

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



PROYECTO DE GRADO
"SISTEMA DE SEGUIMIENTO Y CONTROL DE NUTRICION
PARA CASO:
ZOOLOGICO MUNICIPAL DE LA PAZ"

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

AUTOR: Univ. José Elias Ochoa Piloy
TUTORA: Lic. Fátima Consuelo Dolz de Moreno M.Sc.
REVISOR: Lic. Marcelo Aruquipa Chambi

La Paz – Bolivia
2011

DEDICATORIA

En primer lugar a Dios por su infinito amor y comprensión incondicional hacia todos nosotros.

A mi mamá Eliodora, por estar en las buenas y en las malas, sin su apoyo hubiera resultado imposible haber concluido mis estudios superiores y otras etapas de mi vida.

A mi familia, que estuvo ahí para darme una mano cuando se los necesitaba.

A mi Hijo Daniel, quien fue y será la motivación para seguir los pasos que todavía me falta, HIJO TE LO DEDICO.

A mi nueva familia, que ahora nos une un ser amado en nuestras vidas.

José Elias Ochoa Piloy

AGRADECIMIENTOS

Para finalizar mi Proyecto de Grado, muchos fueron los obstáculos que existieron, pero con el apoyo moral y verbal de algunas personas, pude salir adelante, citare a algunas de ellas:

Primeramente, a Dios que permitió la finalización de mis estudios con un solo propósito que es la honra y gloria a El.

A mi familia que supo colaborarme y apoyarme en los momentos más difíciles, decirles sigan adelante y pronto ya les toca a Ustedes.

A mi tutora, Lic. Fátima Consuelo Dolz de Moreno M.Sc., que con su guía y orientación pude finalizar mi proyecto de grado, muchas gracias.

A mi revisor, Lic. Marcelo Aruquipa Chambi, por su colaboración y orientación con sus conocimientos pude llegar a esta etapa de la vida y además de poder brindarme su amistad muchas gracias.

Al MVZ. Fidel Fernández Anagua, por su apoyo desinteresado, muchas gracias por su desprendimiento intelectual para construir este proyecto y brindarme su amistad.

Al MVZ. Victor Humerez, por su paciencia y tiempo brindado para lograr este paso más que doy en la vida y además de poder brindarme su amistad, muchas gracias.

A las personas que desde el inicio de mis Pre-facultativos me dieron su apoyo incondicional para seguir la carrera que culmino hoy.

A mis amigos que estuvieron presentes en los momentos buenos y malos de mi vida, les agradezco por su paciencia y amistad desinteresada, muchas gracias muchach@s.

José Elías Ochoa Piloy

RESUMEN

La tecnología se apodero de gran magnitud en nuestra forma de vivir, en pleno siglo veinte es casi imposible no tener algún recurso tecnológico. La elaboración de sistemas informáticos hoy en día es necesaria porque agiliza nuestros procesos laborales, y en este medio social que es tan competitivo ya no es un lujo tener uno de ellos.

Abordar el tema de nutrición es enorme, en este trabajo vamos a tocar parte de ella que es la formulación de raciones alimentarias para los animales en cautiverio. La formulación de raciones implica conocer los requerimientos del organismo animal y la disponibilidad en los alimentos. Concuar esas necesidades de nutrientes en los animales y suministro en los alimentos es tarea que necesita sistematizar conocimientos científicos de nutrición. Este trabajo realizado integra los conocimientos científicos de nutrición como el tema de elaboración de sistemas informáticos.

El sistema almacena en la base de datos información clasificada en: alimentos, requerimientos mínimos de mantenimientos y requerimientos mínimos de mantenimientos por rango.

En este trabajo se desarrolla un Sistema de Seguimiento y Control de Nutrición para el Zoológico Municipal de La Paz, específicamente en el área de *Conservación y Manejo de Vida Silvestre*. El sistema de Seguimiento y Control de Nutrición ayuda a la elaboración de dietas alimentarias, tomando en cuenta como principal recurso los alimentos, la taxonomía del animal, peso vivo del animal, requerimientos mínimos para mantenimiento de: aves, mamíferos y reptiles. Una vez elaborada la dieta, el sistema permite realizar el seguimiento y/o aprovechamiento de alimentos de la dieta, para posteriormente anunciar en que condiciones nutricionales se encuentra cada clase de animal.

INDICE GENERAL

CAPITULO I

ANTECEDENTES

1.1 INTRODUCCION.....	1
1.2 ANTECEDENTES.....	2
1.2.1 ANTECEDENTES DE LA INSTITUCION.....	2
1.2.2 ANTECEDENTES DE TRABAJOS ANTERIORES.....	3
1.3 DEFINICION DEL PROBLEMA.....	4
1.4 PLANTEAMIENTO DEL PROBLEMA.....	4
1.5 OBJETIVOS.....	5
1.5.1 OBJETIVO GENERAL.....	5
1.5.2 OBJETIVOS ESPECIFICOS.....	5
1.6 LIMITES Y ALCANCES.....	5
1.6.1 LIMITES.....	5
1.6.2 ALCANCES.....	5
1.7 JUSTIFICACION.....	6
1.7.1 JUSTIFICACION TECNICA.....	6
1.7.2 JUSTIFICACION ECONOMICA.....	6
1.7.3 JUSTIFICACION SOCIAL.....	6
1.8 APORTES.....	6

CAPITULO II

MARCO TEORICO

2.1 INTRODUCCIÓN.....	7
2.1.1 MARCO INSTITUCIONAL.....	7
2.1.1.1 MISION.....	7
2.1.1.2 VISION.....	7
2.1.1.3 ORGANIGRAMA.....	7
2.2 METODOLOGÍA.....	9
2.2.1 RUP Y LOS MODELOS.....	9
2.2.2 FLUJOS DE TRABAJO.....	10
2.2.3 CICLO DE VIDA DEL PROCESO UNIFICADO.....	11

2.2.4 DIAGRAMAS UML.....	14
2.3 HERRAMIENTAS Y LENGUAJES DE DESARROLLO.....	17
2.3.1 HERRAMIENTAS DE CONSTRUCCIÓN DE SOFTWARE.....	17
2.3.1.1 FORMULARIOS WINDOWS FORMS.....	17
2.3.1.2 COMPATIBILIDAD CON XML.....	18
2.3.1.3 LENGUAJE DE PROGRAMACIÓN VISUAL C#.....	18
2.3.1.4 EL ENTORNO .NET FRAMEWORK.....	18
2.3.1.5 LENGUAJE C# Y .NET FRAMEWORK.....	20
2.3.2. GESTOR DE BASE DE DATOS SQL SERVER 2005.....	20
2.3.2.1 PLATAFORMA DE DATOS DE SQL SERVER.....	21
2.4 METRICAS DE CALIDAD DE SOFTWARE.....	21
2.4.1 GESTION DE LA SEGURIDAD.....	21
2.4.2 ANALISIS DE RIESGO DEL SOFTWARE.....	23
2.4.3 CONTROLES DE SEGURIDAD.....	23
2.4.3.1 ¿QUÉ ES CALIDAD?.....	24
2.4.3.2 CALIDAD CON RELACIÓN AL SOFTWARE.....	24
2.4.3.4 METRICAS DE CALIDAD.....	25
2.5 ESTIMACIÓN DE COSTOS.....	29
2.5.1 MODELO COCOMO.....	29

CAPITULO III

MARCO APLICATIVO

3.1 FASE DE INICIO	32
3.1.1 MODELADO DE NEGOCIO.....	32
3.1.1.1 MODELO GENERAL DE CASOS DE USO.....	33
3.1.1.2 MODELO DEL DOMINIO DEL NEGOCIO.....	34
3.1.1.2 IDENTIFICACIÓN DE LOS ACTORES.....	35
3.2 FASE DE ELABORACION.....	37
3.2.1 MODELADO DE REQUISITOS.....	37
3.2.1.1 MODELO DEL NEGOCIO.....	37
3.2.1.2 MODELO DE REQUISITOS.....	40
3.2.2 MODELO DE ANALISIS Y DISEÑO.....	45

3.2.2.1 MODELO FISICO.....	48
3.3 FASE DE CONSTRUCCION.....	49
3.3.1 INTERFASES DE USUARIO.....	49
CAPITULO IV	
CALIDAD DEL SOFTWARE	
4.1 CALIDAD DEL SOFWARE.....	60
4.1.1 FUNCIONALIDAD.....	60
4.1.2 FIABILIDAD.....	62
4.1.3 FACILIDAD DE MANTENIMIENTO.....	63
4.1.4 PORTABILIDAD.....	64
4.1.5 USABILIDAD.....	65
4.2 ANALISIS DE COSTOS.....	66
4.2.1 COSTOS DIRECTOS.....	67
4.2.1 COSTOS INDIRECTOS.....	69
CAPITULO V	
CONCLUSIONES Y RECOMENDACIONES	
5.1 CONCLUSIONES.....	70
5.2 RECOMENDACIONES.....	71
BIBLIOGRAFIA	
ANEXO A: ARBOL DE PROBLEMAS	
ANEXO B: ARBOL DE OBJETIVOS	
ANEXO C: MATRIZ DEL MARCO LOGICO	
ANEXO D: PRUEBA DEL SISTEMA	
ANEXO E: DIAGRAMA DE ACTIVIDADES	
ANEXO F::DIAGRAMA DE CASOS DE USO	
ANEXO G: DIAGRAMA DE ESTADO	
ANEXO E: DIAGRAMA DE SECUENCIA	

INDICE DE FIGURAS

Figura 2.1 Estructura Organizacional Zoológico Municipal de La Paz.....	8
Figura 2.2 Estructura Organizacional Conservación y Manejo de Vida Silvestre....	8
Figura 2.3 Proceso de desarrollo de Software.....	9
Figura 2.4 Flujos de Trabajo y modelos.....	11
Figura 2.5 Un ciclo con sus fases e iteraciones.....	11
Figura 3.1 Modelado de Negocio.....	32
Figura 3.2 Modelo General de Casos de Uso.....	33
Figura 3.3 Modelo del Dominio del Negocio.....	34
Figura 3.4 Diagrama de Actividades – Elaboración de hoja dieta.....	38
Figura 3.5 Diagrama de Casos de Uso – Elaboración de hoja dieta.....	41
Figura 3.6 Diagrama de estado – Elaboración de hoja dieta.....	45
Figura 3.7 Diagrama de Secuencia – Elaboración de hoja dieta.....	46
Figura 3.8 Diagrama de Clases SISECON.....	47
Figura 3.9 Modelo físico SISECON.....	48
Figura 3.10 Inicio de sesión.....	49
Figura 3.11 Ingreso de contraseña.....	50
Figura 3.12 Menú principal.....	50
Figura 3.13 Elaborar Hoja Dieta.....	51
Figura 3.14 Taxonomía de los animales.....	52
Figura 3.15 Requerimiento de energía calculada.....	53
Figura 3.16 Energía aportada por la dieta.....	54
Figura 3.17 Calculo de la energía aportada por la dieta.....	55
Figura 3.18 Modificar hoja dieta.....	56
Figura 3.19 Listado de animales.....	56
Figura 3.20 Adicionar animal.....	57
Figura 3.21 Modificar datos de animal.....	57
Figura 3.22 Lista de usuarios.....	58
Figura 3.23 Adicionar usuarios.....	58
Figura 3.24 Adicionar alimento.....	59

INDICE DE TABLAS

Tabla 2.1 Métrica para medir el factor de funcionalidad.....	26
Tabla 2.2 Métrica para medir el factor de funcionalidad.....	26
Tabla 2.3 Métrica para medir el factor de fiabilidad.....	27
Tabla 2.4 Métrica para medir el factor de fiabilidad.....	27
Tabla 2.5 Métrica para medir el factor de portabilidad.....	28
Tabla 2.6 Métrica para medir el factor de usabilidad.....	29
Tabla 2.7 Niveles de aplicación.....	31

CAPITULO I ANTECEDENTES

1.9 INTRODUCCION.

Hoy en día no se puede poner en duda que los medios informáticos cumplen un papel importante en los servicios ofrecidos por las diferentes organizaciones gubernamentales como privadas. La mayor parte de los servicios ofrecidos por el funcionario público del Gobierno Autónomo Municipal de La Paz, es ahora apoyada por un sistema informático, tanto como para la recepción de trámites, procesos de tramites e incluso en el tema de guía e información al ciudadano.

El Gobierno Autónomo Municipal de La Paz cuenta con unidades dependientes de ella, una es la Oficialía Mayor de Promoción Económica (OMPE) que tiene a cargo la responsabilidad de administrar el zoológico Municipal de La Paz. El zoológico alberga una diversa fauna de animales silvestres nativos del país como de otros países.

La formulación de raciones implica conocer los requerimientos del organismo animal y la disponibilidad en los alimentos. Integrar esas necesidades de nutrientes en los animales y suministro en los alimentos es tarea que necesita sistematizar conocimientos científicos de nutrición. Esta es la capacidad técnica que se requiere para obtener eficiencia biológica y económica en la producción y conservación del animal.

Uno de los principales objetivos del Zoológico Municipal, es el manejo y conservación de la fauna silvestre, de ahí nace la necesidad de contar con un profesional en el tema de nutrición, por consecuente el de utilizar herramientas y medios tecnológicos en apoyo a la elaboración de dietas alimentarias

Los animales silvestres realizan actividades físicas con mayor intensidad y otras no y si hablamos de las *aves* del zoológico en algunos casos su grado de actividad será menor de aquellas que viven en su medio hábitat, lo mismo ocurre con los *mamíferos* y *reptiles*. Si nos interiorizamos en el tema de nutrición alimentaria podemos controlar la cantidad exacta de la energía requerida para cada clase de animal.

Debemos saber que la alimentación es diversa en los animales, las cuales son clasificados de la siguiente manera: etapa de crecimiento, etapa de desarrollo y etapa adulta, cada etapa requiere diferente alimentación. En la actualidad el Zoológico Municipal de La Paz cuenta con veterinarios especializados en el área de conservación y manejo de vida silvestre.

1.10 ANTECEDENTES

1.2.1 ANTECEDENTES DE LA INSTITUCION

El zoológico Municipal de La Paz "VESTY PAKOS SOFRO", lleva el nombre del Señor Silvestre Pakos Sofro, en reconocimiento a su labor y gestión desinteresada para concretar el traslado del zoológico en la zona de Mallasa el 21 de septiembre de 1993. El Zoológico se sitúa aproximadamente a 25 km del centro de la Ciudad de La Paz, se encuentra a una altitud de 3250 m.s.n.m. y presenta una temperatura media de 17° C cuenta con una extensión de 22,4 hectáreas siendo el zoológico mas grande de Bolivia y el más alto del mundo.

Es un parque natural debajo de la sombra de la Muela del Diablo y rodeado por vistas hermosas de las montañas. El zoológico cuenta con jardines, un gran lago donde habitan patos de diversa variedad y peces, áreas de bosque ideal para disfrutar de un picnic campestre. Actualmente, alberga 80 especies de animales y más de 550 individuos, entre fauna nativa del país y fauna perteneciente a otros países.

El zoológico municipal de La Paz es una institución sin fines de lucro que promueve conservación, educación y recreación de la biodiversidad de la fauna regional y nacional. Se constituye en un centro de referencia nacional e internacional especializado en fauna andina.

Tiene por objetivos el transmitir la necesidad de conservación de la biodiversidad de la fauna y flora nacional promoviendo los valores culturales y programas de conservación de la fauna en cautiverio.

1.2.2 ANTECEDENTES DE TRABAJOS ANTERIORES

En cuanto a trabajos relacionados con el tema de nutrición realizada en la carrera de Informática-Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés, encontramos lo siguiente:

- ♦ Sistema de Información y Control para la Unidad de Nutrición y Alimentación Complementaria Escolar, Alejandro Callisaya, 2004 [T.10941].
Tiene por objetivo mejorar las tareas administrativas de nutrición de la UNACE implementando un sistema considerando dos materias primas que son la ración líquida y la ración sólida.
- ♦ Sistema de Programación Nutricional Infantil “SIPRONUTRIN” para el Hospital del Niño “Manuel Villarroel”, Flores Jacqueline [2008]
Este sistema esta encargado de colaborar proceso y análisis del trabajo técnico relacionado con el consumo de alimentos, determinando el valor nutricional, composición química que ellos representan en la dieta de acuerdo al tipo de patología.
- ♦ Sistema de Evaluación y Control y Seguimiento nutricional a niños menores de 5 años Municipio de La Paz, Emerson Vargas [2007].
Su objetivo final es verificar el estado de nutrición con sus respectivas prevalencias utilizando la metodología antropométrica, y se realiza mediante una fórmula que contiene edad, peso y talla.

Después de realizar un estudio bibliográfico a nivel internacional podemos decir que los países que cuentan y además hacen uso de un *sistema de control de dieta alimentaria* son:

En Sudamérica: Brasil, Argentina, Paraguay y Colombia.

En Norteamérica: Estados Unidos, Canadá

Estos países hacen uso de un programa software elaborado en EE.UU. y MEXICO llamado *Zootritión*, este programa ayuda a la elaboración de raciones alimentarias.

Características del Zootriton

- ♦ Zootriton es el programa informático para obtener la dieta ideal para cada especie de animal.
- ♦ Este software permite equilibrar el suministro de alimentos de acuerdo a estándares internacionales que garantizan el desarrollo natural de los animales.
- ♦ Realiza la comparación de la dieta elaborado por el nutricionista y la dieta mínima que debe cumplir la ración alimentaria.
- ♦ La licencia del software tiene un precio relativamente económico.

1.11 DEFINICION DEL PROBLEMA

La elaboración de las dietas de la fauna silvestre cuenta con los siguientes inconvenientes:

- No cuenta con un sistema software específico para la elaboración de dietas alimentarias.
- No cuenta con una de base de datos que almacene las características de los animales que alberga el zoológico.
- No cuenta con una de base de datos que almacene los valores nutricionales de cada alimento que se utiliza en la dieta.

La elaboración de la dieta alimentaria de los animales silvestres se realiza de forma manual y se realiza con tablas de composición de alimentos impresas en hoja de papel, para cada dieta hay que hacer una búsqueda manual y visual de los mismos corriendo el riesgo de cometer errores en la copia de los verdaderos valores nutricionales, además que el tiempo de elaboración de las hojas de dietas alimentarias toma demasiado tiempo.

1.12 PLANTEAMIENTO DEL PROBLEMA

¿El sistema de Seguimiento y Control de Nutrición optimizará el control y elaboración de dietas alimentarias para los animales silvestres del Zoológico Municipal?

1.13 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Desarrollar un Sistema de Seguimiento y Control de Nutrición que elabore las dietas alimentarias para los animales silvestres del **ZOOLÓGICO MUNICIPAL DE LA PAZ**.

1.5.2 OBJETIVOS ESPECIFICOS

- ♦ Automatizar el proceso de elaboración de dietas, formulado para las tres clases de especies que alberga el zoológico Municipal.
- ♦ Elaborar un módulo para el manejo de información de los alimentos con sus respectivos valores nutricionales.
- ♦ Administrar el manejo de la información de las características de los animales mediante módulos amigables y fáciles de manejar.
- ♦ Elaborar un módulo que permita evaluar los requerimientos mínimos de mantenimiento para fauna en cautiverio.
- ♦ Crear un módulo que permita realizar el seguimiento de consumo de alimentos para cada especie, posteriormente almacenarlo en la base de datos.
- ♦ Hacer uso adecuado de los diferentes recursos y herramientas informáticos que posee la institución.

1.14 LIMITES Y ALCANCES

1.6.1 LIMITES

El proyecto se limitara en la ciudad de La Paz, en predios de la institución del Zoológico Municipal de La Paz.

1.6.2 ALCANCES

El sistema estará implementado en el Zoológico Municipal de La Paz, que es dependiente del Gobierno Autónomo Municipal de La Paz.

Con el sistema de Seguimiento Nutricional se pretende alcanzar lo siguiente:

- ♦ Automatizar el proceso de desarrollo de reportes de dietas alimentarias para los animales silvestres del zoológico.
- ♦ Obtener en hojas impresas los reportes de dietas alimentarias correspondientes a cada especie albergada en el zoológico.

1.15 JUSTIFICACION

1.7.1 JUSTIFICACION TECNICA

El proceso de desarrollo, seguimiento y control de nutrición se viene realizando de forma manual con metodologías de procesos morosos, con el nuevo sistema implementado automatizara estos procesos haciendo uso de interfaces amigables y fáciles de manejar a un solo click.

1.7.2 JUSTIFICACION ECONOMICA

La implementación del Sistema de Seguimiento y Control Nutricional implica una reducción económica en la utilización de materiales de escritorio, tales como papeles de pre-elaboración de dietas, bolígrafos, etc., sistematizando todos los datos en un gestor de base de datos, cálculos elaborados implícitamente por el sistema.

1.7.3 JUSTIFICACION SOCIAL

Mejorara el proceso de seguimiento y control de raciones alimentarias, utilizando el tiempo optimizado en necesidades que aún son deficientes en el manejo y conservación de vida silvestre.

1.16 APORTES

El sistema elaborado es un software de carácter aplicativo, en el están plasmados conocimientos de desarrollo de software, automatizando los procesos que se utilizan en la elaboración y seguimiento de los alimentos aprovechados por los animales.

La institución se beneficiara con esta herramienta, específicamente en el área de conservación y manejo de vida silvestre, administrando parte de su documentación del proceso de seguimiento y control de nutrición.

CAPITULO II MARCO TEORICO

2.1 INTRODUCCIÓN

En el presente capítulo describiremos los conceptos más relevantes en el desarrollo del sistema y las herramientas utilizadas para la construcción del Sistema de Seguimiento y control de nutrición para los animales silvestres.

Una de las metodologías de desarrollo de software convencional es el RUP (Proceso Unificado Racional), que va juntamente con el uso del modelo UML (Lenguaje Unificado de Modelamiento).

2.1.1 MARCO INSTITUCIONAL

2.1.1.1 MISION

El zoológico es una institución sin fines de lucro que genera, promueve y facilita la investigación científica, la educación y recreación en pos de la conservación de la biodiversidad, forjando sentimientos de pertenencia en la sociedad, fortaleciendo valores culturales y bio-ecológicos de la fauna regional y nacional.

2.1.1.2 VISION

El zoológico es una institución fortalecida, autosustentable y con autonomía de gestión constituyéndose como un centro de conservación, especializado en fauna nativa y capaz de transmitir mensajes de interacción con la naturaleza, basado en sus líneas de investigación, programas de reproducción y coadyuvando activamente en políticas de conservación.

2.1.1.3 ORGANIGRAMA

Su Estructura Organizacional cuenta con tres áreas especializadas responsables del cumplimiento de los objetivos y misión institucional con dependencia de la Administración General. Dichas áreas se refieren a: conservación y manejo de vida silvestre, finanzas y contabilidad y mantenimiento – recreación:



Figura 2.1 Estructura Organizacional Zoológico Municipal de La Paz
 Fuente: Plan de Manejo del Zoológico Municipal Vesty Pakos Sofro



Figura 2.2 Estructura Organizacional Conservación y Manejo de Vida Silvestre
 Fuente: Plan de Manejo del Zoológico Municipal Vesty Pakos Sofro

2.2 METODOLOGÍA

PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE-RUP

El RUP (Rational Unified Process) es un proceso de software genérico que puede ser utilizado para una gran cantidad de tipos de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de competencia y diferentes tamaños de proyectos.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

El proceso unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

El proceso Unificado hace el uso del Lenguaje Unificado de Modelado (UML) en la preparación de todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado.

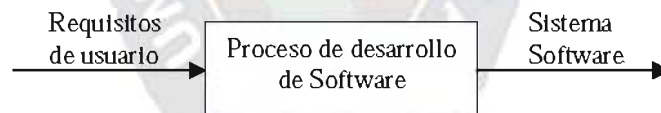


Figura 2.3 Proceso de desarrollo de Software
Fuente: El proceso Unificado de Desarrollo de Software [Jacobson, 00]

2.2.1 RUP Y LOS MODELOS

El proceso unificado considera los siguientes modelos:

1. Modelo de casos de uso
2. Modelo de análisis
3. Modelo de diseño
4. Modelo de despliegue
5. Modelo de implementación
6. Modelo de prueba

1. **Modelo de casos de uso.**- Considera todos los casos de uso y su relación con los usuarios.
2. **Modelo de análisis.**- Considera dos propósitos: refinar los casos de uso con más detalle y establecer la asignación inicial de funcionalidad del sistema a un conjunto de objetos que proporcionan el comportamiento.
3. **Modelo de diseño.**- El modelo de diseño define dos aspectos:
 - ◆ La estructura estática del sistema en la forma de subsistemas, clases e interfaces
 - ◆ Los caso de uso reflejados como colaboraciones entre los subsistemas, clases e interfaces
4. **Modelo de despliegue.**- Define los nodos físicos (PC) y la correspondencia de los componentes con esos nodos.
5. **Modelo de implementación.**- Incluye componentes (que representan al código fuente) y la correspondencia de las clases con los componentes.
6. **Modelo de prueba.**- Describe los casos de prueba que verifican los casos de uso.

2.2.2 FLUJOS DE TRABAJO

El proceso Unificado, considera los siguientes flujos de trabajo fundamentales:

- A. Requisitos
- B. Análisis
- C. Diseño
- D. Implementación
- E. Pruebas

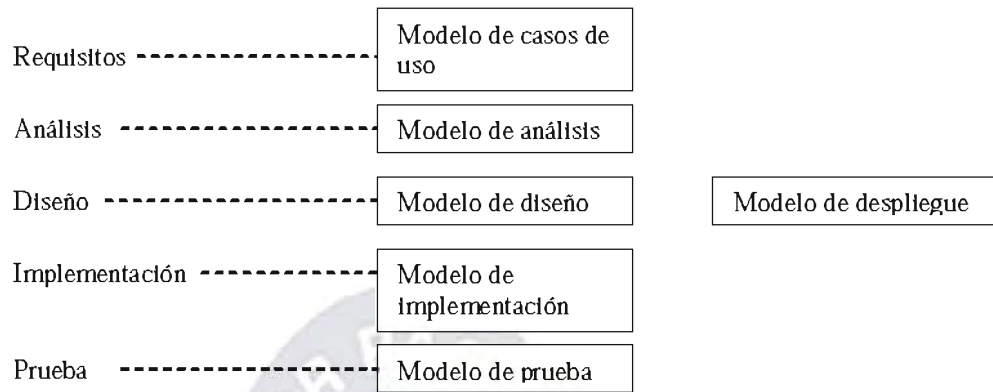


Figura 2.4 Flujos de Trabajo y modelos
 Fuente: El proceso de Desarrollo de Software [Jacobson, 00]

2.2.3 CICLO DE VIDA DEL PROCESO UNIFICADO

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades.

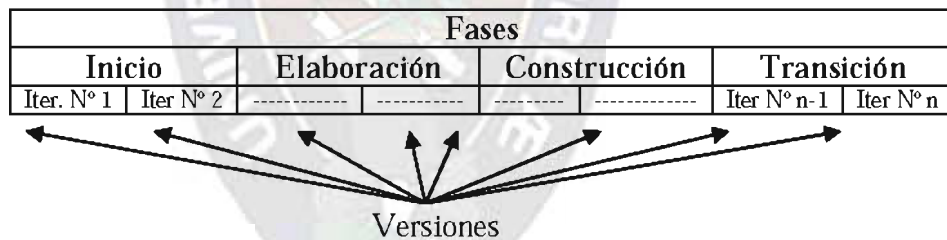


Figura 2.5 Un ciclo con sus fases e Iteraciones
 Fuente: El Proceso Unificado de Desarrollo de Software [Jacobson, 00]

En la figura 2.5 muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (Línea Base) de la arquitectura.

Durante la fase de **inicio** las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

Actividades de la fase de inicio

- ◆ Determinar principales funciones del sistema
- ◆ Desarrollar un esquema tentativo de la arquitectura
- ◆ Desarrollar un plan de desarrollo del proyecto
- ◆ Definir el costo del proyecto

En la fase de **elaboración**, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

Actividades de la fase de de elaboración

- ◆ Analizar el dominio del problema
- ◆ Establecer una base arquitectónica
- ◆ Poner en marcha el plan de desarrollo del proyecto

En la fase de **construcción**, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

Actividades de la fase de construcción

- ◆ Describir que requisitos carece el sistema
- ◆ Refinar el diseño
- ◆ Conclusión del proyecto en un 80-90 % y pruebas del software

En la fase de **transición** se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía. [Jacobson, 00].

Actividades de la fase de transición

- ♦ Poner en prueba la versión beta
- ♦ Obtener el software
- ♦ Desarrollo del manual técnico
- ♦ Mantenimiento y corrección de errores

LENGUAJE DE MODELO UNIFICADO

UML es una especificación de notación orientada a objetos. Divide cada proyecto en un número de diagramas que representan las diferentes vistas del proyecto. Estos diagramas juntos son los que representa la arquitectura del proyecto.

Con UML nos debemos olvidar del protagonismo excesivo que se le da al diagrama de clases, este representa una parte importante del sistema, pero solo representa una vista estática, es decir muestra al sistema parado. Sabemos su estructura pero no sabemos que le sucede a sus diferentes partes cuando el sistema empieza a funcionar. UML introduce nuevos diagramas que representa una visión dinámica del sistema. Es decir, gracias al diseño de la parte dinámica del sistema podemos darnos cuenta en la fase de diseño de problemas de la estructura al propagar errores o de las partes que necesitan ser sincronizadas, así como del estado de cada una de las instancias en cada momento. El diagrama de clases continua siendo muy importante, pero se debe tener en cuenta que su representación es limitada, y que ayuda a diseñar un sistema robusto con partes reutilizables, pero no a solucionar problemas de propagación de mensajes ni de sincronización o recuperación ante estados de error [Larman, 03].

Es importante indicar que UML no es un lenguaje de programación, al contrario es un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

2.2.4 DIAGRAMAS UML

Son varios los diagramas UML, entre los más principales tenemos

Diagrama de clases

Para modelar la estructura estática de las clases en el sistema

Diagrama de casos de uso

Es la técnica para capturar información. Los casos de uso se describen bajo forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario.

Son descripciones de la funcionalidad del sistema independientes de la implementación

Diagrama de componentes

Se utiliza para modelar la vista estática, muestra las organizaciones y dependencias entre un conjunto de componentes.

Diagrama de Despliegue

Muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Diagrama de Estados

Muestra el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro.

Diagrama de secuencia

Describe el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos.

Diagramas de colaboración

Para modelar interacciones entre objetos

UML al ser un lenguaje de modelado de sistemas y no una metodología puede adecuarse en varias metodologías tales como las ágiles para lograr un modelado óptimo y colaborar en el ciclo de vida de estas [Larman, 03]

PROGRAMACIÓN ORIENTADA A OBJETOS (POO)

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

La programación orientada a objetos es una nueva forma de programar. Introduce nuevos conceptos que superan y amplían conceptos antiguos ya conocidos. Entre ellos se destacan los siguientes: Objetos, Clase, Método, Evento, Mensaje, Atributo y Estado Interno

Objeto: entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema.

Clase: definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un *mensaje*. Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.

Mensaje: una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Atributo: contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus

características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

Estado interno: es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.

Abstracción: denota las características esenciales de un objeto, donde se capturan sus comportamientos. Cada objeto en el sistema sirve como modelo de un *agente* abstracto que puede realizar trabajo, informar y cambiar su estado, y *comunicarse* con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción. El proceso de abstracción permite seleccionar las características relevantes dentro de un conjunto e identificar comportamientos comunes para definir nuevos tipos de entidades en el mundo real. La abstracción es clave en el proceso de análisis y diseño orientado a objetos, ya que mediante ella podemos llegar a armar un conjunto de clases que permitan modelar la realidad o el problema que se quiere atacar.

Encapsulamiento: Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.

Polimorfismo: comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en *tiempo de ejecución*, esta última característica se llama *asignación tardía* o *asignación dinámica*. Algunos lenguajes

proporcionan medios más estáticos (en *tiempo de compilación*) de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

Herencia: las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en *clases* y estas en *árboles* o *enrejados* que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

2.3 HERRAMIENTAS Y LENGUAJES DE DESARROLLO

2.3.1 HERRAMIENTAS DE CONSTRUCCIÓN DE SOFTWARE

Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML [Ceballos, 03].

2.3.1.1 FORMULARIOS WINDOWS FORMS

Los formularios Windows Forms sirven para crear aplicaciones de Microsoft Windows en .NET Framework. Este marco de trabajo proporciona un conjunto de clases claro, orientado a objetos y ampliable, que permite desarrollar complejas aplicaciones para Windows. Además, los formularios Windows Forms pueden actuar como interfaz de usuario local en una solución distribuida de varios niveles [Ceballos, 03]

2.3.1.2 COMPATIBILIDAD CON XML

El Lenguaje de marcado extensible (XML) proporciona un método para describir datos estructurados. XML es un subconjunto de SGML optimizado para la entrega a través de Web. El Consorcio World Wide Web (W3C) define los estándares de XML para que los datos estructurados sean uniformes e independientes de las aplicaciones. Visual Studio es totalmente compatible con código XML e incluye el Diseñador XML para facilitar la edición de XML y la creación de esquemas XML [Ceballos, 03].

2.3.1.3 LENGUAJE DE PROGRAMACIÓN VISUAL C#

Microsoft Visual C# 2005 es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C.

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa [Ceballos, 03].

2.3.1.4 EL ENTORNO .NET FRAMEWORK

.NET Framework es un entorno multilingaje que permite generar, implantar y ejecutar aplicaciones y Servicios Web XML. Consta de tres partes principales:

- **Common Language Runtime.**- A pesar de su nombre, el motor en tiempo de ejecución desempeña una función tanto durante la ejecución como durante el desarrollo de los componentes. Cuando el componente se está ejecutando, el motor en tiempo de ejecución es responsable de administrar la asignación de memoria, iniciar y detener subprocesos y procesos, y hacer cumplir la directiva de seguridad, así como

satisfacer las posibles dependencias del componente sobre otros componentes. Durante el desarrollo, el papel del motor en tiempo de ejecución cambia ligeramente; a causa de la gran automatización que permite (por ejemplo, en la administración de memoria), el motor simplifica el trabajo del desarrollador, especialmente al compararlo con la situación actual de la tecnología COM. En concreto, funciones tales como la reflexión reducen de forma espectacular la cantidad de código que debe escribir el desarrollador para convertir la lógica de empresa en componentes reutilizables.

• **Clases de programación unificadas.**- El entorno de trabajo ofrece a los desarrolladores un conjunto unificado, orientado a objetos, jerárquico y extensible de bibliotecas de clases (API). Actualmente, los desarrolladores de C++ utilizan las Microsoft Foundation Classes y los desarrolladores de Java utilizan las Windows Foundation Classes. El entorno de trabajo unifica estos modelos dispares y ofrece a los programadores de Visual Basic y JScript la posibilidad de tener también acceso a las bibliotecas de clases. Con la creación de un conjunto de API comunes para todos los lenguajes de programación, Common Language Runtime permite la herencia, el control de errores y la depuración entre lenguajes. Todos los lenguajes de programación, desde JScript a C++, pueden tener acceso al entorno de trabajo de forma parecida y los desarrolladores pueden elegir libremente el lenguaje que desean utilizar.

ASP.NET.- Construye las clases de programación de .NET Framework, lo que proporciona un modelo de aplicación Web con un conjunto de controles e infraestructura que facilitan la generación de aplicaciones Web. ASP.NET incluye un conjunto de controles que encapsulan elementos comunes de interfaz de usuario de HTML, como cuadros de texto, botones y cuadros de lista. Sin embargo, dichos controles se ejecutan en el servidor Web, y representan la interfaz de usuario en el explorador como HTML. En el servidor, los controles exponen un modelo de programación orientado a objetos que proporciona la riqueza de la programación orientada a objetos al desarrollador Web. ASP.NET también proporciona servicios de

infraestructura, como la administración de estado y el reciclaje de procesos, que reduce aún más la cantidad de código que debe escribir el desarrollador y aumenta la confiabilidad de la aplicación. Asimismo, ASP.NET utiliza estos mismos conceptos para permitir a los desarrolladores la entrega de software como un servicio. Al utilizar características de Servicios Web XML, los desarrolladores de ASP.NET pueden escribir su lógica empresarial y utilizar la infraestructura de ASP.NET para entregar ese servicio a través de SOAP [Ceballos, 03].

2.3.1.5 LENGUAJE C# Y .NET FRAMEWORK

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores crear una amplia gama de aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Puede utilizar este lenguaje para crear aplicaciones cliente para Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y muchas tareas más. Microsoft Visual C# 2005 proporciona un editor de código avanzado, diseñadores de interfaz de usuario prácticos, un depurador integrado y muchas otras herramientas para facilitar un rápido desarrollo de la aplicación basado en la versión 2.0 del lenguaje C# y en .NET Framework [Ceballos, 03].

2.3.2. GESTOR DE BASE DE DATOS SQL SERVER 2005

SQL Server 2005 ofrece una solución integrada de administración y análisis de datos que ayuda a las organizaciones de cualquier magnitud a realizar lo siguiente:

- ♦ Crear, desplegar y administrar aplicaciones empresariales más seguras, escalables y confiables.
- ♦ Maximizar la productividad de IT mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- ♦ Compartir datos en múltiples plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.
- ♦ Controlar los costes sin sacrificar el rendimiento, la disponibilidad, la escalabilidad o la seguridad.

SQL Server 2005 potencia su infraestructura de datos en tres áreas clave: administración de datos empresariales, productividad del encargado del desarrollo e inteligencia empresarial (BI). También abre nuevos caminos en precios y licencias accesibles, rutas de actualización a SQL Server 2005 y el sistema Microsoft Windows Server [Sitio 1]

2.3.2.1 PLATAFORMA DE DATOS DE SQL SERVER

SQL Server es una solución de dato global integrado y de extremo a extremo que habilita a los usuarios en toda su organización mediante una plataforma más segura, confiable y productiva para datos empresariales y aplicaciones de BI. SQL Server 2005 provee herramientas sólidas y conocidas a los profesionales de IT, así como también a trabajadores de la información, reduciendo la complejidad de la creación, despliegue, administración y uso de aplicaciones analíticas y de datos empresariales en plataformas que van desde los dispositivos móviles hasta los sistemas de datos empresariales. A través de un conjunto global de características, la interoperabilidad con sistemas existentes y la automatización de tareas rutinarias, SQL Server 2005 ofrece una solución completa de datos para empresas de todos los tamaños [Sitio 1].

2.4 METRICAS DE CALIDAD DE SOFTWARE

2.4.1 GESTION DE LA SEGURIDAD

Se orienta a preservar los siguientes principios de la seguridad informática:

- a) **Confidencialidad:** Asegurar que únicamente personal autorizado tenga acceso a la información
- b) **Integridad:** Garantizar que la información no sea alterada, eliminada o destruida por entidades no autorizadas.
- c) **Disponibilidad:** Asegurar que los usuarios tendrán acceso a la información cuando la requieran.

A continuación se describe cada una de las áreas de seguridad con el objeto de esclarecer los objetivos de estos controles.

Políticas de seguridad: Proporcionar dirección y apoyo gerencial para brindar seguridad de la información. El nivel gerencial debe establecer una política clara, demostrar apoyo y

compromiso con respecto a la seguridad de la información, mediante la formulación y mantenimiento de una política de seguridad.

Seguridad Organizacional: Administrar la seguridad de la información dentro la organización. Se debe establecer un marco gerencial para iniciar y controlar la implementación de la seguridad de la información.

Seguridad del personal: Reducir los riesgos de error humano, robo, fraude o uso inadecuado de instalaciones. Las responsabilidades en materia de seguridad deben ser explícitas en la etapa de reclutamiento, incluidas en los contratos y monitoreadas durante el desempeño del individuo como empleado. Los candidatos a ocupar los puestos de trabajo deben ser adecuadamente seleccionados, especialmente si se trata de tareas críticas. Todos los empleados y usuarios externos de las instalaciones de procesamiento de información deben firmar un acuerdo de confidencialidad (no revelación).

Seguridad física y de entorno: Impedir accesos no autorizados, daños e inferencia a las sedes e información de la empresa. Las instalaciones de procesamiento de información crítica o sensible de la empresa deben estar ubicadas en áreas protegidas y resguardadas. Deben estar físicamente protegidas contra accesos no autorizados, daños e intrusiones.

Comunicación y administración de operaciones: Integrar los procedimientos de operación de la infraestructura tecnológica y de controles de seguridad documentados, que van desde el control de cambios en la configuración de los equipos, manejo de incidentes, administración de aceptación de sistemas, hasta el control de código malicioso y administración de la red.

Control de acceso: El acceso a la información y los procesos de negocio deben ser controlados sobre la base de los requerimientos. Se deben tener en cuenta las políticas de difusión y autorización de la información.

Desarrollo de sistemas y mantenimiento: La unidad debe disponer de procedimientos que garanticen la calidad y seguridad de los sistemas desarrollados para tareas específicas de la organización.

Continuidad de las operaciones de la organización: El sistema de administración de la seguridad debe integrar los procedimientos de recuperación en caso de contingencias, los cuales deben ser revisados de manera constante y puestos a prueba con la finalidad de determinar las limitaciones de los mismos.

Requerimientos legales: Impedir infracciones y violaciones de las leyes del derecho civil y penal; de las obligaciones establecidas por leyes, estatutos, normas, reglamentos ó contratos; y de los requisitos de seguridad. El diseño, operación, uso y administración de los sistemas de información están sujetos a requisitos de seguridad legal, normativa y contractual.

2.4.2 ANALISIS DE RIESGO DEL SOFTWARE

La explicación adecuada del riesgo siempre implica dos características las cuales son la incertidumbre (acontecimiento que caracteriza al riesgo que ocurra ó no ocurra, no hay un riesgo que sea cien por ciento probable) y la pérdida (si el riesgo se convierte en realidad, ocurrirán consecuencias no deseadas o pérdidas). A continuación se describen los riesgos del software con respecto a la seguridad.

- a) falta de una política sobre el control de acceso a la información; ocasiona que el personal no autorizado tenga acceso libre a la base de datos, la red y el servidor.
- b) Falta de seguridad en la red ocasiona que personas modifiquen la información almacenada en la base de datos, mediante consola.
- c) Falta de encriptación de las contraseñas ocasiona que un usuario acceda al sistema reemplazando a otro usuario.
- d) Falta de políticas de copias de seguridad ocasiona que se pierda información.

2.4.3 CONTROLES DE SEGURIDAD

Los controles de seguridad se describen a continuación:

- Políticas de seguridad: Seguridad de la información e infraestructura física y de datos. Por infraestructura de datos se entiende, a todos los elementos que intervienen en el flujo de comunicación de datos. Estas políticas se aplican a empleados y otras personas que tengan relación con la Unidad de Acreditación y Certificación.
- Seguridad del Personal: Toda persona que ingrese a las oficinas de la Unidad de Acreditación y Certificación, debe ser capacitada e instruida en las políticas de seguridad.

- ♦ Seguridad física y de entorno: Solamente debe ingresar personal autorizado de la Unidad de Acreditación y Certificación. Además siempre debe estar presente en todo momento un responsable de dicha Unidad.
- ♦ Comunicación y administración de operaciones: Todo empleado que introduzca internacionalmente programas maliciosos en la red ó los servidores (virus, gusanos, caballos de Troya y otros) deberá ser sujeto a una acción disciplinaria.
- ♦ Control de acceso: La contraseña del Sistema y del gestor de base de datos debe modificarse de manera confidencial cada semana, de igual manera las contraseñas de acceso a los sistemas de información deben modificarse mensualmente. Pero está modificación debe ser de manera transparente.

2.4.3.1 ¿QUÉ ES CALIDAD?

La primera forma de entender un concepto es mirar su significado en un diccionario: Calidad suele significar el conjunto de cualidades. Cuando se dice que en un caballo es de buena calidad, se da a entender que posee todas las cualidades que constituyen el caballo bueno. Por esta razón llamamos calidad, y no de cualidad, a la nobleza: mujer es de calidad, ha dicho Lope de Vega.” Entonces calidad es un concepto difuso, condicionado por el contexto de la situación, y multidimensional, La adquisición de un bien puede cubrir nuestras expectativas en general, pero basta con que en un momento determinado falle un elemento concreto o que su uso sea “complicado” para que tengamos la sensación de carencia de calidad. [Piattini, 03]

2.4.3.2 CALIDAD CON RELACIÓN AL SOFTWARE

El software, tanto en su vertiente de producto como de aplicación, conlleva una serie de especificidades con relación a la calidad. Si intentamos detallar lo que entendemos por calidad de software tendríamos que hablar de:

- Funcionamiento (El software debe funcionar siempre, debe permitirnos utilizarlo cuando sea necesario)

- Funcionalidad (Debe hacer lo que dice que hace)
- Usabilidad (Debe permitirnos hacerlo de forma adecuada, natural)

En general, con respecto al software, queremos que funciones, que tenga las funcionalidades que dice tener y podamos usar estas funcionalidades de una forma natural. [Piattini, 03]

2.4.3.4 METRICAS DE CALIDAD

Se sugiere que para las propiedades de calidad que se identifiquen en el proyecto se definan métricas para evaluarlas. Estas métricas se basan en el estándar ISO/IEC 9126. El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para el software. El estándar identifica seis atributos clave de calidad, funcionalidad, usabilidad, facilidad de mantenimiento, portabilidad, confiabilidad y eficiencia [sitio2].

FUNCIONALIDAD

El grado en que el software satisface las necesidades indicadas por los siguientes sub atributos: idoneidad, corrección, interoperatividad, conformidad y seguridad.

El factor viene dado por la medida de la subcaracterística de conveniencia, cuyas métricas son la completitud de la implementación funcional y la adecuación funcional.

Este factor viene dado por la medida de la subcaracterística de la madurez, cuyas métricas son: levantamiento de defectos y densidad de defectos que se encuentran explicados en la siguiente tabla.

Característica: Funcionalidad		
Subcaracterística: Conveniencia		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Complettitud de la implementación funcional	¿Cuán completa es la implementación funcional?	$X=1-A/B$ A= Número de <i>casos de uso</i> (o funciones) no implementadas B = Número de <i>caso de uso</i> (o funciones) descriptas en el alcance del sistema final de la fase inicial. $Y=1- A/C$ A=Número de casos de uso (o funciones) no implementadas C= Número de casos (o funciones) descriptas en el alcance del sistema final. $Z=1-A/D$ A=Número de casos de uso (o funciones) no implementadas D= Número de casos de uso (o funciones) descriptas en la especificación de requerimientos

Tabla 2.1 Métrica para medir el factor de funcionalidad

Característica: Funcionalidad		
Subcaracterística: Conveniencia		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Adecuación funcional	¿Cuán adecuadas son las definiciones evaluadas?	$X= 1-A/B$ A = Número de funciones (<i>casos de uso</i>) las cuales se detectaron problemas en la evaluación B=Número de funciones evaluadas

Tabla 2.2 Métrica para medir el factor de funcionalidad

FIABILIDAD

En términos estadísticos como “la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico”.

Este factor viene dado por la medida de la subcaracterística de la madurez, cuyas métricas son: levantamiento de defectos y densidad de defectos que se encuentran explicados en la siguiente tabla.

Característica: Fiabilidad		
Subcaracterística: Madurez		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Levantamiento de defectos (atributo para medir la calidad del proceso)	¿Cuántos defectos han sido corregidos? ¿Cuál es la proporción de defectos corregidos?	$X=A$ A= Número de defectos corregidos en diseño/codificación $Y=A/B$ A=Número de defectos corregidos en diseño/codificación B = Número de defectos detectados en las revisiones.
Densidad de defectos	¿Cuál es la proporción de defectos respecto al tamaño del producto?	$X= 1-A/B$ A = Número de defectos que no fueron corregidos B=Tamaño del producto. Nota: el tamaño del producto se mide en líneas de código.

Tabla 2.3 Métrica para medir el factor de fiabilidad

Característica: Fiabilidad		
Subcaracterística: Madurez		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Densidad de defectos	¿Cuántos defectos fueron detectados durante el periodo de prueba?	$X=A/B$ A= Número de defectos detectados. B = Tamaño del producto. Nota: el tamaño del producto se mide en líneas de código.

Tabla 2.4 Métrica para medir el factor de fiabilidad

FACILIDAD DE MANTENIMIENTO

La facilidad con que una modificación puede ser realizada, está indicada por los siguientes sub-atributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba, y se define con la siguiente fórmula:

$$IMS=(M_t - (F_a+F_c+F_d)) / M_t$$

Donde:

M_t = Número de módulos en la versión actual

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

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

F_d = Número de módulos en la versión anterior que se han borrado en la versión actual

A medida que el IMS se acerca a 1 el producto se empieza a estabilizar.

PORTABILIDAD

La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes atributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Característica: Portabilidad		
Subcaracterística: Facilidad de instalación		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Facilidad de instalación	¿Puede el usuario o quien técnico asignado fácilmente instalar el software en un ambiente operacional?	$X=A/B$ A=Número de casos en que el usuario es exitoso en la operación de instalación. B=Número total de casos en que el usuario intenta ejecutar la operación de instalación.

Tabla 2.5 Métrica para medir el factor de portabilidad

USABILIDAD

Grado en el software es fácil de usar, viene reflejado por los siguientes: facilidad de comprensión, facilidad de aprendizaje y operatividad, cuyas métricas son: completitud de la descripción, consistencia operacional y consistencia operacional en uso.

Característica: Usabilidad		
Subcaracterística: Capacidad de ser entendido (Understandability)		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Complettitud de la descripción	¿Qué proporción de las funciones (casos de uso) o tipo de funciones se describen en la descripción del producto (documentación de usuario, ayuda, etc.)?	$X=A/B$ A= Número de funciones (casos de uso) o tipo de funciones descritas en la descripción del producto. B = Número total de funciones (casos de uso) o tipo de funciones. Nota: Esta medida indica si los usuarios potenciales entenderán la capacidad del producto luego de leer la descripción del producto.
Subcaracterística: Operabilidad		
Métricas		
Nombre	Propósito de la métrica	Medición o fórmula de cálculo
Consistencia operacional	¿Que proporción de las operaciones se comportan de manera similar a operaciones similares en otras partes del sistema?	$X= 1 - A/B$ A = Número de instancias de operaciones con comportamiento inconsistente. B=Número total de operaciones

Tabla 2.6 Métrica para medir el factor de usabilidad

2.5 ESTIMACIÓN DE COSTOS

Existen técnicas para la estimación de costos, pero para ello se requiere experiencia, acceso a una buena información histórica y coraje para confiar en medidas cuantitativas cuando todo lo que existe son datos cualitativos.

El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad.

2.5.1 MODELO COCOMO

Barry Boehm en su libro "economía de la ingeniería de software" detalla un modelo amplio de estimación de costos llamado COCOMO (Constructive Cost Model). La palabra "constructive" se refiere a el hecho que el modelo ayuda a un estimador a comprender mejor

la complejidad del software; este modelo es un ejemplo de variable simple estático y es usado por miles de administradores de proyecto de software.

COCOMO ayuda a estimar el esfuerzo, tiempo, gente y costos (ya sea estos de desarrollo, equipamiento y mantenimiento).

El modelo provee tres "niveles" de aplicación: básico, intermedio y avanzado, basados en los factores considerados por el modelo.

Básico, es un modelo estático simplemente evaluado que calcula el esfuerzo (y costo) del desarrollo del software como función del programa expresado en líneas de código (LDC estimados).

Intermedio, calcula el esfuerzo del desarrollo del software como función del tamaño del programa y un conjunto de "guías de costo" que incluye una evaluación subjetiva del producto, hardware, personal y de los atributos del proyecto.

Avanzado, incorpora todas las características de la versión intermedia con una evaluación del impacto de las vías de costo en cada fase (análisis, diseño, etc) del proceso de la ingeniería de software.

El modelo básico se extiende para considerar un conjunto de atributos de guías de costo que pueden agruparse en cuatro categorías principales:

Producto (por ej. Requerimientos de software, confiabilidad, tamaño de la base de datos, y complejidad del producto).

Computadora (por ej. Restricciones en el tiempo de ejecución y almacenamiento).

Personal (por ej. Capacidad de análisis, experiencia en aplicaciones tanto en lenguajes de programación y capacidad del programador)

Proyecto (por ej. Uso de practicas modernas de programación, uso de herramientas de software y requerimiento de un plan de desarrollo).

En cada nivel de aplicación están definidos para tres tipos de proyectos de software:

Modo orgánico, proyectos de software relativamente pequeños y sencillos en los que pequeños equipos con buena experiencia en la aplicación trabajan en un conjunto de requerimiento poco rígido.

Modo semi-acoplado, un proyecto de software intermedio en tamaño y complejidad en el cual equipos con distintos niveles de experiencia debe satisfacer requerimientos poco y medio rígidos

Modo acoplado, un proyecto de software que debe ser desarrollado dentro un conjunto estricto de hardware, software y de restricciones operativas.

MODO DE DESARROLLO	A	b	c	D
Orgánico	3.2	1.05	2.5	0.38
Semi-acoplado	3.0	1.12	2.5	0.35
Acoplado (empotrado)	2.8	1.20	2.5	0.32

Tabla 2.7 Niveles de aplicación

Estimación del esfuerzo necesario:

$$E = a_b * LDC^b$$

Estimación del tiempo necesario:

$$D = c_b E^d$$

Numero de personas para el proyecto:

$$N = E / D$$

3.1 FASE DE INICIO

En esta fase se hace mayor énfasis en las actividades de modelado del negocio y de requerimientos, para el desarrollo del Sistema de Seguimiento y Control Nutricional.

Por eso es fundamental tener una visión clara de cual es el funcionamiento real en la institución, tanto en la parte administrativa como la parte operativa hasta el final del suministro de los alimentos a los animales silvestres del zoológico.

3.1.1 MODELADO DE NEGOCIO

El modelado es la descripción del contexto de la funcionalidad del manejo del sistema de alimentación llevado hasta ahora en el Zoológico Municipal “VPS”



Figura 3.1 Modelado de Negocio
Fuente: Elaboración propia

Para la elaboración del modelado de negocio utilizamos los siguientes modelos:

- ♦ Modelado de casos de uso general del negocio
- ♦ Modelado de dominio

3.1.1.1 MODELO GENERAL DE CASOS DE USO

En este modelo se representa gráficamente cual es la funcionalidad de los actores que intervienen en la elaboración de “Hoja de Dieta Alimentaria” del Zoológico Municipal “Vesty Pakos Zofro”, tales son los actores indirectos y directos.

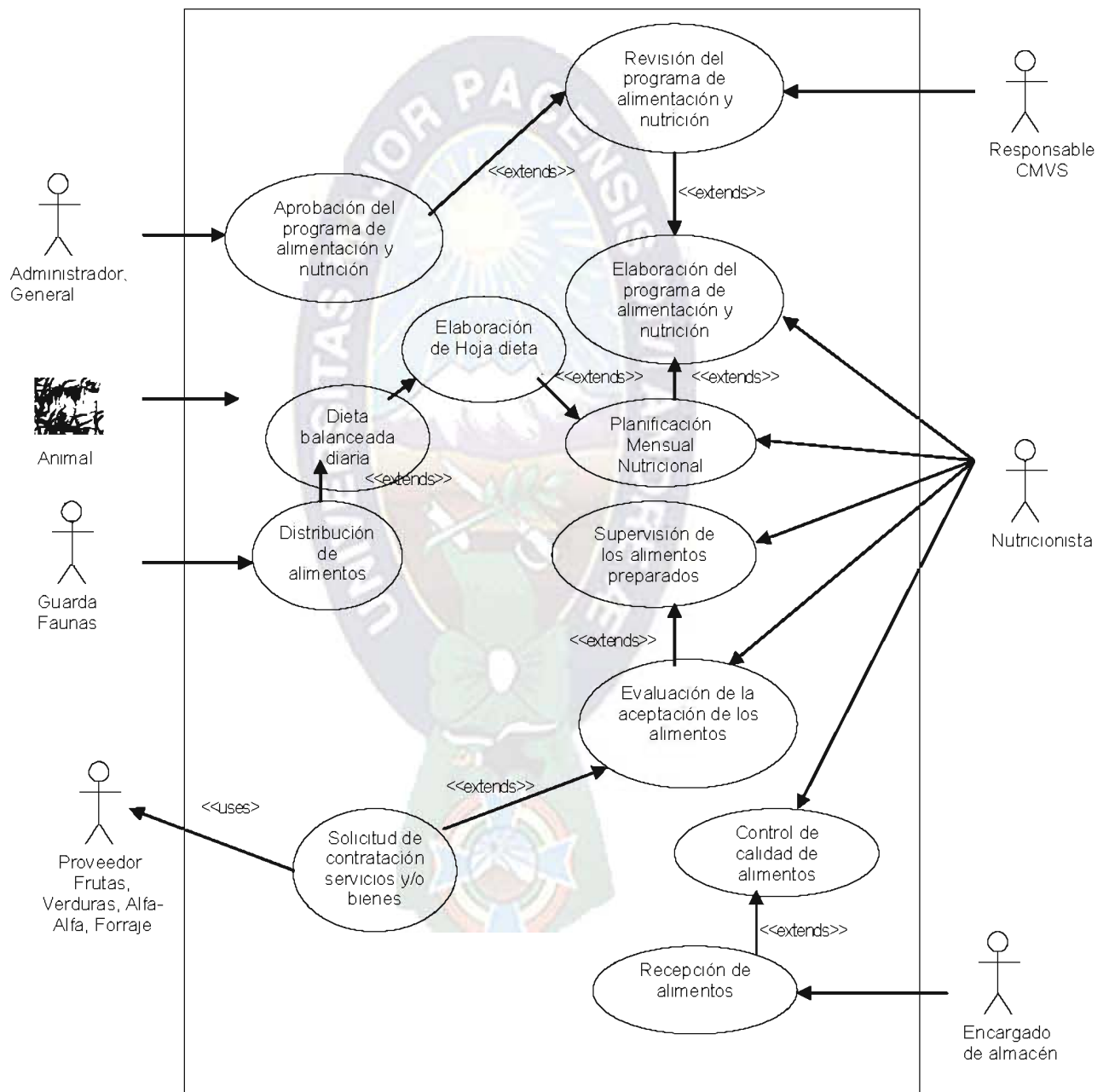


Figura 3.2 Modelo General de Casos de Uso
Fuente: Elaboración Propia

3.1.1.2 MODELO DEL DOMINIO DEL NEGOCIO

Este modelo describe cuales son los enlaces que existen entre los objetos relacionados con el sistema.

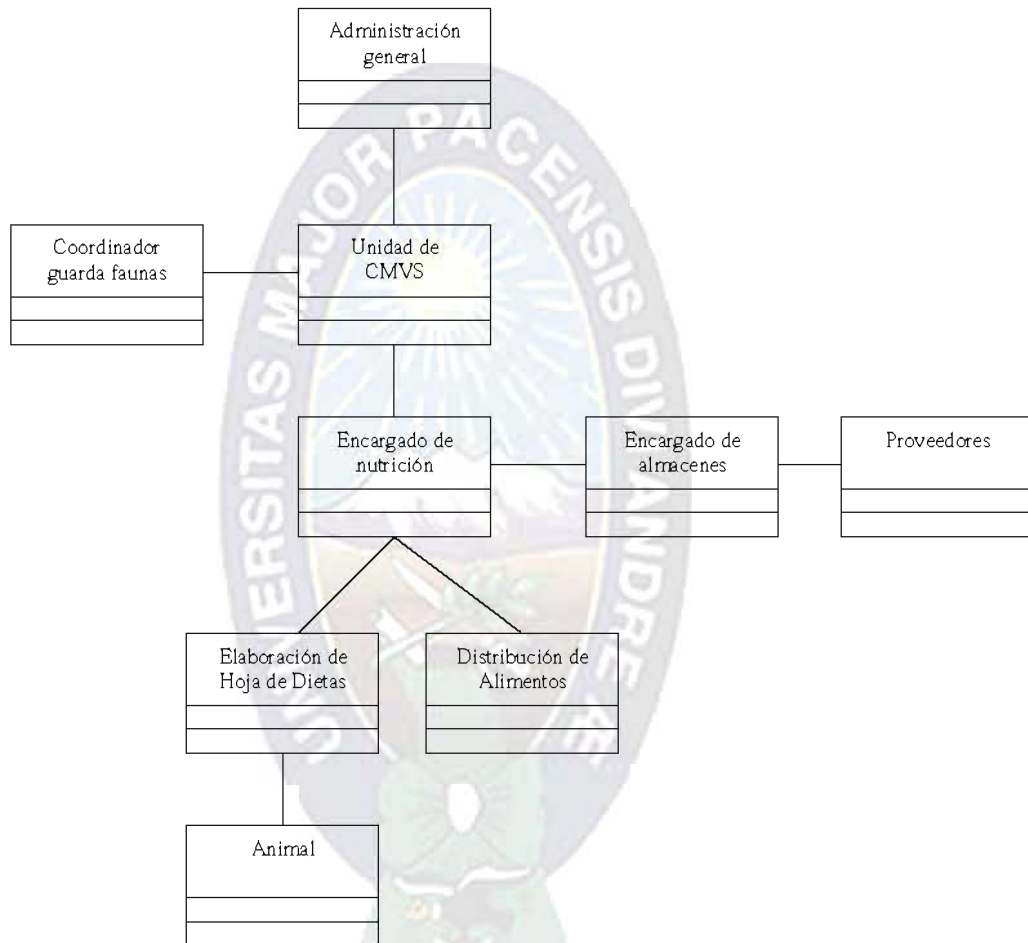

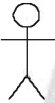
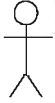
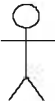


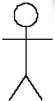


Figura 3.3 Modelo del Dominio del Negocio
Fuente: Elaboración propia

3.1.1.2 IDENTIFICACIÓN DE LOS ACTORES

 <p>Animal</p>	<p>El animal es aquella especie que puede ser: Mamíferos, Aves y Reptiles albergado en el “ZMVPS” que requiere atención veterinaria y dietas alimentarias diarias para la sobrevivencia diaria.</p>
 <p>Administrador General</p>	<p>El administrador General del Zoológico Municipal se encarga de la planificación, diseño, organización, seguimiento y control de políticas definidas para el Zoológico orientadas hacia el cumplimiento de la misión institucional y sus objetivos estratégicos.</p> <p>Uno de los papeles que cumple es la revisión y aprobación (firma) del programa de alimentación y nutrición, considerando el presupuesto para el mismo</p>
 <p>Nutricionista</p>	<p>Es la persona encargada de elaborar el programa de alimentación y nutrición considerando valor nutritivo y cantidad requerida para cada especie e individuo de acuerdo al plan de manejo. Tiene que ajustar (si corresponde) las fichas nutricionales de acuerdo al programa de alimentación y nutrición aprobado y recomendaciones veterinarias.</p>
 <p>Responsable CMVS</p>	<p>Es la persona encargada de revisar el programa de alimentación y nutrición, si ve la pertinencia emite recomendaciones y observaciones al mismo, caso contrario valida mediante firma. Además de supervisar el buen funcionamiento del área de Conservación y Manejo de Vida Silvestre “CMVS”.</p>

 <p>Encargado de almacén</p>	<p>Es la persona encargada de recepcionar los productos alimentarios que traen los diferentes proveedores, verificando que llegue la cantidad y peso exacto de acuerdo al pedido diario que tiene que llegar. El encargado de almacén cuenta con diferentes lugares de almacenamiento de alimentos.</p>
 <p>Guardafaunas</p>	<p>Es la persona encargada de recoger los alimentos y los deposita en las instalaciones ocupados por la fauna, siguiendo los protocolos de seguridad y de acuerdo a cada especie los sitúa en diferentes ubicaciones en cumplimiento al Programa de Enriquecimiento de Ambientes.</p>
 <p>Proveedor: Frutas, Verduras, Alfa- Alfa, Forraje</p>	<p>Persona Natural o Empresa privada que se encarga de abastecer los diferentes alimentos para ZMVPS, que son clasificados en:</p> <ul style="list-style-type: none"> - Proveedor de Frutas - Proveedor de Verduras - Proveedor de Alfa-Alfa - Proveedor de Forraje - Proveedor de Jumentos

3.2 FASE DE ELABORACION

En esta etapa consideramos la especificación de requerimientos y para no dejar suelto ningún detalle usamos el diagrama de actividades, posteriormente podemos visualizar nuestro sistema haciendo el uso de los diagramas de estado y diagramas de casos de uso.

Esta versión es utilizada para la reingeniería de requerimientos, puesto que dada la información por el encargado de área existe la posibilidad de omitir procesos muy importantes en la elaboración de hoja de dieta.

3.2.1 MODELADO DE REQUISITOS

Al hacer uso del diagrama de actividades hacemos la recopilación de las necesidades específicas que requiere la unidad “MANEJO Y CONSERVACIÓN DE VIDA SILVESTRE”.

3.2.1.1 MODELO DEL NEGOCIO

De acuerdo al Modelo General de casos de uso se plantea el caso de uso *Elaboración de Hoja dieta*.

Consideramos como punto de partida esta actividad ya que nuestro objetivo es la elaboración de hojas de dietas alimentarias de acuerdo a la variedad de alimentos de temporada.

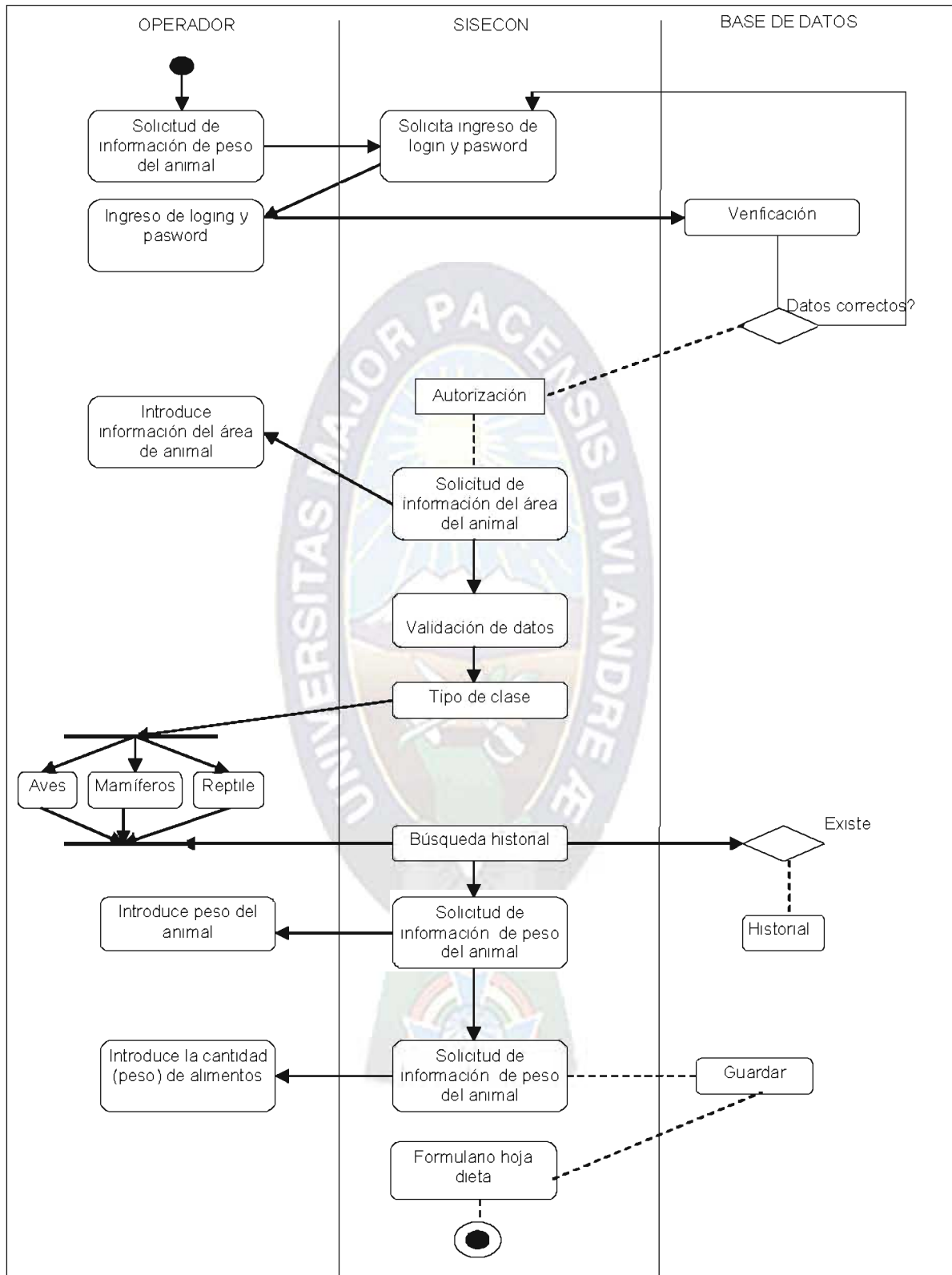


Figura 3.4 Diagrama de Actividades – Elaboración de hoja dieta
Fuente: Elaboración propia

ESPECIFICACION DE LOS CASOS DE USO

DOCUMENTO DE DESCRIPCION DE CASO DE USO 1		
a) Nombre:	Elaboración de hoja dieta	
b) Actor:	Nutricionista	
c) Descripción	Es la propuesta de dietas alimentarias de acuerdo a la especie y cantidad de alimentos con la que se dispone en los almacenes del zoológico.	
d) Flujo Principal	Eventos Actor	Eventos de Sistema
	1. El nutricionista previamente hace un conteo de la variedad de alimentos con las que se dispone en el almacén.	
	2. El nutricionista previamente inserta el área para el cual se elaborara la hoja de dieta alimentaria. Posteriormente puede elegir los datos propios del animal.	El sistema despliega datos actualizados de las tres especies con las que cuenta y son: - Aves - Mamíferos - Reptiles Al elegir una especie el sistema despliega la diversidad de animales. Posteriormente devuelve datos propios del animal que son: - Nombre científico - Nombre común - Familia
	3. El nutricionista después de los eventos calcular el requerimiento de energía calculada y energía aportada por la dieta, añade observaciones de tipo escrito.	El sistema almacena la información en la base de datos la hoja de dieta alimentaria con los siguientes datos. - Nombre común - Nombre científico - Número de individuos (número de animales) - Fecha de elaboración - Requerimiento de energía calculada. - Energía Aportada por la dieta.

		<ul style="list-style-type: none"> - Ubicación <p>Y la tabla que tiene los siguientes datos:</p> <ul style="list-style-type: none"> - Tipo de Alimento - Cantidad/Materia. - Forma de preparación del alimento. - Cantidad de dieta proporcionada.
e) Precondición	El nutricionista deberá conocer el peso vivo estimado del animal.	
b) Poscondición	El nutricionista cuenta con 2 reportes de dieta alimentarias que son: <ul style="list-style-type: none"> - Hoja de dieta con especificaciones que son: nombre de alimento, cantidad y forma de preparación del alimento. - Hoja de dieta con lenguaje científico que tiene: valores nutricionales específicos, total dieta, aporte por la dieta, requerimiento mínimo para la dieta y balance de la dieta. 	
c) Presunción	La hoja de dieta alimentaria está almacenada en la base de datos y dispuesta a ser impresa.	

Tabla 3.1 Caso de uso: Elaboración de dietas alimentarias

3.2.1.2 MODELO DE REQUISITOS

Tras realizar el modelado del negocio la siguiente tarea que nos ocupa es el modelado de requisitos en el cual pasamos a identificar los casos de uso para posteriormente describirlos más profundamente y paralelamente elaborar un modelo conceptual en el que se plasma gráficamente la relación entre las información encontradas en el modelado del negocio.

a) Identificación de Casos de Uso

En este punto tenemos que identificar los casos de uso del proceso del negocio del sistema SISECON.

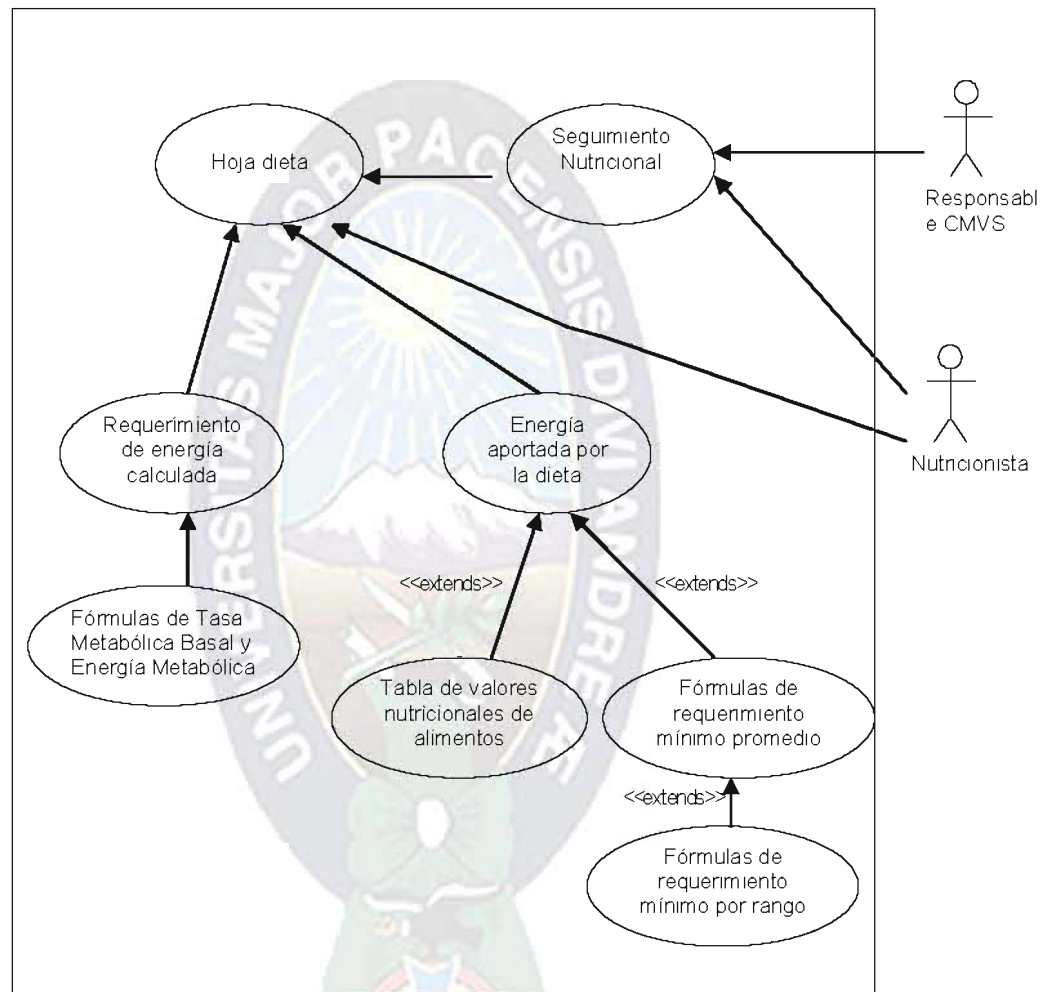


Figura 3.5 Diagrama de Casos de Uso – Elaboración de hoja dieta
Fuente: Elaboración propia

b) Descripción de los casos de uso

En este punto tenemos que describir más profundamente cada uno de los casos de uso que hemos encontrado.

DOCUMENTO DE DESCRIPCION DE CASO DE USO 2		
a) Nombre:	Requerimiento de energía calculada	
b) Actor:	Nutricionista	
c) Descripción	Determinación del cálculo de energía mínima requerida.	
d) Flujo Principal	Eventos Actor	Eventos de Sistema
	El nutricionista mediante cálculos manuales cuenta con la información del peso estimado vivo del animal en unidades de kilogramos e introduce en el sistema.	<p>El sistema hace cálculos con las siguientes formulas: $TMB = 57.6 * 10^{0.76}$ Para el caso de <i>aves</i> utiliza la siguiente formula: $EM = 2 * TMB$ Para el caso de <i>mamíferos</i>: $EM = 2 * TMB$ Para el caso de <i>reptiles</i>: $EM = 0.15 * TMB$</p> <p>En los tres casos que son aves, mamíferos y reptiles el sistema devuelve un valor único que es la variable EM (energía metabólica) que tiene el mismo significado que requerimiento de energía calculada.</p>
e) Precondición	Tener el peso estimado vivo del animal en unidades de kilogramos.	
b) Poscondición	En cifras numéricas se dispone del requerimiento de energía.	
c) Presunción	Es posible acceder a la base de datos para ver el cálculo.	

DOCUMENTO DE DESCRIPCION DE CASO DE USO 3		
a) Nombre:	Energía aportada por la dieta	
b) Actor:	Nutricionista	
c) Descripción	Determinar el cálculo alimenticio con la variedad de alimentos con las que cuenta el almacén.	
d) Flujo Principal	Eventos Actor	Eventos de Sistema
	1. El nutricionista introduce el número de animales en el grupo de la especie.	El sistema multiplica el requerimiento de energía calculada por el número de animales en el grupo y la actualiza en la tabla de requerimiento mínimo promedio.
	2. El nutricionista elige el tipo de alimento con la que preparará la dieta alimentaria y las debe expresar en unidades de gramos.	El sistema expone la variedad de alimentos con la que cuenta en la base de datos. Posteriormente el sistema se encarga de crear nuevas filas donde sus atributos son los valores nutricionales más la cantidad del producto y el tipo de preparación.
e) Precondición	El nutricionista deberá tener a mano el pedido mensual de alimentos.	
b) Poscondición	El nutricionista contará con un reporte de hoja de dieta alimentaria completa y dispuesta para imprimir y/o modificar cuando el así lo vea conveniente.	
c) Presunción	La base de datos esta disponible.	

DOCUMENTO DE DESCRIPCION DE CASO DE USO 4		
a) Nombre:	Seguimiento Nutricional	
b) Actor:	Nutricionista	
c) Descripción	Elaboración de reporte de consumo de alimentos	
d) Flujo Principal	Eventos Actor	Eventos de Sistema
	1. El nutricionista hace una observación visual sobre la cantidad de alimentos consumidos y rechazados.	
	2. El nutricionista introduce el tipo de alimento que no se ha consumido y las cantidades son en gramos.	El sistema hace relación de acuerdo a la fecha elaborada la dieta alimentaria y nos determina la cantidad de energía rechazada por el animal.
e) Precondición	El nutricionista deberá hacer inspección de área y determinar que alimentos no se han consumido y anotar en una libreta.	
b) Poscondición	El nutricionista contará con el reporte seguimiento nutricional aprovechada por el animal.	
c) Presunción	La base de datos esta disponible.	

d) Diagrama de estados

Un diagrama de estados describe todas las secuencias de estados y acciones por las que puede pasar cada uno de los subsistemas durante su ciclo de vida como reacción a distintos eventos.

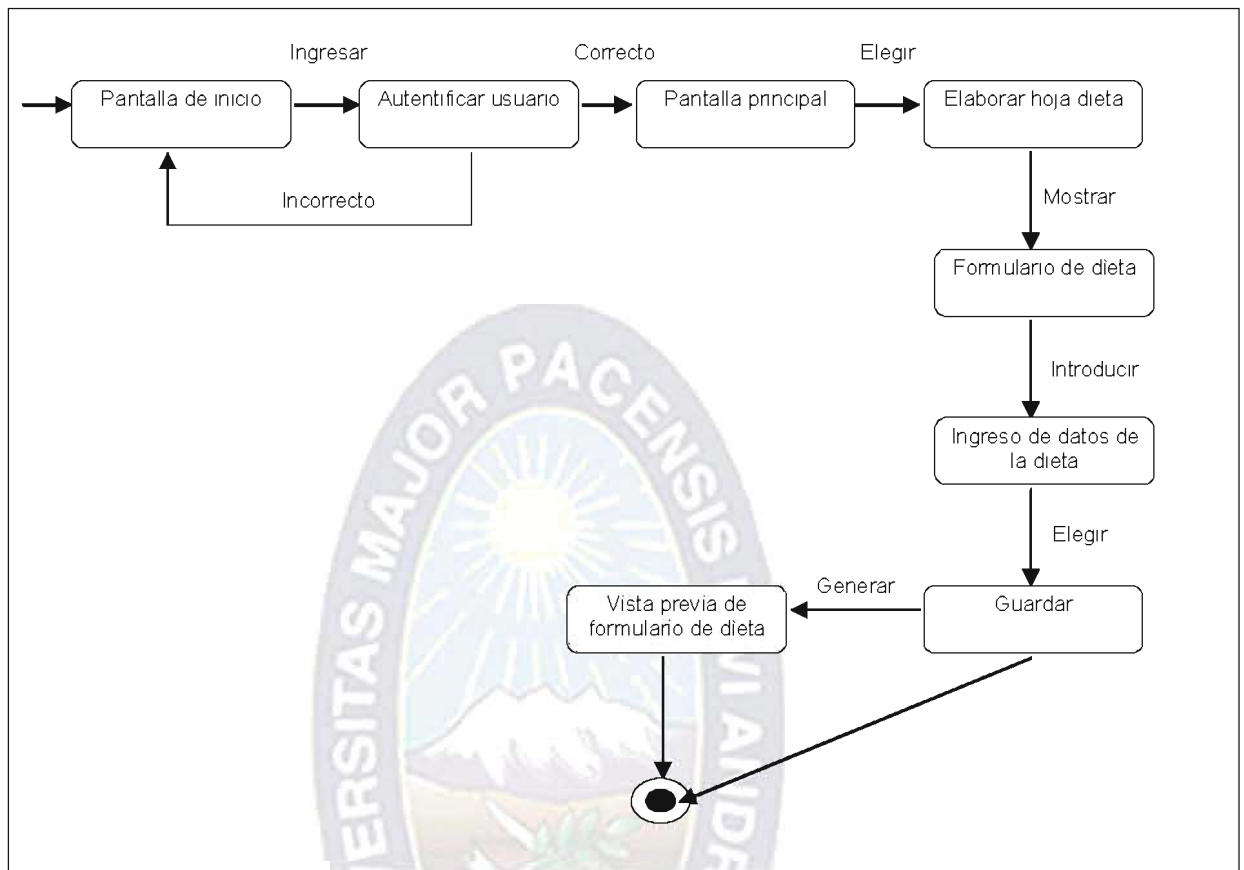


Figura 3.6 Diagrama de estado – Elaboración de hoja dieta
Fuente: Elaboración propia

3.2.2 MODELO DE ANALISIS Y DISEÑO

En análisis y diseño se utiliza el diagrama de clases que son provenientes la elaboración de los diagramas de casos de uso.

a) Diagrama de secuencia

En este punto tratamos de determinar las operaciones que demandan los actores del sistema. Para ello utilizamos diagramas de secuencia en los que intervienen los actores de casos de uso así como las operaciones que reclama cada actor del sistema.

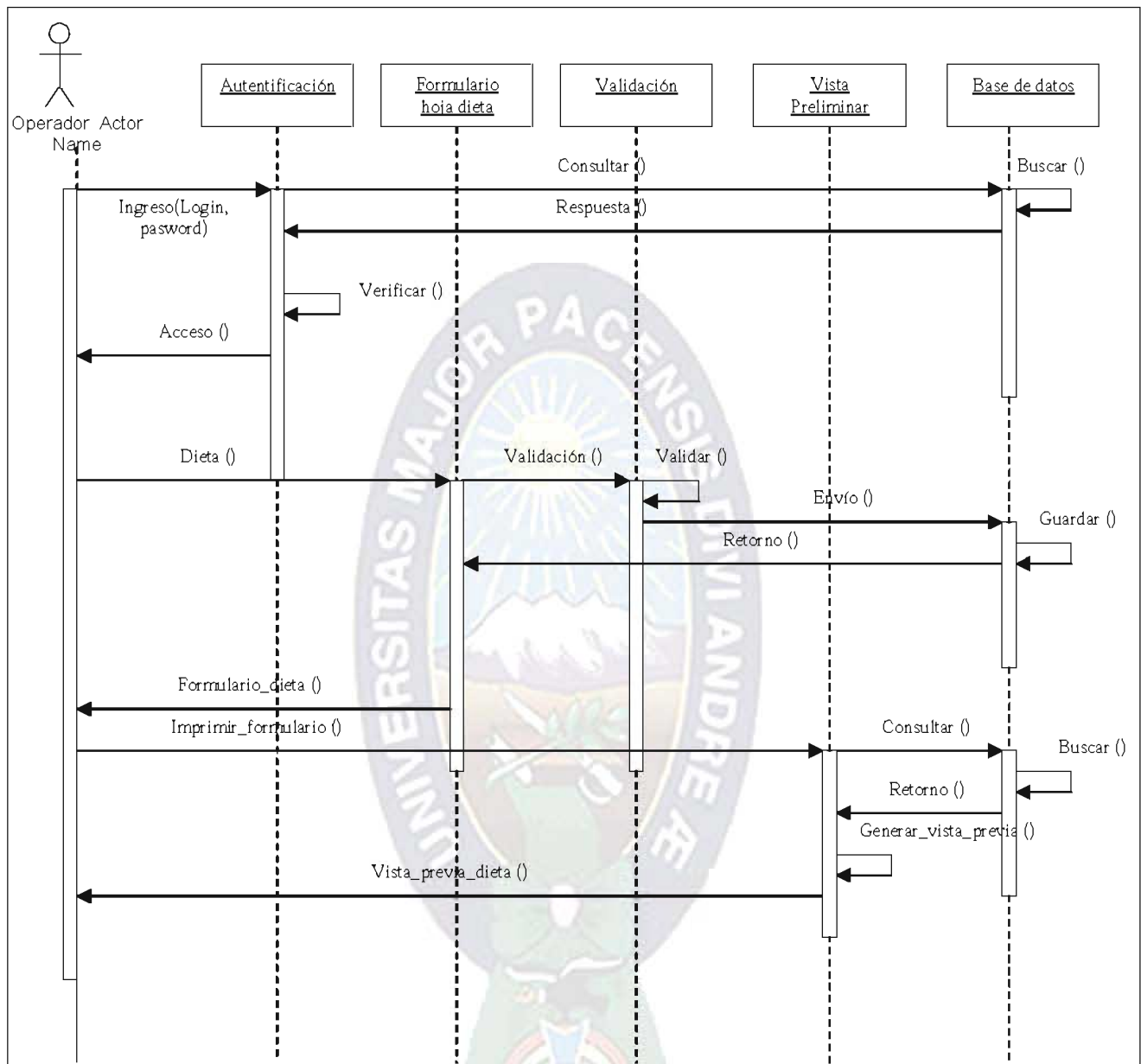


Figura 3.7 Diagrama de Secuencia – Elaboración de hoja dieta
 Fuente: Elaboración propia

b) Diagrama de clases

Figura 3.8 Diagrama de Clases SISECON



3.2.2.1 MODELO FISICO

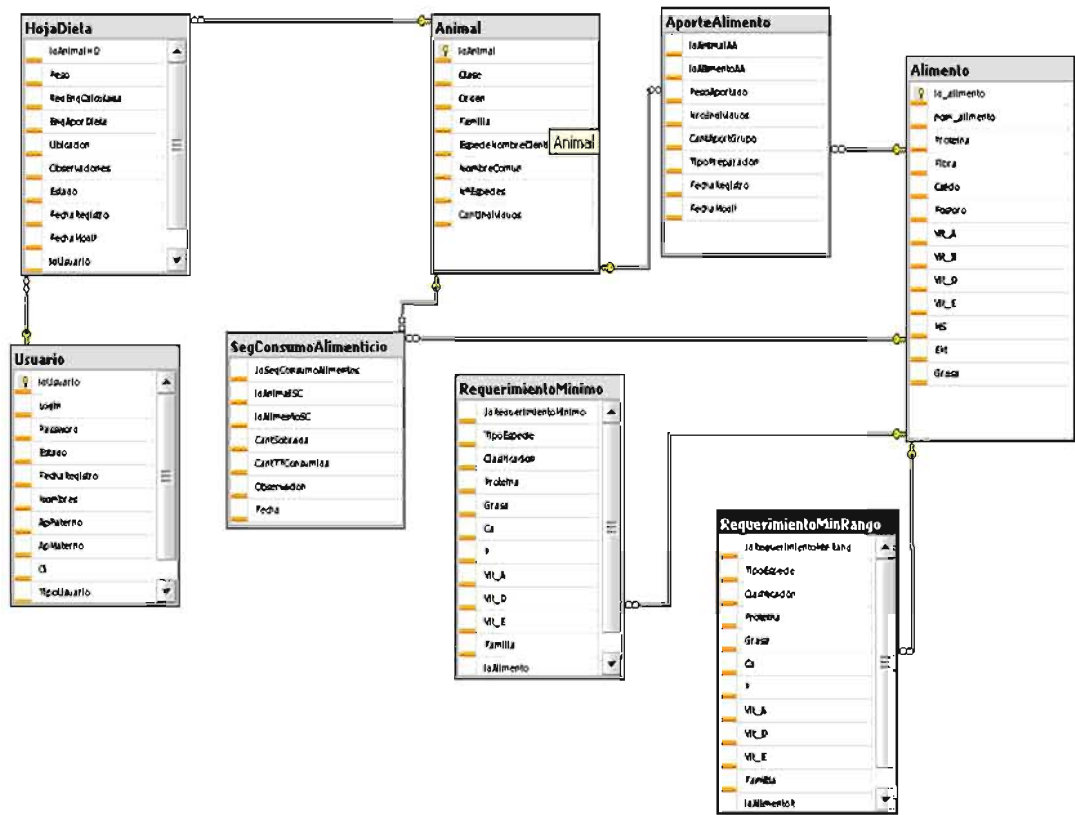


Figura 3.9 Modelo físico SISECON
Fuente: Elaboración propia

3.3 FASE DE CONSTRUCCION

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

a) diagrama de componentes

3.3.1 INTERFASES DE USUARIO

Las interfaces de usuario es un medio de comunicación entre el usuario y el sistema, se debe resaltar que las interfaces diseñadas en el presente sistema son desarrolladas entre el usuario final y el programador.

A. ACCESO AL SISTEMA

El sistema tiene dos tipos de acceso a el que son:

- Acceso directo (sin contraseña)
- Acceso indirecto (introduciendo clave o contraseña)

La siguiente interface muestra cual es la restricción para cada tipo de usuario:



Figura 3.10 Inicio de sesión
Fuente: Elaboración propia

Para cada inicio de sesión o acceso al sistema, el usuario deberá ingresar su nombre de *usuario* y su respectiva *clave*, estos datos son propios y únicos de cada usuario, no pudiendo repetirse por algún otro usuario.



Inicio de Sesión

Bienvenido.

El Acceso a este Sistema requiere, que proporcione un nombre de inicio de sesión y contraseña

Usuario admin

Clave *****

Entrar Salir

Figura 3.11 Ingreso de contraseña
Fuente: Elaboración propia

B. MENU PRINCIPAL

Existen dos tipos de usuarios, el administrador y usuario. El administrador puede hacer cualquier tipo de adición, modificación y eliminación con respecto a los datos de los usuarios, también tiene acceso completo a los datos de los valores nutricionales de los alimentos:



Figura 3.12 Menú principal

Fuente: Elaboración propia

C. ELABORACIÓN DE HOJA DE DIETA

Un papel importante que desempeña el nutricionista es la elaboración de las hojas de dietas para cada especie albergada en el Zoológico Municipal de La Paz, para lograr este objetivo primero introducimos el área al cual corresponde, luego exportamos de la base de datos *Lista de Animales* y muestra las características principales del animal en la hoja de dieta.



Figura 3.13 Elaborar Hoja Dieta

Fuente: Elaboración propia

El sistema muestra un listado de todos los animales albergados en el zoológico al usuario, el usuario debe elegir para que tipo de **animal** desea realizar la hoja de dieta, posteriormente devuelve la taxonomía de cada animal.

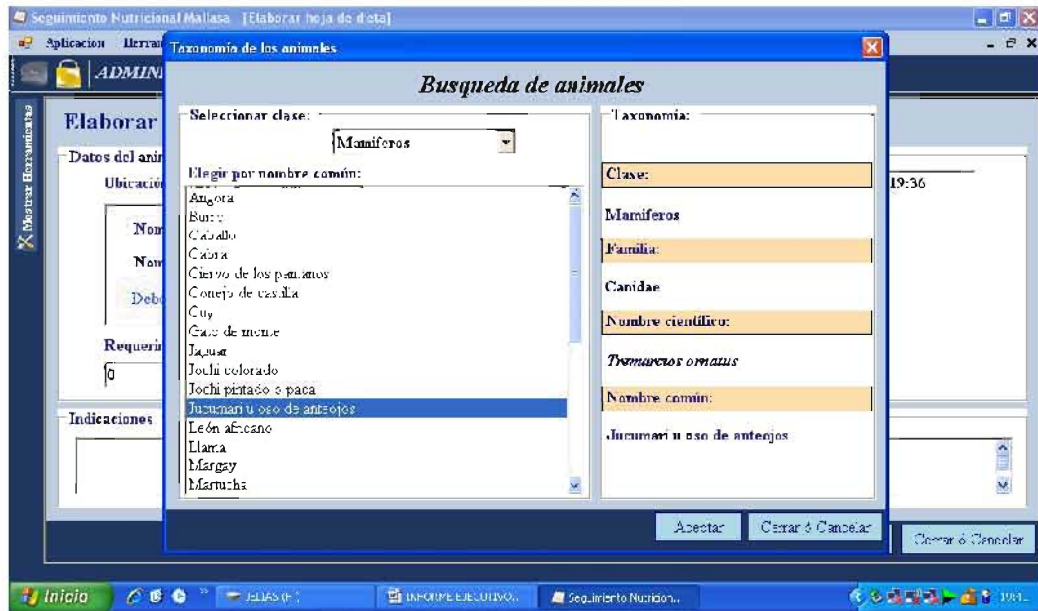


Figura 3.14 Taxonomía de los animales
Fuente: Elaboración propia

Un paso muy importante en la elaboración de la hoja de dieta es encontrar cual es el *Requerimiento de energía calculada* o también conocido como la *Energía Metabólica*. El sistema a través de cálculos específicos busca esta energía metabólica para cada especie (aves, mamíferos y reptiles), el usuario lo único que debe introducir es el *peso estimado vivo del animal* por teclado y el sistema le devolverá el resultado de la energía metabólica.

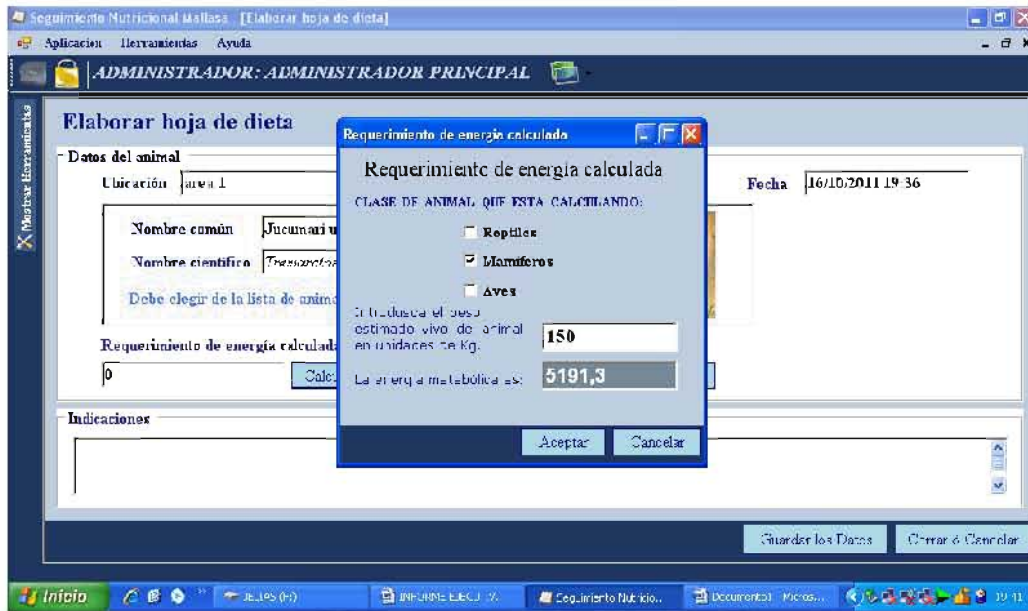


Figura 3.15 Requerimiento de energía calculada

Fuente: Elaboración propia

Las fórmulas utilizadas en el cálculo de requerimiento de energía son las siguientes:

$$TMB=57.6 * (PV)^{0.76}$$

Para el caso de AVES:

$$EM=TMB*2$$

Para el caso de MAMÍFEROS

$$EM=TMB*2$$

Para el caso de REPTILES

$$EM=TMB*0.15$$

Donde:

TMB= Tasa Metabólica Basal

PV= Peso estimado vivo del animal (Kg)

EM= Energía Metabólica (Kcal)

Tasa Metabólica Basal.- Es la cantidad de energía requerida por un organismo para mantener las funciones básicas celulares.

Energía de la dieta.- La energía bruta (EB) de un alimento es la energía producida cuando una muestra del alimento se combustiona completamente en condiciones de laboratorio. Las calorías actuales disponibles para el animal luego de la digestión, se denominan energía digestible y energía metabolizable.

D. CALCULO DE LA ENERGIA APORTADA POR LA DIETA

En cada cálculo de “Energía Aportada por la dieta” es muy importante introducir el número de animales para el cual se realiza la dieta así solo sea un solo animal. Cada especie tiene su respectivo “requerimiento mínimo” almacenado en la base de datos y es mostrado al nutricionista para que evalúe cual es la mejor dieta alimentaria con sus respectivos productos alimentarios.

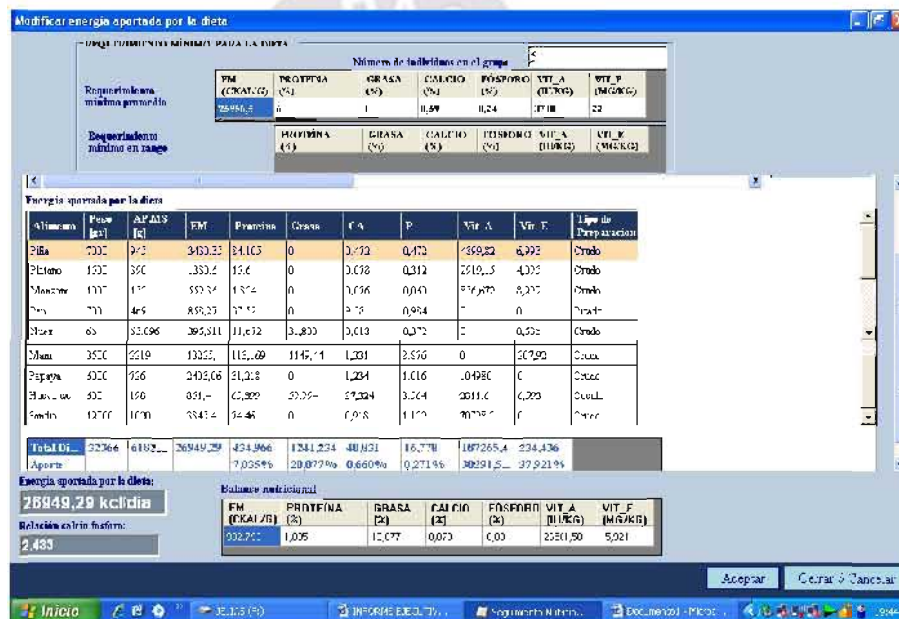


Figura 3.16 Energía aportada por la dieta
Fuente: Elaboración propia

Esta pantalla física del sistema muestra tres características importantes:

1. Requerimientos nutricionales mínimos para la dieta (recomendado por Ellen S. Dierenfeld, 1996)
2. Energía aportada por la dieta (valores encontrados de acuerdo a la dieta)
3. Balance Nutricional (la diferencia de los valores aportados en la dieta menos lo recomendado)

Una vez introducido los alimentos en la hoja de dieta, podemos añadir cualquier observación por el usuario y luego guardamos la Hoja de dieta en la base de datos.

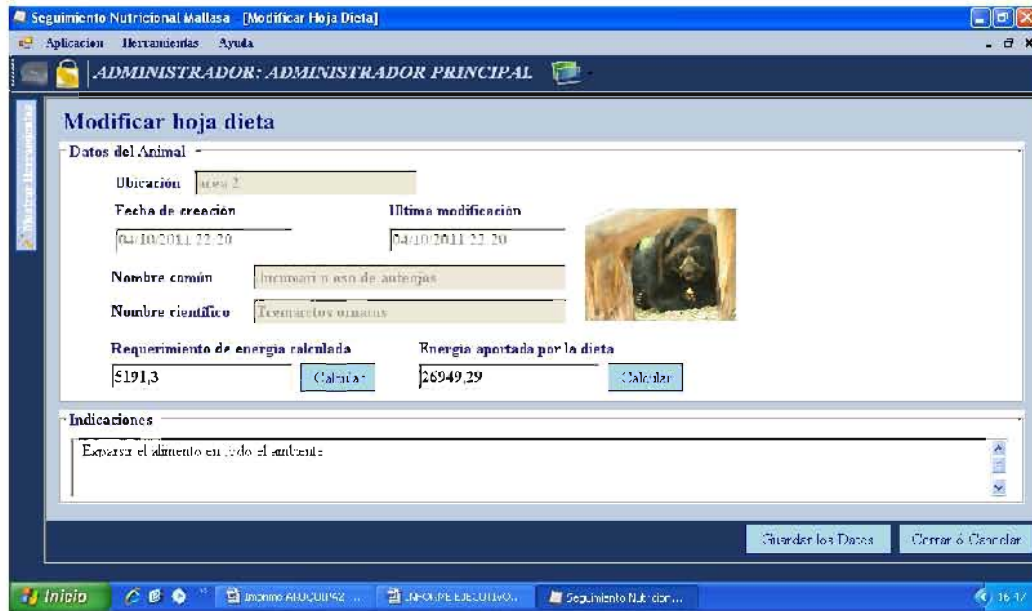


Figura 3.17 Cálculo de la energía aportada por la dieta
Fuente: Elaboración propia

E. MODIFICACION DE HOJAS DE DIETAS ALIMENTARIAS

En la vida cotidiana existe de cuatro a cinco temporadas de frutas que están divididas en:

1. Enero-marzo
2. abril-junio
3. julio-septiembre
4. octubre-diciembre
5. diciembre-febrero

Las temporadas son variables relativamente, por eso la necesidad de hacer uno, dos hasta tres modificaciones en la elaboración de hojas de dietas con respecto a los productos alimentarios, entonces de un mes a otro el nutricionista puede eliminar y/o añadir el producto alimentario que este a su alcance sin repetir todo el proceso de generar nueva hoja de dieta alimentaria.

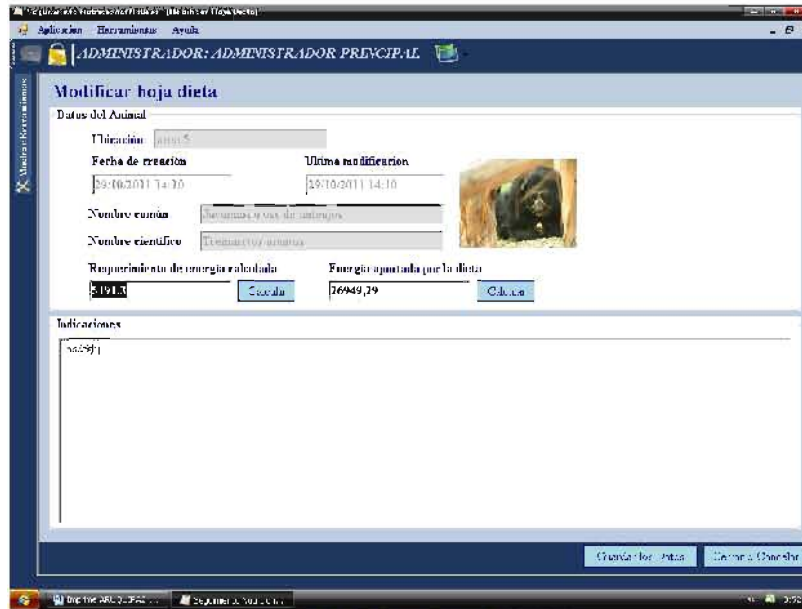


Figura 3.18 Modificar hoja dieta
Fuente: Elaboración propia

F. LISTADO DE ANIMALES

Mediante la opción de búsqueda el usuario podrá buscar las características propias de cada especie.

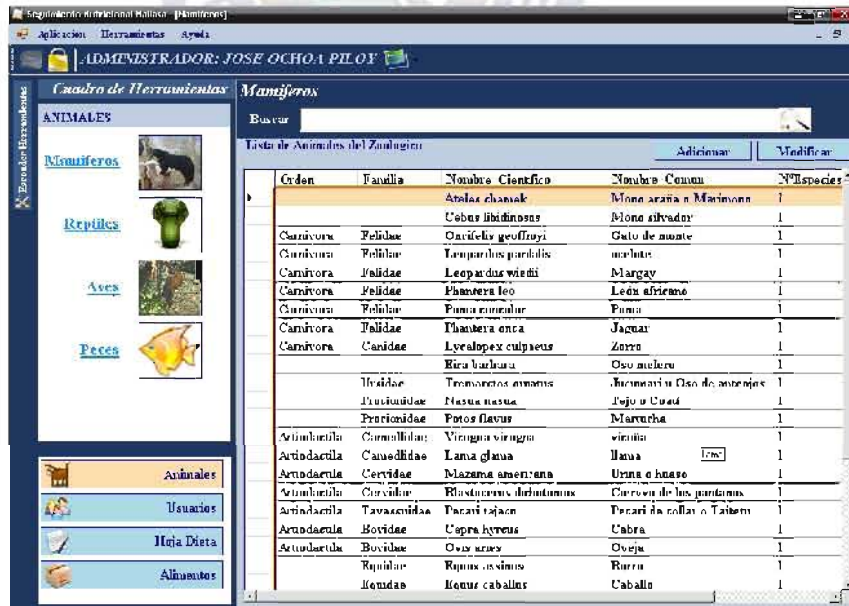


Figura 3.19 Listado de animales
Fuente: Elaboración propia

El administrador podrá adicionar y/o modificar datos propios de cada animal, mediante una interfase incluida en el sistema, este cambio se almacenará en la base de datos.

Adicionar un Animal

Mamiferos

Tipo de Especie

Orden

Familia

Datos Necesarios

Especie ó Nombre Científico

Nombre Comun

Numero de Especies

Cantidad de Individuos [Limpiar](#)

Figura 3.20 Adicionar animal
Fuente: Elaboración propia

Modificar datos un Animal

Mamiferos

Tipo de Especie

Orden

Familia

Datos Necesarios

Especie ó Nombre Científico

Nombre Comun

Numero de Especies

Cantidad de Individuos

Figura 3.21 Modificar datos de animal
Fuente: Elaboración propia

F. ADMINISTRACION DE USUARIOS

El administrador es el único capaz de adicionar, modificar y/o eliminar datos de los usuarios registrados en el sistema



Figura 3.22 Lista de usuarios
Fuente: Elaboración propia



Figura 3.23 Adicionar usuarios
Fuente: Elaboración propia

F. ADMINISTRACION DE ALIMENTOS

La base para realizar las dietas alimentarias es la variedad de alimentos que existe en cada temporada, es por eso que cada vez que el nutricionista vea conveniente adicionar un nuevo producto alimentario, podrá introducir los valores nutricionales de ese producto a través de una interfase denominada lista de alimentos. Este cambio será almacenado en la base de datos una vez introducido sus valores nutricionales.



Administración de Alimentos

ADMINISTRADOR: JOSE OCTAVO PILOY

Buscar por el Nombre del Alimento

Lista de Alimentos

Nombre del Alimento	Proteína	Fibra	Calcio	Fosforo	Vitamina A	Vitamina B
arveja	25,5	0	0,13	0,51	70	0
alfalfa fresca	6,7	0	0,46	0,31	0	0
almendra	23,1	0	0,25	0,57	0	0
arveja (vainas)	15	0	0	0	0	0
humana	4	0	0,03	0,08	7485	0
carrau	20,1	0	0,05	0,16	2048	0
...

Adicionar Modificar

El Sistema de Seguimiento Nutricional está encargado del control de la dieta de todos los Animales del Zoológico.

Hija Dieta Animales Leucos Alimentos

Datos del Alimento

Nombre:

Proteína: Vitamina A:

Fibra: Vitamina E:

Calcio: Vitamina B:

Fosforo: Fecha:

Carrau:



Figura 3.24 Adicionar alimento
Fuente: Elaboración propia

CAPITULO IV CALIDAD DEL SOFTWARE

4.1 CALIDAD DEL SOFTWARE

En este capítulo se verá el desarrollo de la medición de calidad del software construido, haciendo el uso de la métrica mencionada en un anterior capítulo. La métrica esta basada en el estándar ISO/IEC 9126.

4.1.1 FUNCIONALIDAD

FASE INICIAL

No fue implementado ningún caso de uso

FASE DE ELABORACION

Esta versión es utilizada para la reingeniería de requerimientos, puesto que dada la información por el encargado de área existe la posibilidad de omitir procesos muy importantes en la elaboración de hoja de dieta.

Se elaboro 9 casos de uso de uso

El total de casos de uso son 19

Reemplazando en la *ecuación 1*, tenemos:

$$X=1-A/B$$

$$X=1-9/19$$

$$X=1- (0.47)$$

$$X=0.53$$

Este resultado nos indica que falta el 53 % del sistema.

FASE DE CONSTRUCCION

En esta fase se desarrollaron 5 casos de uso más:

Reemplazando en la *ecuación 2*, tenemos:

$$Y=1-A/C$$

$$Y=1-14/19$$

$$Y=1-0.73$$

$$Y=0.27$$

Con este resultado podemos identificar que falta el 27 % del sistema.

FASE DE TRANSICIÓN

En el inicio de la fase de transición todos los casos de uso que son en total 19, por consiguiente se tiene la *ecuación 3*, reemplazando tenemos:

$$Z=1-A/D$$

$$Z=1-1/19$$

$$Z=0.95$$

Este resultado indica que el 95 % de los casos de uso se implementaron.

ADECUACION FUNCIONAL

Esta dada por la siguiente fórmula:

$$X=1-A/B \text{ (ecuación 4)}$$

Donde:

A= Número de funciones (casos de uso) en las cuales se detectaron problemas en la evaluación.

B= Número de funciones (casos de uso) evaluadas.

En esta revisión se detectaron problemas en los flujos de trabajo de pruebas en los siguientes casos de uso:

1. Requerimiento de Energía calculada
2. Registro del animal (nombre común)
3. Requerimiento mínimo promedio

Reemplazando el número de casos de usos en las que se detectaron problemas en la *ecuación 4*, tenemos:

$$X=1-3/19$$

$$X=1-0.15$$

$$X=0.85$$

Tenemos 85% de adecuación del sistema, es lo que nos indica esta cifra encontrada.

Los datos encontrados en la fase de transición, se debe a que el sistema fue sometido a un control.

Por los resultados obtenidos podemos deducir que el **sistema es funcional**.

4.1.2 FIABILIDAD

Esta evaluación es muy importante y es medido por las siguientes métricas:

LEVANTAMIENTO DE DEFECTOS

Como primer paso calculamos el número de defectos encontrados y corregidos en la etapa de diseño/codificación. Con la siguiente fórmula tenemos:

$$X=A$$

A= Número de defectos corregidos en diseño/codificación

$$Y=A/B \text{ (ecuación 5)}$$

Donde:

A=Número de defectos corregidos en diseño/codificación

B = Número de defectos detectados en las revisiones.

Haciendo una evaluación de defectos corregidos en la etapa de diseño y/o codificación tenemos que:

$$A=53$$

$$B=74$$

Reemplazando en *ecuación 5*, tenemos:

$$Y=53/74$$

$$Y=0.71$$

El resultado indica que el 71 % de defectos fueron corregidos en diseño/codificación

DENSIDAD DE DEFECTOS

La densidad de defectos es evaluada con la siguiente fórmula:

$$X= 1-A/B \text{ (ecuación 6)}$$

A = Número de defectos que no fueron corregidos
B=Tamaño del producto (en líneas de código).
Reemplazando en *ecuación 6*, tenemos:

$$X= 1-A/B$$

$$X=1-(74-53)/14018$$

$$X=0.998$$

Con este resultado se puede evidenciar que la densidad de defectos es casi nula, es decir que existe un 99% de efectividad de corrección de errores.

Defectos detectados en el periodo de prueba.

$$X=A/B \text{ (ecuación 7)}$$

A= Número de defectos detectados.

B = Tamaño del producto

Haciendo el reemplazo en la *ecuación 7*, tenemos:

$$X= 53/14018$$

$$X= 0.003$$

El porcentaje que se encontró es de 0.03 % en todo el proceso de elaboración del sistema, es decir que los problemas encontrados en el proceso de elaboración fueron subsanados en su totalidad.

La conclusión que tenemos es el elevado índice de corrección de errores que es del 71 %, una efectividad de corrección de errores del 99 %, y por último un error por línea de código de 0.03%, con un profundo grado de fiabilidad.

Por lo tanto la fiabilidad del sistema es del 99.97 %

4.1.3 FACILIDAD DE MANTENIMIENTO

La fórmula para realizar este cálculo esta dada por la métrica del Índice de Madurez, esta métrica mide la estabilidad del producto:

$$IMS=(M_t - (F_a+F_c+F_d))/ M_t \text{ (Ecuación 8)}$$

Donde:

M_t = Número de módulos en la versión actual

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

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

F_d = Número de módulos en la versión anterior que se han borrado en la versión actual

Esta fórmula debe ser aplicada en cada una de las versiones elaboradas hasta llegar al producto final o sea la versión beta y como resultado tenemos:

$$M_t = 10.$$

$$F_c = 1.$$

$$F_a = 0$$

$$F_d = 1$$

Reemplazando estos valores, tenemos:

$$IMS = (10 - (0+1+1))/ 10$$

$$IMS = 8/10$$

$$IIMS = 0.80$$

Por lo tanto el índice de madurez del sistema se aproxima a 1, esto indica que el producto es estable en un 80%.

4.1.4 PORTABILIDAD

Utilizaremos la métrica de facilidad de instalación, que básicamente calcula el porcentaje de los usuarios que realizan esta operación.

$$X=A/B \text{ (ecuación 9)}$$

Donde:

A=Número de casos en que el usuario es exitoso en la operación de instalación.

B=Número total de casos en que el usuario intenta ejecutar la operación de instalación.

Haciendo una prueba de instalación con el usuario final se pudo evidenciar lo siguiente:

$$A=8$$

$$B=9$$

$$X=8/9$$

$$X=0.88$$

Las probabilidades de hacer una instalación exitosa es del 88 %, por lo tanto el sistema es portable.

4.1.5 USABILIDAD

Para determinar el factor de usabilidad haremos uso de tres métricas las cuales nos ayudaran a decidir cuan usable es el sistema, estas métricas son:

COMPLETITUD DE LA DESCRIPCION

$$X=A/B \text{ (Ecuación 10)}$$

A= Número de funciones (casos de uso) o tipo de funciones descritas en la descripción del producto.

B = Número total de funciones (casos de uso) o tipo de funciones.

Reemplazando en la *ecuación 10*, tenemos:

$$X=17/19$$

$$X=0.89$$

Lo cual nos indica que existe el 89 % de entendimiento de parte de los usuarios con respecto a la capacidad del producto, después de leer la descripción del producto.

CONSISTENCIA OPERACIONAL

$$X= 1-A/B \text{ (Ecuación 11)}$$

A = Número de instancias de operaciones con comportamiento inconsistente.

B = Número total de operaciones

$$X=1-(17/85)$$

$$X=0.8$$

Este indicador nos dice que el 80 % del sistema que no tiene instancias de operaciones con comportamiento inconsistente.

Por lo tanto la usabilidad del producto es de 84.5 %.

En conclusión, el sistema tiene la siguiente métrica de calidad:

Descripción	Promedio
Funcional	85%
Fiabilidad	99.97%
Mantenimiento	80%
Portabilidad	88%
Usabilidad	84.5%

Funcional (85 %).- Significa que el sistema tiene un 85 % de probabilidad que funcione sin riesgos a fallos de operatividad y un 15 % de que el sistema colapse.

Fiabilidad (99,97 %).- Este resultado significa que existe un 0,03 % de probabilidad de que el sistema pueda fallar, debido a que puedan existir errores en la entrada de datos de los requerimientos necesarios.

Usabilidad (84,5 %).- Este resultado nos indica que existe el 89 % de entendimiento de parte de los usuarios con respecto a la capacidad del producto.

Mantenimiento (80 %).- Este resultado significa que el sistema es un 80 % estable, y que la facilidad de mantenimiento en la corrección de fallas o errores del sistema, y su adaptación a nuevas necesidades.

Portabilidad (88%).- Las probabilidades de hacer una instalación exitosa es del 88 %, por lo tanto el sistema es portable.

4.2 ANALISIS DE COSTOS

En este punto se presenta una estimación del costo que implica el desarrollo e implementación del Sistema de Seguimiento y Control de Nutrición. Para tal efecto realizaremos un análisis de estimación de costos. Este procedimiento cuantificará la

inversión económica de cada uno de los recursos que se emplearán para el desarrollo del sistema. Los costos están divididos en 2 grupos: *costos directos* y *costos indirectos*.

4.2.1 COSTOS DIRECTOS

Son aquellos recursos utilizados en el desarrollo del sistema desde el inicio hasta su finalización y entrega al usuario.

Costo del estudio del Sistema: En este punto, el factor más importante, es el tiempo que se toma en cuenta en la construcción del sistema, el tiempo estimado para llevar a cabo esta tarea es aproximadamente 5 meses.

En nuestro medio, el salario que percibe un Informático promedia entre los 1800 a 5000 Bs. o el equivalente en dólares americanos 265 a 730 , por lo tanto el costo estimado para la construcción del sistema sería:

- ♦ Tiempo equivalente en meses es de 5.
- ♦ Costo de estudio del sistema = 5 (meses) * 265 (\$us) = 1325 (\$us)

Costo de mano de obra: Para la programación e implementación del sistema adicionalmente es necesario contratar los servicios de otro personal, entre los que se tienen:

DESCRIPCION	SALARIO (\$us)
Ingeniero de Software	300
Programados en sistemas	130
Transcriptor	80
TOTAL	510

Este costo no es tomado en cuenta por tratarse de un *proyecto de grado* que no tiene fines de lucro, si no el apoyo mutuo correspondido por la sociedad y sus diferentes convenios entre universidades e instituciones públicas.

Costo estimado del Hardware y Software: En este punto, el gobierno Autónomo Municipal de La Paz, dio a conocer que cuenta con las respectivas licencias de software, además de contar con el hardware necesario.

Esfuerzo y costo estimado del desarrollo de software: Tomado en cuenta las características que tendrá el sistema, se decidió utilizar el modelo COCOMO básico, para la implementación de costes del sistema.

Proyecto del Software	a_b	b_b	C_b	d_b
Orgánico	2.4	1.05	2.5	0.38
Semi-acoplado	3.0	1.12	2.5	0.35
Empotrado	3.6	1.20	2.5	0.32

Con el modelo COCOMO se realizara el cálculo del esfuerzo para el desarrollo del sistema en función del tamaño del programa expresado en líneas de código (LDC).

Según la complejidad intermedia y tamaño del proyecto, este será desarrollado en el modo semi-acoplado.

Estimación LDC: Tomando en cuenta el enfoque directo, el tamaño de software se puede medir en líneas de código (LDC). Considerando el número de componentes estándar, como son los módulos, reportes, pantallas, etc., del sistema además de contar con la experiencia y la información de proyectos históricos podemos estimar el tamaño del software.

FUNCION	LDC ESTIMADA
Interfaz de Usuario	3500
Gestión de base de datos	2800
Acceso a consultas	1800
Facilidad para la impresión de la información	900
	9000 LDC estimadas

Estimación del esfuerzo necesario:

$$E = a_b * LDC^b = 3 * (10)^{1.12} = 34 \text{ personas /mes}$$

Estimación del tiempo necesario:

$$D = c_b E^d = 2.5 * (34)^{0.35} = 9 \text{ meses}$$

Numero de personas para el proyecto:

$$N = E / D = 34 / 9 = 4 \text{ personas}$$

4.2.1 COSTOS INDIRECTOS

Estos costos solo serán de apoyo, no se implicaran dentro del proyecto, ya que solo son materiales extras que son necesarios para el desarrollo del sistema.

Materiales de apoyo

MATERIAL	TOTAL (Bs)
Material de Escritorio	200
Otros	600
Total	800

El monto estimado es de 800 bolivianos o su equivalente 117 (\$us) dólares americanos.

4.3.1 COSTO TOTAL DEL PROYECTO

Este costo se determina con suma del costo directo más el costo indirecto.

COSTOS	DETALLE	MONTO
Directos	Estudio del sistema	1325
	Mano de obra	0
	Costo estimado del hardware	0
	Costo estimado del Software	0
Indirectos	Materiales de apoyo	117
TOTAL		1442

Con esta estimación de costos, se puede ver que el desarrollo del sistema es factible para el Gobierno Autónomo Municipal de La Paz.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Concluido el presente trabajo y puesto en funcionamiento el sistema, se alcanzo el objetivo general, que contribuye en la elaboración de dietas y realiza el seguimiento y control de nutrición para los animales silvestres, debido a que el sistema fue sometido a pruebas técnicas que dio resultados esperados por el usuario final, con lo que se logro automatizar el manejo y gestión de la información. De tal manera que el uso de herramientas y metodologías de desarrollo de software descritas en anteriores capítulos contribuyo de forma efectiva.

Por otro lado se cumplió con los siguientes objetivos específicos:

- ♦ Se construyo el modulo *Lista de alimentos*, con este modulo el usuario puede adicionar, modificar el nombre del alimento y sus valores nutricionales.
- ♦ Se creo el modulo *Lista de usuarios*, con el cual se puede adicionar, eliminar y modificar usuarios de tal modo que se pueden crear dos tipos de usuarios que es el de administrador y usuario común.
- ♦ Se construyo el modulo *Animales*, que cuenta con tres eventos que son la de elección de; MAMIFEROS, AVES y REPTILES en cada una de ellas se puede realizar la búsqueda por nombre común, también se puede realizar la adición, modificación y eliminación de datos de los animales registrados en la base de datos.
- ♦ Se elaboro el modulo *Hoja Dieta*, que ayuda en la tarea de la elaboración de dietas, ya que el sistema realiza la importación de los alimentos con sus respectivos valores nutricionales, requerimiento mínimo nutricional para cada especie, requerimiento mínimo nutricional promedio, todos estos datos son extraídos de nuestra base de

datos y luego son procesados por el sistema de acuerdo a formulas utilizadas en el campo de nutrición.

- ♦ Se creo el modulo *Seguimiento nutricional*, el cual permite realizar el seguimiento y de consumo de alimentos, almacenándolo en la base de datos para posteriores consultas y nueva toma de decisiones.
- ♦ Se logro construir y estructurar una base de datos nueva para almacenar toda la información de seguimiento y control nutricional, para esto se hizo uso del gestor de base de datos SQL SERVER 2005.

También se elaboro los siguientes reportes que emite el sistema:

- ♦ *Preparación de dieta*, este reporte filtra la información básica para la elaboración de dietas, de forma que cualquier persona sin grandes conocimientos profesionales pueda entenderlo y luego preparar los alimentos con las cantidades y la forma de preparación indicadas en el reporte.
- ♦ *Composición nutricional*, este reporte filtra la información técnica y específica del balance nutricional, para su respaldo administrativo.
- ♦ *Seguimiento de consumo de alimentos*, este reporte ayuda a evaluar si es conveniente seguir la dieta del animal. De otra manera se procederá a evaluar una nueva dieta.

5.2 RECOMENDACIONES

- ♦ El sistema está elaborado bajo la plataforma Framework 2.0 y la tecnología .NET 2005, para posteriores modificaciones y/o actualizaciones en el sistema se sugiere utilizar la tecnología .NET 2008 en adelante y por consiguiente actualizar el Framework 3.5.
- ♦ Para el mantenimiento del sistema se sugiere actualizar el código fuente con tecnologías recientes del mercado, para su tener mayor usabilidad en lo futuro.
- ♦ Se recomienda realizar backups de la base de datos por lo menos una vez al mes.
- ♦ El sistema es una herramienta elaborado para escritorio y de uso estrictamente personalizado, pero existe la posibilidad de llevarlo vía Web modificando la capa de

presentación, pudiendo así reutilizar los *.dll restantes del sistema y subirlo al ciberespacio.

- ♦ Algunos de los nutrientes de los alimentos no fueron considerados por falta de información científica y de laboratorio, para posteriores estudios de los mismos se recomienda elaborar un sub-modulo para evaluar el balance de los mismos.



BIBLIOGRAFIA

[BOOCH-99] El lenguaje Unificado de Modelado. G. Booch, J. Rumbaugh, I. Jacobson. Addison Wesley Iberoamericana, 1999

[Pressman-2005] Roger S. Pressman “Ingeniería del Software” 5ta Ed. Concepción Fernández Madrid, España, 2005

[Larman-2002] Larman, C. “UML y Patrones: Introducción al Análisis y Diseño Orientado a Objetos” 2da Ed. Prentice Hall, 2002

[Jacobson-00] Jacobson, I., Booch, G., Rumbaugh, J. “El Proceso Unificado de Desarrollo de Software”. Addison-Wesley, 2000

[Navia-1997] Carlos Jorge Navia Alanez, “Elaboración Científica del Perfil de Tesis”. 1ra Ed. EDCON, La Paz-Bolivia, 1997

[Hernández-2006] Roberto Hernández Sampieri “Metodología de la investigación” 4ta Ed. McGRAW-HILL, 2006

[Wallace-2010] Robert B. Wallace, “Distribución, ecología y conservación de los mamíferos medianos y grandes de Bolivia”. Centro de ecología, difusión Simón Patiño, 2010.

[Dierenfeld, Graffman-1996] Ellen S. Dierenfeld, Wendy S. Graffman “Manual de Nutrición y Dietas para animales silvestres en cautiverio”. Wildlife Conservation Society Bronx, NY, 1996

REFERENCIAS ELECTRONICAS

- [Laguna, 2001] Laguna Miguel, 2001: El método de desarrollo, El proceso Unificado
<http://www.infor.uva.es/mlaguna/is2/2-6-Proceso.pdf>
- [Web, 001] Diagramas de UML
<http://jano.unicauca.edu.co/cursos/Especializacion/ApliServicios/docs>
- [Web, 002] Sql Server
<http://www.microsoft.es/sqlserver>
- [Sitio 1] Estándares de seguridad en la información
<http://www.enterate.unam.mx/Articulos/2005/febrero/seguridad.ht>
- [Sitio 2] Estándares de seguridad en la información
<http://www.enterate.unam.mx/Articulos/2005/febrero/seguridad.ht>



ANEXOS

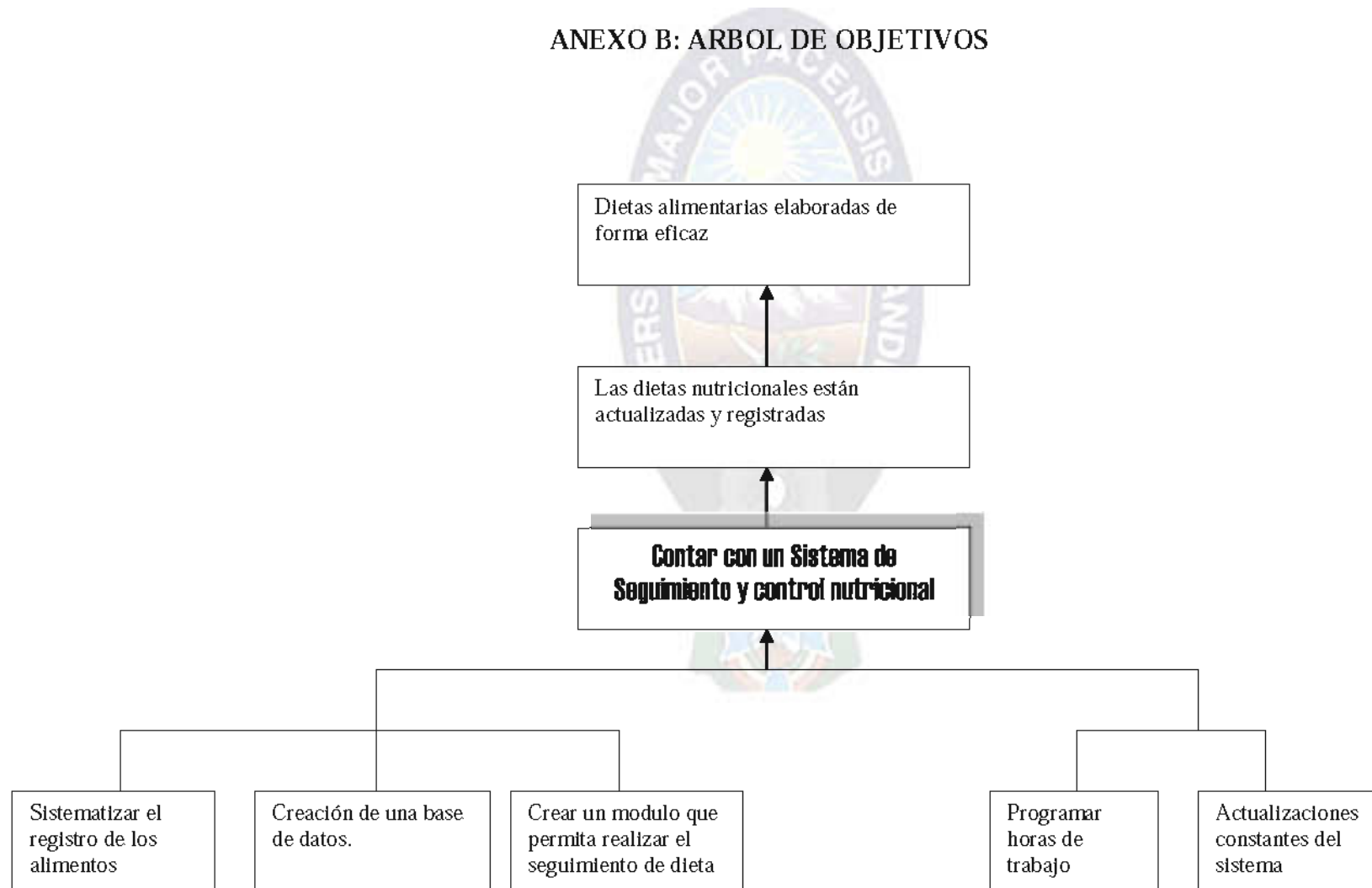
SISTEMA DE SEGUIMIENTO Y CONTROL DE NUTRICION PARA CASO: ZOOLOGICO MUNICIPAL DE LA PAZ

ANEXO A: ARBOL DE PROBLEMAS



SISTEMA DE SEGUIMIENTO Y CONTROL DE NUTRICION PARA CASO: ZOOLOGICO MUNICIPAL DE LA PAZ

ANEXO B: ARBOL DE OBJETIVOS



ANEXO C: MATRIZ MARCO LOGICO

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
FIN Dietas alimentarias elaboradas de forma eficaz	Reportes de dietas constantes	Hacer una revisión constante visual de estas dietas	Animales con salud constante.
PROPOSITO Contar con un Sistema de Seguimiento y control nutricional	Sistema terminado y presentado hasta abril del año 2010.	Aval del tutor Aval del revisor.	Aprobación del proyecto de grado.
COMPONENTES <ol style="list-style-type: none"> a. Reglamento Interno de la Institución y trabajo. b. Modelo de desarrollo de investigación. c. Fundamentos teóricos y prácticos. 	<ol style="list-style-type: none"> 1. Constancia de documentación en archivo. 2. Documento respaldatorio del modelo. 3. Los Médicos quedan satisfechos con los resultados 	<ol style="list-style-type: none"> 1. Administrativos y Veterinarios proporcionan la información. 2. Seguimiento del Veterinario de turno. 3. Seguimiento del Medico de turno. 	<ol style="list-style-type: none"> 1. Se cuenta con el apoyo del Veterinario de turno. 2. Existencia de modelos anteriores y documentos. 3. Conocimiento profundo de las teorías expuestas.

ACTIVIDADES <ol style="list-style-type: none"> 1. Análisis y estudio del proyecto. 2. Desarrollo del modelo 3. Pruebas y análisis del modelo 4. Análisis de resultados 5. Consolidación del sistema. 	<ol style="list-style-type: none"> a. Costo de materiales de escritorio y transporte 200 \$. b. Tiempo de desarrollo 300 \$. c. Solicitud de pruebas y análisis en los animales 500 \$. d. Tiempo invertido 100 \$. e. Complementación con desarrollo 150 \$. 	<ol style="list-style-type: none"> 1. Facturas de gasolina y recibos de compra. 2. Seguimiento personal. 3. Seguimiento personal 4. Seguimiento personal 5. Seguimiento personal 	<ol style="list-style-type: none"> 1. Respaldo de la entidad. 2. Previamente es analizado el tema para el desarrollo. 3. Es aceptado por la Entidad 4. Es aceptado por la Entidad 5. Funcionamiento mínimo del 95 %.

ANEXO D: PRUEBA DEL SISTEMA

INFORME

Nº 1/2011

A : *Lic. Mariana Daza von Boeck*
ADMINISTRADORA ZOOLOGICO
MUNICIPAL DE LA PAZ

VIA : *MVZ. Fidel Fernández Anagua*
RESPONSABLE DE CONSERVACIÓN Y
MANEJO DE VIDA SILVESTRE

DE : *José Elías Ochoa Piloy*

REF : *Informe final sobre el trabajo dirigido del 16 de*
noviembre de 2009 al 5 de octubre de 2011

FECHA : *18 de octubre de 2011*

Dando cumplimiento a la Asignación de Trabajo Dirigido con número DGRH-UCAP-OF. Nº 951/09, tengo a bien elevar el siguiente informe:

1. ANTECEDENTES.

Habiendo sido asignado al Trabajo Dirigido en el Zoológico Municipal de La Paz, se empezó inmediatamente a la recolección de información conjuntamente con MVZ. Victor Humerez, nutricionista del área de Conservación y Manejo de Vida Silvestre supervisado por el MVZ. Fidel Fernández Anagua, para la definición del sistema informático en apoyo a esta área.

2. DETALLE DE LAS GESTIONES TRABAJADAS EN EL AREA DE CONSERVACIÓN Y MANEJO DE VIDA SILVESTRE.

Para lograr la definición del sistema informático, se analizo problemas específicos que tiene esta área y de acuerdo al cronograma elaborado, se empezó con el seguimiento de dietas que tiene la fase de inicio en la preparación de alimentos en cocina (de acuerdo a la dieta elaborada por el nutricionista), posteriormente la distribución de alimentos en ambientes de los animales y la fase final que es la observación y análisis de aprovechamiento de los alimentos.

Habiendo recolectado la información necesaria y analizado el problema y/o necesidad de esta área, se determino elaborar un Sistema de apoyo a la elaboración de dietas y seguimiento de consumo de alimentos para los animales silvestres del Zoológico Municipal

denominado "Sistema de Seguimiento y Control de Nutrición para caso: Zoológico Municipal de La Paz"

La fase de diseño y elaboración del sistema fueron realizados por el software C#.NET y colaborado por el gestor de base de datos SQL SERVER 2005

De acuerdo a lo establecido en el Reglamento Municipal de Prácticas Pre-profesionales (Pasantías y Trabajo Dirigido), los informes elevados en diferentes fechas fueron dirigidos al Lic. Marco Antonio Saavedra Mogro-Director de Gestión de Recursos Humanos, que describe detalladamente el trabajo realizado cada mes vía MVZ. Fidel Fernández Anagua, haciendo registrar por SITRAM.

Debido a la importancia del sistema se determino realizar pruebas al producto final realizando pruebas en diferentes fechas de la gestión 2011, finalizando en una prueba final realizada y documentada en fecha 7 al 21 de septiembre de 2011, siendo presentada al equipo de Conservación y Manejo de Vida Silvestre en fecha 5 de octubre de 2011 como se describe a continuación:

PRUEBA PILOTO

CLASE	:	Mamíferos
FAMILIA	:	Ursidae
NOMBRE COMUN	:	Jucumari u oso de anteojos
NOMBRE CIENTIFICO	:	<i>Tremarctos ornatus</i>
PESO	:	150 Kg
NUMERO DE INDIVIDUOS	:	5

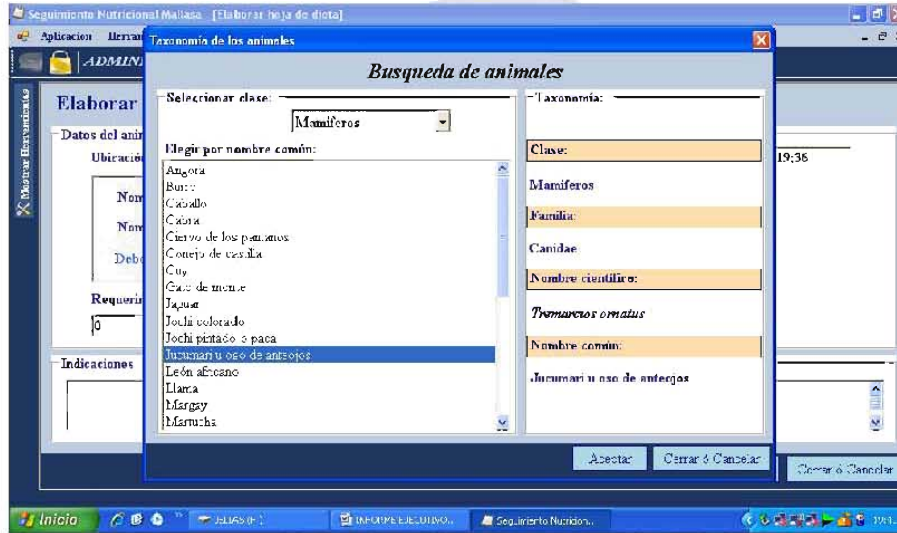
INGRESO AL SISTEMA



Para el ingreso al sistema le solicita un nombre de usuario y su respectiva contraseña.

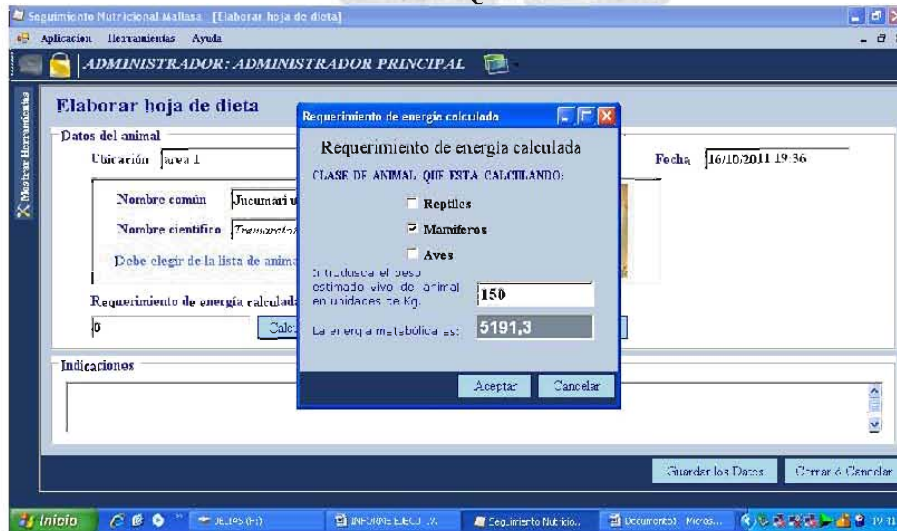
ELABORACIÓN DE DIETA PARA EL OSO JUCUMARI

ELECCION DEL ANIMAL



Esta pantalla le permite seleccionar la clase y el animal, posteriormente el sistema le devuelve la taxonomía del animal.

CALCULO DE LA ENERGÍA REQUERIDA MINIMA



Las fórmulas utilizadas en el cálculo de requerimiento de energía son las siguientes:

$$TMB=57.6 * (PV)^{0.76} \quad (1)$$

Para el caso de AVES:

$$EM=TMB*2 \quad (2)$$

Para el caso de MAMÍFEROS

$$EM=TMB*2 \quad (3)$$

Para el caso de REPTILES

$$EM=TMB*0.15 \quad (4)$$

Donde:

TMB= Tasa Metabólica Basal

PV= Peso estimado vivo del animal (Kg)

EM= Energía Metabólica (Kcal)

Tasa Metabólica Basal.- Es la cantidad de energía requerida por un organismo para mantener las funciones básicas celulares.

Energía de la dieta.- La energía bruta (EB) de un alimento es la energía producida cuando una muestra del alimento se combustiona completamente en condiciones de laboratorio. Las calorías actuales disponibles para el animal luego de la digestión, se denominan energía digestible y energía metabolizable.

ALIMENTOS DE LA DIETA

ALIMENTO	CANTIDAD (gramos)
Piña	7000
Plátano	1500
Manzana	1000
Pan	700
Nuez	66
Maní	3500
Papaya	6000
Huevo con cascara	600
Sandia	12000

REQUERIMIENTOS, NUTRIENTES APORTADOS Y BALANCE DE LA DIETA



Esta pantalla física del sistema muestra tres características importantes:

4. Requerimientos nutricionales mínimos para la dieta (recomendado por Ellen S. Dierenfeld, 1996)
5. Energía aportada por la dieta (valores encontrados de acuerdo a la dieta)
6. Balance Nutricional (la diferencia de los valores aportados en la dieta menos lo recomendado)

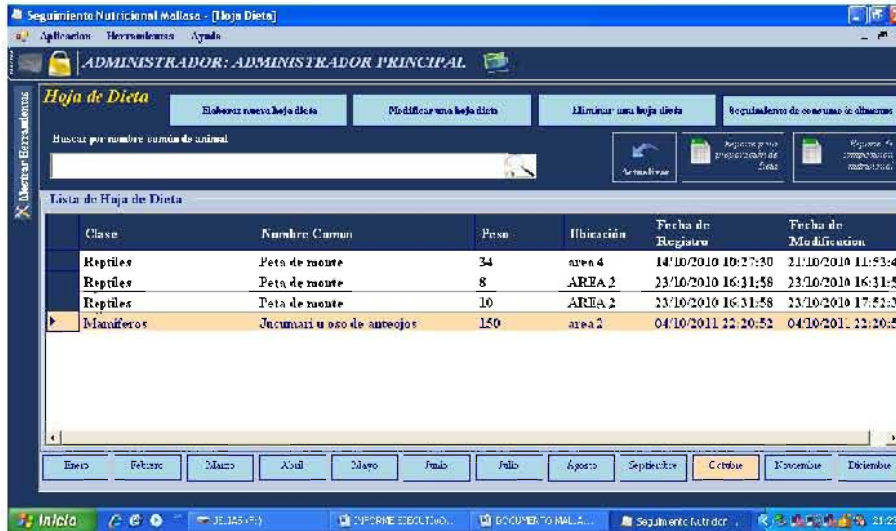
Como se puede observar el aporte total de la dieta cumple los requerimientos mínimos de mantenimiento, obteniendo como balance general un excedente mínimo a lo requerido.

SEGUIMIENTO DE CONSUMO DE ALIMENTOS



Una vez elaborado la dieta para el animal, el sistema elabora el seguimiento de consumo de alimentos que es almacenado en la base de datos para posteriores consultas o dudas en la dieta

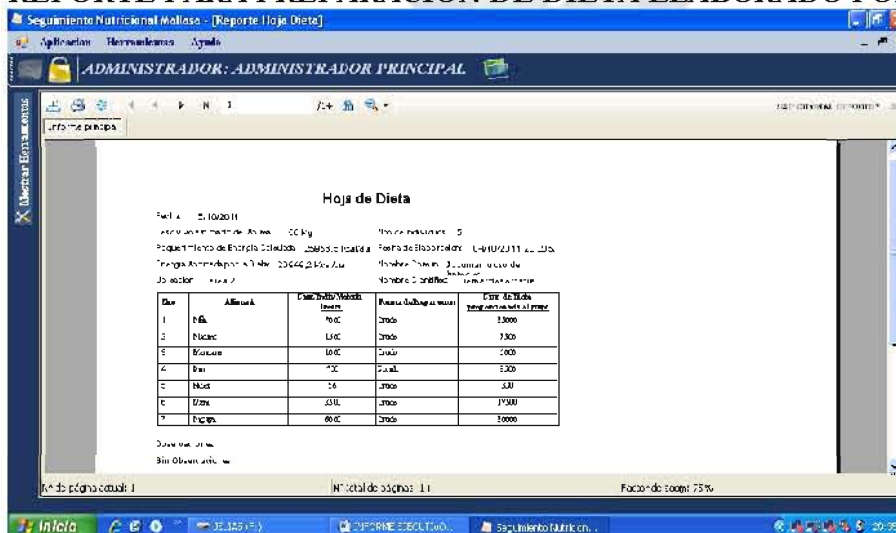
MENU PARA LAS DIFERENTES APLICACIONES



Una vez elaborada y almacenada la hoja de dieta, el sistema nos permite acceder al siguiente menú:

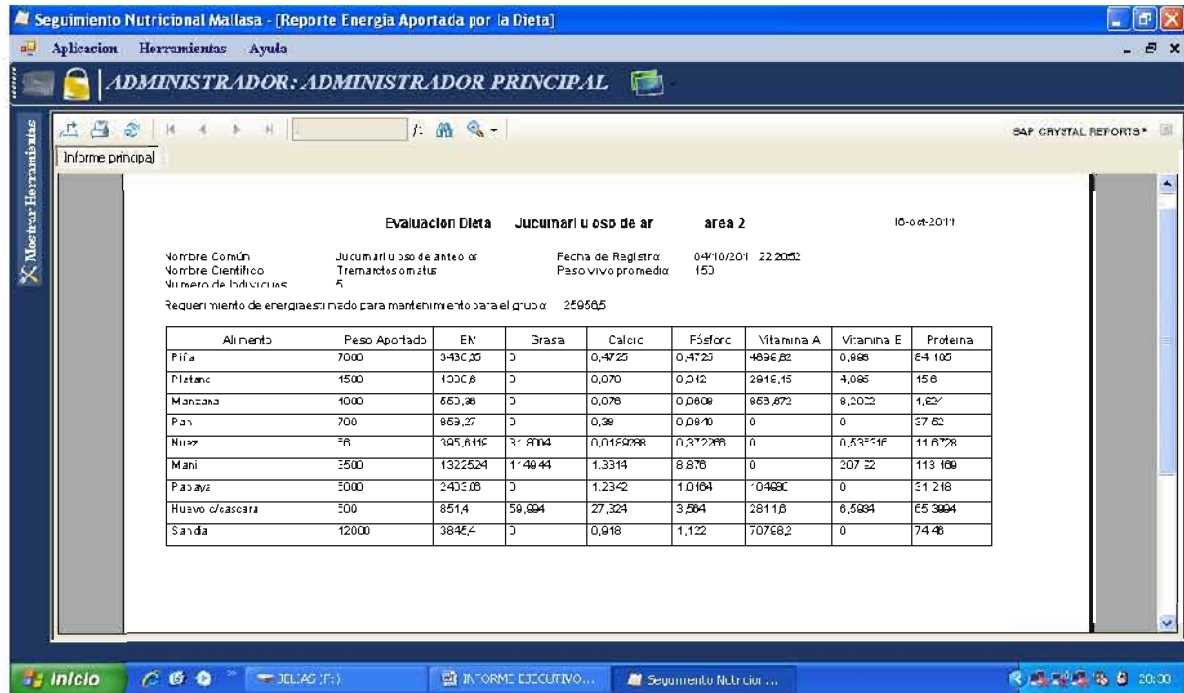
- ◆ *Modificar una hoja de dieta*
- ◆ *Eliminar una hoja de dieta*
- ◆ *Consultar y registrar el seguimiento de consumo de alimentos*
- ◆ *Hacer consultas de las dietas de acuerdo al mes elaborado*
- ◆ *Vista preliminar para la impresión de reporte de preparación de dieta*
- ◆ *Vista preliminar para la impresión de reporte de composición nutricional*

REPORTE PARA PREPARACION DE DIETA ELABORADO POR EL SISTEMA



Hoja lista para ser impresa.

REPORTE DE COMPOSICIÓN NUTRICIONAL ELABORADO POR EL SISTEMA



Seguimiento Nutricional Mallasa - [Reporte Energia Aportada por la Dieta]

Administrador: ADMINISTRADOR PRINCIPAL

Informe principal

Evaluación Dieta Jucumari u oso de ar area 2 10-04-2011

Nombre Común: Jucumari u oso de ar
Nombre Científico: Tremarctos ornatus
Número de Individuos: 5
Fecha de Registro: 04/10/2011 22:20:52
Paso vivo promedio: 150
Requerimiento de energía estimado para mantenimiento para el grupo: 259565

Alimento	Peso Aportado	EN	Grasa	Calcio	Fósforo	Vitamina A	Vitamina E	Proteína
Piña	7000	3430,20	0	0,4723	0,4723	4696,62	0,469	24,100
Piletan	1500	1000,00	0	0,070	0,070	2016,15	4,095	15,6
Manzana	1000	650,00	0	0,078	0,078	863,872	8,202	4,82
Pan	700	859,27	0	0,38	0,0810	0	0	27,62
Niuse	50	105,616	0,816	0,0140998	0,02299	0	0,032516	11,879
Mani	3500	1322,524	1,4944	1,3314	8,876	0	207,22	113,160
Papaya	5000	2400,00	0	1,2342	1,0164	104690	0	21,218
Huevo o/cascara	500	851,4	50,904	27,324	3,504	2811,6	6,5904	65,904
Sanda	12000	3840,0	0	0,918	1,122	707682	0	74,46

Hoja lista para ser impresa.

3. CONCLUSIONES.

De acuerdo a los objetivos planteados, se concluye que el sistema es capaz de evaluar el aporte de nutrientes de la dieta, realizar el balance de los mismos, registrar y almacenar datos de los valores nutricionales de los alimentos, registrar y almacenar datos de los animales, registrar y almacenar datos de los diferentes usuarios para el manejo del sistema y como objetivo principal el sistema apoya la elaboración de dietas como el seguimiento de consumo de alimentos.

El sistema finaliza en la elaboración de reportes que están listas para ser enviadas a una maquina de impresión.

4. RECOMENDACIONES

En el análisis del problema que tuvimos inicialmente con el nutricionista, se pudo evidenciar el problema para la obtención de valores nutricionales de la fibra ya que es un valor que contiene diferentes alimentos; los valores encontrados en los requerimientos hasta ahora aún no satisface las expectativas para la realización de la hoja de dieta. Recomendamos se pueda tomar en cuenta otras investigaciones que consideren este valor

de requerimiento mínimo que fueran analizados y/o estudiados científicamente a partir del año 2011 en adelante.

Es cuanto tengo a bien informar y recomendar para fines consiguientes.

José Elías Ochoa Piloy

Nota: Se adjunta la primera hoja de los 10 informes presentados via SITRAM .



ANEXO E: DIAGRAMA DE ACTIVIDADES

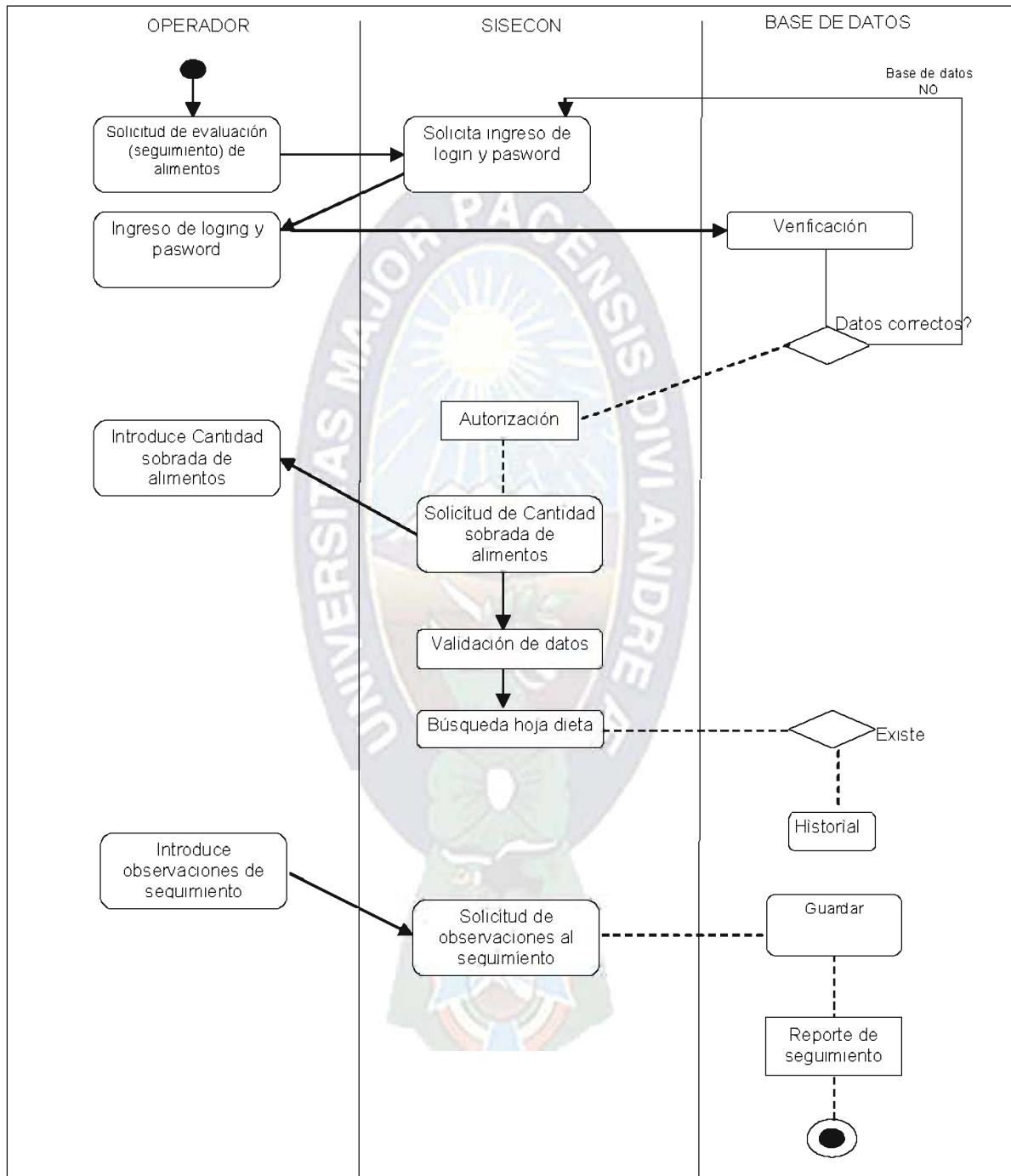


Diagrama de actividades – Evaluación de aceptación de alimentos

Fuente: [Elaboración propia]

ANEXO F: DIAGRAMA DE CASOS DE USO

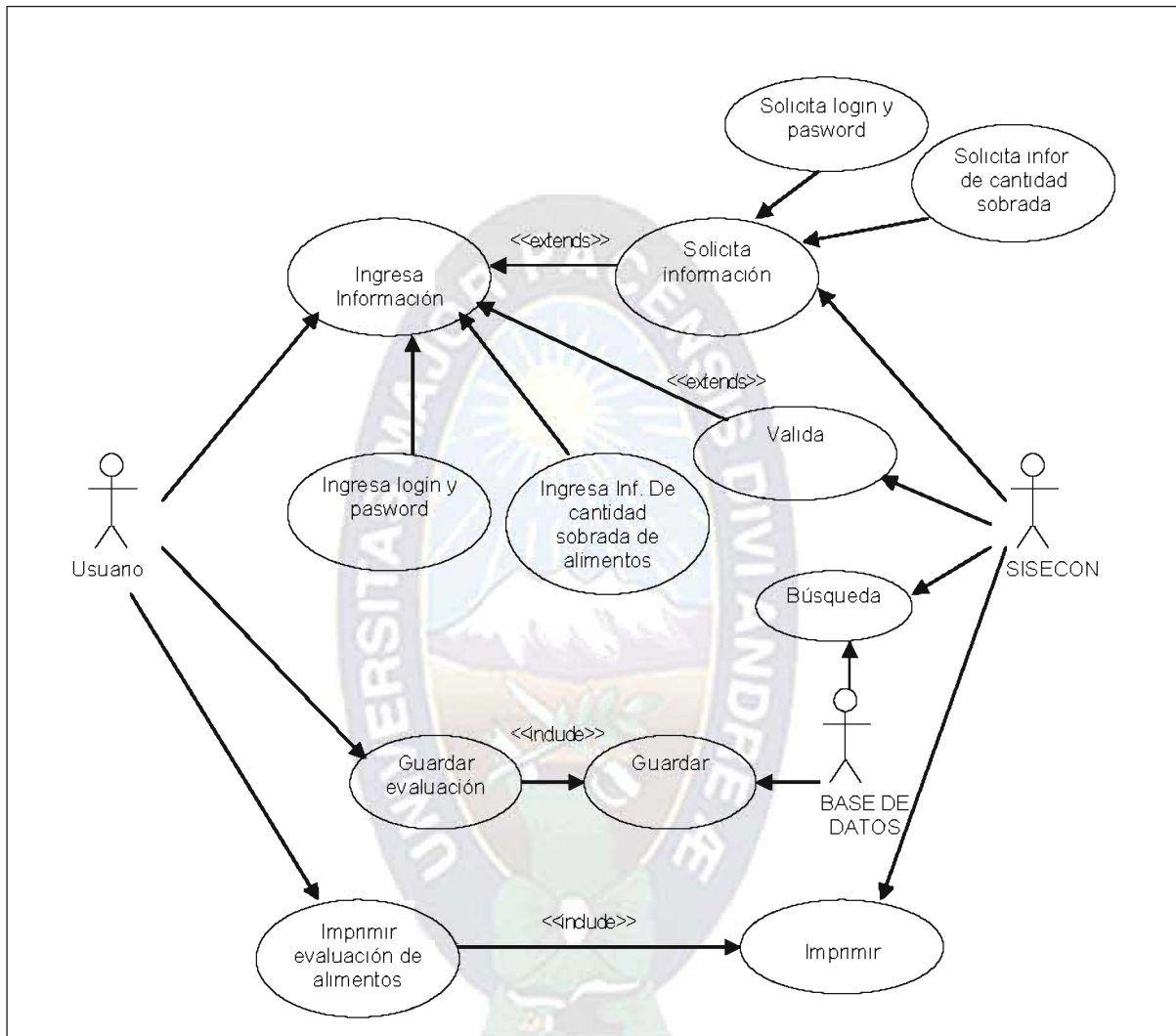


Diagrama de Casos de Uso - Evaluación de aceptación de alimentos
Fuente: [Elaboración Propia]

ANEXO G: DIAGRAMA DE ESTADO

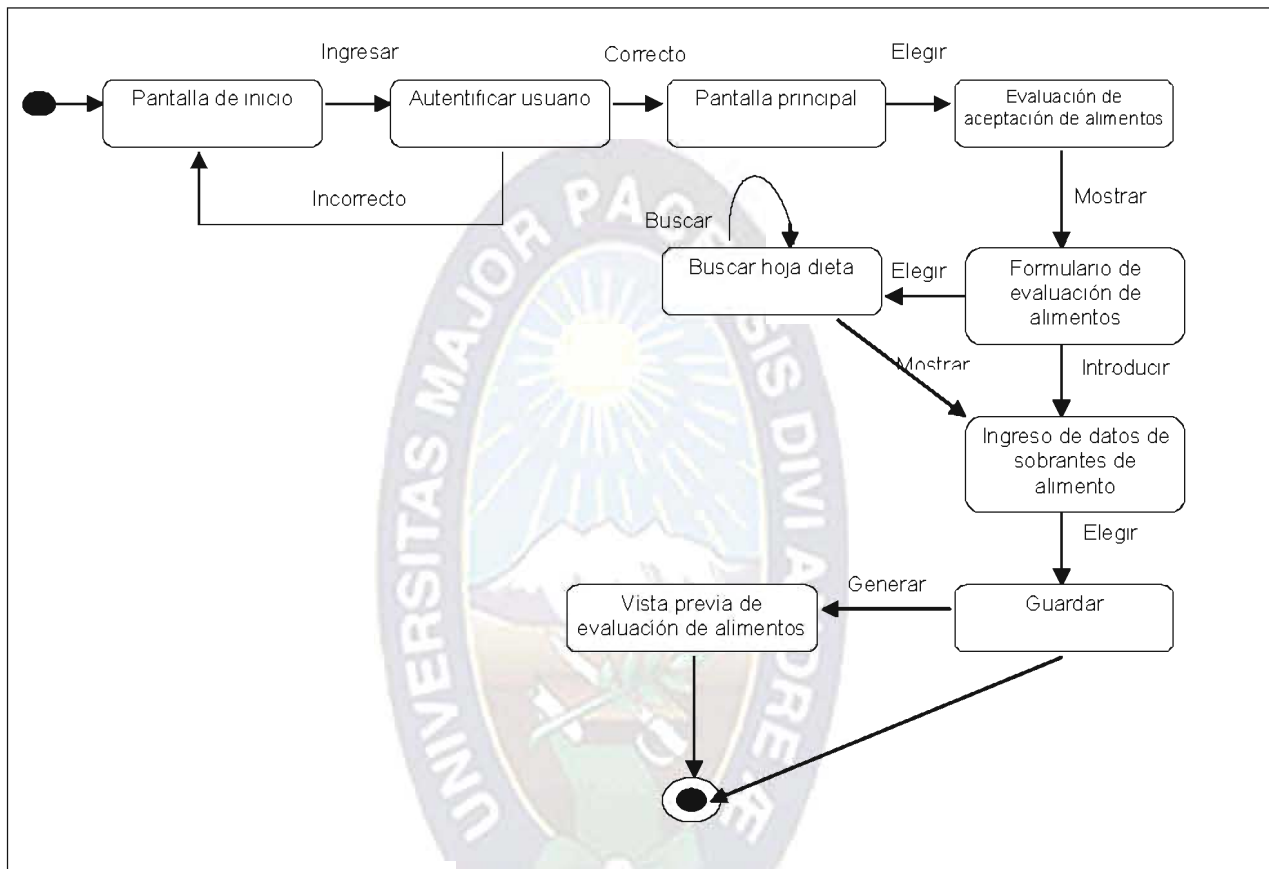


Diagrama de Estado - Evaluación de aceptación de alimentos
Fuente: [Elaboración Propia]

ANEXO H: DIAGRAMA DE SECUENCIA

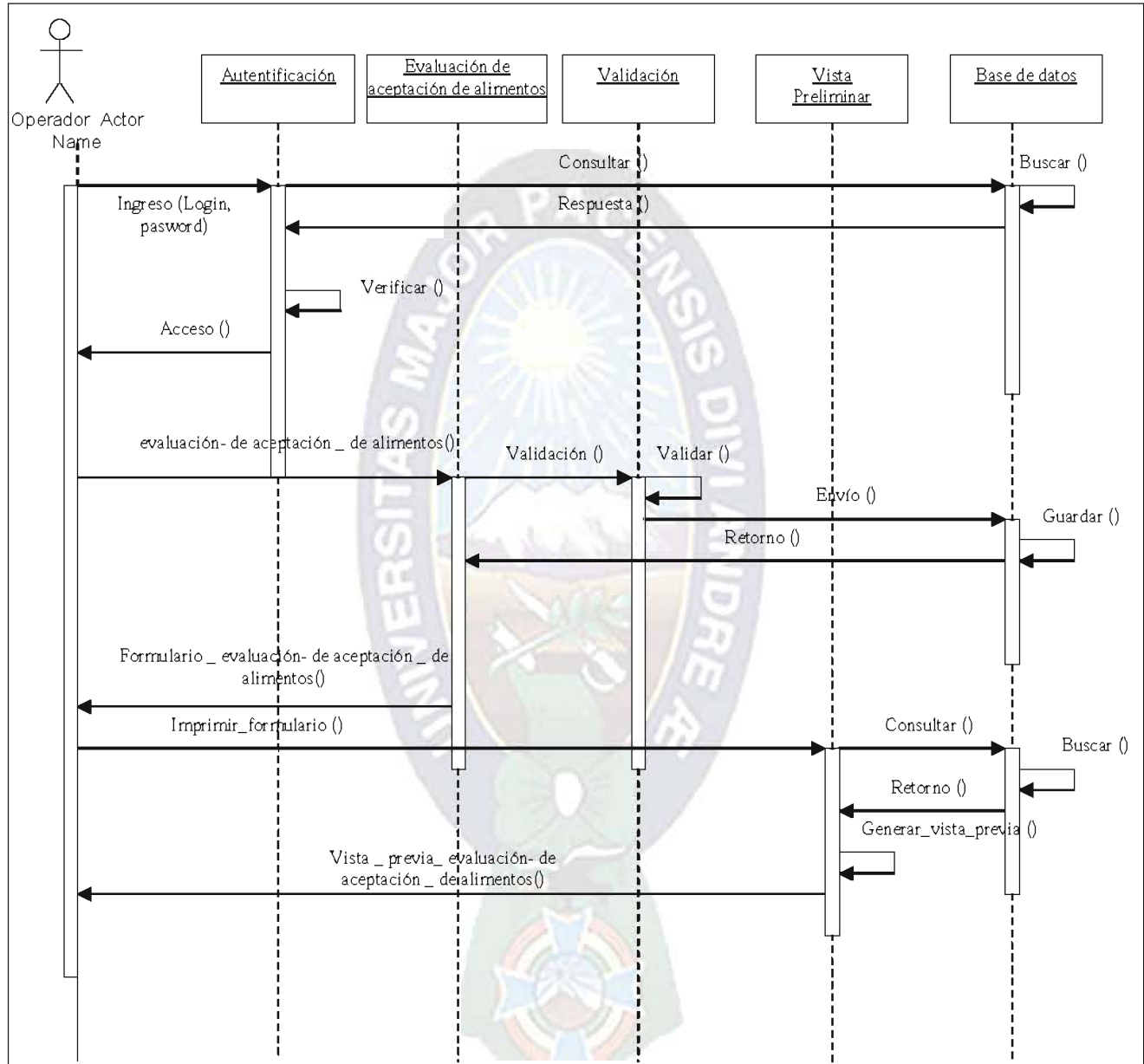


Diagrama de Secuencia-Evaluación de aceptación de alimentos
Fuente: [Elaboración Propia]