

**UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA**



PROYECTO DE GRADO

**SISTEMA GERENCIAL DE CUADRO DE MANDO INTEGRAL (CMI)
CASO: AGENCIA ESTATAL DE VIVIENDA - AEVIVIENDA**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA
MENCION: INGENIERIA DE SISTEMAS INFORMATICOS**

**Postulante : Univ. Alfredo Aruquipa Condori
Tutor : Lic. Grover Alex Rodríguez Ramírez
Revisor : Lic. Juan Gonzalo Contreras Candia**

LA PAZ - BOLIVIA

2015



**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

Dedico este trabajo:

A Dios por guiar mi camino en todo momento y darme fortaleza en los momentos más difíciles y por no dejarme caer y poder seguir adelante.

A mis padres por su apoyo, consejos, amor, ayuda y confiar en mí en todo momento, darme fuerza e inculcarme valores morales, por lo que les estaré eternamente agradecido.

A mis amigos por la comprensión y amistad incondicional que me brindaron.

AGRADECIMIENTOS

Agradezco:

A Dios por guiarme para seguir adelante y superar los obstáculos y por haber puesto en mi camino aquellas personas que me brindaron su ayuda en todo momento.

A mis padres Severo y Antonia, mis hermanos Humberto, Reinaldo y Hugo, quienes fueron impulsores para que pueda concluir con esta etapa de mi vida y lograr este objetivo.

Un agradecimiento al Lic. Grover Alex Rodríguez Ramírez, por su paciencia y consejos en la elaboración del presente proyecto, de igual manera al Lic. Juan Gonzalo Contreras Candia por su apoyo y observaciones que ayudaron a alcanzar los objetivos

En el plano personal expreso mis agradecimientos a mis compañeros y amigos que me apoyaron y colaboraron en todo momento, para culminar este proyecto, Marcelo Ferrufino, Virginia Chino, gracias por su apoyo incondicional, a mis compañeros, Juan Carlos, Ariel, Neils, con quienes compartí momentos gratos en todos estos años.

A todos los docentes de la Carrera de informática, quienes me impartieron sus conocimientos durante todos estos años.

Finamente al personal de la Biblioteca: Don Fernando, Don Willy y Don Daniel, por su colaboración y comprensión que tuvieron desde el inicio de mi carrera.

A todos ellos muchas gracias.

RESUMEN

El presente proyecto fue desarrollado para la Agencia Estatal de Vivienda AEVIVIENDA dependiente del Ministerio de Educación en beneficio de toda la comunidad.

Con la realización de este proyecto una consolidación personal de los conocimientos adquiridos durante la carrera, tanto la planificación y análisis, como de programación y base de datos, así como la involucración en el desarrollo de un proyecto de suficiente magnitud

Se plantea una metodología SCRUM que se caracteriza por adaptarse a proyectos pequeños de largo o corto plazo para el desarrollo de la Aplicación Web, con la finalidad de obtener un diagnóstico de la situación actual de la empresa y poder obtener una solución ante el problema planteado.

Con el presente proyecto se pretende obtener una serie de mejoras y beneficios para la institución, con la información que proporcionara el proyecto se tendrá un mejor análisis del estado de avance de los proyectos, para un mejor análisis a nivel gerencial.

INDICE

CAPITULO I	ANTECEDENTES GENERALES	
1	INTRODUCCION.....	1
2	ANTECEDENTES	2
2.1	Proyectos similares	2
2.2	Resumen de cómo funciona el sistema	3
3	PLANTEAMIENTO DEL PROBLEMA.....	4
3.1	Formulación del problema	4
4	OBJETIVOS	5
4.1	Objetivo general.....	5
4.2	Objetivos específicos	5
5	JUSTIFICACIONES.....	5
5.1	Social	5
5.2	Técnica	6
5.3	Económica.....	6
6	ALCANCES.....	6

CAPITULO II	MARCO TEORICO	
	MARCO INSTITUCIONAL.....	8
a)	Misión	8
b)	Visión	8
c)	Objetivo Estratégico.....	8
d)	Objetivo Institucional	8
2.1	METODOLOGIA.....	9
2.1.1	METODO DE INVESTIGACION CUALITATIVA.....	9
2.1.2	METODOLOGIA SCRUM.....	9
2.1.2.1	Fases de SCRUM	11
2.1.2.2	Componentes de SCRUM	12
2.1.2.2.1	Los Roles.....	12
2.1.2.3	Elementos de SCRUM.....	13
2.1.2.3.1	Producto Backlog	13
2.1.2.3.2	Sprint Backlog.....	14

2.1.2.3.3 Incremento.....	15
2.1.2.4 Desarrollo de un Sprint	15
2.1.2.4.1 Planeación del Sprint.....	16
2.1.2.4.2 Seguimiento del Sprint	16
2.1.2.5 Revisión del Sprint.....	16
2.1.2.6 Modelo de procesos	16
a) Pre-proyecto	17
b) Proyecto.....	18
c) Post-proyecto	18
2.2 TECNICAS DE PRUEBAS	18
2.2.1 Pruebas de caja negra.....	18
2.2.1 Pruebas de caja blanca	19
2.2.1 Pruebas de integridad	19
2.3 UML.....	19
2.3.1 Modelado de objetos	20
2.3.2 Beneficios del UML	20
2.3.3 Metodología de Desarrollo UWE	20
2.3.3.1 Analisis de requisitos	21
2.3.3.2 Modelo conceptual.....	22
2.3.3.3 Diagrama Entidad/Relación.....	23
2.3.3.4 Diagrama Fisico	24
2.3.3.5 Modelo Navegacional	24
2.3.3.5.1 Modelo de Espacio Navegacional.....	24
2.3.3.5.2 Estructura del Modelo Navegacional.....	26
2.3.3.6 Modelo de Presentación	26
2.4 TEMAS DE APORTE	27
2.5 TECNOLOGIAS DE SOPORTE	28
2.5.1 Lenguaje de programación PHP	28
2.5.2 Gertor de Base de datos MySQL.....	29
2.5.3 Framework kohana 2.2 (Modelo Vista Controlador -MVC)	31
2.5.3.1 Modelo Vista Controlador -MVC.....	31
2.5.3.2 Kohana.....	32
2.5.3.3 Linux CENTOS	32
2.6 CALIDAD DE SOFTWARE WEB SITE QEM	33

CAPITULO IV

CALIDAD DEL SOFTWARE

4.1 Introducción	63
4.2 Evaluación elemental WEB SITE QEM	63
4.2.1 Fase de planificación y programación de la evaluación de calidad.....	63
4.2.2 Fase dedefinición y especificación de requerimientos.....	63
4.2.3 Fase de definición e implementacion de la evoluacion elemental	65
4.2.3.1 Criterio elemental absoluto con variables continuas.....	66
4.2.4 Fase de definición e implementación de laevaluación global	69

CAPITULO V

EVALUACION DE COSTOS Y BENEFICIOS

5.1 Introduccion	76
5.2 Analisis de costos	76
5.2.1 Costo de sistma desarrollado	76
5.2.2 Costo de implementación del proyecto.....	79
5.2.3 Costo de elaboración del proyecto.....	79
5.2.4 Costo total	79
5.3 Analisis de beneficios	80
5.3.1 Valor Actual Neto	80
5.3.2 Costo/Beneficio (C/B)	81
5.3.3 Tasa Interna de Retorno TIR	82

CAPITULO VI

SEGURIDAD

6.1 Algortimo MD5	83
6.2 Algoritmo.....	83
6.3 Aplicación del algortimo MD5	85
6.4 Archivos de Log.....	85
6.5 Seguridad del sistema	85
6.5.1 Amenazas	85
6.5.2 Guias de seguridad.....	86

6.5.3 Tipos de seguridad para sistemas WEB	86
6.5.3.1 Seguridad en el cliente	86
6.5.3.2 Seguridad en el servidor	87
6.5.3.3 Seguridad en la comunicación	88
6.5.3.3 Seguridad en la aplicación	88

CAPITULO VII CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones	90
7.2 Recomendaciones	91

INDICE DE FIGURAS

Figura 2.1 Diagrama de Casos de Uso	21
Figura 2.2 Diagrama de Clases	23
Figura 2.3 Diagrama Entidad/Relación	23
Figura 2.4 Diagrama Físico	24
Figura 2.5 Arquitectura MVC.....	32
Figura 3.1 Actores identificados para el Cuadro de Mando Integral.....	47
Figura 3.2 Caso de Uso Fiscal de Obra	48
Figura 3.3 Caso de Uso tecnico Financiero	49
Figura 3.4.Caso de Uso Técnico de Planificación	49
Figura 3.5 Caso de Uso Director.....	50
Figura 3.6 Caso de Uso Administrador del Sistema.....	50
Figura 3.7 Diagrama de Clases	56
Figura 3.8 Diagrama Entidad/Relacion	57
Figura 3.9 Diagrama Fisico	58
Figura 3.10 Diseño Navegacional	59
Figura 3.11 Diseño de Vistas “Listado de Proyectos”	60
Figura 3.12 Diseño de Vistas “Nuevos Proyectos”	60
Figura 3.13 Diseño de Vistas “Reportes”	61
Figura 3.14 Diseño de Vistas “Tablero de Control de Mando”	62
Figura 4.1 Tipos de Criterio Elemental.....	66

INDICE DE TABLAS

Tabla 2.1 Atributos del tamaño de Software.....	36
Tabla 3.1 Producto Backlog.....	43
Tabla 3.2 Primer Sprint.....	44
Tabla 3.3 Segundo Sprint.....	45
Tabla 3.4 Tercer Sprint.....	46
Tabla 3.5 Cuarto Sprrint.....	46
Tabla 3.6 Descripción de Caso de Uso “Registro de información técnica del proyecto”.....	51
Tabla 3.7 Descripción de Caso de Uso “Registro de alcance físico del proyecto”.....	51
Tabla 3.8 Descripción de Caso de Uso “Carga de imágenes de seguimiento a la obra”.....	51
Tabla 3.9 Descripción de Caso de Uso “Registro de ubicación georeferencial del proyecto”.....	52
Tabla 3.10 Descripción de Caso de Uso “Registro de desembolsos de proyecto”.....	52
Tabla 3.11 Descripción de Caso de Uso “Modifica los desembolsos”.....	52
Tabla 3.12 Descripción de Caso de Uso “Genera reportes a partir del Sistema”.....	53
Tabla 3.13 Descripción de Caso de Uso “Realiza seguimiento a la programación física y financiera del proyecto”.....	53
Tabla 3.14 Descripción de Caso de Uso “Verifica la ejecución física y financiera en el Cuadro de Mando Integral”.....	53
Tabla 3.15 Descripción de Caso de Uso “Realiza evaluación a partir del cuadro de mando”.....	54
Tabla 3.16 Descripción de Caso de Uso “Realiza evaluación de la ejecución financiera por departamentales a partir del cuadro de mando”.....	54
Tabla 3.17 Descripción de Caso de Uso “Registra usuarios”.....	54
Tabla 3.18 Descripción de Caso de Uso “Modifica Datos de Usuario”.....	55
Tabla 4.1 Requerimientos de calidad.....	65
Tabla 4.2 Rango de medición.....	70
Tabla 4.3 Resultado de Criterio de Usabilidad.....	72

Tabla 4.4 Resultado de Criterio de Funcionalidad	73
Tabla 4.5 Resultado de Criterio de Confiabilidad	74
Tabla 4.6 Resultado de Criterio de Eficiencia	74
Tabla 4.7 Resumen de resultados obtenidos.....	75
Tabla 5.1 Calculo de punto funcion no ajustado	77
Tabla 5.2 Conversion de punto funcion a KDLC.....	77
Tabla 5.3 Coeficientes a y b	78
Tabla 5.4 Costo de elaboracion del proyecto	79
Tabla 5.5 Costo total el proyecto	80
Tabla 5.6 Calculo del VAN acumulado	81

CAPITULO I
ANTECEDENTES
GENERALES

1 INTRODUCCION

En la actualidad el uso de las nuevas tecnologías informáticas dentro de las empresas privadas, públicas e instituciones gubernamentales es un hecho muy importante. Debido a ello surgen muchas inquietudes respecto a las formas de manejar estos instrumentos y aplicarlas, con el fin de mejorar la productividad y el acceso a la información de manera rápida y cumplir el objetivo, que les permite adecuarse al avance y seguir vigentes en los constantes cambios que ocurren en nuestra sociedad.

Hoy en día las instituciones públicas deben poseer las herramientas necesarias para facilitar la toma de decisiones oportuna y lograr los planes propuestos además de enfrentar cambios futuros. Se ha hecho necesario mantener al día toda la información para conocer el desempeño de gestión en cada periodo y en toda la gestión.

En el presente trabajo se plasma el Perfil del Proyecto de Grado, también se verá un desglose de todo lo concerniente a conceptos y definiciones que se utilizarán, se explicará cómo se dará solución al problema que se pretende resolver, es decir, el análisis, diseño y desarrollo para su posterior implementación y finalmente se llegará a una conclusión y recomendaciones, de acuerdo a los resultados obtenidos a lo largo del desarrollo del Proyecto.

El presente Proyecto de Grado tiene como finalidad el análisis, diseño e implementación de un “SISTEMA GERENCIAL DE CUADRO DE MANDO INTEGRAL (CMI) CASO: AGENCIA ESTATAL DE VIVIENDA – AEVIVIENDA”.

“Es preciso acotar que el modelo gerencial basado en el CMI, es la herramienta necesaria para que las instituciones mejoren su estructura y operatividad organizacional y las conduzca a fortalecer su gestión, en función de mejorar su calidad, su rendimiento y valor agregado”

2 ANTECEDENTES

La Agencia Estatal de Vivienda AEVIVIENDA es una institución pública especializada que tiene como misión disminuir el déficit habitacional mediante la ejecución de programas y/o proyectos de vivienda, priorizando a sectores necesitados, trabajando con compromiso, transparencia, eficacia, responsabilidad y solidaridad.

La AEVIVIENDA cuenta con 9 departamentales donde se ejecutan proyectos de construcción de vivienda nueva, mejoramiento y ampliación de viviendas en sus modalidades de crédito y subsidio de acuerdo a su normativa vigente.

La AEVIVIENDA al ser una institución de reciente creación de acuerdo al Decreto Supremo N° 986, 21 de septiembre de 2011, inicio sus actividades de aprobación y ejecución de viviendas sociales en agosto de 2012.

La Agencia Estatal de Vivienda cuenta con un Sistema de Administración de Proyectos SAP, el cual les sirve para realizar el diseño del proyecto a ejecutar, elaboración de planillas para su pago de acuerdo al avance de ítems aprobados en el diseño, la información que genera este sistema es a nivel técnico operativo, no contando con una herramienta que recolecte toda esta información y la visualice de manera más gerencial y se pueda realizar un seguimiento y monitoreo en tiempo real a la ejecución física y financiera de los proyectos por departamentales.

2.1 Proyectos Similares

En la carrera de Informática se realizaron varios Proyectos de Grado sobre problemáticas similares pero ninguna a nivel gerencial, algunos de los cuales se enumeran a continuación.

- Sistema de información para el control y seguimiento de patrocinio (SIC-PAC) Caso: Avance comunitario (Chilfund Bolivia), trata de un sistema de información para el avance comunitario que servirá de apoyo para la institución y llevar un control y seguimiento de las familias, niños(as), afiliado y patrocinado; tener al día la correspondencia. Desarrollada el año 2013 [Rosario Calle Cachi,

2013].

- Sistema de información para el control de historiales clínicos (SICHC), el problema radica principalmente en la gran cantidad de historiales clínicos del pacientes, expediente y agendas de los doctores con los que cuenta, los pacientes suelen volver periódicamente, por eso surge la búsqueda del historial, con lo que tomaba mucho tiempo, ya que era en forma manual, desarrollada el año 2010 [Amilcar Argollo Lima, 2010].
- Sistema de Administración y control de historiales clínicos para los consultorios de la UMSA, sistema que permite almacenar datos relevantes cantidades de datos desarrollada el año 2014 [Rosmery Lozano Flores, 2014]
- Software de Control de Espacios de Exposición de Mercadería (Caso Ketal SA), el proyecto se desarrolla para el control y gestión de los espacios con respecto a sus productos , proporcionar información y comportamiento de sus espacios, desarrollado el año 2014[Ramiro Jacinto Alcazar, 2014].

2.2 Resumen de cómo funciona el sistema

El Sistema Gerencial de Cuadro de Mando Integral (CMI) Caso: Agencia Estatal de Vivienda, es el que se encargará de obtener y centralizar la información para mostrarla en tableros semafórizados por departamentos, por mes y todas las agrupaciones de información necesarias generando reportes gerenciales rápidos y fiables a todo nivel de usuario dependiendo de la información de proyectos requeridos.

Existen varios proyectos relacionados o casos de estudio donde se aplica la metodología de un cuadro de mando integral para mejorar la gestión y calidad de una entidad, de los cuales podemos señalar como referencia:

- ✓ Cuadro de Mando Integral para el Control de Gestión en Oster de Venezuela S.A. (Universidad Centroccidental “Lisandro Alvarado”, Autora: Yoleida Beatriz Avendaño Briceño).

- ✓ Sistema De Control Gerencial Basado En El Cuadro De Mando Integral – Caso Empresas Asociativas De La Región Junín (Universidad Nacional Mayor de San Marcos – Universidad del Perú, Autor: Cesar Pariona Colonio)

3 PLANTEAMIENTO DEL PROBLEMA

La situación estructural y operativa de las instituciones públicas como la Agencia Estatal de Vivienda requieren información en tiempo real, confiable y segura a la ejecución de sus proyectos a nivel nacional que ayuden a la toma de decisiones preventivas y de acción inmediata para lograr los objetivos de gestión en tal sentido se muestra a continuación los principales problemas .

- Existe un mecanismo manual de centralización de la información generada en las departamentales.
- Se cuenta con información desactualizada de la ejecución de proyectos.
- Las solicitudes de información de entidades externas no son respondidas de manera oportuna y en el plazo establecido.

3.1 Formulación del problema

¿De qué manera se puede resolver la falta de información y seguimiento a los proyectos de manera física y financiera que tiene a cargo la Agencia Estatal de Vivienda, para que se pueda obtener información actualizada, confiable, oportuna y segura?

4 OBJETIVOS

4.1 Objetivo General

Desarrollar un Sistema Gerencial de Cuadro de Mando Integral para realizar el control, seguimiento y monitoreo a la ejecución física y financiera de proyectos que ayuden a la toma de decisiones gerenciales de la Agencia Estatal de Vivienda teniendo información actualizada, confiable oportuna y segura.

4.2 Objetivos Específicos

- Estudiar la teoría del Cuadro de Mando Integral para fundamentar la modelación del sistema de Control de Gestión para las departamentales de la Agencia Estatal de Vivienda a efectos de medir los objetivos estratégicos.
- Integrar los datos y registros de una manera automática para un control eficiente y una buena coordinación de los proyectos y actividades que se desarrolla.
- Contar con un control, seguimiento y monitoreo en tiempo real a la ejecución física y financiera de los proyectos en la Agencia Estatal de Vivienda.
- Canalizar la información por el Cuadro de Mando Integral para que exista un solo medio el cual brinde la información en línea.
- Desarrollar una plataforma web basados en los modelos de estándares web.

5 JUSTIFICACIONES

5.1 Social

La utilización de tecnologías como el aquí propuesto permiten desarrollar iniciativas multidisciplinarias para lograr rápidamente el mejoramiento de la calidad del desarrollo social. Viene a ser necesario desarrollar un sistema, para generar una visión local de la institución basada en datos reales que permita facilitar los distintos procesos en forma objetiva, beneficiando principalmente a los responsables de fiscalización y los técnicos sociales que podrán realizar un seguimiento continuo a la ejecución de los proyectos en tiempo real.

5.2 Técnica

El Ministerio de Obras Públicas Servicios y Vivienda cuenta con los suficientes recursos tecnológicos, materiales y equipos que no son utilizados de manera eficiente ya que muchos de los procesos son en forma semiautomática, es por eso que se pretende dar una utilidad más adecuada a recursos ya existentes.

5.3 Económica

En este contexto, los resultados de este proyecto beneficiarán directamente al sector generador de información, además contribuirá a recursos económicos. El beneficio a estos destinatarios consiste en que les proveerá de una herramienta capaz de generar información responsable y ágil.

6 ALCANCES

- Proporcionar información oportuna
- Registro y publicación de productos.
- Los visitantes del sitio web podrán acceder al sistema para la búsqueda y consulta de las publicaciones, también podrán registrar sus comentarios.
- Para asegurar la calidad del sistema de información se desarrollarán métricas que evaluarán el producto del software terminado; utilizando herramientas y técnicas de la ingeniería del software.
- El sistema permitirá a los encargados y/o administrativos de la institución controlar y administrar todo lo relativo a los procesos administrativos.
- Por último profesionalmente pondrá en manifiesto los conocimientos adquiridos durante la carrera y permitirá sentar las bases para otros estudios que surjan partiendo de la problemática aquí especificada.



CAPITULO II

MARCO TEORICO

MARCO INSTITUCIONAL

a) Misión

La AEVIVIENDA es una institución pública especializada que tiene como misión disminuir el déficit habitacional mediante la ejecución de programas y/o proyectos de vivienda, priorizando a sectores necesitados, trabajando con compromiso, transparencia, eficacia, responsabilidad y solidaridad. [http://www.aevivienda.gob.bo, 2012]

b) Visión

La Agencia Estatal de Vivienda al 2020 es una institución eficiente, transparente, confiable reconocida socialmente que resuelve el acceso de vivienda a la población boliviana y especializada en la dotación de soluciones habitacionales integrales, que trabaja por el acceso de bolivianas y bolivianos a una vivienda digna para vivir bien. [http://www.aevivienda.gob.bo, 2012]

c) Objetivo Estratégico

Contribuir al acceso de la población boliviana a una vivienda y hábitat adecuados, a un costo razonable, basados en la participación, la autogestión, la ayuda mutua, la responsabilidad compartida y la solidaridad social. [http://www.aevivienda.gob.bo, 2012]

d) Objetivo Institucional

Contar con una institución sólida, transparente, comprometida con el cambio y de máxima efectividad en sus procesos internos y en la producción de bienes y servicios a la población. [http://www.aevivienda.gob.bo, 2012]

PROCEDIMIENTOS Y FUNCIONES DE LA UNIDAD

La Dirección General Ejecutiva a través de la Dirección de Planificación realiza el monitoreo y seguimiento a proyectos a nivel nacional en los nueve departamentos, es la encargada de canalizar la demanda de viviendas en base al Plan Plurianual y la Agenda 2025.

OBJETO BAJO ESTUDIO

Seguimiento y monitoreo al cumplimiento de la ejecución física y financiera en todas las direcciones departamentales, actualmente se cuenta con un sistema de administración de proyectos donde se registra toda la información de proyectos, ejecución de planillas para los desembolsos, seguimiento a las pólizas, contratos, etc. Toda esta información se encuentra registrada en una base de datos ORACLE y el sistema está desarrollado en POWER BUILDER aplicación de escritorio que cumple su función siempre y cuando se tenga instalado y configurada la aplicación la cual se conecta a la base de datos en Oracle que se encuentra en línea, el principal objetivo del sistema de cuadro de mando integral es obtener toda esa información y visualizarla en tableros gerenciales y gráficos agrupando por departamentales la información y tenerla disponible desde cualquier dispositivo vía web.

2.1 METODOLOGÍA

La metodología es una de las etapas específicas de este proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas (o *métodos*) acerca del procedimiento para realizar las tareas vinculadas con la investigación de este proyecto.

2.1.1 MÉTODO DE INVESTIGACIÓN CUALITATIVA

El objetivo de este modelo de investigación es desarrollar el diseño arquitectónico de los sistemas, utilizando los requerimientos obtenidos en la primera fase. En el diseño arquitectónico se engloban dos componentes: los datos y los procesos, los cuales serán analizados y diseñados desde una perspectiva conceptual a una física, dentro de las actividades que se encuentran en el desarrollo del presente proyecto.

Para el desarrollo del proyecto se implementará también la metodología de diseño orientado a objetos (UML), donde permite al ingeniero de software expresar un modelo de análisis utilizando una notación de modelado con reglas sintácticas, semánticas, y prácticas [PRESSMAN-2005]

El Lenguaje de Modelado Unificado UML, permitirá visualizar, especificar, construir y documentar el presente proyecto. Se utilizará una estructura cliente/servidor y características visuales en la interfaz del usuario.

Las técnicas que se utilizarán son:

- El lenguaje UML su notación será de vital utilidad para la construcción de los diagramas para visualizar el sistema.
- Scrum, desarrollo detallado de la fase de aprobación de un proyecto informático mediante el uso de metodologías ágiles, se debe definir:
 - 1.- El cliente.- Será la persona que nos está pidiendo una solución para un problema.
 - 2.- El usuario.- la persona que utilizara esta nueva solución.
 - 3.- El tiempo.- El proyecto se atenderá a unas fechas de comienzo y unas fechas de fin.

4.- Jefe de Proyecto.- Figura necesaria para la gestión de los recursos, así como la planificación del proyecto.

2.1.2 METODOLOGIA SCRUM

En el año 1987 Takeuchi y Nonaka publicaron el artículo (the new product development game el cual dará a conocer una nueva forma de gestionar proyectos) en la que la agilidad, flexibilidad y la incertidumbre son los elementos principales.

Takeuchi y Nonaka se fijaron en empresas tecnológicas que, estando en el mismo entorno en el que se encontraban otras empresas realizaban productos en menos tiempo, de buena calidad y menos coste. SCRUM al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo que comúnmente se llaman iteraciones y que en SCRUM se llaman "sprint".

2.1.2.1 Fases de SCRUM

Para entender el ciclo de desarrollo de Scrum es necesario conocer las cinco fases que tienen el ciclo de desarrollo ágil.

1. **Concepto:** se define de forma general las características del producto y se asigna al equipo que se encargará de su desarrollo.
2. **Especulación:** en esta fase se hacen disposiciones con la información obtenida y se establece los límites que marcaran el desarrollo del producto, tales como el coste y agendas. Se construirá a partir de ideas principales y se comprueban a las partes realizadas y su impacto a su entorno. Esta fase se repite en cada iteración y consiste en rasgos generales en:
 - a. Desarrollar y revisar los requisitos generales.
 - b. La lista de las funcionalidades que se esperan

- c. Plan de estrategia. Se establece las fechas de las versiones, hitos e iteraciones, medirá el esfuerzo realizado en el proyecto.
3. **Exploración:** se incrementa en el producto en el que se añaden las funciones de la fase de especulación
 4. **Revisión:** el equipo revisa todo lo que se ha construido y se contractará en el objetivo deseado.
 5. **Cierre:** se entrega en la fecha acordada una versión del producto deseado. Al tratarse de una versión, de cierre no indica que se ha finalizado, sino que seguirá haciendo cambios, denominados “mantenimientos”, que hará que el producto final se acerque al producto final.

2.1.2.2 Componentes de SCRUM

2.1.2.2.1 Los roles

Personas que están comprendidas con el proyecto y el proceso Scrum

- **Productowner:** es la persona que toma las decisiones, y la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el productbacklog.
- **Scrummaster:** es el encargado de comprobar que el modelo y la metodología funciona. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuara con el cliente y con los gestores.
- **Equipo de desarrollo:** Suele ser un equipo pequeño de unas 5 – 9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del backlog. Personas que no son parte del proceso de Scrum, es necesario que parta de la retroalimentación de la salida del proceso y así poder revisar y plantear cada sprint.

- **Usuarios:** es el destinatario final del proceso.
- **Stakeholders:** las personas a las que el proyecto les producirá un beneficio. Participarán durante las revisiones de sprint.
- **Managers:** toma las decisiones finales participando en la selección de los objetivos y de los requisitos.

2.1.2.3 Elementos de SCRUM

2.1.2.3.1 Producto Backlog

Es el inventario en el que se almacena todas las funcionalidades o requisitos en forma de lista priorizada. Estos requisitos son los que tendrán el producto o los que irá adquiriendo en sucesivas iteraciones. La lista será gestionada y creada por el cliente con la ayuda del Scrum master, quien indicará el coste estimado para completar un requisito, y además contendrá todo lo que aporte un valor final al producto.

Las 3 características principales de esta lista de objetivos serán:

1. Contendrá los objetivos del producto, se suele usar para expresar las historias de usuario.
2. En cada objetivo se indicará el valor que le da el cliente y el coste estimado, de esta manera, se realizará la lista, priorizado por el valor y el coste, se basará en el rol.
3. En esta lista se tendrá que indicarla posibles iteraciones que se han indicado al cliente.
4. La lista ha de incluir los posibles riesgos e incluir las tareas para solventarlos.

Es necesario que antes de empezar el primer sprint se definirá cuáles van a ser los objetivos del producto y tener las listas de los requisitos ya definidas. No es necesario que sea muy detallada, simplemente deberá contener los requisitos principales para que el equipo pueda trabajar. Realizar este orden de tareas tiene como beneficios:

1. El proyecto no se paraliza simplemente por no tener claro los requisitos menos relevantes, y el cliente podrá ver resultados de forma más rápida.
2. Los requisitos secundarios aparecerán a medida que se va desarrollando el proyecto, por lo tanto, no se pierde tanto tiempo en analizarlos al principio y el cliente será más consciente de sus necesidades.
3. Los requisitos secundarios puede que no lleguen a necesitar por que se han sustituido o por que no reportan un retorno rol interesante

Una vez definidos los requisitos se tendrán que acordar cuando se tiene que entender un objetivo terminado o completo.

Se entiende que un producto está completo si:

1. Asegura que se puede realizar un entregable para realizar la demostración de los requisitos y ver que se han cumplido.
2. Incluirá todo lo necesario para indicar que se está realizando el producto que el cliente desea.

Como complemento a la definición de completado, se debería de asociar una condición de aceptación o no aceptación a cada objetivo en el mismo momento en el que se crea la lista.

Finalmente el producto backlog ira evolucionando mientras el producto exista en el mercado. Esta es la forma para evolucionar y tener un valor de producto para el cliente suficiente para ser competitivo.

2.1.2.3.2 Sprint Backlog

Es la lista de tareas que elabora el equipo durante la planificación de un sprint. Se asignan las tareas a cada persona y el tiempo que queda para terminarlas.

De esta manera el proyecto se descompone en unidades más pequeñas y se puede determinar o ver en que tareas no se está avanzando e intentar eliminar el problema.

Cómo funciona la lista:

1. Es una lista ordenada por prioridades para el cliente.
2. Puede haber dependencias entre un atarea y otra, por lo tanto se tendrá que diferenciar de alguna manera.
3. Todas las tareas tienen que tener un coste semejante que sea entre 4-16 horas.

Generalmente, las tareas a completar se suelen gestionar mediante el Scrumyboard, a cada objetivo se le asignan las tareas necesarias para llevarlo a cabo, se usan post-its que se van moviendo de una columna a otra para cambiar el estado.

Se debe incluir:

1. Lista de tareas
2. Personas responsables de cada tarea, el estado en el que se encuentran y el tiempo que queda por terminarla.
3. Permite la consulta diaria del equipo.

Permite tener una referencia diaria del tiempo que queda a cada tarea.

2.1.2.3.3 Incremento

Representa los requisitos que se han completado en una iteración y que son perfectamente operativos.

Según los resultados que se obtengan, el cliente puede ir haciendo los cambios necesarios y replantando el proyecto.

2.1.2.4 Desarrollo de un Sprint

En los sprints. El equipo trabaja para conseguir un incremento del producto. El tiempo más conveniente está entre 2 y 4 semanas, o 30 días consecutivos como máximo. Estos intervalos de tiempo, son los que se consideran más apropiados para que el stakeholders no pierda interés. Durante la ejecución del sprint se va a realizar reuniones.

2.1.2.4.1 Planeación del Sprint

Se definirá un documento en el que se refleja los requisitos del sistema por prioridades. En esta fase se definirá también la planificación del sprint, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esta iteración. Se obtendrá además en esta reunión un sprint back log, que es la lista de tareas y que es el objetivo más importante de un sprint.

Definirá que tareas se tienen que realizar y cuáles son los objetivos del backlog producto. Una vez definidos, el equipo comienza su desarrollo, pero teniendo en cuenta una serie de normas:

- El equipo puede realizar consultas de agenda fuera del sprint
- No se permite a nadie gobernar al equipo durante el sprint. El equipo se auto gestionará
- Si durante el desarrollo del sprint no se puede realizar, porque no es viable, se puede realizar una nueva planificación para realizar un nuevo sprint.
- Si el equipo no puede comprometerse a cumplir todo el backlog, realizará una consulta con el productowner para decidir que ítems eliminar.

Si de la misma manera. El equipo será capaz de realizar más ítems del backlog durante el sprint, que el indicado inicialmente, consultara también con el product owner que ítems podrán añadir.

2.1.2.4.2 Seguimiento del Sprint

En esta reunión los componentes del equipo comparten información relativa al desarrollo y colaboración para hacer las adaptaciones necesarias, aumentando así su productividad

En esta reunión se tendrá como referencia el backlog del sprint y el equipo gráfico burndown con la información de la reunión anterior y además, que tareas hizo cada persona del equipo. L reunión no podrá consumir más de 15 minutos y contestará a tres preguntas básicas.

¿Qué se ha hecho de nuevo con respecto a la última reunión diaria?

¿Qué será lo siguiente en realizar?

¿Qué problemas hay para realizarlos?

Se usará como herramientas de apoyo, con la lista de tareas del sprint actualizada y con el esfuerzo pendiente de cada tarea. También se tendrá un gráfico con las tareas pendientes en la iteración.

2.1.2.5 Revisión del Sprint

En esta reunión, los desarrolladores presentan el producto entregable que han implementado, los gestores clientes, usuarios y product owner analizan esa entrega y escuchan al equipo sobre los problemas que han tenido durante el proceso. Esta reunión servirá para tomar decisiones que ayudan a escoger el camino más adecuado para alcanzar las metas.

2.1.2.6 Modelo de procesos

La metodología Scrum emplea una estructura incremental basada en iteraciones y revisiones. El ciclo de vida del Scrum está compuesto de tres tareas o etapas las cuales son:

a. Pre – Proyecto

Antes del desarrollo del proyecto, se especifica lo que va a realizar cada sprint, al planear todos los miembros del equipo incluyendo al cliente que contribuye a la creación de una lista de características del sistema, para el análisis y la conceptualización del sistema. La recopilación de requerimiento para conformar el backlog del producto y la definición de fechas de entrega del sprint será realizada en la primera etapa.

b. Proyecto

Des pues del pre- proyecto se llevará a cabo la elaboración del proyecto con un continuo seguimiento a cargo del mismo grupo de desarrollo. En cada sprint se realizara las siguientes tareas ya mencionadas en “Desarrollo del Sprint”:

- Planeación del sprint
- Desarrollo del sprint
- Revisión del sprint

c. Post - proyecto

Luego de haber culminado todas las iteraciones, resta la revisión final. Esta última etapa denominada “cierre” en esta etapa se realiza la preparación operacional, incluyendo la documentación necesaria para la documentación necesaria para la presentación. También se encuentran las típicas actividades de fin de proyecto como hacer una versión distribuible, testear, marketing, etc.

2.2 TECNICAS DE PRUEBAS

2.2.1 Pruebas de caja negra: Realizar pruebas de forma que se compruebe que cada función es operativa.

2.2.2 Pruebas de caja blanca: Desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

2.2.3 Pruebas de integridad: Asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos.

La base de datos y los procesos de bases de datos deben ser probados como sistemas separados del proyecto.

2.3 UML

Lenguaje Unificado de Modelado, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está respaldado por el OMG (Object Management Group). UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. Según Grady Booch: UML es un “lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software...”. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

Visualizar: UML permite expresar una forma gráfica un sistema de forma que otro lo puede entender.

Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.

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

2.3.1 Modelado de Objetos

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación. Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión

2.3.2 Beneficios del UML

- Provee a los desarrolladores un lenguaje de modelado visual listo para utilizar. Consolida un conjunto de conceptos generalmente aceptado por muchos métodos y herramientas.
- Proporciona mecanismos de extensión y de especialización para ampliar los conceptos básicos.
- Es independiente de los lenguajes programación y de metodologías de desarrollo de software.
- Proporciona una base formal para entender el lenguaje de modelado.
- Utiliza conceptos de alto nivel de desarrollo tales como colaboraciones, armazones, modelos y componentes.
- Integra las mejores prácticas de desarrollo de software. **[Matsukawa, 94]**

2.3.3 Metodología de Desarrollo UWE

La propuesta de Ingeniería WEB basa en UML (UWE [Kock,2000]) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso. Iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con

las propuestas en el Proceso Unificado de Modelado

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de características adaptativas de la navegación en función de las preferencias, conocimientos o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un meta- modelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre modelo

2.3.3.1 Análisis de requisitos

Fija los requisitos funcionalidades de la aplicación para reflejarlos en un modelo de casos de uso

Las tareas que son requeridas para el desarrollo e ingeniería son las siguientes

- Definir actores
- Definir relaciones entre actores
- Definir Casos de Uso para cada actor mostrado

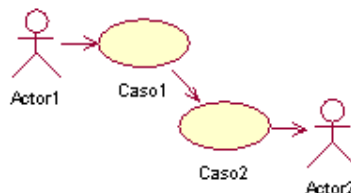


Figura 2.1 Diagrama de Casos de Uso

Fuente [Elaboración propia]

2.3.3.2 Modelo conceptual

Se define mediante un modelo de dominio, considerado los requisitos plasmados en los casos de uso, el diagrama de clases representará los conceptos con un gran porcentaje de detalle. Esto se puede representar en UML con un grupo de diagramas tanto para la parte estática como para las operaciones que define el comportamiento o sea la parte dinámica. Para ello muestra conceptos, asociaciones entre conceptos y atributos de los conceptos. La creación del esquema conceptual ayuda a descubrir y conocer la terminología del dominio del problema, modelando la comunicación y relación entre los términos importantes.

La definición del modelado conceptual para asociar y definir el modelo desde un punto de vista de análisis y diseño, se debe considerar que el termino *concepto* equivale al de clases y tipo para la notación de modelado, y partiendo desde especificaciones obtenidas de un sistema existente o sea desde un punto de vista de implementación lo más correcto es modelar las estructuras abstraídas por la ingeniería inversa desde el punto de vista de diseño o sea clases, las cuales representan los conceptos con un poco más de detalle en cuanto a atributos, operaciones, métodos y relaciones.

Las asociaciones son las relaciones significativas entre dos clases, que pueden ser comunes o eventuales pero que se preservan y son reconstruidas en el tiempo. En el modelo visual se representa como una línea entre conceptos con el nombre de la asociación, en sentidos bidireccional. En los extremos puede incluirse una expresión de multiplicidad que indica las instancias de la clase. Se puede indicar el sentido de lectura con una flecha de dirección sin sentido bidireccional. En los extremos puede incluirse una expresión de multiplicidad que indica las instancias de la clase. Se puede indicar el sentido de lectura con una flecha de dirección sin sentido semántico.

El modelo conceptual proporciona una visión más amplia del problema, más que limitarse a enumerar los requisitos. También se trata de mantener esos requisitos en contexto y tomar decisiones racionales. El diseño captura las tareas y la información esencial necesaria de las actividades del negocio, dando como resultado una visión de la solución con un enfoque sobre el proceso y centrada en los usuarios. Especifica la ubicación, las capacidades y las expectativas, de los usuarios. Se considera como un análisis de actividades y consiste en la solución de negocios para el usuario y se expresa con casos de uso.

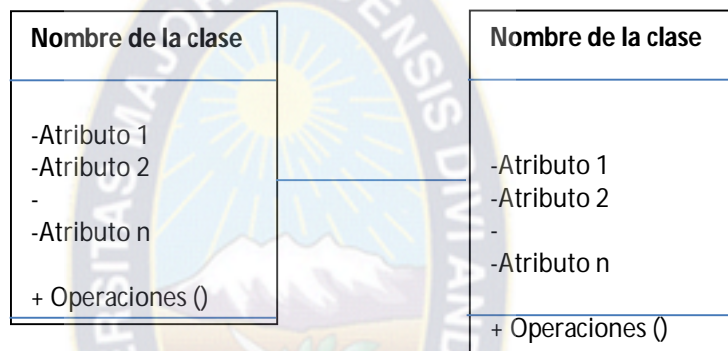


Figura 2.2 Diagrama de Clases

Fuente [Ludwig, 2010]

2.3.3.3 Diagrama Entidad/Relación

La siguiente figura nos muestra cómo se desarrolla el diagrama

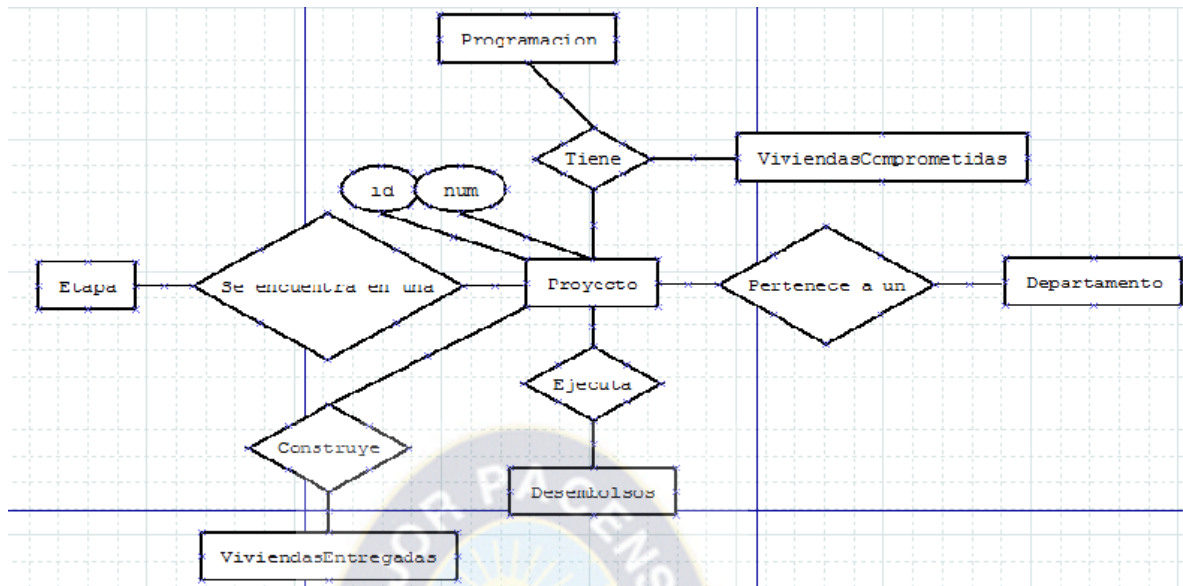


Figura 2.3 Diagrama Entidad/Relación

Fuente [Elaboración propia]

2.3.3.4 Diagrama físico

Generado automáticamente por su herramienta a partir del anterior, el cual ya incluye claves foráneas, tablas intermedia, etc.

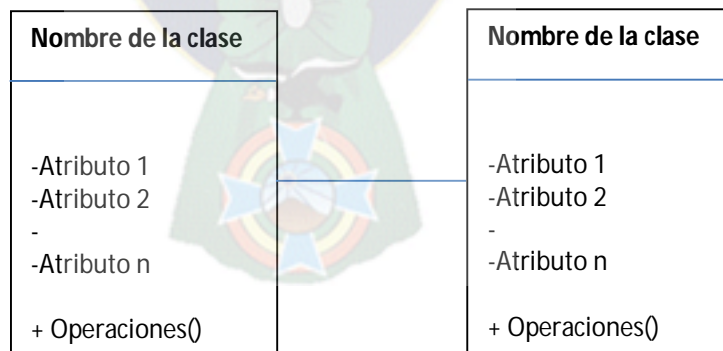


Figura 2.4 Diagrama Físico

Fuente [Ludwig, 2010]

2.3.3.5 Modelo Navegacional

En un sistema es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos y enlaces.

Nodos son unidades de navegación y están conectados por medio de enlaces. Nodos pueden ser presentados en diferentes páginas o en una misma página.

2.3.3.5.1 Modelado del Espacio Navegacional

El primer objetivo es especificar cuáles objetivos pueden ser visitados a través de la aplicación. Incorporado a este diagrama construcciones adicionales que muestran como el usuario puede alcanzar estos elementos navegando.

El modelo de navegación es representado por diagramas estereotipados de clases. Este incluye las clases de esos objetos, los cuales pueden ser visitados a través de la aplicación UWE provee un conjunto de guías y mecanismos semiautomáticos para el modelado de navegación de una aplicación, las cuales son detalladas en manuales de referencias.

UWE define un conjunto de elementos de modelado en la construcción del modelo de navegación. Como primer paso la <clase de navegación> y el <enlace de navegación> y luego los estereotipados <clase de proceso> y <enlace de proceso>, los cuales se definen con la siguiente semántica.

La clase proceso modela una clase cuya instancia es visitada por el usuario durante la navegación. Se le asigna el mismo nombre que su clase correspondiente conceptual. Este posibilita definir un mapa funcional entre la clase de proceso y los

casos de uso en una vía similar para el mapeo funcional definido entre la clase de navegación y la clase conceptual.

El enlace de proceso modela la asociación entre una clase de navegación y una clase de proceso. Este enlace de proceso necesita tener asociada información acerca del estado de proceso. Esta permite resumir actividades que contienen el proceso sobre condiciones acertadas. Las relaciones entre el espacio de navegación presentan una navegabilidad directa, indicando el punto de inicio y el punto final de navegación. Para fijar las direcciones de navegación se usan flechas, en las que se indican la multiplicidad y el rol.

Cuando falta el nombre de la relación, por convenio se determina de la siguiente manera, si la multiplicidad es menor o igual a uno, se toma el nombre de la clase destino para el rol, y si es mayor que uno se toma el plural del nombre.

2.3.3.5.2 Estructura del Modelo Navegacional

El segundo paso en la construcción del modelo de navegación, consiste en la ampliación del modelo, por un conjunto de estructuras de acceso necesarias para la navegación, esta ampliación es parcialmente automatizada, esto consiste en introducir índices, visitas guiadas y consultas. Los estereotipos y los iconos asociados para estos elementos de navegación son:

- ❖ **Índice:** Permite un acceso directo a las instancias de las clases de navegación. Se modela mediante un objeto compuesto de enlaces con los nombres de las instancias de las clases de navegación a donde apuntan
- ❖ **Consultas:** Su diseño consiste en una clase cuyo atributo es una cadena de consultas.

- ❖ **Visitas Guiadas:** Representa como acceder secuencialmente las instancias de una clase de navegación y deben ser controladas por el usuario o por el sistema

2.3.3.6 Modelo de Presentación

El modelo de presentación en UWE permite la especificación de la lógica de presentación de una aplicación. Basada sobre el modelo lógico una presentación física puede ser construida, la cual contiene afinaciones adicionales de elementos, para el diseño físico por ejemplo letras y colores. Esta presentación física, no es el objetivo de este trabajo, pues no puede ser capturada por algún modelo UML. Dentro del modelo de presentación se puede distinguir dos vistas diferentes: la estructura de vista que muestra la estructura del espacio de presentación, y la interfaz de usuario la cual presenta detalles acerca de los elementos de la interface de usuario.

Este diseño se guía por algunas reglas que a continuación se presentan:

- ❖ Crear una clase de presentación por cada clase de navegación que se presente en el modelo de estructura de navegación
- ❖ Crear una clase de presentación por cada clase menú, índice, visita guiada o consulta
- ❖ Como soporte de navegación se debe crear una clase de presentación
- ❖ Adicionar enlaces a las clases de presentación
- ❖ Señalar que elementos deberán presentarse en alguna ventana o sección
- ❖ Crear los escenarios mediante serie de bocetos, representado la sucesión de vistas de interfaz de usuario

2.3 TEMAS DE APORTE

Marco teórico sobre el modelo, cuadro de mando integral, finanzas, gerencia y toma de decisiones: Los modelos representan una colección de herramientas para describir datos explícitamente.

Un modelo muestra la relación causa y efecto entre objetivos y restricciones de tal manera que permita resolver problemas que no se pueden efectuar en el sitio debido a su magnitud, complejidad, estructura, tamaño, paso, volumen, situación geográfica o características importantes.

Un modelo puede ser tan sencillo como una simple explicación con palabras de lo fundamental de una realidad. También son utilizados los diagramas los cuales dibujan en forma simplificada, los componentes del sistema señalado con flechas, las acciones de unos sobre otros.

Gobierno electrónico: El Gobierno Electrónico, según lo define la Organización de las Naciones Unidas (ONU), es el uso de las Tecnologías de la Información y la Comunicación (TIC), por parte del Estado, para brindar servicios e información a los ciudadanos, aumentar la eficacia y eficiencia de la gestión pública, e incrementar sustantivamente la transparencia del sector público y la participación ciudadana.

2.4 TECNOLOGIAS DE SOPORTE

Las herramientas que se utilizarán en este proyecto son:

- Lenguaje de programación PHP 5
- Gestor de base de datos MySQL.
- Framework Kohana 2.2 (Modelo Vista Controlador-MVC)
- Servidor Linux Centos 6.4.

2.5.1 Lenguaje de programación PHP

La definición oficial del lenguaje PHP dice que esto son las siglas de "PHP: Hypertext Preprocessor". Es un lenguaje script (no se compila para conseguir códigos máquina, si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), que desarrolla páginas Web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto y a que es de código abierto, es el más popular y extendido en la web. Se trata de un lenguaje cuyos programas se ejecutan en la parte del servidor, y cuyo código escribiremos incrustado con el código HTML.

PHP es un lenguaje que se ejecuta en el ambiente del servidor, lo que quiere decir que ejecuta comandos en el servidor antes de dar un resultado a la vista del usuario o visitante, lo que nos permite aprovechar todos los recursos del servidor.

Características:

- Muy sencillo de aprender.
- El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.
- Se puede incrustar código PHP con etiquetas HTML.
- Excelente soporte de acceso a base de datos.
- La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar que chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, xml, creación de PDF, etc.)

- Es multiplataforma, funciona en todas las plataformas que soporten apache.

2.5.2 Gestor de base de datos MYSQL

MYSQL es un gestor de bases de datos SQL (Structured Query Language). Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Podemos agregar, acceder, y procesar datos grabados en una base de datos. Actualmente el gestor de base de datos juega un rol central en la informática, como única utilidad, o como parte de otra aplicación.

Es un Sistema de Gestión de Base de Datos Relacional. El modelo relacional se caracteriza por disponer de toda la información que está contenida en tablas, y las relaciones entre tablas deben ser representadas explícitamente en esos mismos datos. Esto añade velocidad y flexibilidad en el momento de realizar consultas. MYSQL es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo.

MYSQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los usuarios. Sin embargo, bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de MYSQL altamente conveniente para acceder a bases de datos en Internet.

Los controladores de Conectividad Abierta de Base de Datos (ODBC) para MYSQL permiten conectar la PC al servidor de MySQL y también permiten exportar/importar bases de datos.

Características:

- Mayor rendimiento.
- Mayor velocidad tanto al conectar con el servidor como al servir consultas y demás funciones.
- Mejores utilidades de administración (backup, recuperación de errores, etc.).
- Mejor integración con PHP.
- No hay límites en el tamaño de los registros.
- Mejor control de acceso, es decir, los usuarios a que tablas tienen acceso y qué permisos tienen sobre ellas.
- MYSQL se comporta mejor que otros motores de BD a la hora de modificar o añadir campos a una tabla.
- Lo mejor de MYSQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor.

2.5.3 Framework Kohana 2.2 (Modelo Vista Controlador-MVC)

2.5.3.1 Modelo Vista Controlador (MVC)

El patrón de diseño MVC organiza el código en base a su función. De hecho, este patrón separa el código en tres capas:

- La capa del **modelo** define la lógica de negocio (la base de datos pertenece a esta capa).
- La capa de la **vista** es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa).
- La capa del **controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario.

Como Mejora el proceso de desarrollo de Software:

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones.

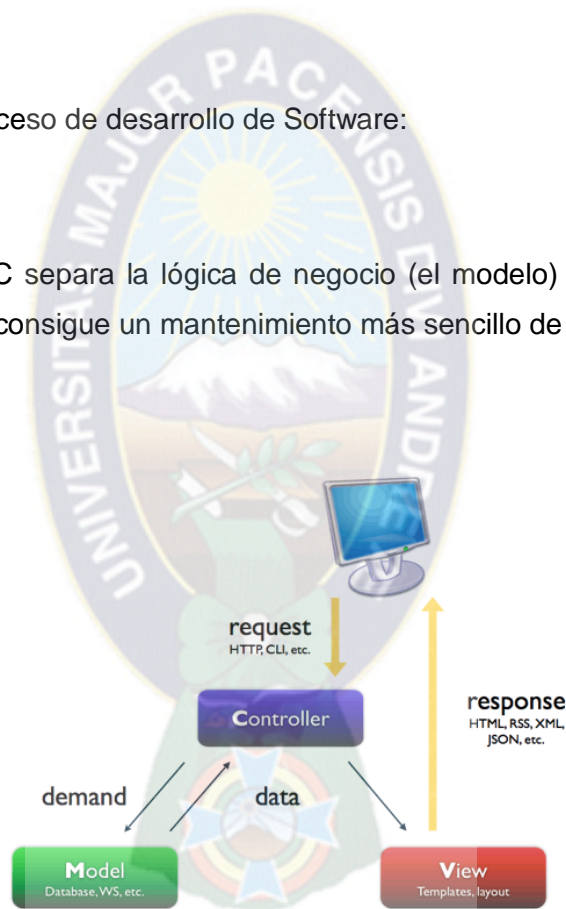


Figura 2.5 Arquitectura MVC

Fuente :[K&K, 2003]

2.5.3.2 Kohana

Es un framework para aplicaciones web para PHP5 que implementa el patrón de Modelo Vista Controlador Jerárquico (HMVC). Sus principales objetivos se basan en ser seguro, ligero, y fácil de utilizar.

Características

- Extremadamente seguro
- Extremadamente ligero
- Mínima curva de aprendizaje
- Utiliza el patrón MVC y HMVC
- Compatibilidad UTF-8 100%
- Arquitectura con bajo acoplamiento
- Extremadamente sencilla de extender

2.5.3.3 Linux Centos

CentOS (Community **ENT**erprise **O**perating **S**ystem) es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux **RHEL**, compilado por voluntarios a partir del código fuente publicado por Red Hat.

Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. Se define como robusto, estable y fácil de instalar y utilizar. Desde la versión 5, cada lanzamiento recibe soporte durante diez años, por lo que la actual versión 7 recibirá actualizaciones de seguridad hasta el 30 de junio de 2024.

2.6 CALIDAD DE SOFTWARE WEB SITE QEM (Quality Evaluation Methodology)

Aplicar una Métrica de Calidad, en Aplicaciones Web comprende un amplio rango de actividades diversas, estas son algunas:

- Aseguramiento y control de calidad
- Modelos de fiabilidad
- Modelos y evaluación de ejecución
- Modelos y medidas de productividad

En nuestro proyecto realizaremos la medición de acuerdo a los siguientes atributos como enlaces rotos, promedio de enlaces por página, páginas de acceso rápido, imagen con título, páginas muertas, entre otros. A seguir, presentamos el listado de las métricas que lograremos automatizar mediante el empleo de la herramienta Website QEM:

- Cantidad de Enlaces Rotos. Este atributo representa la cantidad de enlaces rotos (broken o dangling links) internos al sitio, como la cantidad de enlaces rotos que referencian a sitios (URLs) externos. Website MA permite almacenar los URLs de los enlaces rotos, tanto internos como externos, para un análisis posterior y posible corrección. El chequeo de un enlace roto se comprueba por medio del tipo de error 404, conforme a los códigos de estado del protocolo HTTP. Asimismo, en consideración del código de error devuelto, se puede determinar la cantidad de enlaces que conducen a páginas no accesibles.

- Cantidad Total de Enlaces de un Sitio. Se puede recolectar automáticamente la cantidad total de enlaces que posee un sitio Web. La herramienta permite calcular con gran efectividad esta métrica empleando procedimientos recursivos en el seguimiento de todos los enlaces que posee el sitio que se está analizando.
- Porcentaje de Enlaces Rotos de un Sitio. Mediante el empleo de las métricas directas mencionadas arriba (Cantidad de Enlaces Rotos y Cantidad Total de Enlaces de un Sitio), se computa el porcentaje de enlaces rotos de un sitio, establecido por la fórmula siguiente:

$$\text{Porcentaje Enlaces Rotos} = \frac{\text{Cantidad Enlaces Rotos Internos} + \text{Cantidad Enlaces Rotos Externos}}{\text{Cantidad Total Enlaces}} \times 100$$

Métricas de Calidad.- El Principal objetivo de los ingenieros de software es producir sistemas, aplicaciones o productos de alta calidad.

Para este proyecto aplicaremos el **WQM: Modelo de Calidad para Aplicaciones Web**, debido a la importancia que la calidad de software en Internet ha adquirido en los últimos años, la Conferencia Internacional de la Ingeniería de Software del año 2002 (ICSE 2002) se centró en los aspectos de Calidad para los Sistemas en Internet (Dávila y Mejía, 2002). En esta conferencia se concluyó que las métricas más importantes son las siguientes: Fiabilidad, Usabilidad, Mantenibilidad, Seguridad, Disponibilidad y Escalabilidad (Covella, 2005).

La estrategia propuesta, denominada Metodología de Evaluación de Calidad de Sitios WEB (o en inglés, Web-site Quality Evaluation Method, o, metodología Web-site QEM), pretende realizar un aporte ingenieril al proponer un enfoque sistemático, disciplinado y cuantitativo que se adecue a la evaluación, comparación

y análisis de calidad de sistemas de información centrados en la Web (más o menos complejos).

La metodología WebQEM, propuesta por Olsina el año 1999, especifica un árbol de requerimientos de características y atributos de calidad, para determinar los criterios de medición elementales e implementarlos, para realizar la agregación apropiada de manera de producir un indicador global. [OLSINA, 1999]

2.7 ESTUDIO DE COSTOS Y BENEFICIOS

2.7.1 MÉTRICAS ORIENTADAS AL TAMAÑO

La métrica del software es un factor realmente importante en el análisis de un proyecto. Las métricas orientadas al tamaño proporcionan medidas directas del software y del proceso por el cual se desarrolla. Se basan en la medición del número de Líneas De Código - LDC - que contiene el desarrollo, entendiendo por línea de código una sentencia del lenguaje de programación (se excluyen comentarios y líneas en blanco de las fuentes).

Los atributos de acuerdo a su calificación se muestran en el cuadro siguiente:

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	

Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,23	1,08	1,00	1,04	1,10	

Tabla 2.1 Atributos del Tamaño de Software

Fuente: [Software Engineering Economics (Prentice-Hall, 1981)]

2.7.2 COCOMO II

El modelo original COCOMO (Constructive Cost Model), se publicó por primera vez en 1981 por Barry Boehm y reflejaba las prácticas en desarrollo de software de aquel momento. En la década y media siguiente las técnicas de desarrollo software cambiaron drásticamente. Estos cambios incluyen el gasto de tanto esfuerzo en diseñar y gestionar el proceso de desarrollo software como en la creación del producto software, un giro total desde los mainframe que trabajan con procesos Batch nocturnos hacia los sistemas en tiempo real y un énfasis creciente en la reutilización de software ya existente y en la construcción de nuevos sistemas que utilizan componentes software a medida.

Estos y otros cambios hicieron que la aplicación del modelo COCOMO original empezara a resultar problemática. La solución al problema era reinventar el modelo para aplicarlo a los 90. Después de muchos años de esfuerzo combinado entre USC-CSE1, IRUS y UC Irvine²² y las Organizaciones Afiliadas al Proyecto COCOMO II, el resultado es COCOMO II, un modelo de estimación de coste que refleja los cambios en la práctica de desarrollo de software profesional que ha surgido a partir de los años 70. Este nuevo y mejorado COCOMO resultará de gran ayuda para los estimadores profesionales de coste software:

1. Orgánico: proyectos relativamente sencillos, menores de 50.000 líneas de código.

Se tiene experiencia en proyectos similares y se encuentra en un entorno estable.

2. Semiacoplado: proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable, y las restricciones intermedias.

3. Empotrado: proyectos bastante complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

2.8 SEGURIDAD

La seguridad es uno de los aspectos más importantes y conflictivos en el uso de Internet. La falta de una política de seguridad global está frenando el desarrollo de Internet en diferentes áreas, por lo que es necesario crear entornos seguros.

Para el desarrollo del sistema se desarrolla un entorno de seguridad para proteger la información enviada del servidor hacia el cliente y viceversa, de esta manera evitar el uso no autorizado de las funciones que ofrece el sistema, dotando al mismo de servicios de seguridad.

MD5 es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (Massachusetts Institute of Technology, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4, después de que Hans Dobbertin descubriese su debilidad. A pesar de su amplia difusión actual la sucesión de problemas de seguridad detectados desde 1996.

2.8.1 Algoritmo MD5

Se comienza suponiendo que se tiene un mensaje de b bits de longitud, escritos m_0, m_1, \dots, m_{b-1} , el algoritmo se estructura en cinco pasos:

1. Adición de bits de relleno.

El mensaje es rellenado con n bits, de tal manera que le falte a su longitud 64 bits para ser múltiplo de 512. El primer de los nbits es 1 y el resto son 0.

2. Adición de la longitud.

La nueva longitud es una representación de 64 bits y es añadida en forma de dos palabras de 32 bits, en primer lugar se muestran los bits menos significativos. Si la longitud del mensaje es mayor que 264, se usan los 64 bits menos significativos

3. Iniciar cuatro buffer A,B,C y D que son registros de 32 bits.

Inicializados con los siguientes valores:

A: 01 23 45 67

B: 89 ab cd ef

C: fe dc ba 98

D: 76 54 32 10

4. Procesar el mensaje en bloques de 16 bits(se tendrá una entrada y una salida de 32 bits)

$$F(X,Y,Z)=(X \text{ AND } Y) \text{ OR } ((\text{NOT } X) \text{ AND } Z)$$

$$G(X,Y,Z)=(X \text{ AND } Z) \text{ OR } (Y \text{ AND } (\text{NOT } Z))$$

$$H(X,Y,Z)=X \text{ XOR } Y \text{ XOR } Z$$

$$I(X,Y,Z)= Y \text{ XOR } (X \text{ OR } (\text{NOT } Z))$$

Se usa una tabla de 64 elementos $T[1...64]$ construida con la función seno, siendo T_i la parte entera $294967296 * \text{abs}(\text{sen}(i))$ (i en radianes)

5. Salida

Mensaje producido por A, B, C y D, empezando con los bits menos significativos de A y terminado con los más significativos de D. Independientemente de la longitud del mensaje, su tamaño será de 128 bits.

Lo que hace el algoritmo MD5 en relación a lo anterior es mostrar la contraseña de manera codificada, y cuando se quiere ingresar a alguna cuenta compara la contraseña con de la base de datos. De esta manera cualquier persona malintencionada que quiera obtener alguna contraseña solo podrá ver un grupo de caracteres y ninguna lógica para descifrarla ya que MD5 es un algoritmo de un solo sentido, es imposible decodificarlo.

2.8.2 Principios de seguridad

La seguridad informática se dedica principalmente a proteger la:

- Confidencialidad
- Integridad
- Disponibilidad
- Control
- Autenticación

2.8.3 Seguridad Física

Cuando hablamos de seguridad física nos referimos a todos aquellos mecanismos generalmente de prevención y detección, destinados a proteger físicamente cualquier recurso del sistema; estos recursos desde un simple teclado hasta una cinta de backup con toda la información que hay en el sistema, pasando por la

propia CPU de la máquina. La seguridad física consiste en la aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial.

A continuación mencionaremos algunos de los problemas de seguridad física con los que nos podemos enfrentar:

- Acceso Físico
- Desastres naturales
- Alteraciones del entorno
- Alimentación eléctrica
- Ruido eléctrico
- La corriente estática
- Temperaturas extremas

2.8.4 Seguridad Lógica

La seguridad lógica consiste en la aplicación de barreras y procedimientos que resguarden el acceso a datos y solo permita acceder a las personas autorizadas para hacerlo.

Los objetivos planteados por la seguridad lógica son:

- Restringir el acceso a los programas y archivos.
- Asegurar que los operadores puedan trabajar sin una supervisión minuciosa y no puedan modificar los programas ni los archivos que no correspondan.
- Asegurar que se estén utilizados los datos, archivos y programas correctos en y por el procedimiento correcto.
- Que la información transmitida sea recibida sólo por el destinatario al cual ha sido enviada y no a otro.
- Que la información recibida sea la misma que ha sido transmitida.

- Que existan sistemas alternativos secundarios de transmisión entre diferentes puntos.
- Que se disponga de pasos alternativos de emergencia para la transmisión de información.





CAPITULO III

MARCO Apl iCA tiv0

MARCO APLICATIVO

3.1 Introducción

En el presente capítulo se realizará el análisis y diseño del sistema utilizando la metodología Scrum de desarrollo adaptándose a las circunstancias de evolución del proyecto.

El proyecto consta de cuatro iteraciones (sprint), en estas se determina que partes se van a construir, priorizando los criterios del negocio y la cantidad de trabajo y tiempo que se abordara durante cada iteración.

1. Fase pre-proyecto
 - Producto backlog
 - Recopilación de requerimientos
2. Fase proyecto
 - Planeación del sprint
 - ✓ Cuatro iteraciones
 - Desarrollo del sprint
 - ✓ Análisis de requisitos
 - ✓ Modelo Conceptual
 - ✓ Modelo Navegacional
 - ✓ Modelo de Presentación
3. Fase post- proyecto
 - Cierre

3.2 Fase Pre-Proyecto

3.2.1 Recopilación de Requerimientos

En esta etapa se genera el “sprint backlog” o lista de tareas a realizar, los requisitos del sistema son parte de la visión general de los resultados que quieren obtener durante el desarrollo del

sistema. A continuación se presenta un listado con los requerimientos básicos del sistema:

- Creación de la base de datos
- Desarrollo de la interfaz de ingreso al sistema
- Desarrollo de los tableros de cuadro de mando a la ejecución física y financiera
- Desarrollo de la interfaz de adición de nuevos proyectos
- Desarrollo de la interfaz donde se muestra los registros de todos los proyectos del Cuadro de Mando Integral
- Desarrollo del módulo de monitoreo de los proyectos
- Desarrollo del módulo de seguimiento físico y financiero de los proyectos
- Desarrollo de reportes de acuerdo a requerimiento del usuario.

3.2.2 Producto Backlog

A continuación se presenta el producto backlog, que contiene los requerimientos y características del software:

Prioridad	Descripción
Alta	Creación de la base de datos
Alta	Desarrollo de la interfaz de ingreso al sistema
Alta	Desarrollo de los tableros de cuadro de mando a la ejecución física y financiera
Media	Desarrollo de la interfaz de adición de nuevos proyectos
Alta	Desarrollo de la interfaz donde se muestra los registros de todos los proyectos del cuadro de Mando Integral CMI
Alta	Desarrollo del módulo de monitoreo de los proyectos
Alta	Desarrollo del módulo de seguimiento físico y financiero de los proyectos
Alta	Desarrollo de reportes de acuerdo a requerimiento del usuario

Tabla 3.1 Producto Backlog

Fuente: [Elaboración propia]

3.3 Fase Proyecto

En esta fase de desarrollo las iteraciones, cada una de ellas corresponde a un elemento de la plataforma, contenidos y sistemas de comunicación.

3.3.1 Planeación del sprint

3.3.1.1 Primer sprint

El primer sprint desarrolla los elementos pertenecientes a las plataformas de registro, control y seguimiento. Las actividades durante este sprint constituyen el backlog siguiente:

	Sprint	Inicio	Duración
	1	02-02-2015	25 días
TAREA	TIPO	DIAS DE TRABAJO	ESTADO
Realizar la planificación del sprint	Planificación	3	Terminado
Analizar los requerimientos del backlog del producto	Planificación	3	Terminado
Construir el diseño conceptual	Desarrollo	2	Terminado
Construir el diseño de navegación	Desarrollo	2	Terminado
Construir el diseño de presentación	Desarrollo	2	Terminado
Desarrollo de la interfaz de ingreso al sistema	Desarrollo	6	Terminado
Desarrollo de los tableros de cuadro de mando a la ejecución física y financiera	Desarrollo	6	Terminado

Tabla 3.2 Primer Sprint
Fuente: [Elaboración propia]

Las funcionalidades del Sprint son:

- La creación de la base de datos
- Interfaz de ingreso con control de roles de cada usuario

3.3.1.2 Segundo Sprint

En la segunda iteración se desarrolló la interfaz de adición de proyectos, la interfaz del listado de los registros de proyectos y el módulo de monitoreo de proyectos.

	Sprint	Inicio	Duración
	2	02-03-2015	25 días
TAREA	TIPO	DIAS DE TRABAJO	ESTADO
Realizar la planificación del sprint	Planificación	2	Terminado
Analizar los requerimientos del backlog del producto	Planificación	2	Terminado
Desarrollo de la interfaz de adición de nuevos proyectos	Desarrollo	4	
Construir el diseño conceptual	Desarrollo	2	Terminado
Construir el diseño de navegación	Desarrollo	2	Terminado
Construir el diseño de presentación	Desarrollo	2	Terminado
Desarrollo de la interfaz de adición de nuevos proyectos	Desarrollo	3	Terminado
Desarrollo de la interfaz de registro de proyectos	Desarrollo	3	Terminado
Desarrollo del módulo de monitoreo de proyectos	Desarrollo	9	Terminado

Tabla 3.3 Segundo Sprint
Fuente: [Elaboración propia]

Las funcionalidades del Sprint son:

- Módulo de adición de nuevos proyectos
- Módulo de listado de registros de todos los proyectos
- Módulo de monitoreo de proyectos

3.3.1.3 Tercer Sprint

En la tercera iteración se desarrolló el módulo de seguimiento físico y financiero de los proyectos

	Sprint	Inicio	Duración
	2	02-03-2015	25 días
TAREA	TIPO	DIAS DE TRABAJO	ESTADO
Realizar la planificación del sprint	Planificación	3	Terminado
Analizar los requerimientos del backlog del producto	Planificación	2	Terminado
Construir el diseño conceptual	Desarrollo	2	Terminado
Construir el diseño de navegación	Desarrollo	2	Terminado
Construir el diseño de presentación	Desarrollo	2	Terminado
Desarrollo del módulo de seguimiento físico y financiero de los proyectos	Desarrollo	14	Terminado

Tabla 3.4 Tercer Sprint
Fuente: [Elaboración propia]

Las funcionalidades del Sprint son:

- Módulo de seguimiento físico y financiero de los proyectos

3.3.1.4 Cuarto Sprint

En la cuarta iteración se desarrolló los reportes de acuerdo a requerimiento del usuario.

	Sprint	Inicio	Duración
	2	02-03-2015	25 días
TAREA	TIPO	DIAS DE TRABAJO	ESTADO
Realizar la planificación del sprint	Planificación	3	Terminado
Analizar los requerimientos del backlog del producto	Planificación	2	Terminado
Construir el diseño conceptual	Desarrollo	2	Terminado
Construir el diseño de navegación	Desarrollo	2	Terminado
Construir el diseño de presentación	Desarrollo	2	Terminado
Desarrollo del módulo de reportes	Desarrollo	14	Terminado

Tabla 3.5 Cuarto Sprint
Fuente: [Elaboración propia]

Las funcionalidades del Sprint son:

- Módulo de reportes a requerimiento del usuario

3.3.2 Desarrollo del Sprint

De acuerdo a las cuatro iteraciones anteriores se forma la pila de productos no necesariamente detallada simplemente deberá contener los requisitos principales para poder trabajar de acuerdo a la información

3.3.2.1 Análisis de requisitos

3.3.2.1.1 Diagrama de casos de uso del cliente

Este diagrama responde a los procesos directos del cliente y sus principales actores que intervienen en el proceso de seguimiento y monitoreo de la información:

Los actores (usuarios del sistema) identificados para el uso del sistema propuesto son los siguientes:

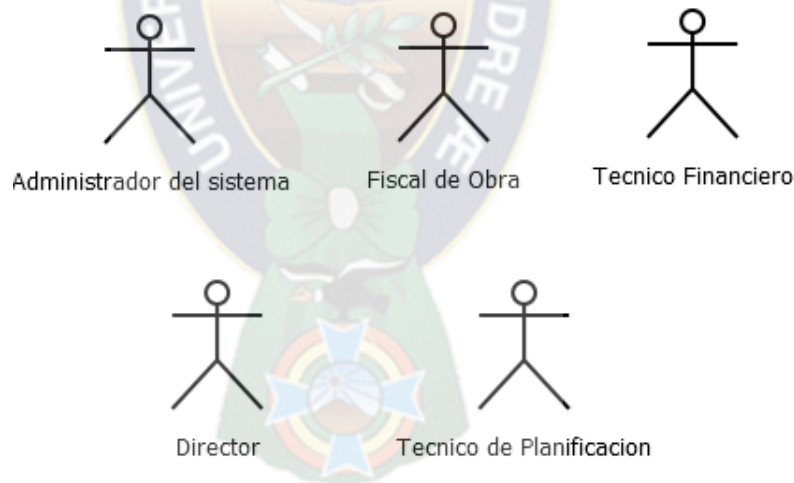


Figura 3.1 Actores identificados para el Cuadro de Mando Integral

Fuente: [Elaboración propia]

Cada uno de los actores identificados cumple una función o un rol dentro del uso del sistema, ya sea para el cargado de información o reportes generados por el sistema o para el seguimiento a la información.

3.3.2.1.2 Identificación de funciones

A continuación se muestra las funciones o roles que cumplen los actores visualizados en los diagramas de casos de uso:

Diagrama de casos de usos de Fiscal de Obras. El diagrama responde a los procesos que realiza el fiscal de obra, para el cargado de la información de proyectos en todas sus etapas.

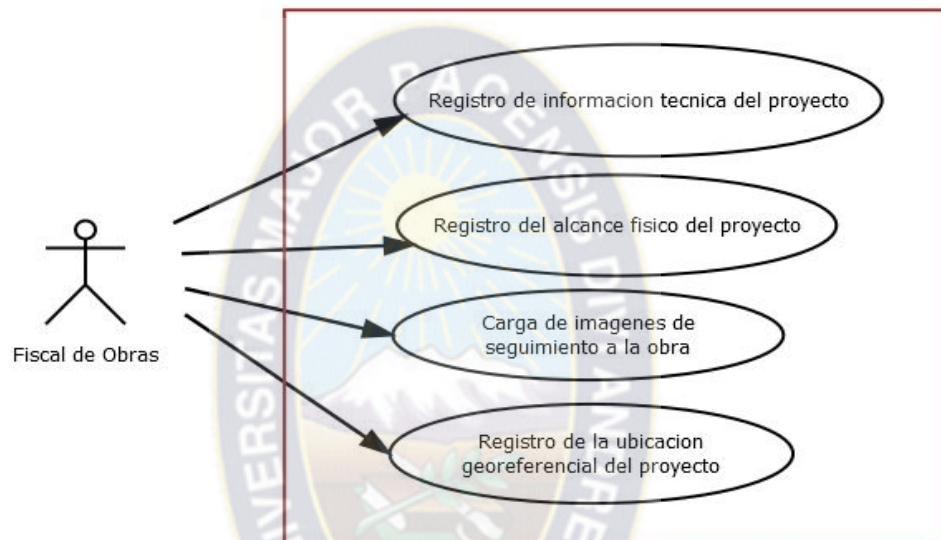


Figura 3.2 Caso de uso Fiscal de obra

Fuente: [Elaboración propia]

Diagrama de casos de uso de Técnico Financiero. El diagrama responde a los procesos administrativos financieros que realiza el técnico financiero.

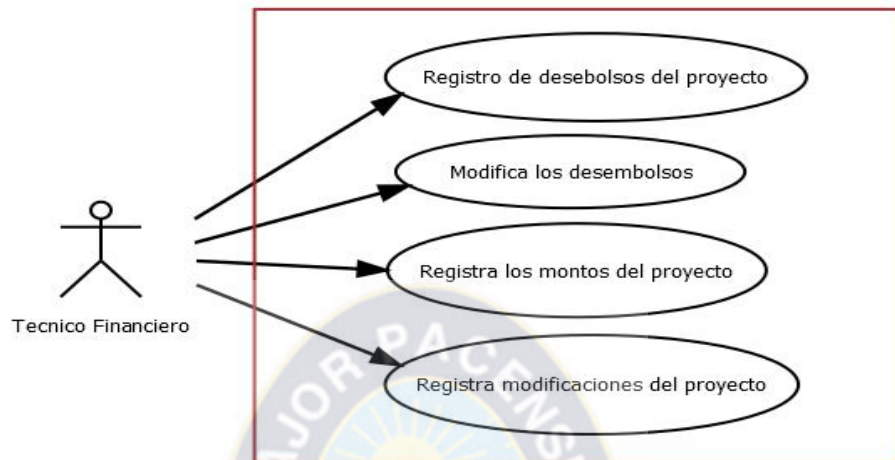


Figura 3.3 Caso de uso Técnico Financiero
Fuente: [Elaboración propia]

Diagrama de caso de uso de Técnico de planificación. Este diagrama responde a los procesos que realiza el Técnico de planificación.

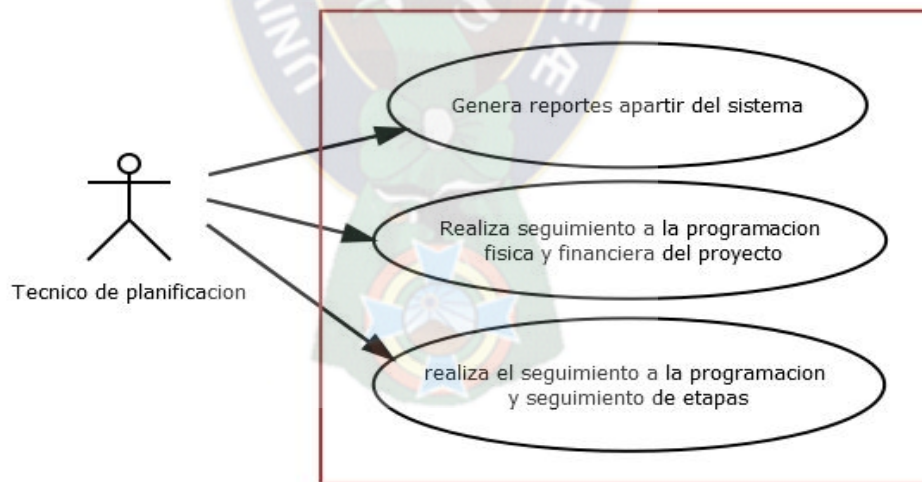


Figura 3.4Caso de uso Técnico de Planificación
Fuente: [Elaboración propia]

Diagrama de caso de uso de Director. Este diagrama responde a las consultas del director para la posterior toma de decisiones.

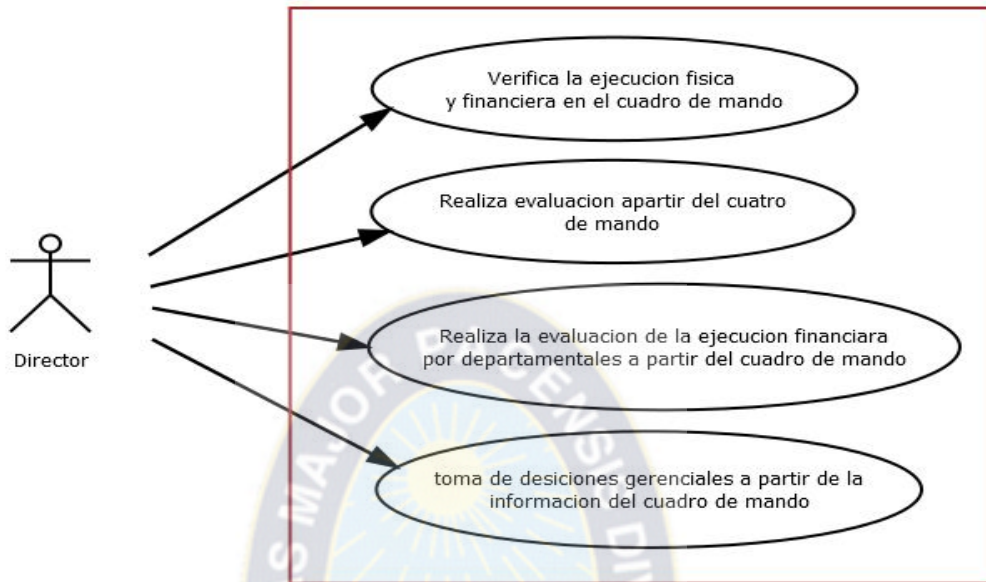


Figura 3.5 Caso de Uso Director
Fuente: [Elaboración propia]

Diagrama de caso uso Administrador del Sistema. Este diagrama responde al Administrador.

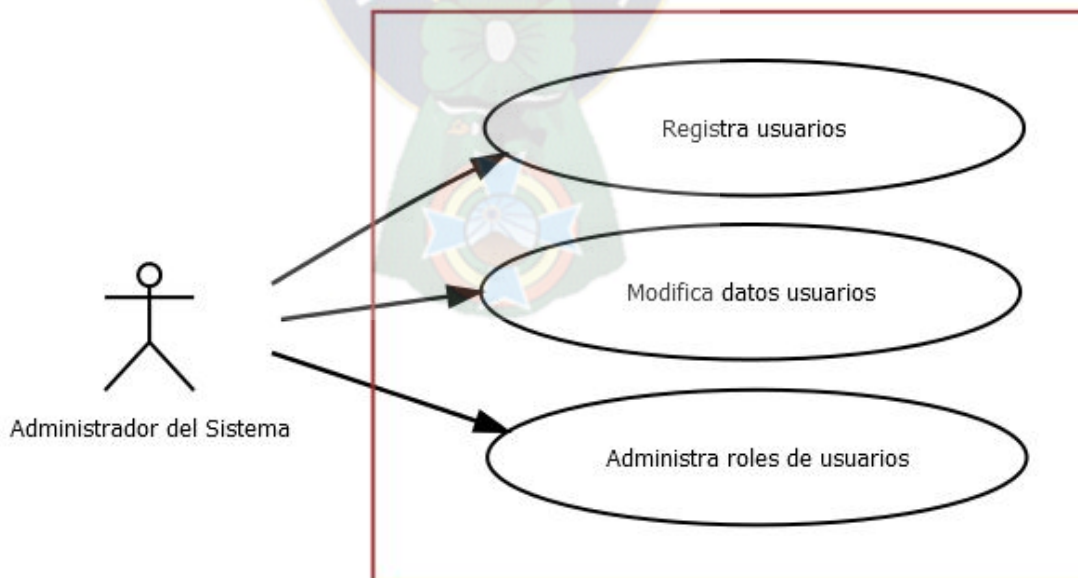


Figura 3.6 Caso de Uso Administrador del sistema
Fuente: [Elaboración propia]

Descripción de los casos de uso Fiscal de Obra

En esta sección se tiene la descripción de los casos de uso tipo texto. En este caso de uso se describe la funcionalidad de registrar información.

CASO DE USO	Registro de información técnica del proyecto
Actor	Fiscal de Obra
Descripción	El fiscal registra la información técnica del proyecto: Nombre del proyecto, Departamento, Gestión, Contratista
Tipo	Primario

Tabla 3.6 Descripción de Caso de Uso “Registro de información técnica del proyecto”

Fuente: [Elaboración propia]

En este caso de uso se describe el alcance del proyecto.

CASO DE USO	Registro del alcance físico del proyecto
Actor	Fiscal de Obra
Descripción	El fiscal verifica el alcance físico del proyecto y el avance de obras.
Tipo	Primario

Tabla 3.7 Descripción de Caso de Uso “Registro del alcance físico del proyecto”

Fuente: [Elaboración propia]

En este caso de uso se registran las imágenes del proyecto

CASO DE USO	Carga de imágenes de seguimiento a la obra
Actor	Fiscal de Obra
Descripción	El fiscal registra las imágenes del avance de obras de los proyectos.
Tipo	Primario

Tabla 3.8 Descripción de Caso de Uso “Carga de imágenes de seguimiento a la obra”

Fuente: [Elaboración propia]

En este caso de uso se registra la ubicación geográfica del proyecto

CASO DE USO	Registro de ubicación georeferencial del proyecto
Actor	Fiscal de Obra
Descripción	El fiscal registra la ubicación exacta georeferencial del proyecto.
Tipo	Primario

Tabla 3.9 Descripción Caso de Uso “Registro de ubicación georeferencial del proyecto”

Fuente: [Elaboración propia]

Descripción de Casos de uso del Técnico Financiero.

En este caso se registra los desembolsos del proyecto

CASO DE USO	Registro de desembolsos del proyecto
Actor	Técnico Financiero
Descripción	El técnico financiero registra los desembolsos asignados a cada proyecto
Tipo	Primario

Tabla 3.10 Descripción Caso de Uso “Registro de desembolsos del proyecto”

Fuente: [Elaboración propia]

En este caso de uso se modifican los desembolsos de los proyectos.

CASO DE USO	Modifica los desembolsos
Actor	Técnico Financiero
Descripción	El técnico financiero modifica los desembolsos asignados a los proyectos.
Tipo	Primario

Tabla 3.11 Descripción Caso de Uso “Modifica los desembolsos”
Fuente: [Elaboración propia]

Descripción de casos de uso del Técnico de planificación.

En este caso de uso se genera reportes del sistema.

CASO DE USO	Genera reportes a partir del sistema
Actor	Técnico de planificación
Descripción	El técnico de planificación genera reportes de acuerdo a sus requerimientos.
Tipo	Primario

Tabla 3.12 Descripción Caso de Uso “Genera reportes a partir del sistema”
Fuente: [Elaboración propia]

En este caso de uso se realiza la programación física y financiera del proyecto.

CASO DE USO	Realiza seguimiento a la programación física y financiera del proyecto.
Actor	Técnico de planificación
Descripción	El técnico de planificación realiza el seguimiento al desarrollo físico y financiero del proyecto.
Tipo	Primario

Tabla 3.13 Descripción Caso de Uso “Realiza seguimiento a la programación física y financiera del proyecto”

Fuente: [Elaboración propia]

Descripción de casos de uso del Director.

En este caso de uso se verifica la ejecución física y financiera del proyecto.

CASO DE USO	Verifica la ejecución física y financiera en el cuadro de mando
Actor	Director
Descripción	El Director verifica las ejecuciones físicas y financieras de los proyectos en el cuadro de mando
Tipo	Primario

Tabla 3.14 Descripción Caso de Uso “Verifica la ejecución física y financiera en el cuadro de mando”

Fuente: [Elaboración propia]

En este caso de uso se realiza la evaluación en el cuadro de mando

CASO DE USO	Realiza evaluación a partir del cuadro de mando
Actor	Director
Descripción	El director realiza la evaluación de los proyectos a partir del cuadro de mando.
Tipo	Primario

Tabla 3.15 Descripción Caso de Uso “Realiza evaluación a partir del cuadro de mando”

Fuente: [Elaboración propia]

En este caso de uso se evalúa la ejecución financiera por departamentos a partir del cuadro de mando.

CASO DE USO	Realiza evaluación de la ejecución financiera por departamentales a partir del cuadro de mando
Actor	Director
Descripción	El director evalúa la ejecución financiera de los proyectos asignados por departamentales en el cuadro de mando.
Tipo	Primario

Tabla 3.16 Descripción Caso de Uso “Realiza evaluación de la ejecución financiera por departamentales a partir del cuadro de mando”

Fuente: [Elaboración propia]

Descripción de casos de uso Administrados del Sistema.

En este caso de uso se verifica el registro de usuarios al sistema

CASO DE USO	Registra usuarios
Actor	Administrador del sistema
Descripción	El administrados del sistema registra a los usuarios para el manejo del sistema
Tipo	Primario

Tabla 3.17 Descripción Caso de Uso “Registra usuarios”

Fuente: [Elaboración propia]

En este caso de uso se verifica la modificación de datos de usuario.

CASO DE USO	Modifica datos de usuario
Actor	Administrador del sistema
Descripción	El administrador del sistema modifica datos de usuarios registrados.
Tipo	Primario

Tabla 3.18 Descripción Caso de Uso “Modifica datos de usuario”
Fuente: [Elaboración propia]



3.3.3 Modelo Conceptual

3.3.3.1 Diagrama de Clases

El diseño conceptual se basa en el análisis de requerimientos basados en los casos de uso.



Figura 3.7 Diagrama de Clases “diseño conceptual”

Fuente: [Elaboración propia]

3.3.3.2 Diagrama Entidad/Relación

El modelo entidad relación del Sistema Gerencial de Cuadro de Mando Integral, donde se muestran las entidades, relaciones y atributos.

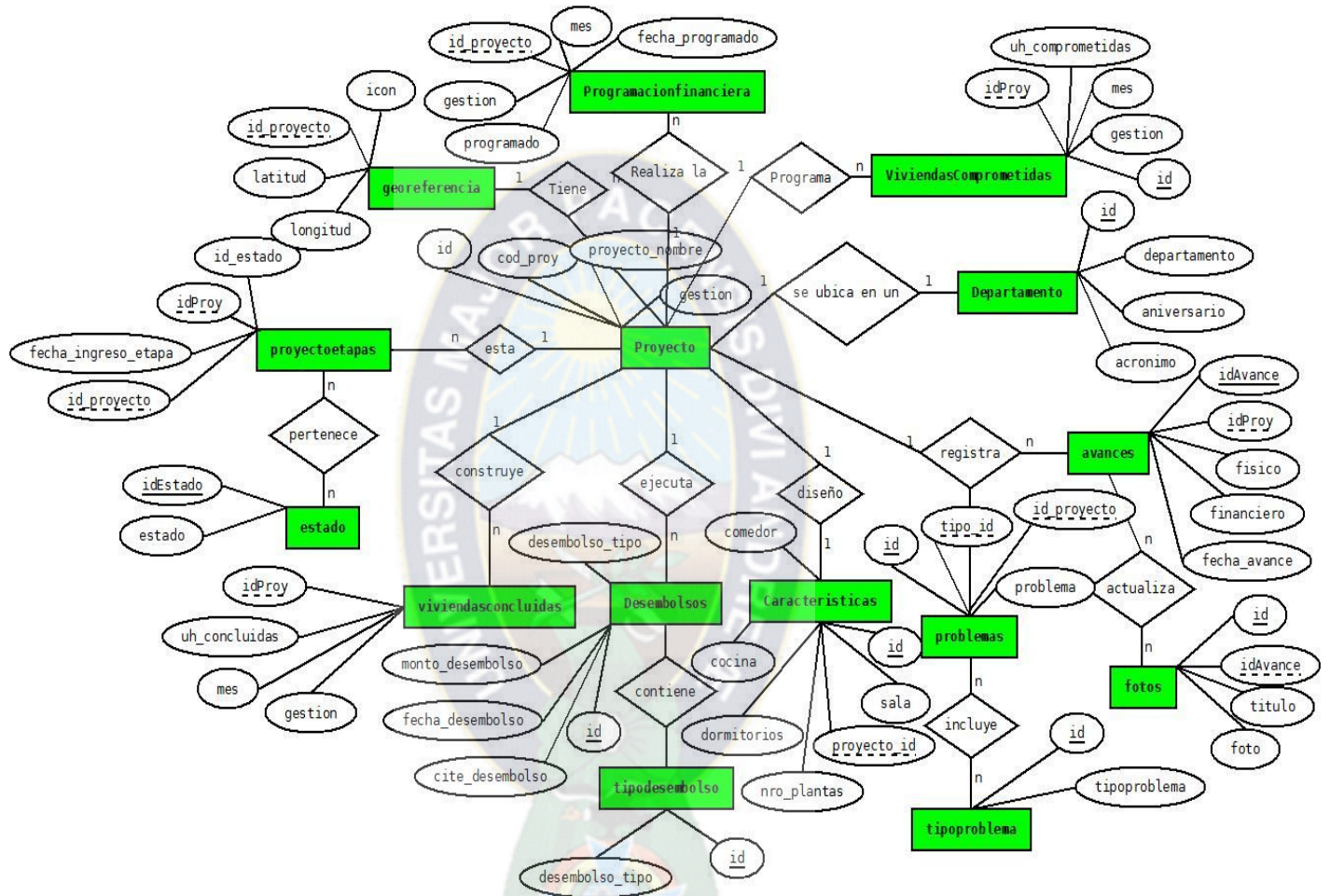


Figura 3.8 Diagrama Entidad Relación
Fuente: [Elaboración propia]

3.3.3.3 Diagrama físico

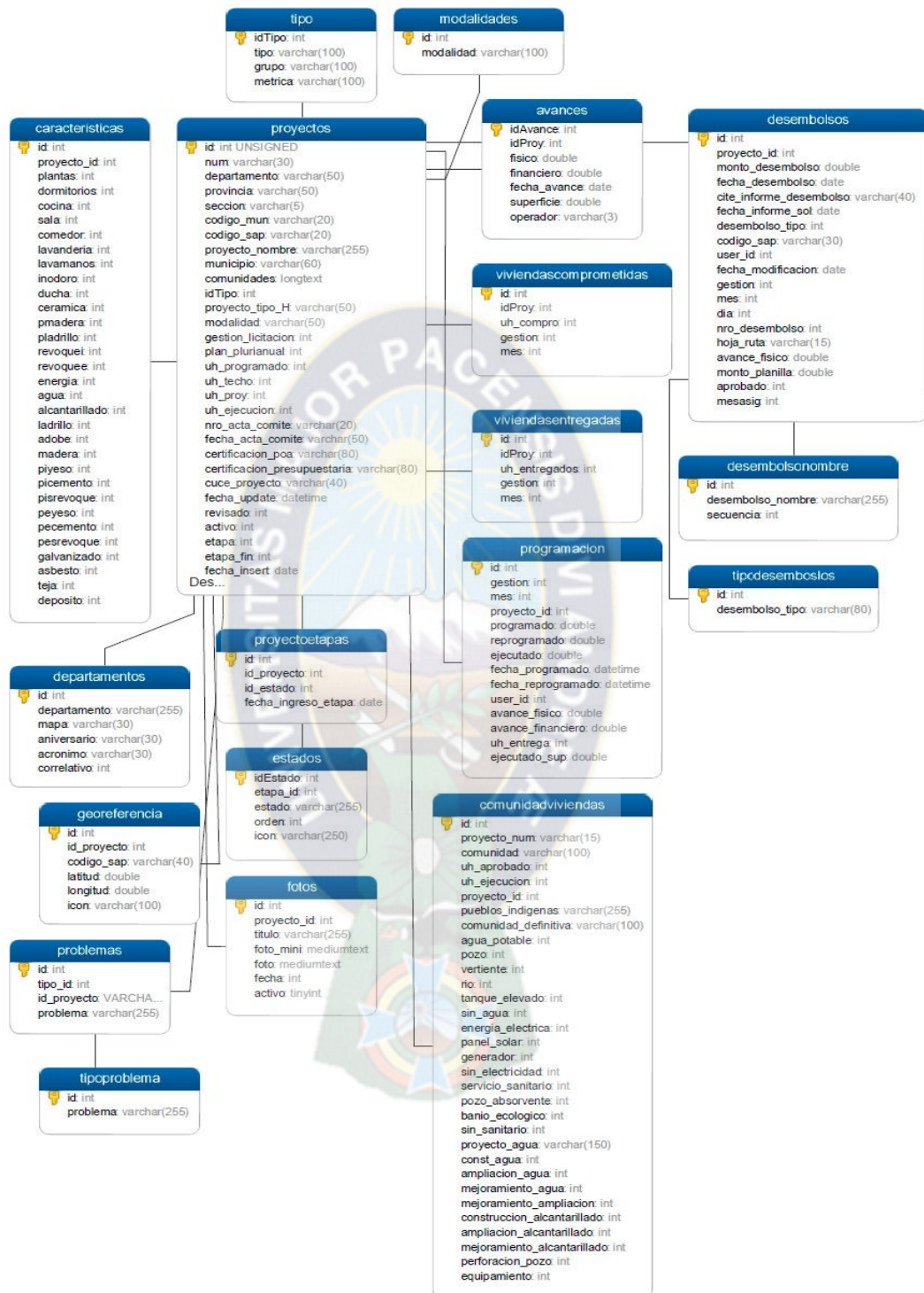


Figura 3.9 Diagrama Físico “Base de Datos”
Fuente: [Elaboración propia]

3.3.3.4 Diseño navegacional

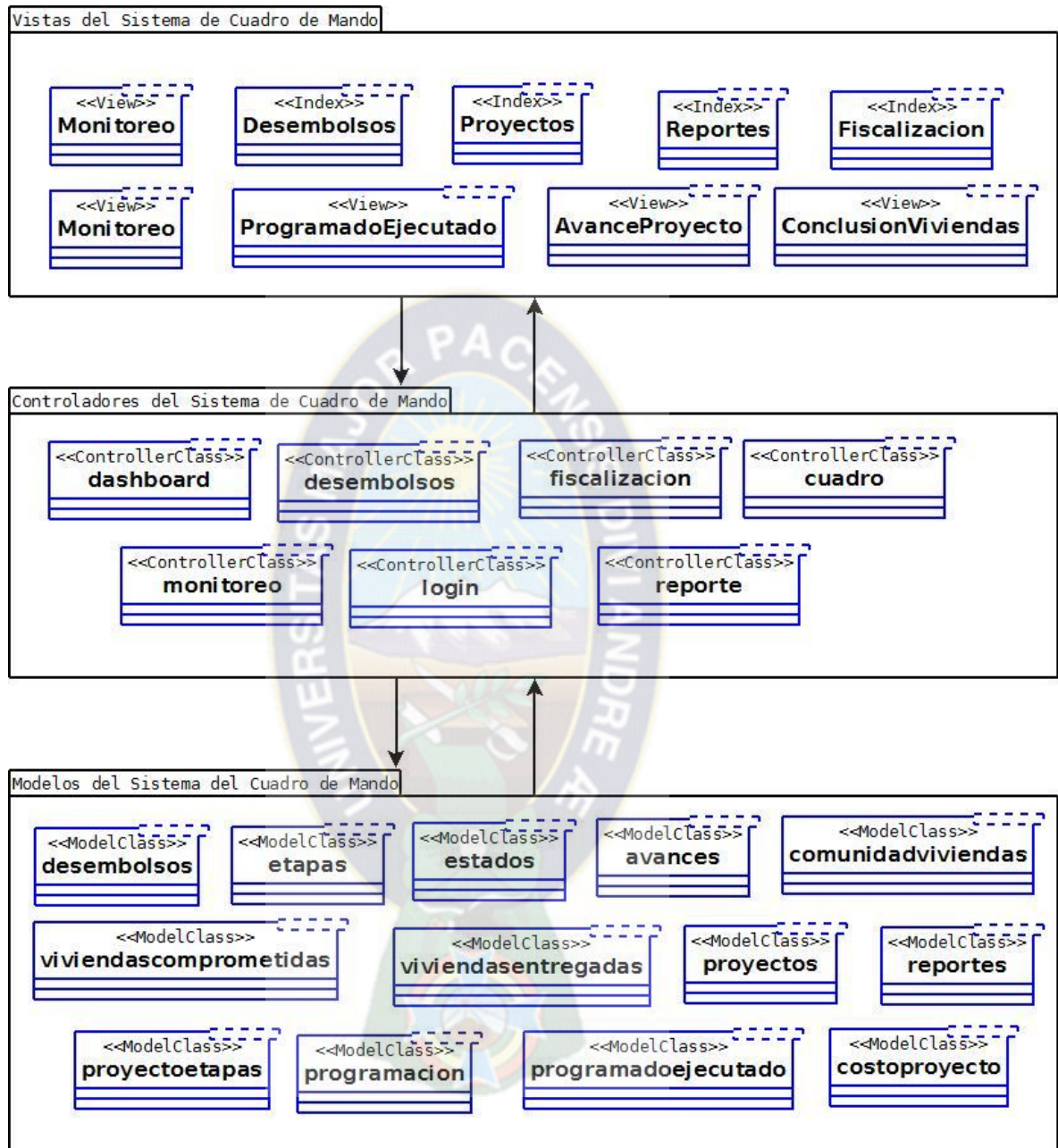


Figura 3.10 Diseño Navegacional

Fuente: [Elaboración propia]

3.4 Fase Post – Proyecto

3.4.1 Diseño de interfaces

Lista de proyectos:

			SICOES	IMPRIMIR	EJECUTIVO	Quitar filtro
COD	CUSE SICOES	Codigo SAP	Nombre	Departamento	Municipio	Estado
AEV-BNI-0001	13-0342-00-347363-2-2	AEV-02-00000112	CONSTRUCCION DE 100 VIVIENDAS EN E	BENI	Guayaramerin	En ejecucion
AEV-BNI-0002	13-0342-00-347363-2-2	AEV-02-00000112	CONSTRUCCION DE 50 VIVIENDAS PARA	BENI	Loreto	Viviendas construidas

Figura 3.11 Diseño de vistas “Listado de proyectos”

Fuente: [Elaboración propia]

Nuevo proyecto:

NUEVO PROYECTO

INGRESAR DATOS

NOMBRE DEL PROYECTO

GESTION PARA EL PROYECTO

CODIGO INE

Departamento

Provincia

TIPO

MODALIDAD

ETAPA

Figura 3.12 Diseño de vistas “Nuevos proyectos”

Fuente: [Elaboración propia]

Reportes, ficha técnica de proyecto generada a partir de la información del cuadro de mando.



ESTADO PLURINACIONAL DE BOLIVIA
AGENCIA ESTATAL DE VIVIENDA



CONSTRUCCIÓN DE 150 VIVIENDAS EN EL MUNICIPIO DE MAGDALENA - BENI
Fecha Actualizada al 09/11/2014 09:11:11

Aniv. Dep: 18 de Noviembre / Aniv. Mu:

UBICACION DEL PROYECTO

DEPARTAMENTO	BENI
MUNICIPIO(s)	Magdalena
COMUNIDADES BENEFICIADAS	1. Nueva Calama (41 u.h) 2. Orobayaya (48 u.h) 3. Bella Vista (61 u.h)
	
LATITUD	-13.2583333
LONGITUD	-64.0566666



DATOS DEL PROYECTO

TIPO DE PROYECTO:	MODALIDAD:	MONTO LICITADO:
Vivienda Nueva	SUBSIDIO	12.204.535,50

DETALLE DE PRESUPUESTO EN Bs

DETALLE	AEVIVIENDA	BENEFICIARIO	CONCURRENCIA MUNICIPAL	CONCURRENCIA DEPARTAMENTAL	COSTO PROYECTO
OBRA (Bs.)	12.204.535,50	2.059.279,50	0,00	0,00	14.263.815,00
OBRA (%)	85,56	14,44	0,00	0,00	100,00
SUPERVISION	340.430,75	0,00	0,00	0,00	340.430,75
COSTO TOTAL DEL PROYECTO					14.604.245,75
COSTO TOTAL POR VIVIENDA					95.092,77

* Monto concurrencia: 558.935 Bs, Sin concurrencia y/o recuperacion

FECHA INICIO DE OBRA	PLAZO EJECUCION	FECHA DE CONCLUSION
07/03/2013	666	01/01/2015

PLAZO EJECUCION:270, AMPLIACION: 395 DIAS

POBLACION BENEFICIADA

No DE FAMILIAS BENEFICIADAS	No DE HABITANTES
150	614

GENERACION DE EMPLEO

EMPLEOS DIRECTOS	EMPLEOS INDIRECTOS
83	453

VERIFICACION FISICA Y PRESUPUESTARIA

ESTADO DEL PROYECTO	% AVANCE FISICO	% AVANCE FINANCIERO
Paralizado	86,00	88

CARACTERISTICAS DEL PROYECTO

DESCRIPCION EL PROYECTO	SUP. CONSTRUIDA
El proyecto de construcción de viviendas sociales ha contemplado el diseño de una planta que consta de 3 dormitorios, estar/comedor, con piso de ladrillo cerámico, un baño con lavamanos e inodoro, ducha y cocina. Con acabados de revoque de cemento y pintura interior, carpintería metálica en ventanas y carpintería de madera en puertas; revestimiento de cerámica en baños y cocina, además de las instalaciones eléctricas e hidrosanitarias.	61.51 m2

OBSERVACIONES

El proyecto actualmente se encuentra en ejecución en las 3 comunidades intervenidas. En la comunidad de Bella Vista los trabajos se encuentran con mayor avance debido a que existe mejor vinculación (terrestre y fluvial), en las comunidades de Orobayaya y Nueva Calama se están ejecutando viviendas que se encuentran en distintos niveles de avance y se encuentra culminando con el acopio del total de los materiales, aprovechando la accesibilidad que existe a la fecha, ya que los caminos se deterioran con las precipitaciones pluviales, por lo que es menester, agilizar el acopio de materiales y ejecutar las viviendas antes que comience la "época de lluvias". Cabe mencionar que en el cuadro de ampliación de plazo se encuentran adicionados los días paralizados, por lo que la fecha de conclusión da el 02 de Diciembre del presente. Actualmente los avances se están realizando de forma regular para poder lograr los objetivos en la presente gestión.

Figura 3.13Diseño de vistas "Reporte del Proyecto"

Fuente: [Elaboración propia]

Reportes, tablero de control de mando con la ejecución financiera
 “Programado vs Ejecutado” por departamentales.



Figura 3.14 Diseño de vistas “Tablero de Control de Mando”
 Fuente: [Elaboración propia]



CAPITULO IV

Calidad de software

CALIDAD DE SOFTWARE

4.1 Introducción.

En este capítulo se hará el análisis posterior al desarrollo e implementación del cuadro de control de mando integral en este análisis se comprobará la calidad del software mediante el uso de estándares.

4.2 Evaluación elemental WEB SITE QEM

La metodología WEB SITEQEM sirve para la evaluación de calidad de sitios web, cuyo. Las fases de la metodología que intervienen en el proceso de evaluación y ordenamiento de calidad son:

- ✓ Planificación y programación de la evaluación de calidad
- ✓ Definición y especificación de requerimiento de calidad
- ✓ Definición e implementación de la evaluación elemental.
- ✓ Definición e implementación de la evaluación global.

4.2.1 Fase de planificación y programación de la evaluación de calidad

Contiene actividades y procedimientos de soporte con el fin de determinar objetivos estratégicos, tácticos u operativos. Establece las principales estrategias y metas del proceso en un contexto organizacional; permite seleccionar un modelo de proceso de evaluación, asignar métodos, agentes, recursos a las actividades programas y replanificar el proceso de evaluación cuando ya está en marcha.

4.2.2 Fase de definición y especificación de requerimientos

En esta etapa, se consideran aspectos de la fase de definición y especificación de requerimientos de calidad. Esa fase trata con actividades y procedimientos para la licitación, modelado y especificación de los requerimientos de calidad.

A partir de un modelo de evaluación con el fin de analizar, comparar, comprender y potencialmente mejorar características y atributos de una determinada aplicación, los requerimientos deben responder a necesidades y deseos de un perfil (o perfiles) de usuario y dominios establecidos.

El proceso de determinación de requerimientos finaliza con un documento en el cual se utilizan todas las características y atributos cuantificables. De manera que a partir de esas características se derivan características y a partir de estas siguiendo un proceso de descomposición recursivo, se especifican atributos. Por otra parte para cada uno de los atributos cuantificables ahí podemos asociar una variable X_i que puede tomar valores reales a través de un proceso de medición.

El valor final computado corresponderá a una preferencia elemental. Por lo tanto, los requerimientos de calidad quedaran completos, luego de acordar un conjunto de valores y rangos para cada atributo, que son los siguientes:

1. Usabilidad	2.2.2.1 Permanencia de los controles contextuales
1.1 Comprensibilidad global del sitio	2.2.2.2 Estabilidad
1.1.1 Esquema de organización global	2.2.2.3 Nivel de desplazamiento
1.1.1.1 Mapa del sitio	2.2.2.3.1 Desplazamiento vertical
1.1.1.2 Índice global (por temas, etc.)	2.2.2.3.2 Desplazamiento horizontal

1.1.1.3 Tabla de contenidos	2.2.3 Predicción navegacional
1.1.2 Calidad en el sistema de etiquetado	2.2.3.1 Enlace con título (texto explicatorio)
1.1.2.1 Etiquetado textual	2.2.3.2 Calidad de la fase de enlace
1.1.2.2 Etiquetado con iconos	2.3 Funciones misceláneas y específicas del dominio
1.1.3 Visita guiada	2.3.1 Relevancia de enlaces
1.1.3.1 Visita convencional	2.3.2 Relevancia de enlaces
1.1.3.2 Visita virtual(con tecnología VR*)	2.3.3 Aspecto de imágenes
1.1.4 mapa de imagen	2.3.1 Indicador de tamaño
1.2 Mecanismos de ayuda u retroalimentación en línea	2.3.2 Zooming
1.2.1 Calidad de ayuda	3 Confiabilidad
1.2.1.1 Ayuda explicaría acerca del sitio	3.1 No diferencia
1.2.1.2 Ayuda de la búsqueda	3.1.1 Número de diferencias o cualidades ausentes debido a diferentes navegadores
1.2.2 Indicador de la última actualización	3.1.2 Errores de enlaces
1.2.2.1 Global	3.1.3 Enlaces rotos
1.2.2.2 Restringido subsitio o pagina	3.1.4 Enlaces inválidos
1.2.3 Directorio de direcciones	3.1 Errores de diferencias varias
1.2.3.1 Directorio E-mail	4 Eficiencia
2 Funcionalidad	4.1 Accesibilidad de información
2.1 Aspectos de búsqueda y recuperación	4.1.1 Soporte de versión solo texto
2.1.1 Mecanismos de búsqueda en e l sistema	4.1.2 Legibilidad al desactivar propiedad imagen browser
2.1.1.1 Búsqueda restringida	4.1.2.1 Imagen con título
2.1.1.1.1 De proyectos (Nombre, descripción)	4.2 Performance
2.1.1.2 Búsqueda global	4.2.1 Paginas rápidas

Tabla 4.1 Requerimientos de calidad
Fuente: [OLSINA, 1999]

4.2.3 Fase de definición e implementación de la evolución elemental

La elección del tipo de criterio de evaluación elemental resulta de importancia en consideración de los niveles de precisión depende del grado de criticidad de algunos o de todos los componentes del producto en un proyecto de evaluación. Dos tipos básicos de criterios elementales son los absolutos y los relativos y dentro de los primeros se pueden descomponer en criterios con variables continuas y criterios con variables discretas [OLSINA, 1999] como vemos en la siguiente figura:

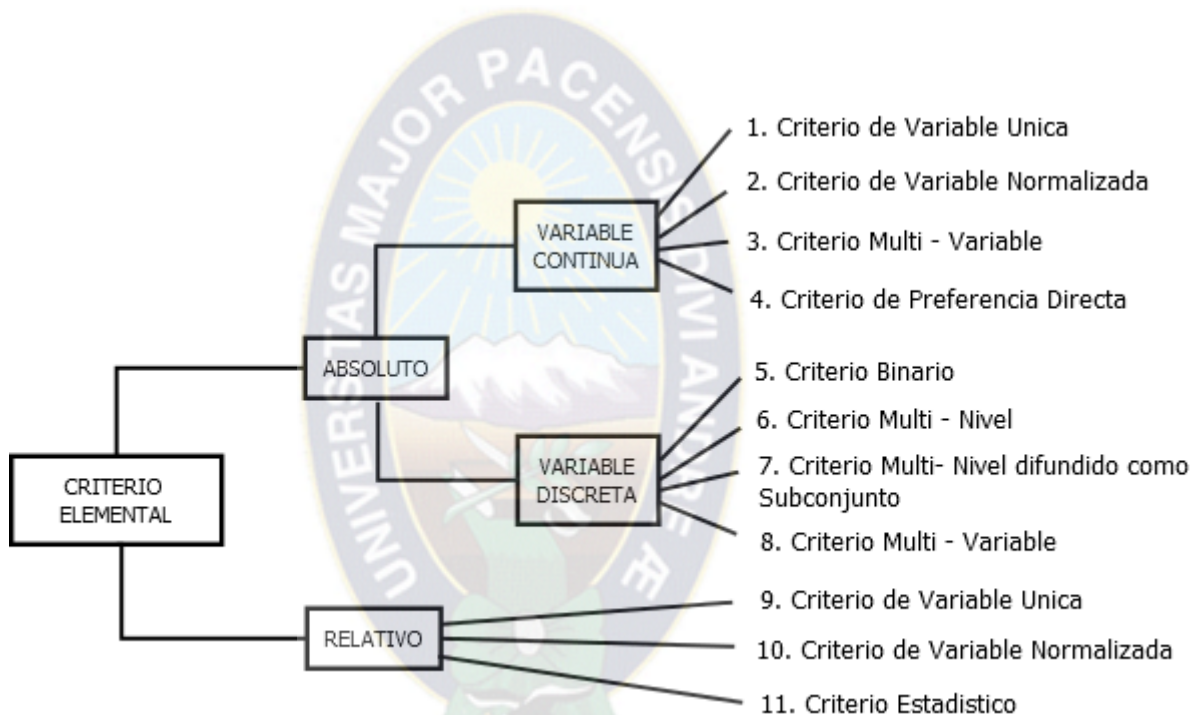


Figura 4.1 Tipos de Criterio elemental
Fuente: [OLSINA, 1999]

4.2.3.1 Criterio elemental absoluto con variables continuas

Criterio de variable única

En un criterio elemental común. Se asume que la variable X es única y continua, como por ejemplo, el tiempo medio entre dos fallas; el tiempo total transcurrido de un programa de prueba, el tiempo activo de un microprocesador durante una prueba, etc.

Con el fin de determinar el criterio elemental, el primer paso consiste en definir el rango de valores de interés para la evaluación de la variable continua. El siguiente paso, consiste en determinar las coordenadas de los puntos más relevantes y sus preferencias de calidad.

Criterio de variable normalizada

Este es un criterio elemental que se suele utilizar para evaluar la relación entre dos atributos con criterio absoluto de un mismo sistema definido por:

$$U_{mi} = X_i / X_j$$

Donde $X_i = E$ de puntos máximos y $X_j = E$ de atributos con puntaje obtenido.

Criterio de multi-variable continuas.

En este tipo de criterio, la variable X es restante de algunas otras variables y constantes (el valor de X corresponderá una métrica indirecta)

Criterio de preferencia de calidad directa

Este tipo de criterio es subjetivo y basado en la experiencia y criterios de los evaluadores. Desde el punto de vista de la precisión y objetividad, es el peor criterio, debido a que pueden introducir errores de valoración intencionales e involuntarios.

No obstante, dentro de los requerimientos algunos atributos solo podrán comportarse de un modo subjetivo, a partir del juicio de evaluadores expertos. Es decir, puede ser difícil y costoso modelar la descomposición del atributo para

determinar la preferencia de calidad.

El criterio para la variable X se mapea en una preferencia trivial cuyas coordenadas son:

$$\text{CrE}(X_i) = \{(0,0), (100,100)\}$$

Donde un valor de $X_i = 0$, se interpreta como la ausencia del atributo de calidad; en cambio un valor de $X_i = 1$, se interpreta como la presencia o disponibilidad del mismo.

Criterio de multi-nivel

Este criterio es una generalización del criterio binario. La variable discreta puede tomar más de dos valores, cada uno de los cuales se corresponde a una preferencia de calidad.

$$\text{CrE}(X_i) = \{(0,0), (1,60), (2,100)\}$$

Donde un valor de $X_i = 0$, se interpreta como la ausencia del atributo de calidad en cambio un valor de $X_i=1$, se interpreta como la presencia parcial de la versión solo texto; y finalmente un valor $X_i=2$ se interpreta como la presencia total de la versión solo texto para todo el sitio Web.

Criterio de multi-nivel definido como subconjunto

Este criterio es uno multi-nivel definido como un subconjunto de los números naturales (es una escala estrictamente ordinal). La variable discreta puede tomar más de dos valores, cada uno de los cuales corresponde a una preferencia de calidad

$$\text{CrE}(X_i) = \{(0,0), (1,60), (2,100)\}$$

En donde el listado de valores para X_i es como sigue:

0 = ausencia del mecanismo de búsqueda restringida

1 = búsqueda básica: mecanismo de búsqueda por nombre / apellido

2 = 1 + búsqueda extendida o avanzada: mecanismo de búsqueda por unidad académica y disciplinas o materia.

Criterio de multi-variables discretas

Este criterio permite agrupar varias variables discretas y modelar el resultado en una única variable X . de este modo se puede reducir la cantidad de criterios elementales. Sea el conjunto de variables discretas D_1, \dots, D_m , entonces se puede definir una variable compuesta X , también discreta como función de las anteriores.

$$X = F(D_1, \dots, D_n) \text{ y } E_e\{X_1, \dots, X_n\}$$

4.2.4 Fase de definición e implementación de la evaluación global

Trata con actividades modelos y herramientas para determinar los criterios de agregación de las preferencias de calidad elemental para producir la preferencia global, para cada sistema seleccionado.

Se consideran tipos de funciones de agregación para modelar diferentes relaciones entre atributos y características.

Para caracterizar el sistema a evaluar y comparar se identifican n atributos necesarios cuya preferencia o indicador elemental (EI) se debe computar y los valores individuales de IE_1, \dots, IE_n , están normalizados de manera que: $0 \leq IE_i \leq 100\%$ además que las características están asociadas a pasos definidos como P_i .

En este caso se expresa el indicador o preferencia global (IG) mediante el uso de una sumatoria:

$$IP = P_1IE_1 + \dots + P_nIE_n \text{ para } 0 \leq IE_i \leq 1$$

Donde $0 \leq P_i \leq 1$, para $i=1 \dots n$ y $P_1 + \dots + P_n = 1$

Tanto los puntajes excedentes, como el indicador de calidad global están un uno de los tres niveles de aceptabilidad

Insatisfecho	De 0 a 40 %
Marginal	de 40 a 60 %
Satisfecho	de 60 a 100 %

Tabla 4.2 Rango de Medición
Fuente [OLSINA, 1999]

Evaluación elemental de calidad de software WEB SITEQEM

Título: Búsqueda de registro de proyectos

Tipo: Atributo

Características de alto nivel: Funcionalidad

Súper – característica: Orientación

Definición y comentarios: algunas veces, tiene sentido brindar el usuario con la facilidad de búsqueda restringida a un subsitio o parte de un sitio, debido a que el

mismo es altamente cohesivo o distinto del resto de la información del sitio web global.

Tipo de criterio elemental: es un criterio multi-nivel, discreto y absoluto, definido como subconjunto. Podemos decir que 0=no disponible mecanismo de búsqueda registrada; 1 = búsqueda básica, 2=1+ búsqueda extendida o avanzada: mecanismo de búsqueda (con filtro).

Escala de preferencias: 0 a 40% mala, 40 a 60% regular, 60 a 100 buena.

Tipo de recolección de datos: Observacional

Los criterios usados son los siguientes:

*CVN: $IE = (X/Y) * 100$ con $X = \sum$ puntajes máximos, y $Y = \sum$ Puntaje obtenidos*

*CV: $IE = (X/Y) * 100$ con $X =$ cantidad total de datos para la variable y $Y =$ cantidad total*

CB: $IE = 0$ si no existe, $IE = 1$ si existe

CPD: Sujeto a la objetividad del observador

CMN: $IE = 0 = 1$ si existe

CPD: Sujeto a la objetividad del observador

CMN: $IE = 0 = 0$ ausente

CMN = 1 = 60 presente parcial

CMN: $IE = 2 = 100$ presente

Antes de implementar la evaluación global dividiremos por criterios

CODIGO	ATRIBUTO	CRITERIO ELEMENTAL	IEi (%)
1	USABILIDAD	CVN	95.8
1.1	Comprensibilidad global del sitio	CVN	100
1.1.1	Esquema de organización global	CVN	100
1.1.1.1	Mata del sitio	CB	1≈100
1.1.1.2	Menú de contenidos	CB	1≈100
1.2	Mecanismos de ayuda y retroalimentación en línea	CVN	95
1.2.1	Calidad de ayuda	CVN	95
1.2.1.1	Ayuda explicando orientación al usuario	CPD	90
1.2.1.2	Ayuda de la búsqueda	CPD	100
1.2.2	Indicador de última actualización	CVN	100
1.2.2.1	Global todo el sitio web	CMN	2≈100
1.2.3	Retroalimentación	CVN	90
1.2.3.1	Formulario de entrada	CPN	90
1.2.3.2	Reportes	CPN	90
1.3	Aspectos de interfaces y estéticos	CVN	92,5
1.3.1	Cohesividad al agrupar los objetos de control principales	CPN	90
1.3.2	Permanencia y estabilidad en la presencia de los controles principales	CVN	90
1.3.2.1	Permanencia de controles directos	CPN	90
1.3.2.2	Permanencia de controles internos	CPN	90
1.3.3	Aspectos de estilos	CVN	100
1.3.2.1	Uniformidad en el color de enlaces	CMN	2≈100
1.3.2.2	Uniformidad en el sitio global	CMN	2≈100
1.3.3.3	Guía de estilo global	CMN	2≈100
1.3.4	Preferencia estética	CPN	90
1.4	Misceláneas	CVN	66.67
1.4.1	Soporte de lenguaje extranjero	CB	0≈0
1.4.2	Atributo "que es lo bueno"	CMN	2≈100
1.4.3	Indicador de resolución de pantalla	CB	1≈100

Tabla 4.3 Resultado del criterio de Usabilidad

Fuente [Elaboración propia]

De acuerdo a la comprensibilidad del sitio, mecanismos de ayuda para el usuario y aspectos de la interfaz gráfica del sistema, los niveles de indicadores global del criterio USABILIDAD tiene un porcentaje del 95.8% y está dentro del margen satisfactorio.

CODIGO	ATRIBUTO	CRITERIO ELEMENTAL	IEi (%)
2	FUNCIONALIDAD	CVN	90.5
2.1	Aspectos de búsquedas y recuperaciones	CVN	100
2.1.1	Mecanismos de búsquedas en el sitio web	CVN	100
2.1.1.1	Búsqueda restringida	CVN	100
2.1.1.1.1	De productos funciones (nombre, descripción)	CVN	100
2.1.1.1.2	De clientes	CB	1≈100
2.1.1.2	Búsqueda global	CB	1≈100
2.2	Aspectos de navegación y exploración	CMN	1≈100
2.2.1	Navegabilidad	CVN	76.6
2.2.1.1	Orientación	CVN	100
2.2.1.1.1	Indicador del camino	CVN	100
2.2.1.1.2	Etiqueta de posición actual	CB	1≈100
2.2.1.2	Promedio de enlaces por pagina	CB	1≈100
2.2.2	Objetos de control de navegación	CMN	1≈100
2.2.2.1	Nivel de desplazamiento	CVN	50
2.2.2.1.1	Desplazamiento vertical	CVN	50
2.2.2.1.2	Desplazamiento horizontal	CB	0≈0
2.2.3	Predicción navegacional	CB	1≈100
2.2.3.1	Enlace con titulo	CVN	80
2.2.3.2	Calidad de frase del enlace	CMN	2≈100
2.3	Aspecto del dominio orientado al usuario	CMN	1≈60
2.3.1	Relevancia del contenido	CVN	95

2.3.1.1	Información del contenido	CVN	95
2.3.1.1.1	Información del funcionamiento	CVN	100
2.3.1.1.2	Listado de funciones	CB	1≈100
2.3.1.1.2	Información de estudio, bajas altas, etc.	CB	1≈100
2.3.1.2	Información del estado actual del funcionamiento	CVN	90
2.3.1.2.1	Datos personales	CMN	2≈100
2.3.1.2.2	Descripción de reportes	CMN	1≈60

Tabla 4.4 Resultado del criterio de Funcionalidad
Fuente [Elaboración propia]

De acuerdo a la búsqueda de productos o navegación y exploración del sistema, los niveles de indicador global el criterio FUNCIONALIDAD tiene un porcentaje del 90.5% y está dentro del margen satisfactorio

CODIGO	ATRIBUTO	CRITERIO ELEMETAL	IEi (%)
3	CONFIABIIDAD	CVN	90
3.1	No deficiencia	CVN	90
3.1.1	Errores de enlaces	CVN	100
3.1.1.1	Enlaces rotos	CMN	2≈100
3.1.1.2	Enlaces inválidos	CMN	2≈100
3.1.1.3	Enlaces no implementados	CMN	2≈100
3.1.2	Errores no implementados	CVN	80
3.1.2.1	Deficiencias o cualidades ausentes debido a diferentes navegadores (Browser)	CMN	2≈100
3.1.2.2	Diferencias o resultados inesperados independientes de Browser (ej. Error de búsquedas imprevistos diferencias con macros, frames)	CMN	1≈60
3.1.2.3	Nodos destinados (inesperados en construcción)	CMN	1≈60

3.1.2.4	Nodos muertos (sin enlace de retorno)	CMN	2≈100
---------	---------------------------------------	-----	-------

Tabla 4.5 Resultado del criterio de Confiabilidad
Fuente [Elaboración propia]

De acuerdo al soporte de diferentes navegadores y no hay errores de conexión de los diferentes páginas, los niveles de indicador global el criterio CONFIABILIDAD tiene un porcentaje del 90% y está dentro del margen satisfactorio.

CODIGO	ATRIBUTO	CRITERIO ELEMENTAL	IEi (%)
4	EFICIENCIA	CVN	90
4.1	Performance	CVN	90
4.1.1	Paginas de acceso rápido	CPD	90
4.2	Accesibilidad	CVN	90
4.2.1	Accesibilidad de información	CVN	80
4.2.1.1	Soporte versión solo texto	CB	1≈60
4.2.1.2	Legitimidad al desactivar propiedad imagen Browser	CVN	100
4.2.1.2.1	Imagen con titulo	CB	1≈100
4.2.1.2.2	Legitimidad global	CB	1≈100
4.2.2	Accesibilidad de	CMN	2≈100

Tabla 4.6 Resultado del criterio de Eficiencia
Fuente [Elaboración propia]

De acuerdo a la velocidad de carga de las diferentes páginas a imágenes de los proyectos, los niveles de indicador global el criterio EFICIENCIA tiene un 90% y está dentro del margen satisfactorio.

Criterio global

CRITERIO	IE (%)
USABILIDAD	95.8
FUNCIONABILIDAD	90.56
CONFIABILIDAD	90
EFICIENCIA	90
CALIDAD GLOBAL	91.5

Tabla 4.7 Resumen de resultados obtenidos

Fuente [Elaboración propia]

De acuerdo con la valoración de la calidad global del sitio web, aplicando la metodología Web Site QEM el valor de calidad global total del sistema es de 91.5% ya que de 20 persona que utilizaron el sistema 18 están conformes y satisfechos con el mismo.





CAPITULO v

*Estudio de costos y
beneficios*

EVALUACION DE COSTOS Y BENEFICIOS

5.1 Introducción

El propósito, es mostrar a los usuarios el nuevo software, al igual que otros grupos de administradores de la organización que los beneficios que se espera obtener con el nuevo sistema superan los costos superados.

En las secciones siguientes examinaremos diversos aspectos de los cálculos de costo/beneficio:

- Análisis de costo
- Análisis de beneficio

5.2 Análisis de costos

Se deben calcular todos los costos anticipados asociados con el sistema. Para determinar el costo total del proyecto se tomaran en cuenta los siguientes costos.

- Costo del sistema desarrollado
- Costo de la implementación del sistema
- Costo de la elaboración del proyecto

5.2.1 Costo de sistema desarrollado

Para determinar el costo del software desarrollado se utiliza el modelo constructivo COCOMO II, orientado a los puntos de función.

Estimación de puntos función:

Parámetro de Medición	Cuenta	Factores de ponderación			Total
		simple	medio	complejo	
Nro. de entradas de usuario	5 *	3	4	6	= 15
Nro. de salidas de usuario	5 *	4	5	7	= 20
Nro. de Peticiones de usuario	5 *	3	4	6	= 30
Nro. de archivos	5 *	7	10	15	= 50
Nro. de interfaces externas	1 *	3	4	6	= 4
Cuenta total					119

Tabla 5.1 Cálculo de punto función no ajustado

Fuente: [Elaboración propia]

Cálculo de valores de ajuste de la complejidad, se determina la complejidad y se obtiene los siguientes resultados, será la base para calcular el costo.

$$\text{Factor de Ajuste} = (0.65 + 0.01 * 51)$$

$$\text{Factor de ajuste} = 1.16$$

El cálculo del punto función se basa en la fórmula

$$PF = 119 * 1.16 = 138.04$$

Conversión de los puntos función a KDLC

Ahora convertimos los PF a miles de líneas de código. Para ello veremos la tabla 5.2

LENGUAJE	NIVEL	FACTOR LCD/PF
JAVA	6	53
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Tabla 5.2 Conversión de punto función a KDLC

Fuente: [Pressman, 2005]

$$\text{LCD} = \text{PF} * \text{Factor LCD/PF}$$

$$\text{LCD} = 138.04 * 29$$

$$\text{LCD} = 4003.16$$

$$\text{KLCD} = 4$$

Aplicando la fórmula de esfuerzo, tiempo calendario y personal requerido.

La ecuación de COCOMO II tiene la siguiente forma:

$$E = a * (\text{KLCD})^b$$

$$D = c (E)^d$$

Dónde:

E: Esfuerzo aplicado en personas por mes

D: Tiempo de desarrollo en meses

KLDC: Número estimado de líneas de código distribuido (en miles)

PROYECTO DE SOFTWARE	A	B	C	D
Orgánico	2.4	1.05	2.5	0.38
Semi-acoplado	3	1.12	2.5	0.35
Empotrado	3.6	1.2	2.5	0.32

Tabla 5.3 Coeficientes a y b

Fuente:[Elaboración propia]

En la tabla 5.3, se muestra los tipos de proyecto. Como este es proyecto intermedio en tamaño y complejidad se elige semi-acoplado.

$$E = 3 * (4)^{1.12}$$

$$E = 14.17$$

Redondeando:

$$E = 14$$

$$D = 2.5 * (14)^{0.35}$$

$$D = 6.29$$

Redondeando

$$D = 6 \text{ meses}$$

El personal requerido se obtiene con la siguiente fórmula:

$$\text{Numero Programadores} = E/D = 2 \text{ personas}$$

El salario de un programador aproximadamente es de 500\$US a 800\$US en nuestro caso se tomara el promedio con un valor de 650\$US, cifra que se tomara en cuenta para la estimación siguiente

$$\text{Costo mensual de desarrolladores} = \text{Número de programadores} * \text{salario promedio}$$

$$\text{Costo mensual de desarrolladores} = 2 * 650\$US$$

$$\text{Costo mensual de desarrolladores} = 1.300\$US$$

Como el desarrollo de software se lo estima en 6 meses tenemos:

$$\text{Costo total de desarrollo} = \text{Costo del Software por mes} * \text{Numero de meses}$$

$$\text{Costo total de desarrollo} = 1.300\$US * 6 \text{ meses}$$

$$\text{Costo total de desarrollo} = 7.800 \$US$$

5.2.2 Costo de implementación del proyecto

La Agencia Estatal de Vivienda cuenta con equipos y servidores donde se realizará la instalación del sistema, por lo que el costo de implementación es cero.

5.2.3 Costo de elaboración del proyecto

Se refiere a los costos de estudios del sistema, se presenta en la tabla 5.4

DESCRIPCION	COSTO TOTAL(\$US)
Análisis y diseño del proyecto	500
Material de escritorio	90
Internet	80
Otros	50
TOTAL	720

Tabla 5.4 Costo de elaboración del proyecto
Fuente: [Elaboración Propia]

5.2.4 Costo total

El costo es la suma del costo de software de desarrollo y el costo de elaboración del proyecto que se puede observar en la siguiente tabla

DESCRIPCION	COSTO TOTAL(\$US)
Costo de software de desarrollo	7.800
Costo de implementación	0
Costo de elaboración de proyecto	720
TOTAL	8.520

Tabla 5.5 Costo total del proyecto
Fuente: [Elaboración Propia]

5.3 Análisis de Beneficios

Los beneficios del proyecto se calcularán en base a su rentabilidad, utilizando los siguientes coeficientes:

- **Valor Actual Neto (VAN)** es el valor de beneficios que se recibirá al final del proyecto. Cuando el resultado es mayor que cero significa que el proyecto es conveniente
- **Costo/Beneficio(C/B)**, al dividir los beneficios entre los costos actualizados el resultado puede tomar dos valores: menor a uno indica que se tienen pérdidas económicas reales y mayor a uno indica que se tiene ganancias
- **Tasa Interna de Retorno (TIR)**, es la tasa interna de retorno que permite conocer el porcentaje que rinde la inversión y se puede comparar si es conveniente invertir en el proyecto. El resultado mayor a cero significa que el proyecto es conveniente. El resultado menor a cero significa que no es conveniente

5.3.1 Valor actual neto (VAN)

La fórmula para el cálculo del VAN está dada por:

$$VAN = \sum_{t=0}^n \frac{Bt}{(1+i)^t} - \sum_{t=0}^n \frac{Gt}{(1+i)^t}$$

Donde:

t: Período

Bt: Ingreso generado durante el periodo *t*

i: Tasa de interés correspondiente al periodo *t*

n: Número de períodos

Gt: Costos exigidos durante el periodo *t*

Tomando el interés del 3% equivalente al interés otorgado en un banco por caja de ahorros, se consigue los siguientes datos, representados en la tabla 5.6

VAN	Año 1	Año 2	Año 3	Año 4	Total
Ahorro	0.00	5500.00	8500.00	9500.00	23500.00
VAN beneficio	0.00	5184.28	12962.98	21403.61	39550.87
Gasto	9430.00	500.00	200.00	200.00	10330.00
VAN gasto	9155.34	9626.64	9809.67	9987.36	38579.01
VAN acumulado	-9155.34	-4442.36	3153.32	11416.25	971.86

Tabla 5.6 Cálculo del VAN Acumulado

Fuente: [Elaboración Propia]

Por tanto se puede decir que el valor acumulado hoy de los que esperamos recibir del sistema en cinco años, es de 39550 \$US. Otra interpretación puede ser que es de conveniencia realizar el proyecto en lugar de invertir ese dinero en el banco, ya que se obtendrá una mejor ganancia

5.3.2 Costo/Beneficio (C/B)

El valor costo beneficio entonces se calcula de la siguiente forma:

$$\frac{C}{B} = \frac{\sum_{t=0}^n \frac{Bt}{(1+i)^t}}{\sum_{t=0}^n \frac{Ct}{(1+i)^t}}$$

$$\frac{C}{B} = \frac{39550.87}{3857901} = 1.2$$

Esto significa que cada dólar invertido en el proyecto será recuperado, y a la vez se tiene una ganancia por dólar de 0.2 centavos.

5.3.3 Tasa Interna de Retorno TIR

Ahora, calculamos el interés que hubiese sido necesario para obtener el VAN Beneficio, de haber depositado el dinero en el banco, en otras palabras la TIR, tenemos:

$$\sum_{t=0}^n \frac{Bt}{(1+i)^t} = VAN \text{ Beneficio}$$

$$TIR = 0.11$$

Es decir, la tasa de interés necesaria para lograr el beneficio actual de un depósito bancario hubiere sido de 11%.

El costo es la suma del costo de software de desarrollo y el costo de elaboración del proyecto llega a 8.520 \$US, para una Institución que desembolsa mensualmente entre 60 y 120 millones de Bs., contar con una herramienta que ayude a tener un control de su ejecución física y financiera, ayude a la toma de decisiones gerenciales a nivel departamental y nacional; cumple con el objetivo al ser el costo de proyecto mínimo en relación al tipo de inversiones que se realizan, además el mantenimiento del software es mínimo ya que todos sus servicios están basados en software libre que no incluye pago de licencias.



CAPITULO VI
SEGURIDAD

SEGURIDAD

6.1 Algoritmo MD5

Este algoritmo fue desarrollado por Ronald Rivest en 1995 está basado en dos algoritmos anteriores MD2 y MD4. MD5 comienza rellorando el mensaje a una longitud congruente con $448 \bmod 512$, es decir la longitud del mensaje es de 64 bits, el relleno empieza con un 1, seguido de tantos 0 como sean necesarios.

La codificación del MD5 de 128 bits es representada como un número de 32 dígitos hexadecimal. El siguiente código de 28 bytes ASCII será tratado con MD5 y veremos su correspondiente salida.

6.2 Algoritmo

Terminologías y notaciones, Se entenderá como palabra a una entidad de 32 bits y un byte tiene 8 bits, una secuencia de bits puede ser interpretada de manera natural como una secuencia de bytes, donde cada grupo consecutivo de ocho bits se interpreta como un byte con el bit más significativo al principio. Similarmente, una secuencia de byte puede ser interpretada como una secuencia de 32 bits (palabra), donde cada grupo consecutivo de cuatro bytes se interpreta como un apalabra en la que el byte menos significativo esta al principio.

Descripción del algoritmo MD5, Empecemos suponiendo que tenemos un mensaje de 'b' bits de entrada, y que nos gustaría encontrar su resumen. Aquí 'b' es un valor arbitrario entero no negativo, pero puede ser cero, no tiene por qué ser múltiplo de ocho, y puede ser muy largo. Imaginemos los bits del mensaje escrito así:

$$m_0, m_1, m_2 \dots m_{b-1}$$

Los siguientes cinco pasos son efectuados para calcular el resumen del mensaje:

Paso 1. Añadiendo Bits. El mensaje será extendido hasta que su longitud en bits sea congruente con $448 \bmod 512$. Esto es, si se le resta 448 a la longitud del mensaje tras este paso se obtiene un múltiplo de 512. Esta extensión se realiza siempre, la extensión se realiza como sigue: un solo bit "1" se añade al mensaje, y después bits "0" se añaden hasta que la longitud en bits del mensaje extendido se haga congruente con $448 \bmod 512$.

Paso 2. Longitud del Mensaje. Un entero de 64 bits que representa la longitud 'b' del mensaje, se concatena al resultado del mensaje anterior. En el supuesto de 'b' sea mayor que 2^{64} , entonces solo 64 bits de menos paso que 'b' se usarán.

Paso3. Inicializar el Buffer MD. Un buffer de cuatro palabras (A,B,C,D) se usa para calcular el resumen del mensaje aquí cada una de las letras A,B,C,D representa un registro de 32 bits. Estos registros se inicializan con los siguientes valores hexadecimales, los bits de mejor peso primero.

Paso 4. Procesando el Mensaje en bloque de 16 palabras, Primero definimos cuatro funciones que toman como entrada tres palabras de 32 bits y su salida es una palabra de 32 bits.

$$F(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$G(X, Y, Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$$

$$H(X, Y, Z) = X \ominus Y \ominus Z$$

$$I(X, Y, Z) = Y \ominus (X \vee \neg Z)$$

En cada posición de cada bit F actúa una condicional: si X, entonces Y sino Z la función F podría haber sido definida usando + en lugar de v ya que XY y notXnunca tendrán unos en la misma posición de bit.

Paso 5. Salida. El resumen del mensaje es la salida producida por A,B,C, y D. Se comienza el byte de menor peso de A y se acaba con el byte de mayor peso D.

6.3 Aplicación del algoritmo md5

En PHP la codificación para ingresar al password al sistema sería de la siguiente manera:

```
$this->password = md5( $_POST['password'] );
```

6.4 Archivos de Log

Los archivos Log se encargan de registrar los eventos que ocurren en el sistema, aplicaciones incluidas como se r: registros, modificaciones y bajas junto a ellos la fecha y hora de realización de sus eventos.

La importancia de crearlos está en poder realizar un rastreo de alguna acción que probablemente afecta el normal funcionamiento del sistema, proveyéndole así un grado de seguridad más.

La implementación del presente proyecto se tomo esta medida de seguridad por el continuo crecimiento de esta tabla, se recomienda revisas mensualmente y dar de baja los archivos que no signifiquen riesgo para no saturar la base de datos.

6.5 Seguridad del sistema

Los problemas de seguridad de este sistema pueden venir de las herramientas que se utilizaron en su desarrollo o pueden ser una falla en el diseño lógico, a menudo es la segunda falla la que ocasiona fallas en el funcionamiento del sistema.

6.5.1 Amenazas.

Existen distintas amenazas las cuales son:

- Ingreso a usuarios no validos
- Control de acceso roto

- Administración de sección y autenticación rota
- Desbordamiento de buffer
- Inyección de código
- Manejo inadecuado de errores
- Almacenamiento inseguro
- Administración de configuración insegura

6.5.2 Guías de seguridad

Estos son algunos principios de seguridad para el diseño del sistema:

- Validar todas las entradas y salidas
- Mantener un esquema de seguridad simple
- Manejar las fallas y errores de forma adecuada
- Utilizar componentes solo de confianza
- Controlar las excepciones

6.5.3 Tipos de seguridad para sistemas WEB

6.5.3.1 Seguridad en el cliente

Uno de los mecanismos de seguridad que se implementa son las validaciones por el lado del cliente.

Existen mecanismos de validación provistas por la herramientas que utilizamos para hacer la aplicación, en el caos de PHP se tienen los controles de validación para la información introducida por el cliente, estas validaciones son realizadas antes de que la información introducida llegue al servidor, esto evita que se envíen datos incorrectos al servidor además se ahorra tiempo ya que si la información es incorrecta simplemente no se envía al servidor.

6.5.3.2 Seguridad en el servidor

La validación del lado del cliente no es suficiente, también debe realizarse otro tipo de controles por el lado del servidor, ya sea del servidor de aplicaciones o el servidor de base de datos.

➤ Seguridad en el servidor de aplicaciones

Un servidor de aplicaciones proporciona muchos servicios y no todos son necesarios para el funcionamiento del sistema

Es conveniente deshabilitar lo que no se necesite, para ello debe configurarse adecuadamente el servidor de aplicaciones.

➤ Seguridad en el servidor de base de datos

Existen muchos problemas a nivel de base de datos uno de ellos es la inyección SQL (Lenguaje de consulta estructurada), son los ataques realizados contra base de datos.

En este caso un usuario utiliza debilidades en el diseño de la base de datos o del sistema para extraer información o más aun para manipular información dentro de la base de datos. Una forma de evitar este tipo de debilidades es restringiendo los caracteres que el usuario introduce, por ejemplo: las comillas, los puntos y coma y otros

También está el acceso no autorizado a la información, esto podría solucionarse asignado correctamente los roles que tiene cada usuario en el sistema. El acceso de los usuarios a los formularios es cargado desde la base de datos, cuando un usuario se autentifica el sistema le da acceso solo a los formularios asignados de acuerdo al nivel que tenga asignado el usuario.

6.5.3.3 Seguridad en la comunicación

SSL es un protocolo y diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TC/IP y de los protocolos HTTP, FTP, SMTP. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo descifrado simétrico, típicamente el RC4 o IDEA y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción lo cual permite que aunque sea reventada por un atacante en una transacción dada no sirva para descifrar futuras transacciones. MD5 se utiliza como algoritmo de hash.

Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y opcionalmente autenticación de cliente para conexiones TC/IP.

Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado protocolo SSL, Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el protocolo el SSL record y el puerto abierto para comunicarse de forma segura con el cliente.

6.5.3.4 Seguridad en la aplicación

El control de acceso en los usuarios es una parte fundamental para un sistema.

- Autenticación, determina si un usuario es quien dice ser. Autenticación HTTP básica cuando se quiere ingresar a un formulario protegido el servidor devuelve en código "HTTP/1.1 401 Authentication required". El cliente debe enviar sus datos al servidor.
- Autenticación basada en aplicación, en este caso la aplicación implementa su propio mecanismo de autenticación. Es más costosa pero es más flexible porque permite establecer diferentes permisos y niveles de acceso asignados al usuario.
- Passwords se recomienda restringir los valores para los nombres de los usuarios, almacenar los passwords de forma segura protegiendo el

acceso de la base de datos. Bloquear una cuenta cuando se detecta un número determinado de accesos incorrectos. Tener una política de recuperación de passwords en caso de olvido por parte del usuario.

- Sesiones, después de que el usuario se a autenticado se debe mantener esa autenticación en cada conexión subsiguiente. Para esto se utiliza las variables de sesión que permiten mantener el estado entre las diferentes peticiones HTTP. El procedimiento es el siguiente, después de autenticarse el usuario recibe un identificador de sesión este identificador es invisible y lo acompaña en cada petición. Este identificador se almacena en la máquina del cliente, mediante una cookie.





CAPITULO VII

*CONCLUSIONES Y
RECOMENDACIONES*

CONCLUSIONES Y RECOMENDACIONES

7.1 Conclusiones.

La culminación del presente proyecto y conforme a las actividades definidas para el análisis e implementación del cuadro de mando integral para la Agencia Estatal de Vivienda- AEVIVIENDA.

Con la implementación del sistema ahora es posible obtener información pertinente.

- Se realizó un Estudio a la teoría del Cuadro de Mando Integral para fundamentar la modelación del sistema de Control de Gestión para las departamentales de la Agencia Estatal de Vivienda a efectos de medir los objetivos estratégicos.
- Se integraron los datos y registros de una manera automática para un control eficiente y una buena coordinación de los proyectos y actividades que se desarrolla.
- Se logró realizar un control, seguimiento y monitoreo en tiempo real a la ejecución física y financiera de los proyectos en la Agencia Estatal de Vivienda.
- Se logró canalizar la información por el Cuadro de Mando Integral para que exista un solo medio el cual brinde la información en línea.
- Se desarrolló una plataforma web basados en los modelos de estándares web.
- Se desarrolló un Sistema Gerencial de Cuadro de Mando Integral para realizar el control, seguimiento y monitoreo a la ejecución física y financiera de proyectos que ayuden a la toma de decisiones gerenciales de la Agencia Estatal de Vivienda.

7.2 Recomendaciones.

Para ampliar el presente proyecto de grado se hace las siguientes recomendaciones:

- ✓ Desarrollar el módulo de georeferenciación del proyecto para visualizar en un mapa la intervención en el territorio nacional el impacto de la construcción de viviendas por parte de la Agencia Estatal de Vivienda.
- ✓ Desarrollar el módulo de registro y control de beneficiarios a nivel nacional para evitar el doble beneficio,
- ✓ Desarrollar un sistema de archivo digital relacionada a los proyectos, con información digitalizada de contratos, actas de aprobación de proyectos, pólizas, instructivos de desembolso y planillas de pago de avance obra, para contar con respaldos del proyecto.
- ✓ Incorporar normas y políticas de uso del sistema de acuerdo al siguiente detalle:
 - Generar los manuales técnicos tanto a nivel de instalación y configuración del Sistema, manual de Base de Datos y los manuales de usuario para todos los niveles de usuario que se tienen en el Sistema.
 - Definir tiempos periódicos y generar scripts para realizar copias de seguridad (backups) automáticos de la base de datos como del código fuente necesario para volver a levantar el sistema en casos de contingencias como caídas del servicio del Sistema en producción.
 - Documentar y manejar versionamientos del código fuente cuando se realicen mantenimientos y nuevas adecuaciones al Sistema.



BIBLIOGRAFIA

BIBLIOGRAFÍA

- [PRESSMAN, 2005] Pressman R., 2005, "Ingeniería del Software", 5ta. Edición, Editorial Concepción Fernández, Madrid.
- [OLSINA, 1999] OLSINA, D.L.(1999)."Métricas, Criterios y Estrategias para Evaluar Calidad" Argentina: Gidis, Facultad de Ingeniería, UNL Pan.
- [QUINTERO, 2005] Quintero, J. (2005). "Diseño de un modelo gerencial basado en la metodología del cuadro de mando integral para el Instituto Universitario Tecnológico de Ejido", Mérida, Venezuela: Universidad de losAndes.
- [VOGEL, 2004] Vogel, M. (2004). Reinventado los gobiernos con apoyo de los tableros de comando y control (parte 1), <http://www.tablero-decomando.com>
- [K&K, 2003] Kendall & Kendall, (2003). "Análisis y Diseño de Sistema", 3ra. Edición, Editorial Pearson Education.
- [VERDU, 1998] Verdú M., 1998, "Tesis de Doctorado",Facultad E. T. S. I. Telecomunicación,Universidad de Valladolid, Portugal.
- [ISPO, 1996] ISPO, (1996), "Information Society Project Office", Bruselas.

[TESIS, 2007]

Tesis y más, 27 de diciembre del 2007; 11:53 AM,
Tesis, Tesinas, Monográficos y **Páginas Web**

<http://tesisymas.blogspot.com/>

[Sabino C., 1994]

“Como Hacer Una Tesis” Ed. Caracas,

