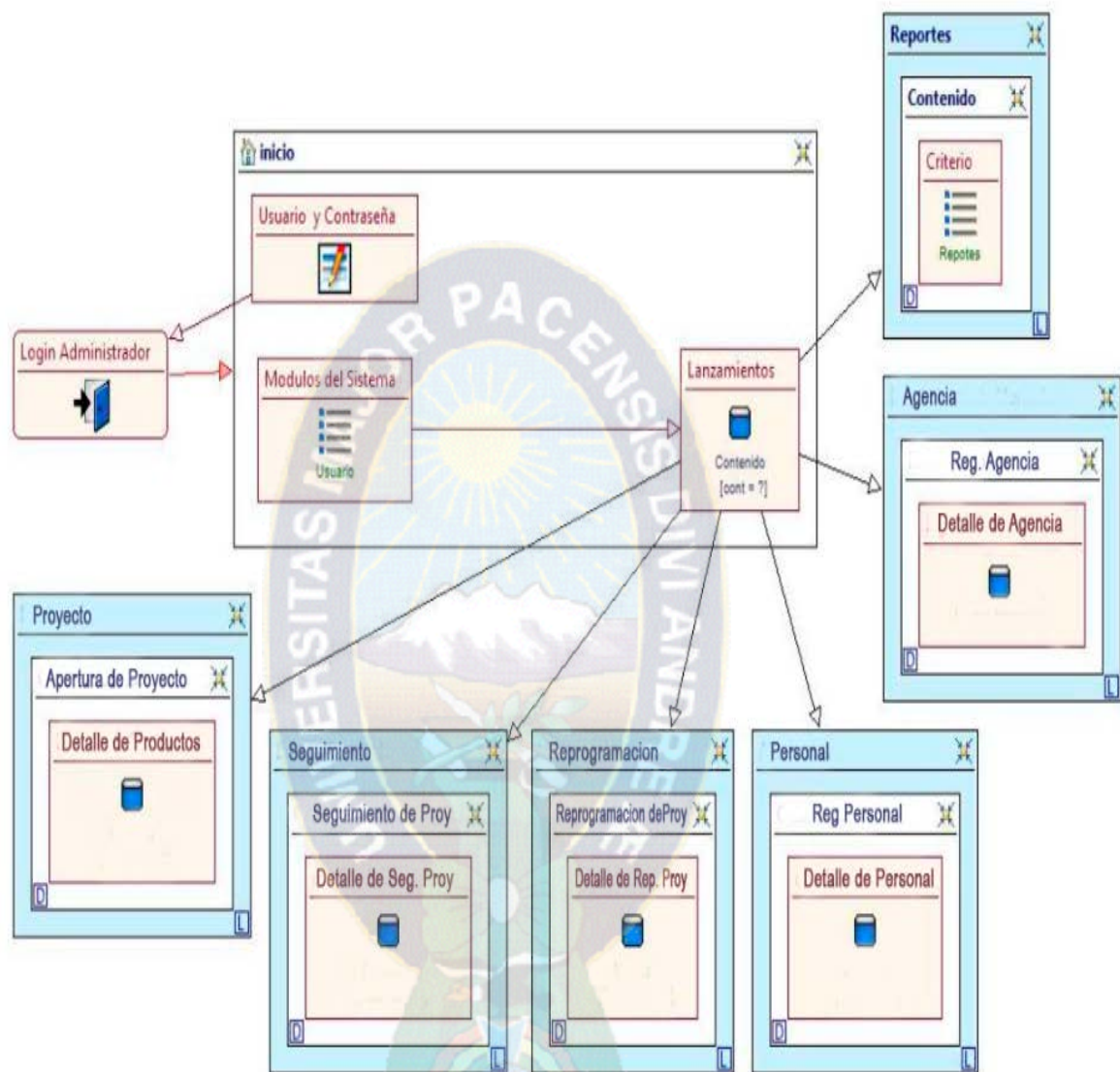


Figura 3. 6 Modelado Web: Pagina Principal
Fuente: Elaboración propia

Luego de identificar los usuarios internos que manejaran el sistema según los roles y tareas que cumplen cada uno de ellos, mostramos a continuación los escenarios más relevantes de navegación e interfaz directa para con el sistema.

Diagrama navegacional del usuario Encargado de proyecto dentro de la empresa de una determinada Agencia que cuenta con los módulos que tendrá acceso el mencionado usuario, así como podemos ver en la siguiente Figura 3.7.

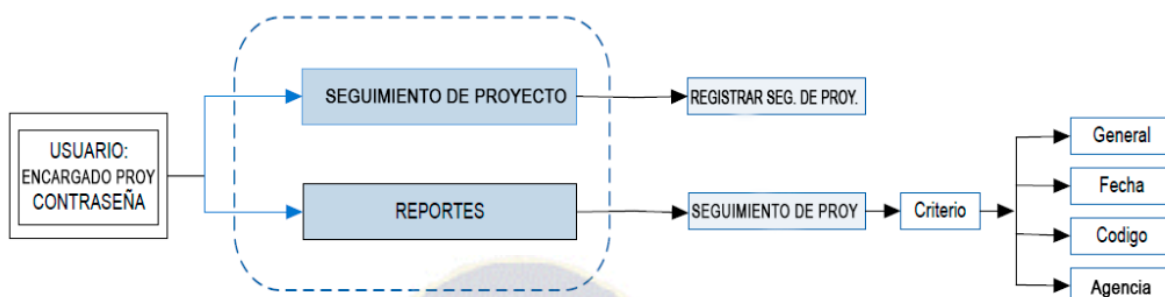


Figura 3.7 Diagrama Navegacional: Usuario Encargado de proyecto
Fuente: Elaboración propia:

La siguiente Figura 3.8 muestra el modelado de proyectos el cual muestra las operaciones que se realizan dentro del modulo.

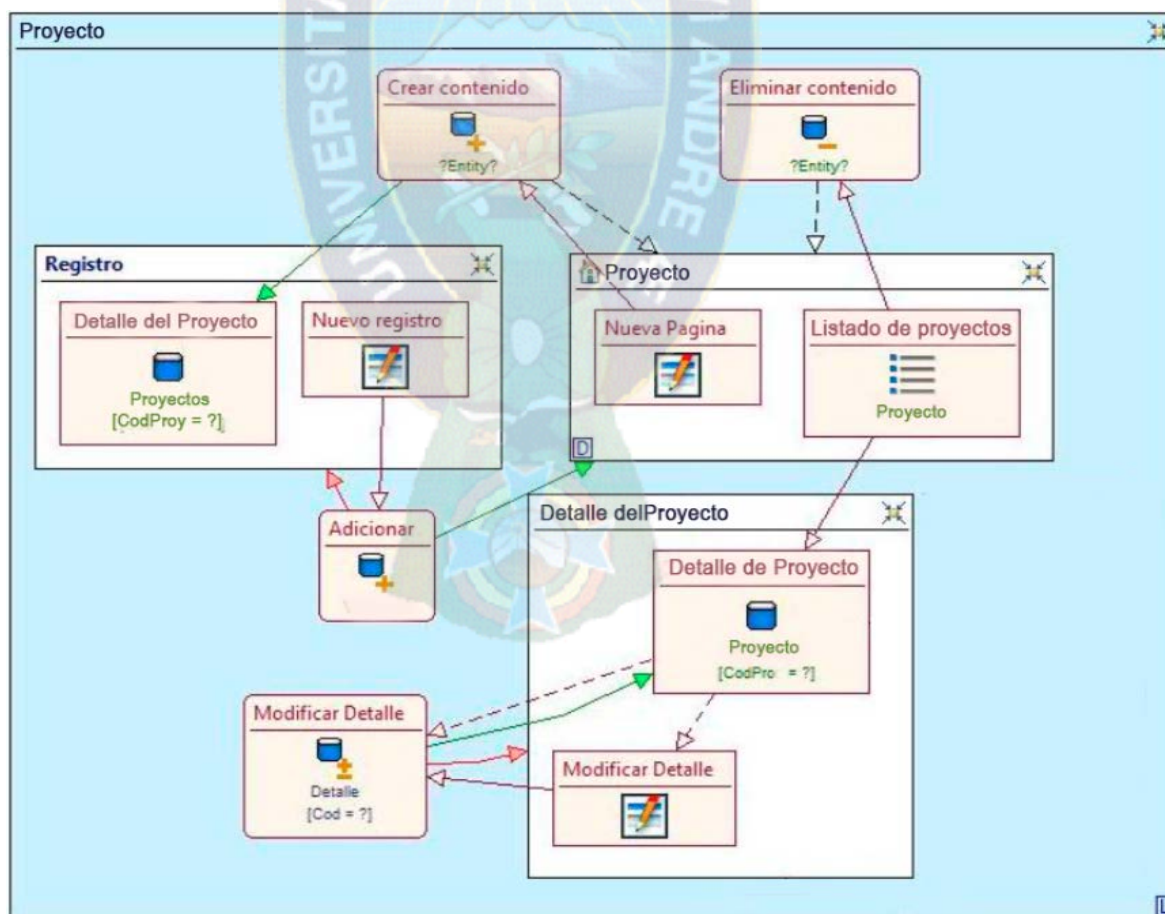


Figura 3.8 Modelado Web: Proyectos
Fuente: Elaboración propia

3.4 DESARROLLO

En esta fase de la metodología XP, la codificación de cada una de las historias de usuario se realiza junto al cliente.

Permitiendo de esta forma una retroalimentación en la comprensión de lo que el cliente quiere priorizar más en las entregas del sistema.

El desarrollo de las funcionalidades del sistema fue realizado con el lenguaje de programación PHP en una versión 5.3.0, con symfony que es uno de los frameworks PHP más populares entre los usuarios y las empresas, ya que permite que los programadores sean mucho más productivos a la vez que crean código de más calidad.

Además de utilizar como gestor de base de datos a MySQL versión 5.1.36, capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir la necesidad de cualquier tipo de organización.

A continuación se muestran pantallas de los módulos que se desarrollaron para el sistema que se realizaron través de las historias de usuario y requerimientos del cliente.

Pantalla de inicio:

La pantalla de inicio tiene lo siguiente:

Login: Donde identifican al usuario teniendo que introducir nombre y contraseña dependiendo de los tipos de usuario y permisos que tienen.

Galería de fotos: Son las obras que ha realizado la empresa.

También son fotos de la empresa.

Consta también de un menú dependiendo del tipo de usuario que sea.

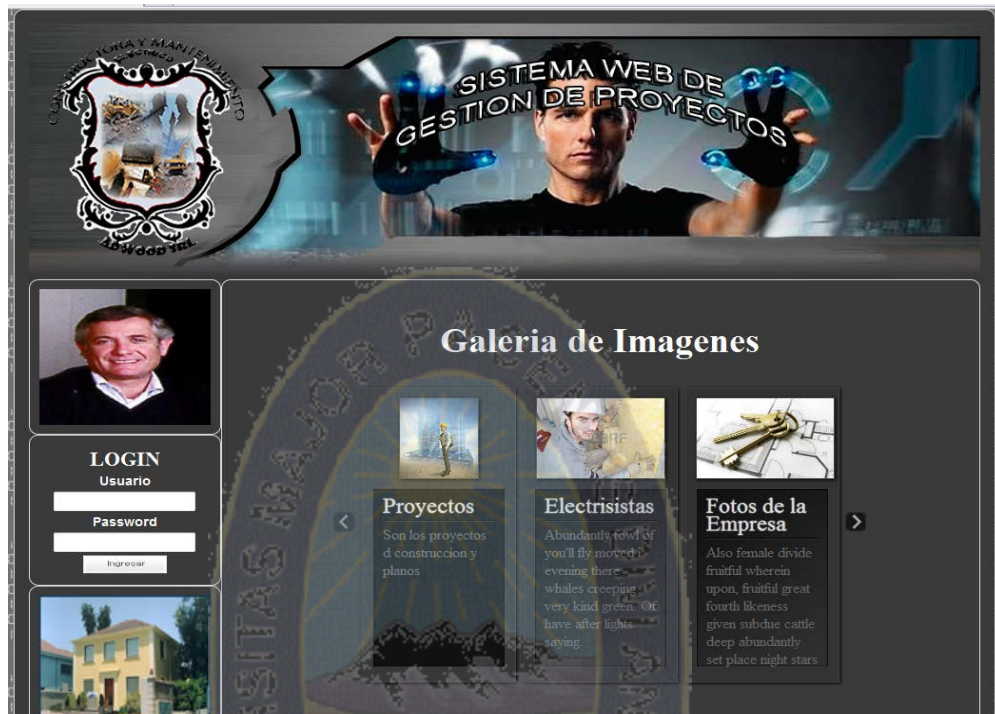


Figura 3. 9 Inicio del Sistema
Fuente: Elaboración propia

Modulo de **Apertura de Proyectos** donde se introduce los datos de los proyectos en ejecución ver Figura 3.10

Figura 3. 10 Apertura de Proyectos
Fuente: Elaboración propia

Modulo de **Seguimiento de Proyectos** donde los responsables de proyectos tienen que reportar el avance de los proyectos asignados ver Figura 3.11

SEGUIMIENTO DE PROYECTOS

Buscar

*Codigo:

*Fecha Inicio:

*Seg. hasta la Fecha:

*Estado de avance del proyecto:

*Presupuesto Establecido:

*Total Invertido Saldo:

*Observación para la Conclusión del Proyecto:

Eliminar Registrar

Figura 3. 11 Registro Seguimiento de Proyectos
Fuente: Elaboración propia

Modulo de **Reprogramación de Proyectos** es para cuando el proyecto necesita una reprogramación por diferentes factores, ver Figura 3.12.

REPROGRAMACION DE PROYECTOS

Buscar

*Codigo:

*Fecha Inicio:

*Fecha Fin:

*Justificación de la Reprogramación:

*Presupuesto:

*Nuevo Presupuesto:

*Observación del Proyecto:

Eliminar Registrar

Figura 3. 12 Registro Reprogramación de Proyectos
Fuente: Elaboración propia

El **modulo de Personal** es para registrar a todos los empleados que constituyen la empresa, ver Figura 3.13

REGISTRO DE PERSONAL

*Codigo:

*Nombres:

*Ap Paterno:

*Cedula:

Telefono:

*Pais:

Sexo: ☐ Hombre ☐ Mujer

*Fecha Nacimiento:

*Ap Materno:

*Cargo:

Celular:

*Profesion:

Nro Hijos:

Est. Civil: ☐ Soltero ☐ Casado ☐ Divorciado

*Direccion:

Eliminar Registrar

Debe Seleccionar

Figura 3. 13 Registro de Personal
Fuente: Elaboración propia

El modulo de Agencia es para registrar las nuevas agencias que la empresa requiera en un futuro, ver Figura 3.14

REGISTRO DE AGENCIAS

*Codigo:

*Nombre:

*Direccion:

*Ciudad:

Eliminar Registrar

Figura 3. 14 Registro de agencias
Fuente: Elaboración propia

3.5 PRUEBAS

En esta fase de la metodología de programación extrema XP, la fase de pruebas es una de las fases más importantes ya que nos permite verificar junto al cliente que se pudo atender los requerimientos especificados en las historias de usuario.

También sirven como retroalimentación para ver que las historias de usuario que fueron implementadas en versiones anteriores y que necesitan ser modificadas, mejoradas o simplemente descartadas.

El tipo de pruebas realizadas en esta fase son pruebas de aceptación descritas a continuación.

3.5.1 Pruebas de aceptación

Este tipo de pruebas fueron realizadas para cada entrega del software en las distintas iteraciones que se tuvo, ya que fueron definidas en el reverso de cada historia de usuario.

Las siguientes tablas muestran todas las pruebas de aceptación requeridas por el cliente en cada historia de usuario, además de la iteración en la cual fueron solucionadas correctamente.

Registro de Apertura de Proyectos: esta prueba busca encontrar todo tipo de errores que se generan durante el proceso de registro, modificación y eliminación de proyectos en el sistema.

Ya sea en el llenado del formulario de registro con datos no validos o también por la falta de datos obligatorios.

También se verifico con el usuario de la empresa solicitante con diversos requerimientos.

Prueba de aceptación	
Número: 1	Usuario: Administrador
Nombre historia : Registro y modificación de proyectos	
Descripción: <ul style="list-style-type: none"> • Llenar el formulario olvidando introducir algún dato obligatorio y verificar que se muestra en el mensaje de error. • Introducir contraseñas distintas y verificar el control de las mismas. • Introducir todos los datos correctamente e informarle que se completo su registro. 	
Test de aceptación: aceptado por el cliente en la primera iteración.	

Tabla 3. 32 Prueba de aceptación: Registro y modificación de proyectos
Fuente: Elaboración propia

Registro de Seguimiento de proyectos: esta prueba busca encontrar todo tipo de errores que se generen durante el proceso de registro, modificación y eliminación de los registros de seguimiento de proyectos y listado, además de la verificación que la información generada es almacenada correctamente.

Prueba de aceptación	
Número: 2	Usuario: Encargado del Proyecto
Nombre historia : Registro y modificación de Seguimiento de proyectos	
Descripción: <ul style="list-style-type: none"> • Mostrar mensajes para introducir los datos obligatorios en caso de no ser llenados los campos. • Mostrar mensajes de confirmación en la actualización de datos de los registros de seguimiento de proyectos. • Mostrar alertas cuando el administrador este apunto de eliminar un seguimiento de proyecto. 	
Test de aceptación: aceptado por el cliente en la primera iteración.	

Tabla 3. 33 Prueba de aceptación: Registro y modificación seguimiento proyectos
Fuente: Elaboración propia

Registro de Reprogramación de proyectos: Esta prueba busca encontrar todo tipo de errores que se generen durante el proceso de registro y eliminación de reprogramación de proyectos.

Prueba de aceptación	
Número: 3	Usuario: Administrador
Nombre historia : Registro y Eliminación de Reprogramación de proyectos	
Descripción: <ul style="list-style-type: none"> • Llenar el formulario olvidando introducir algún dato obligatorio y verificar que se muestra en el mensaje de error. • Introducir código del proyecto y registrar la reprogramación. • Introducir todos los datos correctamente e informarle que se completo su registro. 	
Test de aceptación: aceptado por el cliente en la segunda iteración.	

Tabla 3. 34 Prueba de aceptación: Registro reprogramación de proyectos
Fuente: Elaboración propia

Registro de Personal: Esta prueba busca encontrar todo tipo de errores que se generen durante la creación, actualización, eliminación y listado del personal para que se tengan datos precisos del personal que tiene la empresa.

Prueba de aceptación	
Número: 4	Usuario: Administrador
Nombre historia : Registro Personal	
Descripción: <ul style="list-style-type: none"> • Mostrar un mensaje de confirmación para registro del personal. • Mostrar mensajes para introducir los datos obligatorios en caso de no ser llenados los campos obligatorios. • Solicitar una confirmación antes de eliminar a un registro de personal. • Mostrar una lista del personal según la agencia. 	
Test de aceptación: aceptado por el cliente en la segunda iteración.	

Tabla 3. 35 Prueba de aceptación: Registro de Personal
Fuente: Elaboración propia

Registro de Agencia: Esta prueba busca encontrar todo tipo de errores que se generen durante la creación, actualización, eliminación y listado de las agencias para que se tengan datos de las agencias que constituyen la empresa.

Prueba de aceptación	
Número: 5	Usuario: Administrador
Nombre historia : Registro Agencia	
Descripción: <ul style="list-style-type: none"> • Mostrar un mensaje de confirmación para registro de la nueva agencia. • Mostrar mensajes para introducir los datos obligatorios en caso de no ser llenados los campos obligatorios. • Solicitar una confirmación antes de eliminar una agencia. • Mostrar una la lista de las agencias que tiene la empresa. 	
Test de aceptación: aceptado por el cliente en la tercera iteración.	

Tabla 3. 36 Prueba de aceptación: Registro de Agencia
Fuente: Elaboración propia

Reportes: Esta prueba busca encontrar todo tipo de errores que se generaron durante todos procesos realizados en el sistema mediante el listado de reportes.

Prueba de aceptación	
Número: 6	Usuario: Administrador
Nombre historia : Reportes	
Descripción: <ul style="list-style-type: none"> • Mostrar alertas de mensajes en caso de no realizar la búsqueda adecuadamente con datos precisos. • Verificar que los reportes generen los datos requeridos. 	
Test de aceptación: aceptado por el cliente en la tercera iteración.	

Tabla 3. 37 Prueba de aceptación: Reportes
Fuente: Elaboración propia

3.5.2 Pruebas de Stress

Una prueba de stress es aquella que forsa al sistema al máximo punto para poder medir sus capacidades y las condiciones en las cuales trabaja realizando una cantidad definida de peticiones y procesos.

La prueba de stress para este sistema se realizo utilizando el Software JMeter, configurándolo adecuadamente utilizando el árbol de configuración de la figura 3.15, para que se llevaran a cabo 6 peticiones por usuario las cuales estaban dadas en el siguiente orden:

- Logeo en el sistema, que consta de ingresar el Usuario y el Password y desplegar la pagina de bienvenida.
- Desplegar la lista de los proyectos.
- Ver la información detallada de cada proyecto.
- Desplegar la lista de personal de la empresa.
- Ver la información detallada del personal de la empresa.
- Salir del sistema.



Figura 3. 15 Configuración del JMeter
Fuente: Elaboración propia

Entre cada una de las peticiones se dejaba un tiempo de 2 a 3 segundos para no saturar al sistema de manera simultánea, y para poder dar un poco de realismo a la prueba.

Los usuarios se conectan al mismo tiempo, cada uno con una sesión diferente y llevan a cabo estas 6 actividades, para lo cual se registraron los tiempos de respuesta y se tomaron algunos datos estadísticos que proporciona el JMeter en el Informe agregado.

Para encontrar el número correcto de usuarios de varias pruebas incrementales, es decir se comenzó probando para un número de usuarios reducidos, y se fue aumentando para medir el desempeño del sistema.

Hasta que se encontró que el caso óptimo fue con 325 usuarios, y el caso crítico con 350. A pesar de que la diferencia que existe entre estos dos números es muy pequeña, para 350 usuarios se comienzan a reportar errores mínimos pero que si afectan a algunos de los usuarios que se encuentran utilizando el sistema.

El informe agregado contiene varias columnas y renglones, los renglones representan cada uno de los 6 *request* o actividades que se realizaron y las columnas representan ciertas medidas dentro de la prueba:

- URL: es la actividad que se desempeña, el *request* o la petición.
- # Muestras: es la cantidad de veces que se realizó la actividad (una vez por cada usuario).
- Media: el promedio o media aritmética del tiempo en milisegundos.
- Mediana: del tiempo en milisegundos.
- Min: tiempo mínimo de todos los *requests* de ese tipo.
- Max: tiempo máximo de todos los *requests* de ese tipo.
- Porcentaje de error: en el cual se muestra el porcentaje de los *requests* fallidos.
- Rendimiento: esta medido en *request*/segundo.
- KB/Sec: medida de velocidad de kilobytes/seg.

En la Tabla 3.38 la media total fue de 2788 ms, esto quiere decir que el sistema en promedio se tardo en responder 2.7 segundos, lo cual es tiempo bastante bueno tomando en cuenta que son 350 usuarios conectados al mismo tiempo.

URL	#Muestras	Media	Mediana	Min	Max	%Error	Rendimiento	KB/Sec
Login	350	13969	13735	312	27500	9.71%	11.2/sec	148.80
Lista de Proyectos	350	2231	1141	0	8422	0.00%	21.6/sec	105.93
Proyecto	350	9	15	0	47	0.00%	19.6/sec	47.53
Lista de Personal	350	514	16	0	5688	0.00%	18.4/sec	47.59
Personal	350	8	0	0	47	0.00%	20.9/sec	46.96
Salir	350	0	0	0	16	0.00%	21.0/sec	60.11
Total	2100	2788	16	0	27500	1.62%	13.3/sec	65.48

Tabla 3. 38 Informe Agregado para la prueba de 350 usuarios
Fuente: Elaboración Propia

En la tabla 3.38 también se puede apreciar que la actividad Login existe un porcentaje de error del 9.71%, este representa 33 usuarios de los 350 con los que se realizo la prueba. Estos 33 usuarios obtuvieron una página de error al intentarse conectar al sistema. Esta es una cantidad importante de usuarios por lo que se tuvo que considerar como caso critico, la media total fue de 2788 ms, esto quiere decir que el sistema en promedio se tardo en responder 2.7 segundos. Este tiempo es bastante bueno tomando en cuenta que son 350 usuarios conectados al mismo tiempo.

Para el caso óptimo se utilizaron 325 usuarios y como se muestra en la tabla 3.39 no hay porcentaje de error y es un mejor rendimiento del sistema. Además se pudo apreciar que el porcentaje de error en toda las peticiones es de 0.00%, esto indica que no fue desplegada ninguna pagina de error, ya que todas las peticiones fueron respondidas de manera adecuada y correcta. Si se toma la media de la tabla 3.39, que es de 2788 ms, y se compara con la de la tabla 3.40, que es de 2478, se puede apreciar que existe una mínima diferencia de 0.3 segundos, lo cual es un poco menos de tiempo. Por lo que se

considera como un tiempo de respuesta muy pequeño, lo que clasifica al sistema como rápido.

URL	#Muestras	Media	Mediana	Min	Max	%Error	Rendimiento	KB/Sec
Login	325	12913	12640	93	24906	0.00%	10.9/sec	159.10
Lista de Proyectos	325	1839	1063	0	6140	0.00%	16.2/sec	62.75
Proyecto	325	8	0	0	46	0.00%	18.3/sec	40.17
Lista de Personal	325	97	16	0	2563	0.00%	19.0/sec	58.83
Personal	325	9	0	0	63	0.00%	16.9/sec	44.21
Salir	325	1	0	0	16	0.00%	21.2/sec	60.53
Total	1950	2478	16	0	24906	0.00%	16.3/sec	79.51

Tabla 3. 39 Informe Agregado para la prueba de 325 usuarios
Fuente: Elaboración Propia

Como se puede apreciar en los informes agregados los tiempos de respuesta son pequeños, ese tiempo también ya incluye el acceso a la base de datos por lo que se considera que el sistema se tarda muy poco tiempo en realizar el acceso a la base de datos para recuperar la información necesaria.

Esta prueba de *stress* sirvió para poder concluir que el sistema soportara la cantidad de 330 usuarios, ya que la prueba de stress soporto con esa cantidad, teniendo en cuenta que no todos harán uso de este servicio al mismo tiempo.

CAPÍTULO 4

CALIDAD Y SEGURIDAD

4.1 QUE ES CALIDAD DEL SOFTWARE

Definición de calidad de software:

Consecuencia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente. [Pressman, 2005]

Además de las anteriores definiciones se puede decir que la calidad de software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. Donde la calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad integral. [Pressman, 2005]

De acuerdo al estándar ISO 9126 identifica cinco atributos clave de calidad y que se desarrollan a continuación.

4.1.1 Funcionalidad

Las métricas orientadas a la función son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcular las líneas de código, las métricas orientadas a la función se centran en la funcionalidad o utilidad del software, midiendo la aplicación desde la perspectiva del usuario, dejando de lado los detalles de codificación. Para calcular la funcionalidad del sistema mediante la métrica Punto Función, se utiliza la siguiente ecuación:

$$PF = \text{Calidad Total} * [0.65 + 0.001 * \sum F_i]$$

Donde:

PF = Medida de Funcionalidad

Calidad Total = es la suma de todas las entradas de PF obtenidas en la Tabla 4.1

0.01 = Factor de Conversión, es decir un error del 1%

0.65 = Valor mínimo de ajuste de complejidad.

F_i = Son valores de ajuste de complejidad.

Para hacer uso de la ecuación anterior se empieza por determinar los siguientes parámetros de medición.

- **Número de entradas de usuario**, presenta las entradas de control de usuario que proporciona diferentes datos a la aplicación.
- **Número de salidas de usuario**, estas se refieren a informes, mensajes de error, es decir salidas que proporcionen al usuario información orientada a la aplicación.
- **Número de peticiones o consultas de usuario**, es una combinación de ingreso de dato con una devolución inmediata de información por el usuario en pantalla.
- **Numero de archivos, grupo de datos lógicos**, presenta a cada archivo parte de la base de datos o a un archivo independiente.
- **Número de interfaces externas**, se cuenta todas las interfaces legibles por el ordenador que son solicitadas para transmitir información a otro sistema.

De acuerdo a lo mencionado es que se tiene los resultados de la Tabla 4.1

Donde muestra los datos obtenidos de las entradas, salidas y peticiones.

Parámetros de medición	Cantidad	Factor de ponderación			
		Simple	Medio	Complejo	Total
Número de entradas de usuario	21	3	4	6	86
Número de salidas de usuario	34	4	5	7	170
Número de peticiones de usuario	10	3	4	6	40
Número de archivos o Tablas	14	7	10	15	140
Número de interfaces externas	0	5	7	10	0
Cantidad Total					436

Tabla 4. 1 Cálculo de los Parámetros de Medición
Fuente: Elaboración Propia

Se calcula los valores de ajuste de complejidad (F_i) basado en las respuestas a las 14 preguntas que aparecen en la tabla 4.3 evaluando las respuestas en una escala de 0 a 5, tomando en cuenta la tabla 4.2

Grado	Ponderación
Sin influencia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Tabla 4. 2 Escala de evaluación para los valores del Ajuste de Complejidad
Fuente: 8

Nº	Pregunta	Factor
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?	5
2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones de procesos distribuidos?	1
4	¿Es crítico el rendimiento?	2
5	¿Sera ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	3
6	¿Requiere el sistema de datos interactivos?	3
7	¿Requiere el sistema de entrada de datos interactiva sobre múltiples ventanas?	3
8	¿Se actualizan los archivos maestros de manera interactiva?	2
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?	1
10	¿Es complejo el procesamiento interno?	2
11	¿Sea diseñado el código para ser reutilizable?	5
12	¿Están incluidas en el diseño la conversión y la instalación?	2
13	¿Se ha diseñado el sistema para soportar múltiples plataformas?	5
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
Total ($\sum F_i$)		44

Tabla 4. 3 Valor de Ajuste de Complejidad
Fuente: Elaboración Propia

Con el valor de ajuste de complejidad (F_i) ya calculado procedemos a calcular el valor de Punto Función (PF) reemplazando los valores en la Ecuación de Punto Función.

$$PF = 436 \times [0.65 + 0.01 \times 44]$$

$$PF = 475.24$$

Entonces se puede concluir que la funcionalidad del sistema es óptima con el resultado obtenido, ya que es mayor a 300, esto de acuerdo con la tabla 4.4

Escala	Observación
PF > 300	Óptimo
200 < PF < 300	Bueno
100 < PF < 200	Suficiente
PF < 100	Deficiente

Tabla 4. 4 Escala de Punto Función
Fuente: Elaboración Propia

Además se halla el punto máximo para comparar los valores de funcionalidad del sistema, entonces se tiene:

$$PF_{\max} = 475.24 \times [0.65 + 0.01 \times 44] = 475.24 \times [1.09] = 518.01$$

Así, la funcionalidad real es:

$$\text{FUNCIONALIDAD} = \frac{PF}{PF_{\max}} = \frac{475.24}{518.01} = 0.9174$$

Por lo tanto, la funcionalidad del Sistema de Web de Gestión de Proyectos para la empresa Constructora y Mantenimiento Eléctrico ATWOOD SRL se estima que es del **91%**. En otras palabras, del total de requerimientos funcionales solicitados por el usuario, un **91%**, satisfacen a cabalidad las necesidades del usuario.

4.1.2 Confiabilidad

La confiabilidad en términos de software se refiere a la probabilidad de operación libre de fallos de un programa de computación en un entorno determinado y un tiempo específico.

Para realizar el cálculo de la confiabilidad del sistema, se toma en cuenta $t_0=0$ como el instante en que se empieza a trabajar con el sistema, observando detalladamente el trabajo hasta que se produce una falla en el instante t , el tiempo que se trabaja sin que exista una falla o cuando esta se produce, la relación es la siguiente:

Probabilidad de hallar fallas: $P(T \leq t) = F(t)$

Probabilidad de que no ocurran fallas $P(T > t) = 1 - F(t)$

Se calcula un periodo t , periodo en el que se aprueba el sistema hasta que se produzca una falla, para tal efecto se utiliza la función exponencial:

$$F(t) = Fc * e^{(-\frac{\lambda}{10} \times t)}$$

Donde:

$Fc=0.9174$ Es la funcionalidad del Sistema ya calculado.

$\lambda=0.003(3\%)$: La probabilidad de error que puede tener el sistema.

$t= 12$ meses. Es el tiempo durante los próximos 12 meses:

$$F(t) = 0.9174 \times \left[e^{(-\frac{3}{10} \times 12)} \right]$$

$$F(t) = 0.9174 \times [0.02732]$$

$$F(t) = 0.0250$$

La probabilidad de que sucedan fallas en el sistema es de un 2.5% durante los próximos 12 meses.

$$P(T > t) = 1 - F(t)$$

$$P(T > t) = 1 - 0.0250$$

$$\text{CONFIABILIDAD} = P(T > t) = 0.975$$

Se estima que el sistema tiene una confiabilidad del 97% durante un periodo de 12 meses de procesamiento. Lo que significa, si el sistema fuese ejecutado 100 veces durante el periodo de 12 meses, es probable que operaría correctamente 97 veces.

4.1.3 Mantenibilidad

La mantenibilidad de un software se asocia a cambios que van de la mano correcciones de errores, cambios para adaptarse a las necesidades cambiantes del cliente, mejorar para adaptarse a el hardware que constantemente evoluciona.

El índice de madurez de software proporciona una indicación de la estabilidad de un producto de software y nos sirve de métrica de la calidad del mantenimiento del sistema.

Calculamos el índice de madurez de software con la siguiente relación:

$$IMS = \frac{[M_t - (F_c + F_a + F_e)]}{M_t}$$

Donde:

M_t = Número de módulos en la versión actual.

F_c = Número de módulos en la versión actual que se han cambiado.

F_a = Número de módulos en la versión actual que se han añadido.

F_e = Número de módulos en la versión que se han eliminado en la versión actual.

Se tiene que el índice de madurez de software es:

$$IMS = \frac{[11 - (0 + 1 + 0)]}{11}$$

$$IMS = 0.91$$

MANTENIBILIDAD = 91%

Por lo tanto, el sistema es estable en un **91%**, lo que significa que la facilidad de mantenimiento con la que se puede corregir el software en caso de encontrar un error o se puede adaptar si su entorno cambia o mejor de acuerdo a los requerimientos del usuario.

4.1.4 Portabilidad

Según Roger Pressman la portabilidad es el esfuerzo necesario para transmitir un programa de un entorno de sistema de hardware y/o software a otro. [8]

1-(Número de días para portar Sistema / Número de días para implementar el sistema)
(Formula para el cálculo de la portabilidad [8])

$$\text{Portabilidad} = (1 - (0.5 \text{ dias} / 5 \text{ dias})) \times 100\%$$

$$\text{Portabilidad} = (1 - (0.1)) \times 100\% = 0.9 \times 100 = 90\%$$

PORTABILIDAD = 90% (Significa que en un 90% es Portable)

Por lo tanto el software tiene una portabilidad de un **90%** en sus entornos tanto en Hardware y Software.

4.1.5 Usabilidad

La medición de la facilidad de uso se puede entender como la facilidad que el usuario tiene para conocer el sistema, tanto como para comprenderlo, aprenderlo y operarlo. A continuación se presenta en la tabla 4.6 los resultados obtenidos de la encuesta realizada a cinco usuarios.

Cuestionario:

1. ¿Considera que el sistema maneja la información según sus requerimientos?
2. ¿Usted cree que la información que le muestra el sistema cumple con sus expectativas?
3. ¿Cómo considera el ingreso de datos al sistema?
4. ¿Cómo considera los reportes que elabora el sistema?
5. ¿Considera que el sistema facilita su trabajo?

Ponderación	
1	Pésimo
2	Malo
3	Regular
4	Bueno
5	Muy Bueno

Tabla 4. 5 Escala Ponderación para responder a la encuesta
Fuente: Elaboración Propia

	Respuesta					Total	Promedio
	Usuario1	Usuario2	Usuario3	Usuario4	Usuario5		
Pregunta1	4	4	4	4	4	20	4
Pregunta2	4	4	4	4	4	20	4
Pregunta3	4	4	4	5	5	22	4.4
Pregunta4	4	4	5	5	5	23	4.6
Pregunta5	5	4	5	5	5	24	4.8
				Promedio Total			21.8

Tabla 4. 6 Resultados de la encuesta
Fuente: Elaboración Propia

Partiendo de los resultados obtenidos con el cuestionario, calculamos la facilidad de uso con la ayuda de la siguiente fórmula:

$$FU = \frac{\left[\frac{\sum x_t}{n} \times 100 \right]}{5}$$

Donde:

n = Número de cuestionarios.

$\sum x_t$ = Total de ponderación de las preguntas.

Entonces, reemplazando los valores en la ecuación:

$$FU = \frac{\left[\frac{21.8}{5} \times 100 \right]}{5}$$

$$\text{USABILIDAD} = FU = 87.2$$

Por lo tanto la facilidad de uso la usabilidad es de **87.2%** se puede entender como la facilidad que el usuario tiene para entender y conocer al sistema.

4.1.6 Resultado Final

En base a los resultados obtenidos anteriormente, se puede establecer la calidad del sistema.

Resultados	
Funcionalidad	0.91
Confiabilidad	0.97
Mantenibilidad	0.91
Portabilidad	0.90
Usabilidad	0.87

Donde:

F=Funcionalidad, C=Confiabilidad, M=Mantenibilidad, P=Portabilidad, U=Usabilidad.

Así:

$$\text{Calidad Global} = \frac{F + C + M + P + U}{5}$$

$$\text{Calidad Global} = \frac{0.91 + 0.97 + 0.91 + 0.90 + 0.87}{5}$$

$$\text{CALIDAD GLOBAL} = 0.912$$

Por lo tanto, con el resultado obtenido después de aplicar los atributos de la norma ISO 9126 se puede afirmar que el Sistema Web de Gestión de Proyectos tiene una calidad total del **91%**.

4.2 SEGURIDAD

4.2.1 Seguridad del software

Para dar seguridad al software en sí, se utilizó un control de acceso a usuarios, utilizando el método de encriptación MD5 para el manejo de las contraseñas de los usuarios.

La encriptación consiste en transformar un valor legible en uno ilegible mediante un algoritmo lo cual nos ayuda a que cierta información sea inútil para aquellas personas que por alguna razón tengan acceso al sistema o a la base de datos, pero que el administrador del sistema esta información tenga sentido ya que seguramente contara con otro algoritmo que descifre este dato encriptado.

El algoritmo MD5 es una función que encripta una cadena. En nuestro caso acepta la cadena del campo de entrada contraseña, y devuelve un número de 128 bits, para luego poder compararla con la contraseña almacenada en la base de datos en la que también se

utilizó el mismo método para su almacenamiento. Este método es de un solo sentido, es decir, una vez encriptado no se puede descryptar, en este caso nos sirve para comparar el dato ingresado con el dato almacenado.

Además podemos mencionar que teniendo de todos los usuarios su nombre de usuario y contraseña se puede restringir el ingreso a algunas páginas de acuerdo a su nivel de acceso con el manejo de sesiones en PHP, si un usuario no está autenticado no podrá ingresar la administración del sistema ni a los reportes, solo al sitio web público.

En el Sistema considera a 3 tipos de usuarios con distinto nivel de acceso, los cuales describimos a continuación:

Administrador: Acceso a todos los módulos del sistema.

Responsables de Proyecto: Acceso al modulo de seguimiento de proyectos y sus reportes.

Cliente: es un usuario cualquiera con acceso al sitio web público de la empresa.

Otro aspecto de seguridad es que no existe un link directo de la administración sistema por la página web, impidiendo que personas ajenas a la fundación intenten ingresar.

4.2.2 Seguridad de la base de datos

La seguridad que se tiene a nivel de base de datos la da el gestor de base de datos a utilizar, en este caso XAMPP el cual nos ayuda a gestionar la base de datos de nuestro proyecto.

XAMPP consiste principalmente en Servidor Apache, base de datos MySQL, e intérpretes de programas escritos en los lenguajes de programación PHP y Perl, ofrece varios niveles de seguridad que el administrador de base de datos puede realizar.

Es así que para el proyecto se hace uso de uno de los algoritmos más conocidos como es el md5 que está implementando en el lenguaje PHP y en el gestor de base de datos MySql, y de esta manera su uso nos resulta sencillo de usar y aplicar

La información más sensible a ser utilizada por intrusos son datos como números de tarjetas de crédito, números de seguro social y, como en nuestro caso, contraseñas de acceso al sistema

4.2.3 Seguridad del hardware

Para la implementación de este sistema se hará uso de un servidor mail para la empresa, el cual tendrá la respectiva seguridad de:

- Aislamiento, para restringir el acceso a personas no autorizadas.
- Administración, existe personal autorizado.
- Ambiente, adecuado para su resguardo.

Por tanto se cuentan con todos los mecanismos de seguridad básicos para poder implementar este sistema, que podrá ser accedido a nivel local dentro la empresa como también vía web.

4.3 ANÁLISIS DE COSTO BENEFICIO

4.3.1 Análisis Utilizando COCOMO II

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.

- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluarán el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

COCOMO II está compuesto por tres modelos denominados:

Composición de Aplicación, Diseño Temprano y Post-Arquitectura

Éstos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software. Esta diversidad podría representarse con el siguiente esquema. Figura 4.1.

Aplicaciones desarrolladas por usuarios finales		
Generador de Aplicaciones	Aplicaciones con Componentes	Sistemas integrados
Infraestructura		

Figura 4. 1 Distribución del Mercado de Software Actual y Futuro.

Fuente: [Boehm 1995/1]

- **Aplicaciones desarrolladas por Usuarios Finales:** En este sector se encuentran las aplicaciones de procesamiento de información generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como planillas de cálculo, sistemas de consultas, etc. Estas aplicaciones surgen debido al uso masivo de estas herramientas, conjuntamente con la presión actual para obtener soluciones rápidas y flexibles.
- **Generadores de Aplicaciones:** En este sector operan firmas como Lotus, Microsoft, Novell, Borland con el objetivo de crear módulos pre-empaquetados que serán usados por usuarios finales y programadores.
- **Aplicaciones con Componentes:** Sector en el que se encuentran aquellas aplicaciones que son específicas para ser resueltas por soluciones pre-empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables. Componentes típicas son constructores de

interfaces gráficas, administradores de bases de datos, buscadores inteligentes de datos, componentes de dominio-específico (medicina, finanzas, procesos industriales, etc.). Estas aplicaciones son generadas por un equipo reducido de personas, en pocas semanas o meses.

- **Sistemas Integrados:** Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Porciones de estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones. Entre las empresas que desarrollan software representativo de este sector, se encuentran grandes firmas que desarrollan software de telecomunicaciones, sistemas de información corporativos, sistemas de control de fabricación, etc.
- **Infraestructura:** Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos, etc. Incrementalmente este sector direccionará sus soluciones, hacia problemas genéricos de procesamiento distribuido y procesamiento de transacciones, a soluciones middleware. Firmas representativas son Microsoft, Oracle, SyBase, Novell y NeXT.

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

Se puede afirmar que para las aplicaciones desarrolladas por usuarios finales no se justifica la utilización de un modelo de estimación de costos. Estas aplicaciones normalmente se construyen en poco tiempo, por lo tanto requieren solamente una estimación basada en actividades.

El modelo Composición de Aplicación, es el modelo de estimación utilizado en los proyectos de software que se construyen a partir de componentes pre-empaquetadas. En este caso, se emplean Puntos Objeto para estimar el tamaño del software, lo cual está

acorde al nivel de información que generalmente se tiene en la etapa de planificación, y el nivel de precisión requerido en la estimación de proyectos de esta naturaleza.

Para los demás sectores del mercado se aplica un modelo mixto, combinación de los tres modelos. El modelo Composición de Aplicación se emplea en desarrollos de software durante la etapa de prototipación.

El modelo Diseño Temprano se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto.

En estas etapas se tiene poca información, lo que concuerda con el uso de Puntos Función, para estimar tamaño y el uso de un número reducido de factores de costo.

El modelo Post-Arquitectura se aplica en la etapa de desarrollo propiamente dicho, después que se define la arquitectura del sistema, y en la etapa de mantenimiento. Este modelo utiliza:

- Puntos Función y/o Líneas de Código Fuente para estimar tamaño, con modificadores que contemplan el rehúso, con y sin traducción automática, y el "desperdicio" (breakage)
- Un conjunto de 17 atributos, denominados factores de costo, que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, etc., que tienen injerencia en los costos.
- Cinco factores que determinan un exponente, que incorpora al modelo el concepto de deseconomía y economía de escala. Estos factores reemplazan los modos Orgánico, Semiacoplado y Empotrado del modelo COCOMO II

4.3.2 Análisis del Costo

A través del COCOMO II se determina el costo del software desarrollado, tomando en cuenta lo siguiente, se tiene como punto función.

$$PF = 544.7 \ 475.24$$

Para la conversión de los puntos de función a KLDC. Se convierte los puntos función a miles de líneas de código, teniendo en cuenta lo siguiente:

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2,5	128
Ansi Basic	5,0	64
Java	6,0	53
PL/I	4,0	80
Ansi Cocol 74	3,0	107
Visual Basic	7,0	46
ASP	9,0	36
PHP	11,0	29
Visual C++	9,5	34

Tabla 4. 7 Conversión de puntos de Función
Fuente: [Elaboración Propia]

Calculando las líneas de código LCD, tenemos:

$$LDC = PF \times \text{Factor LDC/PF}$$

$$LDC = 475.24 \times 29$$

$$LDC = 13781.96$$

El total de código es el 13781.96 donde se estima que un 40 % del código es reutilizable.

Calculando el número de líneas distribuidas en el sistema KLCD se tiene:

$$KLDC = (\text{Total LCD} - \text{LCD reutilizable}) / 1000$$

$$KLDC = (13781.96 - 5512.78) / 1000$$

$$KLDC = 8,27$$

Por tanto existen 8 líneas de código distribuidas para el proyecto.

Se aplican las formulas básicas de esfuerzo, tiempo, tiempo calendario y personal requerido en las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E = a_b (KLDC)^{b_b}$$

$$D = c_b (E)^{d_b}$$

Donde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo en meses cronológicos.

KLDC: Número estimado de líneas de código distribuidas (en miles).

PROYECTO DE SOFTWARE	a _b	b _b	c _b	d _b
Orgánico	2.4	1.05	2.5	0.38
Semi – acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Tabla 4. 8 Relación de valores en el modelo COCOMO
Fuente: [PRESSMAN, 2005]

Reemplaza los datos en las ecuaciones se tiene:

$$E = 2.4 \times (8.27)^{1.05}$$

$$E = 22,05 \text{ programadores / mes}$$

$$D = 2.5 \times (22.05)^{0.38}$$

$$D = 8 \text{ meses}$$

El personal requerido, en este caso el número de programadores se obtiene con la siguiente fórmula:

$$\text{Numero de programadores} = E / D$$

$$\text{Numero de programadores} = 22.05 / 8$$

$$\text{Numero de programadores} = 2.756 = 3 \text{ programadores}$$

El salario de un programador puede oscilar entre los 2000 y 4000 Bs., cifra que es tomada en cuenta para la estimación siguiente:

Costo del software desarrollado = Numero de programadores x salario de un programador.

$$\text{Costo del software desarrollado por persona} = 3 \times 3000 \text{ Bs.}$$

$$\text{Costo del software desarrollado por persona} = 9000 \text{ Bs.}$$

$$\text{Costo total del software desarrollado} = 9000 \text{ Bs.} \times 6$$

$$\text{Costo total del software desarrollado} = 54000 \text{ Bs.}$$

4.3.3 Costo de Implementación del Sistema

Se observan que las herramientas del software son de Php y MySQL son de uso libre, son herramientas que muchas empresas utilizan por ser software libres que van con el presupuesto de la empresa.

4.3.4 Costo de Elaboración del Proyecto

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema en la etapa de recopilación y análisis principalmente, estos costos se presentan en la siguiente Tabla.

DETALLE	IMPORTE (\$us)
Análisis y diseño del proyecto	300
Material de Escritorio	30
Internet	40
Otros	20
TOTAL	440

Tabla 4. 9 Costo de elaboración del Proyecto
Fuente: [Elaboración Propia]

4.3.5 Costo Total

El costo total es la sumatoria del costo del software desarrollado, costo de implementación del Sistema y costo de elaboración del proyecto, detallado en las siguientes Tabla 4.10.

DETALLE	IMPORTE (\$us)
Costo de software desarrollado	7637
Costo de elaboración del proyecto	440
Total	8077

Tabla 4. 10 Costo Total del Proyecto
Fuente: [Elaboración Propia]

Por todo esto se observa que el costo más alto es el “costo del software desarrollado. Por lo tanto el costo total del proyecto es **igual 8077 (\$us)** su equivalente en **bolivianos 57104.39 Bs.**

4.3.6 CALCULO DE BENEFICIOS CON EL VAN y el TIR

Para evaluar el beneficio que se obtendrá al implementar el proyecto se calcula con el método del Valor Actual Neto (VAN) y la tasa Interna de Retorno (TIR).

4.3.6.1 Calculo del VAN

El VAN (Valor Actual Neto) es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión.

La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto.

A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto. La fórmula que utilizaremos para hallar nuestro Valor Actual Neto será.

La fórmula que utilizaremos para hallar nuestro Valor Actual Neto será.

$$VAN = \sum \frac{Ganancia}{(1+k)^n} - \sum \frac{Costos}{(1+k)^n}$$

Dónde:

VAN: Valor Actual Neto Ganancias: Ingreso de flujo anual

Costos: Salidas de flujo anual n: Numero de periodo

k: Tasa de descuento o tasa de interés al préstamo

Los Gastos y Ganancias que se estiman en un lapso de 4 años los mostraremos en la Tabla 4.15, para este caso en particular utilizaremos una tasa de descuento del 12% ($k = 0.12$) ya que es la tasa actual de interés de préstamo en las entidades financieras.

Año	Costos	Ganancias	$\frac{\text{Costos}}{(1+k)^n}$	$\frac{\text{Ganancia}}{(1+k)^n}$	Resultado
1	8077	0	7211.61	0	
2	1200	2000	956.63	1594.38	
3	600	6000	427.06	4270,68	
4	300	8500	190.65	5401.90	
	10177	16500	8785.95	11266.96	
$VAN = \sum \frac{\text{Ganancia}}{(1+12\%)^n} - \sum \frac{\text{Costos}}{(1+12\%)^n}$					2481.01

Tabla 4. 11 Cálculo del VAN

Fuente: Elaboración propia

Interpretación del VAN.

Para ver si el proyecto es rentable utilizaremos la siguiente tabla:

Valor del VAN	Interpretación
$VAN > 0$	El proyecto es rentable
$VAN = 0$	El proyecto también es rentable, porque ya está incorporada ganancia de la Tasa de Interés.
$VAN < 0$	El proyecto no es rentable

Tabla 4. 12 Interpretación del VAN

Fuente: Elaboración Propia

Como el resultado que obtuvimos es de: $VAN = 2481.01$ podemos afirmar que nuestro proyecto es rentable.

4.3.6.2 Tasa Interna de Retorno (TIR)

Cuando el VAN toma un valor igual a 0, k pasa a llamarse TIR (Tasa Interna de Retorno). La TIR es la rentabilidad que nos está proporcionando el proyecto. La ecuación que utilizaremos es la siguiente:

$$TIR = \sum \frac{Ganancia - Costos}{(1 + k)^n}$$

Dónde:

TIR: Tasa Interna de Retorno

Ganancias: Flujos de entrada de un periodo

Costos: Flujo de salida de un periodo

k : Tasa de interés al ahorro

n : Numero de periodo

Para una más fácil comprensión de esta ecuación, la expresaremos en la tabla 4.12

Año	Costos	Ganancias	$\frac{Ganancia - Costos}{(1 + k)^n}$
1	8077	0	-7211.61
2	1200	2000	637.76
3	600	6000	3843.61
4	300	8500	5211.25
			2481.01

Tabla 4. 13 Tasa Interna de Retorno

Fuente: Elaboración Propia

Es así que tras el resultado obtenido en la Tabla 4.17 se puede concluir que:

El proyecto nos dará una rentabilidad de 2481.01 \$us.

Si hubiéramos ahorrado la inversión inicial de 8077 \$us en cualquier entidad a una tasa de interés de 3% que es la tasa actual al ahorro, al cabo de 4 años contaríamos con la suma de 9046.24 \$us., lo que hace una ganancia de 969.24 \$us.

Claramente vemos que la rentabilidad es superior a la inversión realizada en ahorro. Por lo tanto esto demuestra que el porcentaje de interés ganado es mayor al 3%. Lo que nos indica que es conveniente invertir en el proyecto mucho más que dejar nuestro dinero en el banco.

4.3.7 COSTO / BENEFICIO

Para hallar el costo/beneficio de un proyecto aplicamos la siguiente ecuación:

$$C / B = \sum \text{Ganancias} / \sum \text{Costos}$$

De ahí, reemplazando la ecuación con los valores que ya conocemos de la Tabla 4.15, tenemos:

$$C/B = 16500 / 10177$$

$$C / B = 1.62 \text{ \$us}$$

Este resultado se lo interpreta de la siguiente forma:

Por cada dólar invertido en el Proyecto de Software, la empresa tiene una ganancia de **0.62 \$us.**

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

Durante el desarrollo del presente proyecto, se pudo evidenciar que la implementación de un Sistema Web de Gestión de Proyectos no es una tarea sencilla pero es imprescindible adecuar los procesos que tienen para mejorar las perspectivas empresariales.

Finalmente este capítulo detalla las conclusiones y recomendaciones a las cuales llegarnos con la culminación del proyecto, así como las recomendaciones pertinentes.

5.1 CONCLUSIONES

Con la implementación del sistema se lograron alcanzar todos los objetivos que se tenían planteados en un principio, las cuales son:

- Se pudo automatizar el registro de proyectos de la empresa de manera rápida y segura.
- Se puede realizar las distintas consultas de los proyectos que se realizan en las agencias con un criterio de búsqueda.
- Ahora se cuenta con información mucho más detallada del seguimiento de los proyectos que se realizan o se realizaron en las distintas agencias.
- Se puede generar informes y realizar la reprogramación de los proyectos.
- Se cuenta con una tabla dentro de la base de datos para el registro del personal de trabajo.
- Se realizó el módulo de reportes para tener una búsqueda actualizada de las operaciones que se realizan en los proyectos.
- Dentro de cada módulo están los sub módulos los cuales permiten tener un módulo completo, que completan los requerimientos necesarios para el sistema.

Todos los puntos señalados muestran que el trabajo se realizó según los objetivos trazados con los requerimientos de la empresa.

En conclusión según las pruebas y evaluaciones realizadas se concluye:

- La calidad del software se obtuvo una funcionalidad de 91%, confiabilidad de 97%, mantenibilidad de 91%, portabilidad de 90% y usabilidad de 87%, mostrando como resultado que el sistema se encuentra enmarcada en la norma ISO/IEC 9126
- La seguridad del sistema se utiliza la autenticación de usuarios, manejo de sesiones y encriptación de la información mediante el algoritmo de encriptación md5 de php.
- La seguridad de la base de datos se contempla el manejo de transacciones y backups.

5.2 RECOMENDACIONES

Para que puedan seguir creciendo los beneficios obtenidos con la implementación del sistema, es necesario que cada uno de los usuarios realice correctamente las tareas del sistema para afrontar el mercado actual y poder mejorar la gestión de la empresa. Las recomendaciones hechas a continuación se realizan de modo personal conforme a la experiencia adquirida en el desarrollo del proyecto.

- Recopilar información, para crear nuevas historias de usuario e implementarlas en futuras versiones del sistema.
- Se debe adaptar un sistema de facturación para tener un sistema más compacto en el área de contabilidad.
- Desarrollar sistemas basados en la logística de negocios así como: CRM (Customer Relationship Management) que es la Gestión de las Relaciones con el Cliente, EPM (Enterprise Project Management) que es una solución o herramienta de gestión de proyectos además de otros métodos.

- Se recomienda ampliar el modulo de personal para su mejor control de los proyectos.
- Integrar el sistema o desarrollar el modulo de Gestión del Proyecto a nivel Financiero para control los ingresos y egresos del proyecto.
- Se recomienda en el futuro el modulo de facturación donde registren todas las facturas.



BIBLIOGRAFÍA

- [1] Asteasuain, F., (2009), UML, 1ra edición, Users, Argentina
- [2] Abud, F.M.A. (2009), Calidad en la industria del Software La Norma ISO-9126. Instituto técnico Sistemas de información y de estudios superiores, México
- [3] Conh, M., (2009), UserStoriesAppliedfor Agile Software Developmente, 1era Edicion, Pearson-Educato. Inc., Estados Unidos
- [4] Farley, J., (2007), JBossSeamProjects, 1ra Edición, Steve Anglin, Estados Unidos.
- [5] Fasal, J., (2007), JBossSeamFromNoviceto Profesional, 1ra Edicion, Steve Anglin, Estados Unidos.
- [6] Gilifillan, I., (2003), La biblia del Mysql, 1ra Edicion, Tinypic, España.
- [7] Pressman, R., (2005), Ingeniería de Software; Un enfoque practico, 6ta Edición, McGraw-Hill, España.
- [8] Priolo, S., (2009), Métodos Ágiles, 1ra Edición, Users, Argentina.

DIRECCIONES ELECTRÓNICAS

- [9] Canos, J., Letelier, P. y Panadés C., (2003, Nombre 12), Metodologías Agiles en el Departamento de Software, [en línea], Universidad Politécnica de Valencia, <http://issi.dsic.upv.es/archives/f-1069162627/actas.pdf>, [consulta: 5 de enero de 2014]