

UNIVERSIDAD AUSTRAL DE CHILE  
Facultad de Ciencias de la Ingeniería  
Escuela de Ingeniería en Computación



Modulo de tratamientos  
para el sistema de control y manejo de peces  
superior control 5.11

Seminario de Titulación  
para optar  
al título de  
Ingeniero de Ejecución en Computación

Profesor Patrocinante:  
Sr. Carlos Arenas Soto

Profesor Co-Patrocinante:  
Sra. Sandra Ruíz Aguilar

Ana María Moris Valdebenito  
Puerto Montt Chile 2002

## INDICE

Síntesis

Synthesis

1. Introducción.....	1
2. Objetivos.....	6
2.1 Objetivo General.....	6
2.2 Objetivos Específicos.....	7
3. Planteamiento del problema.....	8
3.1 Antecedentes.....	8
3.1.1 Definición del Problema.....	8
3.1.1.1 Ambito General.....	8
3.1.2 Identificación de Esfuerzos Anteriores.....	11
3.1.3 Solución Propuesta.....	12
3.1.4 Equipo de Trabajo.....	15
3.2 Justificación.....	16
3.2.1 Situación sin proyecto.....	16
3.2.2 Situación con proyecto.....	20
3.3 Delimitación.....	22
4. Metodología.....	25
4.1 Planificación de la Base de Datos.....	28
4.2 Definición del Sistema.....	28
4.3 Colección y Análisis de Requerimientos.....	28

4.4	Diseño de la Base de Datos.....	29
4.4.1	Diseño de Base de Datos Conceptual.....	30
4.4.2	Diseño de Base de Datos Lógico.....	31
4.4.3	Diseño de Base de Datos Físico.....	32
4.5	Selección del DBMS (Opcional).....	33
4.6	Diseño de la Aplicación.....	33
4.7	Prototipo (opcional).....	34
4.8	Implementación.....	34
4.9	Conversión.....	34
4.10	Prueba.....	34
4.11	Mantenimiento Operacional.....	35
5.	Recursos.....	37
5.1	Hardware.....	37
5.2	Software.....	37
5.3	Arquitectura.....	38
6.	Planificación de Base de Datos.....	42
6.1	Tiempo.....	43
6.2	Dinero.....	44
6.3	Materia Prima y Recursos.....	45
7.	Definición de Sistema.....	47
8.	Análisis y Recolección de Requerimientos.....	59
8.1	Entrevistas.....	61
8.2	Examinar Documentos.....	62

8.3 Requerimientos.....	64
9. Diseño de la Base de Datos.....	66
9.1 Diseño de Base de Datos conceptual.....	67
9.1.1 Identificación de Entidades.....	67
9.1.2 Identificación de Relaciones.....	71
9.1.3 Identificación y Asociación de Atributos con tipos de Entidades y Relaciones.....	76
9.1.4 Determinación de Dominios de Atributos.....	93
9.1.5 Identificación de Claves Candidatas y Elección de Claves Primarias para las Entidades.....	100
9.1.6 Modelo Conceptual E-R.....	103
9.2 Diseño de Base de Datos Lógico para el Modelo Relacional.....	105
9.2.1 Transformación del Modelo Conceptual al Modelo Lógico.....	106
9.2.1.1 Eliminación de Entidades Muchos-a-Muchos...	106
9.2.1.2 Eliminación de Relaciones con Atributos.....	107
9.2.1.3 Eliminación de Atributos Multivalóricos.....	107
9.2.2 Derivación de Relaciones del Modelo de Datos Lógico.....	108
9.2.3 Validación del Modelo utilizando Normalización.....	113
9.2.3.1 Primera Forma Normal (1FN).....	114
9.2.3.2 Segunda Forma Normal (2FN).....	114

9.2.3.3 Tercera Forma Normal (3FN).....	120
9.2.4 Validación del Modelo contra las Transacciones de Usuario.....	126
9.2.5 Diagrama E-R del Diseño Lógico.....	134
9.2.6 Definición de Restricciones de Integridad.....	136
9.2.6.1 Datos Requeridos.....	136
9.2.6.2 Restricciones de Dominios de Atributos.....	136
9.2.6.3 Integridad de Entidades.....	137
9.2.6.4 Integridad Referencial.....	137
9.2.6.5 Restricciones de Empresa.....	140
9.3 Diseño de Base de Datos Físico para el Modelo Relacional.....	141
9.3.1 Transformación del modelo de datos lógico global para un DBMS específico.....	142
9.3.1.1 Diseño de relaciones bases para un DBMS específico.....	143
9.3.1.2 Diseño de restricciones de la empresa para un DBMS específico.....	147
9.3.2 Diseño de representación física.....	148
9.3.2.1 Análisis de transacciones.....	148
9.3.2.2 Elección de organización de archivos.....	151
9.3.2.3 Elección de índices secundarios.....	151

9.2.3.4 Considerar la introducción de redundancia controlada (Denormalización)....	152
9.3.2.5 Estimación de espacio de requerimiento en disco.....	152
9.3.3 Diseño de mecanismos de seguridad.....	153
9.3.3.1 Diseño de vistas de usuario.....	153
9.3.3.2 Diseño de reglas de acceso.....	154
10 Diseño de la Aplicación.....	155
10.1 Diseño Transaccional.....	155
10.2 Diseño de interfaces de usuario.....	158
11 Prueba.....	167
11.1 Plataforma de cuatro pruebas para el desarrollo de sistemas.....	167
11.1.1 Pruebas de unidad.....	169
11.1.1.1 Caja Negra.....	169
11.1.2 Pruebas de integración.....	174
11.1.3 Prueba de validación.....	174
11.1.4 Pruebas de sistema.....	176
11.1.4.1 Prueba de recuperación.....	176
11.1.4.2 Prueba de seguridad.....	178
11.1.4.3 Prueba de resistencia.....	179
11.1.4.4 Prueba de rendimiento.....	180
11.1.5 Pruebas de aceptación de usuario.....	181

12 Conclusiones y/o recomendaciones.....	182
13 Bibliografía.....	184
14 Anexos.....	186
A Procesos .....	186
A.1 Importación/exportación de Datos.....	186
A.1.1 Exportar datos a Superior Control.....	187
A.1.2 Importar datos desde Superior Control.....	191
A.1.3 Exportar datos a una base de datos externa.....	192
A.1.4 Fórmulas utilizadas en el Módulo de Tratamientos.....	195
A.1.5 Software de creación de Archivos de Ayuda.....	200
B Documentos.....	204

## INDICE DE TABLAS

Tabla N° 1	“Identificación de Entidades”.....	68
Tabla N° 2	“Identificación de Relaciones”.....	72
Tabla N° 3	“Identificación y asociación de atributos con tipos de entidades y relaciones”.....	77
Tabla N° 4	“Determinación de dominios de atributos”.....	94
Tabla N° 5	“Identificación de claves candidatas y elección de claves primarias para las entidades”.....	101
Tabla N° 6	“Lista de colores para las trazas que satisfacen los requerimientos de usuario del Módulo de Tratamientos”.....	127
Tabla N° 7	“Integridad referencial del Módulo de Tratamientos”....	138
Tabla N° 8	“Frecuencia de acceso a las tablas del Módulo de Tratamientos”.....	149
Tabla N° 9	“Tablas bases utilizadas en las transacciones”.....	156
Tabla N° 10	“Menús de usuario para el Módulo de Tratamientos”...	170
Tabla N° 11	“Resultados Prueba de Caja Negra”.....	173
Tabla N° 12	“Validación de Transacciones v/s formularios”.....	175



## INDICE DE FIGURAS

Figura N° 1	“Situación sin proyecto”.....	19
Figura N° 2	“Situación con proyecto”.....	21
Figura N° 3	“Relación entre los departamentos que participan en el proyecto”.....	24
Figura N° 4	“Ciclo de vida de una Base de Datos”.....	36
Figura N° 5	“Arquitectura del Módulo de Tratamientos”.....	40
Figura N° 6	“Funcionamiento del Módulo de Tratamientos”.....	48
Figura N° 7	“Funciones que incluirá el Módulo de Tratamientos”.....	55
Figura N° 8	“Diagrama E-R del Modelo Conceptual del Módulo de Tratamientos”.....	104
Figura N° 9	“Mapa transaccional que satisface requerimiento T(1)”.....	129
Figura N° 10	“Mapa transaccional que satisface requerimientos T(2), T(3)”.....	130
Figura N° 11	“Mapa transaccional que satisface requerimientos T(4), T(5), T(6), T(7)”.....	131
Figura N° 12	“Mapa transaccional que satisface requerimientos T(7), T(8), T(9), T(12), T(14), T(15)”.....	132
Figura N° 13	“Mapa transaccional que satisface requerimientos T(10), T(13), T(16)”.....	133

Figura N° 14	“Diagrama E-R del Modelo Lógico del Módulo de Tratamientos”.....	135
Figura N° 15	“Creación de tablas en Microsoft Access 2000”.....	145
Figura N° 16	“Creación de claves primarias e índices en Microsoft Access 2000”.....	146
Figura N° 17	“Pantalla de ingreso al Módulo de Tratamientos”.....	158
Figura N° 18	“Pantalla Menú principal”.....	159
Figura N° 19	“Pantalla exportar datos a una Base de Datos Externa”.....	159
Figura N° 20	“Pantalla importar datos desde Superior Control”.....	160
Figura N° 21	“Pantalla crear nuevos tratamientos”.....	160
Figura N° 22	“Pantalla Panorámica de un tratamiento ingresado”.....	161
Figura N° 23	“Pantalla exportar información de los tratamientos directamente a Superior Control”.....	161
Figura N° 24	“Pantalla Declaración de Garantía”.....	162
Figura N° 25	“Pantalla agregar nueva empresa”.....	162
Figura N° 26	“Pantalla panorámica de los centros”.....	163
Figura N° 27	“Pantalla agregar nuevas medicinas para los Tratamientos”.....	163
Figura N° 28	“Pantalla agregar nuevos proveedores de Tratamientos”.....	164

Figura N° 29	“Pantalla agregar nuevos veterinarios” .....	164
Figura N° 30	“Pantalla panorámica de los reportes que emite el Módulo de Tratamientos” .....	165
Figura N° 31	“Pantalla del archivo de ayuda desarrollado para el Módulo de Tratamientos” .....	166
Figura N° 32	“Secuencia de pruebas de un sistema” .....	168
Figura N° 33	“Menú Archivo” .....	171
Figura N° 34	“Menú Actividades” .....	171
Figura N° 35	“Menú Registro” .....	171
Figura N° 36	“Vista del Panel de Trabajo de RoboHelp Office 9.0” .....	200
Figura N° 37	“Interacción de RoboHelp Office 9.0 con Microsoft Word 6.0” .....	201
Figura N° 38	“Pantalla de configuración del proyecto” .....	202
Figura N° 39	“Pantalla para generar archivos de ayuda con distintas extensiones” .....	203

## **SINTESIS**

El siguiente documento detalla el trabajo realizado en el desarrollo e implementación del “Módulo de Tratamientos para el sistema de control y manejo de peces Superior Control 5.11”

Este sistema se encuentra dirigido al Departamento de Salud de la empresa Fjord Seafood para ser utilizado por los veterinarios que componen dicho departamento.

Durante mucho tiempo las empresas del rubro salmonero han registrado la información referente a los Tratamientos de forma manual lo que genera una serie de inconvenientes, principalmente al momento de consultar dicha información ya que por lo general se encuentra en planillas de Excel o formularios que requieren de una clasificación especial lo cual hace complicada su búsqueda.

El objetivo de este sistema es proveer una solución que permita registrar información de los tratamientos en los centros de manera rápida y confiable, agilizar la búsqueda de información y mantener actualizados los datos referentes a dosis de medicamentos, alimentación, mortalidad,

períodos de carencia, entre otros datos, permitiendo de esta manera obtener una visión real del estado de los centros. El sistema permitirá además la emisión de documentos legales necesarios al momento de autorizar las cosechas que posteriormente serán enviadas a las plantas de proceso y a los consumidores finales.

Las entidades involucradas en el proyecto son el Departamento de Salud de la empresa Fjord Seafood y el Departamento de Información y Tecnología, IT&C, de la empresa Akvasmart Chile S.A. Esta última tiene un papel muy importante ya que es la empresa que provee el sistema de control y manejo de peces llamado "Superior Control 5.11" desde el cual se obtiene información relevante que permite el funcionamiento del nuevo sistema.

El Módulo de Tratamientos permitirá registrar y obtener información referente a tratamientos de manera rápida y confiable asegurando eficiencia y control sobre las operaciones que comprende dicha actividad.

Como punto a destacar se señala que en el desarrollo del proyecto se utilizó la metodología de Thomas Connolly "Ciclo de Vida de Base de Datos" [Connolly1999].

## **SYNTHESIS**

The following document details the work carried out in the development and implementation of the Module of Treatments for the control system and handling of fish Superior Control 5.11"

This system is directed to the Department of Health of the company Fjord Seafood and is to be used by the veterinarians that comprise this department.

For a long time the companies of the fish farming have registered the information with respect to the Treatments in a manual way that generates a series of inconveniences, mainly due to the time needed to consult with this information because generally it is in Excel files or forms that require a special classification that makes searching complicated.

The objective of this system is to provide a solution that allows registering information of the treatments in the centers in a quick and reliable way, to speed up the search of information and to maintain updated the relating of data for the dose of medications, feeding, mortality, periods of quarantine, among other data, allowing this way to obtain a real vision of the state of the centers. The system will also allow the emission of necessary

legal documents instantly authorize the harvest that later on will be sent to the process plants and the final consumers.

The entities involved in the project are the Department of Health of the company Fjord Seafood and the Department of Information and Technology, IT&C, of the company Akvasmart Chile S.A. The latter has a very important paper because it is the company that provides the control system and handling of fish called Superior Control 5.11 from which excellent information is obtained which will allow the operation of the new system.

The Module of Treatments will allow the registration and obtaining of information with respect to treatments in a quick and reliable way assuring efficiency and control on the operations by the relevant personnel

The basis of this project highlights the methodology was used of Thomas Connolly "Cycle of database Life " [Connolly1999].

## **1. INTRODUCCION**

Diversos estudios estadísticos realizados a nivel mundial, proyectan que para poder satisfacer las necesidades de la población humana será necesario un significativo aumento en la producción de alimentos, siendo el área acuícola uno de los principales agentes productivos.

La existencia de áreas de cultivo controlado en las aguas chilenas a permitido un creciente desarrollo del área acuícola lo cual, asociado al sostenido aumento de la población mundial, representa un desafío en términos de sustentabilidad del medio ambiente. El desarrollo de los cultivos artificiales de especies pasa por la investigación de nuevas técnicas y sistemas aplicados cada vez más eficientes que permitan cultivar nuevas especies, y aumentar las tradicionales, con costos más bajos y mejores procesos tecnológicos, sin dañar el ecosistema.

La tendencia mundial de un crecimiento acelerado de la acuicultura ha permitido que el consumo de productos provenientes de la pesca y la acuicultura se haya mantenido en constante desarrollo y crecimiento, lo cual necesariamente se traduce en un permanente esfuerzo por mantener y mejorar la eficiencia productiva de los diversos cultivos.



Actualmente, el Departamento de Salud de la empresa Fjord Seafood maneja grandes volúmenes de información asociados a tratamientos y vacunas aplicados en los centros. Es por esto que necesita implementar un sistema que permita reducir el trabajo y agilizar la búsqueda de información relacionada con los tratamientos y vacunas. Este proceso normalmente implica registrar información detallada de cada uno de los grupos de peces sometidos a algún tipo de tratamiento con sustancias de origen farmacológico, lo cual significa muchas horas de digitación y complicadas planillas electrónicas que realicen los cálculos asociados a las dosis, alimentación y períodos de carencia.

La empresa AKVAsmart Chile S.A. se encuentra abocada al proceso de desarrollar e implementar una solución a este problema, siendo la responsable de proveer todos los recursos necesarios para que el alumno tesista logre desarrollar una herramienta que permita satisfacer los requerimientos de los usuarios.

En el desarrollo del sistema participará el Jefe del Departamento de Salud de la empresa Fjord Seafood, un veterinario y el alumno tesista el cual pertenece a la empresa AKVAsmart Chile S.A. e involucrará a los Departamentos de Salud y de Producción. Por lo tanto serán estas personas las encargadas de desarrollar e implementar el nuevo sistema denominado

“Módulo de Tratamientos” el cual debe satisfacer los requerimientos expresados por los futuros usuarios.

El sistema ofrecerá una solución que permitirá registrar y consultar información referente a los Tratamientos que se aplican a los peces en los centros de cultivo permitiendo elaborar mecanismo que contribuyan a superar cualquier eventualidad sanitaria en el centro. De esta manera la información podrá ser consultada de manera rápida y eficiente permitiendo además generar documentos legales necesarios el momento de autorizar la venta a mercados finales.

La metodología utilizada en el desarrollo del Módulo de Tratamientos lleva por título “Ciclo de Vida de Base de Datos” del autor Thomas Connolly[Connolly1999], la cual consta de las siguientes etapas : Planificación de Base de Datos, Definición de Sistemas, Colección y Análisis de Requerimientos, Diseño de la Base de Datos, Selección de DBMS (Data Base Manager System), Diseño de Aplicación, Prototipo, Implementación, Conversión, Prueba, Mantenimiento Operacional.

A continuación se describe brevemente el contenido de cada capítulo presente en este trabajo.

El Capítulo 2 del presente trabajo detalla los objetivos generales y específicos del Sistema.

El Capítulo 3 corresponde al planteamiento del problema, incluyendo antecedentes, definición del problema y el ámbito general. También se indica una descripción de esfuerzos anteriores para resolver el problema y la solución propuesta. Se detalla el equipo de trabajo involucrado en la solución y posteriormente la justificación y delimitación del Proyecto.

El Capítulo 4 detalla la metodología utilizada para el desarrollo del Proyecto.

En el Capítulo 5 se detalla los recursos y la arquitectura que tendrá el Sistema.

El Capítulo 6 corresponde a planificación de la base de datos, incluyendo algunos puntos considerados al momento de decidir la factibilidad del Proyecto.

El Capítulo 7 corresponde a la definición del Sistema y sus alcances.

El Capítulo 8 detalla la recolección y análisis de requerimientos para el Módulo de Tratamientos.

El Capítulo 9 muestra el diseño e implementación de la base de datos para el Módulo de Tratamientos.

El Capítulo 10 presenta el diseño de la aplicación para el Módulo de Tratamientos.

El Capítulo 11 presenta la aplicación de la Metodología de Prueba en el Módulo de Tratamientos.

Como parte de apoyo para la comprensión del Sistema se incluye en el capítulo 14 un anexo con algunos procedimientos utilizados en el desarrollo del presente Proyecto. Como punto final del anexo se muestra la documentación utilizada en el desarrollo del Módulo de Tratamientos.

## **2. OBJETIVOS**

### **2.1. Objetivo General**

Proveer de un nuevo sistema denominado “Módulo de Tratamientos”, para ser utilizado por el departamento de salud de Fjord Seafood.

Con esta herramienta, los veterinarios podrán registrar información de los tratamientos en los centros, obtener automáticamente información actualizada, calcular dosis de medicamentos, alimentación, mortalidad, períodos de carencia, entre otros datos, permitiendo de esta manera un control real de los tratamientos aplicados en los centros, además de permitir la emisión de documentos legales necesarios al momento de autorizar las cosechas que posteriormente serán enviadas a las plantas de proceso y a los consumidores finales.

## **2.2. Objetivos Específicos**

- ✓ Recopilar antecedentes que permitan diseñar un sistema según las necesidades de los usuarios.
- ✓ Registrar información referente a los Tratamientos que se suministran en un centro de cultivo.
- ✓ La información del Módulo de Tratamientos debe trabajar directamente con la base de datos del Sistema de Producción Superior Control.
- ✓ La información de los tratamientos registrada en el módulo debe ser exportada a Superior Control.
- ✓ El sistema deberá ser multi-empresa y permitir el registro de todos los centros de cada una de las empresas. El registro de información deberá ser a nivel de jaulas. Deberá, además, permitir seleccionar el idioma del entorno de trabajo: inglés o español.
- ✓ El sistema deberá emitir algunos informes, entre ellos la “Prescripción Veterinaria” y la “Declaración de Garantía” los cuales serán requeridos por las plantas de proceso al momento de recepcionar la cosecha.

### **3. PLANTEAMIENTO DEL PROBLEMA**

#### **3.1. Antecedentes**

##### **3.1.1. Definición del problema**

###### **3.1.1.1 Ambito General**

El departamento de salud de la empresa Fjord Seafood, basado en las leyes chilenas, necesita introducir nuevas funcionalidades y cambios en Superior Control (Software para el control y manejo de peces en centros de cultivo), para obtener y registrar información y reportes mejorados para su departamento de veterinarios.

En la actualidad, no existe un software que permita registrar toda esta información, por lo que todo el historial de salud (Tratamientos) de la empresa se encuentra en formularios, archivadores, planillas Excel, cuadernos, y es a través de ellos que la mayoría de los veterinarios realizan el registro de sus prescripciones médicas [ver anexo B], vacunas, tratamientos y su duración, así como el período de carencia asociado.

La actual normativa chilena (Programa de Aseguramiento de Calidad: Norma Técnica Sección 2) exige una serie de documentos que certifiquen el tratamiento que se la ha dado a los peces provenientes de centros de cultivo en cuanto a salud. La “Declaración de Garantía” [ver anexo B] es obligatoria y en ella se debe detallar la empresa, el centro de cultivo, el grupo de peces, los tratamientos, datos del médico veterinario, y debe estar respaldada en los archivos del centro de cultivo, por alguno de los siguientes documentos : Prescripción Médico Veterinaria, Informe de Análisis, Declaración Jurada Simple o Informe de Análisis Contaminantes del Laboratorio, los cuales deben estar disponibles para SERNAPESCA.

Toda la documentación antes descrita es preparada en base a la información que registra el médico veterinario por lo que es el responsable, además, de mantener al día los registros y documentos, desde donde se recopila y organiza la información que ya fue ingresada y que se encuentra en distintos medios de respaldo (formularios, planillas, cuadernos, etc.).

La diversidad de software presentes en el mercado para el manejo de centros de cultivos, no permiten en la actualidad llevar un control eficiente de los tratamientos que se registran en ellos, debiendo resguardar la información de manera manual en formularios y planillas electrónicas.



AKVAsmart ASA es una empresa noruega presente en el mercado desde 1980. Siempre cumpliendo con su rol de optimizar el proceso de cultivo de salmones, AKVAsmart a través de su departamento Information Technology and Consultancy, IT&C, desarrolla y provee sistemas de tecnología de información orientados a facilitar la eficiencia productiva de la industria salmonera nacional. Su oficina principal está en Trondheim, Noruega, pero también está presente en Chile, Canadá, Estados Unidos y Escocia, entre otros. Cuenta con empleados con experiencia y conocimiento en el manejo de peces, biología, alimentación, economía e ingeniería de software moderno. Los productos han estado en el mercado desde 1988 y bajo plataforma Windows desde 1995.

AKVAsmart a través de su producto Superior Control, provee software con diseño a la medida, entrenamiento y servicios de consultoría así como también sistemas de información estándar que cubren el control de la producción y presupuesto para la cadena biológica completa – desde el desove a cosecha. Los sistemas se desarrollan en estrecha cooperación con los usuarios, empleando metodología probada y herramientas de desarrollo moderno sobre la plataforma de Microsoft Windows.

La Familia Superior está construida en módulos, lo que permite una completa interacción entre ellos ofreciendo sistemas con los requerimientos necesarios para cubrir las distintas necesidades de la empresa. Por esto, en su constante búsqueda de soluciones, se encuentra trabajando en el diseño y desarrollo de un nuevo módulo de Superior Control, el Módulo de Tratamientos, el cual permitirá registrar información referente al tema Tratamientos del centro de cultivo, prescribir recetas, determinar dosis, tipos de tratamientos, emitir informes y documentos legales asociados al tema, entre otras funcionalidades.

### **3.1.2 Identificación de esfuerzos anteriores**

A través del departamento de Salud de la empresa Fjord Seafood se gestó la posible solución a un problema que se encontraba presente desde siempre en todas las empresas del rubro salmonero: registrar los tratamientos y mantener un historial médico del centro de manera accesible para todos los veterinarios de la empresa. Es por esto que luego de analizar la situación se sugirió que la empresa responsable de diseñar la solución fuera la misma que provee el Sistema de Control de Producción: AKVAsmart Chile S.A.. Esta decisión también fue apoyada por el hecho de que gran parte de la información que se manejará en el sistema proviene de información que se digita en

Superior Control, lo cual significará un mejor aprovechamiento de la información que ya se encuentra ingresada.

Sin embargo, el proyecto fue postergado en varias oportunidades debido a que no se disponía de algunos recursos, entre ellos el factor tiempo y recurso humano, por lo que la materialización se produce a partir de la disposición de estos factores.

### **3.1.3 Solución Propuesta**

Se plantea como solución un sistema que permita registrar y consultar toda la información referente a los tratamientos que se suministran en un centro de cultivo.

Para ello se propone que el sistema disponga de un mecanismo que permita el flujo bidireccional de datos con Superior Control, sistema desde el cual obtendrá información importante al momento de determinar el estado del centro de cultivo. Para lograrlo se deberá implementar una solución que permita “importar y exportar” datos del sistema y para ello se debe considerar las herramientas utilizadas en el desarrollo del sistema de control de la producción, Superior Control, desde el cual se obtendrán datos. En el desarrollo de Superior

Control se utilizó como lenguaje de desarrollo Microsoft Visual Basic y Btrieve de Pervasive como gestor de Base de Datos, el cual permite mediante sentencias de código y archivos "dll", abrir (o acceder) las tablas de Superior y obtener la información de los campos requeridos, así como también incorporar datos. El sistema mediante la utilización de estas sentencias de código obtendrá e incorporará los datos de manera transparente para el usuario, facilitando de esta manera el ingreso de información.

Esta utilidad permitirá además exportar los datos (extraer datos) hacia otros centros de cultivo y hacia la oficina central donde se encontrará un consolidado de todos los centros, el cual será actualizado mediante exportaciones que serán creadas en los centros y luego estos datos serán importados (incorporados) en la oficina central de la empresa. De esta manera, se dispondrá de la información que se maneja en los distintos centros de la empresa, permitiendo de esta forma estudiar mecanismos que permitan mejorar la gestión en el tema salud.

El ingreso de los nuevos tratamientos permitirá registrar en una misma pantalla la información tanto para las especies de Agua Mar como para las especies provenientes de Pisciculturas (Agua Dulce), así como también, no será necesario ingresar a una nueva opción para seleccionar la vía mediante la cual se suministrará el tratamiento (oral o baño).

El sistema contemplará además la opción de ingresar nuevas empresas, centros, veterinarios, proveedores de medicamentos, medicamentos, dietas medicadas, de manera que los usuarios dispongan en todo momento de las herramientas que les permitan registrar su información sin inconvenientes. Cabe destacar que el proceso de importar datos desde Superior Control incorpora información referente a estos puntos pero puede no ser suficiente cuando se desea registrar información que no se encuentra en Superior y para el registro que se desea hacer en el centro. La funcionalidad de importar/exportar datos es un proceso que facilitará el trabajo de las personas encargadas de registrar la información en los centros, pero se consideró además la posibilidad de que no se utilicen datos provenientes desde Superior Control, ante lo cual el sistema contará con todos los procesos necesarios para trabajar de manera independiente.

El sistema permitirá obtener la “Prescripción Medica” y la “Declaración de Garantía” como documentos legales emitidos por la empresa y que luego serán exigidos por la entidad que regula la producción de peces provenientes de centros de cultivo (SERNAPESCA).

En definitiva, existen una serie de procesos que serán plasmados en el sistema y que permitirán mejorar de manera considerable el proceso de registrar información de los Tratamientos, al permitir que la información se

encuentre siempre disponible y alcance de los usuarios que lo requieran, el tiempo que antes se tardaban en buscar la información se reducirá al mínimo ya que solo dependerá de la habilidad del usuario en utilizar el sistema, el cual por lo demás contará con un entorno totalmente amigable y archivos de ayuda.

#### **3.1.4 Equipo de Trabajo**

El equipo de trabajo estará compuesto por un conjunto de veterinarios pertenecientes a la empresa Fjord Seafood, tres Consultores en Acuicultura pertenecientes al departamento IT&C de la empresa AKVAsmart Chile S.A., los cuales entregarán las bases médicas y biológicas al sistema.

El desarrollo de la parte informática del proyecto estará a cargo del alumno tesista, el cual pertenece al departamento IT&C de la empresa AKVAsmart Chile S.A.

## **3.2 Justificación**

### **3.2.1 Situación sin proyecto**

Todo el proceso de control de salud de los peces pertenecientes a un centro debe estar a cargo del departamento de veterinarios de la empresa, el cual debe proveer los mecanismos necesarios tendientes a prevenir y superar cualquier posible emergencia que afecte al centro, referente al tema salud. La no detección a tiempo de una enfermedad puede llevar a la pérdida total de la producción de dicho centro con sus respectivos costos asociados.

Es por ello que el veterinario debe permanecer en constante observación de la evolución de los peces, emitiendo prescripciones médicas y administrando los medicamentos y tratamientos necesarios tendientes a prevenir cualquier eventualidad. También es el responsable de registrar el historial médico del centro, a través de documentos escritos manualmente y en planillas electrónicas [ver anexo B] lo cual significa que debe resguardar esos registros en un lugar seguro, siendo muy poco expedita la búsqueda de información para consultar datos específicos referente a algún tratamiento administrado a un grupo de peces o en un centro en especial.

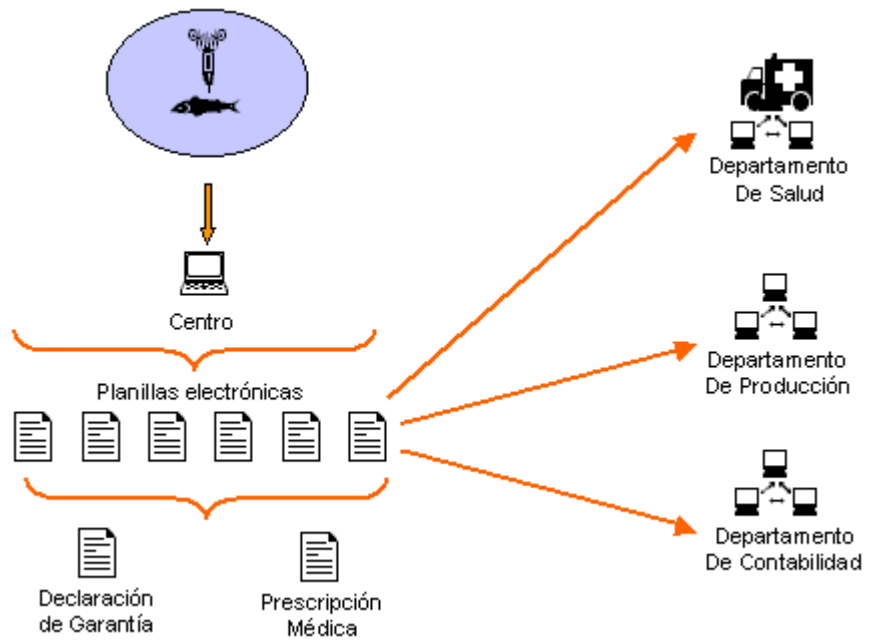
Esta forma de registrar la información implica un trabajo adicional ya que se deben calcular de manera manual las dosis de medicamento, el tiempo que se deben administrar los tratamientos, la alimentación de los peces durante el tratamiento y el período de resguardo mínimo necesario para alcanzar niveles inferiores a los límites establecidos por las normativas para residuos de productos farmacéuticos de uso veterinario y que se conoce como “carencia”. La carencia requiere especial cuidado ya que se calcula en base a Unidades Térmicas Acumuladas (UTA) y las cuales consisten en un muestreo de la temperatura del agua en el centro y luego distribuirlas en “n” días. Estos “n” días será el tiempo que deberán esperar los peces hasta que se encuentren en condiciones de ser consumidos de manera segura para la salud humana.

Un ejemplo de un documento emitido por el Veterinario es la “Prescripción Veterinaria” [anexo B], la cual es utilizada para comenzar a administrar un determinado medicamento, el cual una vez que ha comenzado a ser consumido por el grupo de peces implica que éstos no podrán ser extraídos del centro hasta que concluya su período de carencia. Cabe destacar que existen diferencias en los tratamientos entre un centro de Agua Dulce (Piscicultura o Lago) y un centro de Agua Mar (Mar), por lo tanto esta diferencia significa formularios, registros, documentos y planillas diferentes, lo cual se traduce en una mayor cantidad de material que debe ser correctamente



clasificado y archivado para luego disponer de la información cuando sea requerida.

SERNAPESCA es el organismo que regula a las empresas que producen pescados provenientes de centros de cultivo. En su normativa establece una serie de requisitos y documentos que deben ser emitidos y presentados por la empresa al momento de enviar a las plantas de proceso y al mercado su producción. Un ejemplo es la “Declaración de Garantía” [ver anexo B] que extiende el veterinario. Dicho documento representa una herramienta legal que permitirá a la empresa comercializar el grupo de peces certificando su calidad. Cualquier trasgresión a la veracidad de la información de dicho documento haría incurrir a la empresa emisora en una infracción que puede ser sancionada incluso con el cierre del centro involucrado.



**Figura N° 1: Situación sin Proyecto**

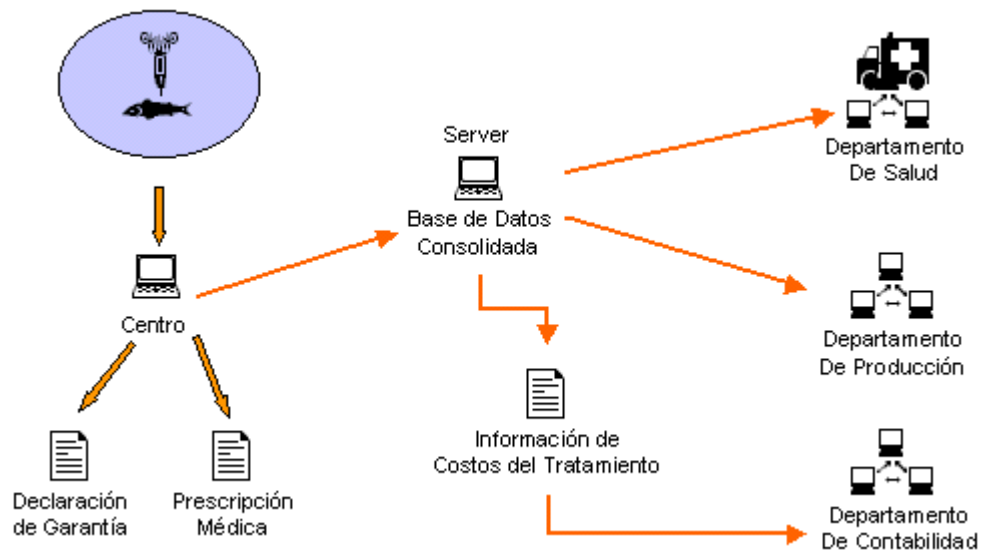
### **3.2.2 Situación con proyecto**

La implementación de un sistema que permita registrar toda la información referente al tema Tratamientos de un centro de cultivo significará un aporte real al trabajo que realizan los veterinarios ya que podrán registrar y consultar información de forma rápida y segura, disponiendo de reportes que les permitirán obtener información referente a los tratamientos del centro.

Las “Prescripciones Médicas” se podrán obtener directamente desde el sistema y a partir de ellas se podrá obtener automáticamente información relevante respecto del historial médico del centro, los períodos de carencia posteriores al tratamiento podrán ser asignados inmediatamente y permitirá realizar una estimación más exacta de su duración.

El registro de los tratamientos se realizará en un formulario que será común tanto para los tratamientos de Agua Mar y Dulce. La información necesaria para emitir la “Declaración de Garantía”, entre otros, se podrá obtener de manera fácil y rápida ya que toda la información se encontrará registrada con anterioridad, además el sistema permitirá emitirla de manera inmediata.

Los datos que arroje el sistema permitirán obtener información actualizada referente al tema salud en todo momento, siendo esto de gran ayuda para el médico Veterinario ya que le ayudará al momento de decidir sobre la salud de los peces del centro.



**Figura N° 2: Situación con Proyecto**

### **3.3 Delimitación**

El desarrollo del proyecto abarca desde la definición de objetivos hasta la implementación del Módulo de Tratamientos en la empresa Fjord Seafood. La etapa de Conversión no será realizada ya que no existe un sistema previo que justifique un análisis. La etapa de Mantenimiento estará a cargo de la empresa AKVAsmart Chile S.A. como entidad a la cual pertenece el alumno tesista y desarrollador, debido a una decisión interna de la empresa Fjord Seafood.

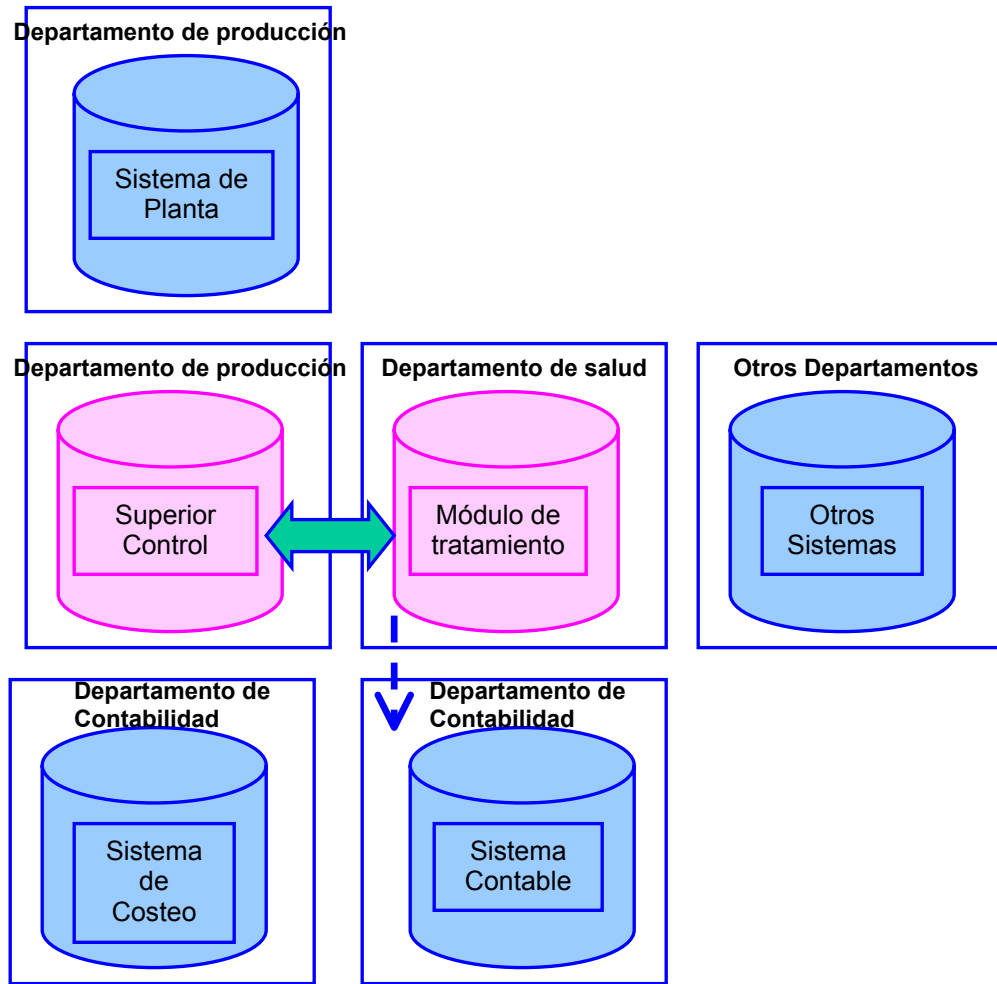
La Fase de Prueba se desarrollará en conjunto con el jefe del departamento de salud de la empresa.

El sistema se encuentra relacionado al sistema de producción de peces Superior Control, el cual provee información importante al momento de analizar la situación de los centros, es por esto que la información que se registre en el sistema debe ser incorporada de alguna manera en este sistema de producción, ya que todas las proyecciones, estadísticas e informes de la empresa referentes a la producción de peces se obtienen desde ahí.

Los departamentos involucrados serán el Departamento de Salud y el Departamento de Producción de la empresa. A pesar de que el sistema permite valorizar algunos costos asociados al tratamiento, esta información no será

importada al sistema contable de la empresa, por lo tanto dicho departamento solo participa del proyecto de manera indirecta, es decir, analiza la información que se puede obtener desde el Módulo de Tratamiento en la opción de costos del tratamiento y la incorpora en sus registros. El desarrollo del Módulo no contempla analizar ni implementar la manera de incorporar información a ningún otro sistema de la empresa, ante lo cual los sistemas que requieran información de su gestión, deberán obtenerla de los reportes y luego digitarla donde corresponda.

La siguiente figura esquematiza la interacción de los sistemas y departamentos pertenecientes a la empresa y su relación con el Módulo de Tratamientos.



**Figura N 3: Relación entre los departamentos que participarán en el proyecto**

#### 4. METODOLOGIA

Para lograr determinar cuales serían los pasos a seguir en el desarrollo del proyecto, se analizaron las metodologías de Roger Pressman “Ciclo de Vida Clásico” y “El Modelo en Espiral” y de Thomas Connolly “Ciclo de vida de una Base de Datos”.

La metodología de Roger Pressman [Pressman1993] “Ciclo de Vida Clásico” comprende las siguientes etapas:

- ✓ Ingeniería y análisis del sistema
- ✓ Análisis de los requisitos del software
- ✓ Diseño
- ✓ Codificación
- ✓ Prueba
- ✓ Mantenimiento



La siguiente metodología de Roger Pressman [Pressman1993] analizada corresponde a “El Modelo en Espiral” y comprende las siguientes etapas:

- ✓ Planificación
- ✓ Análisis de riesgo
- ✓ Ingeniería
- ✓ Evaluación del cliente

Ambas metodologías del autor Roger Pressman [Pressman1993] analizan el diseño de un sistema desde la perspectiva de una interacción continua y recurrente con el usuario. Considerando el tiempo de que se dispone para desarrollar el proyecto, esto puede conllevar una serie de retrasos debido a que el usuario siempre considera agregar nuevos requerimientos y cambios en los prototipos que se le presenten y lo que en un principio se estableció como delimitación del proyecto puede quedar absolutamente descartado y convertirse cada vez en un proyecto con delimitaciones diferentes.

La metodología “Ciclo de Vida de una Base de Datos” del autor Thomas Connolly [Connolly1999] analiza el desarrollo de un sistema desde la perspectiva de Base de Datos. Esta metodología, al igual que las metodologías de Pressman analizadas [Pressman1993], considera la participación de los usuarios finales, pero solo en la toma de requerimientos y luego de haber

concluido el sistema en la fase de Prueba y posterior Mantenimiento Operacional. El diseño del sistema es un punto que no es analizado en profundidad en esta metodología y solo se considera en el Diseño de la Aplicación y en la Implementación lo cual también tiene sus desventajas ya que todo el trabajo que implica desarrollar un sistema muchas veces queda reflejado solo en función de la Base de Datos sin resaltar todo el trabajo que implicó el implementar el sistema completo.

Considerando el análisis que se realizó de metodologías se eligió para el desarrollo del Módulo de Tratamientos de Superior Control el “Ciclo de vida de una Base de Datos” del autor Thomas Connolly [Connolly1999], la cual fue elegida debido a factores tan importante como el trabajo a fondo que se realiza con la Base de Datos y a que el tiempo estimado para el proyecto es limitado lo cual no permitirá una interacción continua con los usuarios ya sea para presentar prototipos o considerar nuevos requerimientos. El desarrollador deberá por lo tanto trabajar arduamente en una completa recolección de requerimientos para captar de la mejor manera posible lo que los usuarios necesitan para solucionar la situación presentada como problema. Cabe mencionar que esta metodología también plantea la utilización de prototipos pero como un paso opcional dentro del desarrollo del sistema.

Las etapas que componen esta metodología son las siguientes:

#### **4.1 Planificación de la Base de Datos**

En esta fase se identifica el trabajo necesario para realizar el proyecto y los recursos con que se cuenta (humanos y económicos).

#### **4.2 Definición del Sistema**

Definición del alcance y límites de la aplicación y la forma de comunicarse con las otras fuentes de información de la empresa.

#### **4.3 Colección y Análisis de Requerimientos**

En esta etapa se recopila y analiza la información de la organización que será apoyada por el sistema o la aplicación de Base de Datos. Esta información también es utilizada para identificar los requerimientos de los usuarios del sistema [Pressman1999]. Para lograr obtener la información necesaria en esta fase se pueden utilizar los siguientes procedimientos:

- ✓ Entrevistas con los futuros usuarios
- ✓ Observar el funcionamiento de la empresa
- ✓ Examinar procedimientos manuales (si existen) y los reportes existentes
- ✓ Realizar cuestionarios a los usuarios
- ✓ Consultar a expertos que tengan experiencia en el uso y manejo de soluciones similares

#### **4.4 Diseño de la Base de Datos**

Corresponde al proceso de crear un diseño de Base de Datos que apoye las operaciones y objetivos de la empresa. Las metas claves del diseño son:

- ✓ Representar y relacionar datos requeridos por todas las áreas de aplicaciones y grupos de usuarios.
- ✓ Entregar un modelo de datos que apoye las transacciones requeridas en los datos.

- ✓ Especificar un diseño mínimo el cual, estructurado apropiadamente, pueda obtener los rendimientos identificados en los requerimientos.

Durante el desarrollo del Diseño de la Base de Datos los contenidos que se deben abarcar son:

#### **4.4.1 Diseño de Base de Datos Conceptual**

Se identifican las entidades involucradas en el proceso; se analiza las relaciones existentes entre ellas, así como también los atributos para cada una de las entidades.

Se documenta cada entidad presente con sus respectivos atributos y dominios correspondientes. En esta etapa se construye el primer diagrama E-R, el cual se irá refinando posteriormente. Debe elegirse un conjunto de claves candidatas de las cuales se elegirá la que más se acerque a ser clave primaria.

Lo que se pretende en esta etapa es confeccionar un modelo de datos local y conceptual para cada vista específica de usuario.

#### **4.4.2 Diseño de Base de Datos Lógico**

En esta etapa se realizarán los siguientes pasos:

- ✓ Construcción y validación del modelo de datos lógico local para cada requerimiento de usuario
- ✓ Construcción y validación del modelo de datos lógico global

El objetivo en este punto es refinar el diseño conceptual para eliminar las características no deseadas que permita luego, transformar este modelo al modelo de datos lógico local.

En esta etapa se construye el Diagrama E-R final, es decir, se refina aquel construido en el diseño conceptual. Es necesario analizar los tipos de relaciones existentes entre las entidades, de modo de eliminar aquellas que causen problemas, como son las relaciones muchos-a-muchos, atributos multivalóricos, etc., es decir, todas aquellas relaciones que puedan producir inconsistencias. En esta etapa se definen los esquemas para las entidades presentes. Se deben identificar las relaciones padre-hijo para identificar las claves foráneas. De manera posterior se validará el modelo utilizando las técnicas de normalización, el cual permite identificar qué atributos deben estar junto a un tipo de entidad.

El proceso de normalización incluye:

- ✓ Primera Forma Normal (1FN) que elimina grupos repitentes.
- ✓ Segunda Forma Normal (2FN) que elimina dependencias parciales en clave primaria
- ✓ Tercera Forma Normal (3FN) que elimina las dependencias transitivas en la clave primaria

En esta etapa se definen las restricciones de integridad que permiten eliminar la inconsistencia en la base de datos.

#### **4.4.3 Diseño de Base de Datos Físico**

Dentro de esta etapa la labor a realizar implica diseñar la representación Física. Se deben analizar las transacciones, elegir la organización para los archivos existentes. Se debe considerar el proceso de redundancia controlada (denormalización).

Es necesario diseñar mecanismos de seguridad para las interfaces de usuario y las reglas de acceso.

#### **4.5 Selección del DBMS (Opcional)**

Un estudio previo determinó que la empresa Fjord Seafood no incurrirá en costos de implementación en equipos computacionales debido a que esto ya fue realizado hace poco tiempo, por lo tanto el sistema deberá considerar la implementación que actualmente tienen los centros de la empresa, además considerando este punto y el hecho de que ya se dispone de licencias, se determinó que el gestor de Base de Datos sería Microsoft Access y el lenguaje de programación Visual Basic 6.0, como una manera de aprovechar los recursos existentes y no agregar nuevos costos para el proyecto. Debido a lo anterior este punto no será abordado por el alumno tesista.

#### **4.6 Diseño de Aplicación**

Implica diseñar interfaces de usuario y programas de aplicaciones que utilicen y procesen la base de datos.



#### **4.7 Prototipo (Opcional)**

No se considerarán en el desarrollo de este sistema debido a que se dispone de un período de tiempo limitado para implementarlo, por este motivo se trabajará sobre la base de un buen Diseño de Base de Datos para no invertir recursos en prototipos.

#### **4.8 Implementación**

Esta etapa implica la realización física de la base de datos y diseño de aplicaciones.

#### **4.9 Conversión**

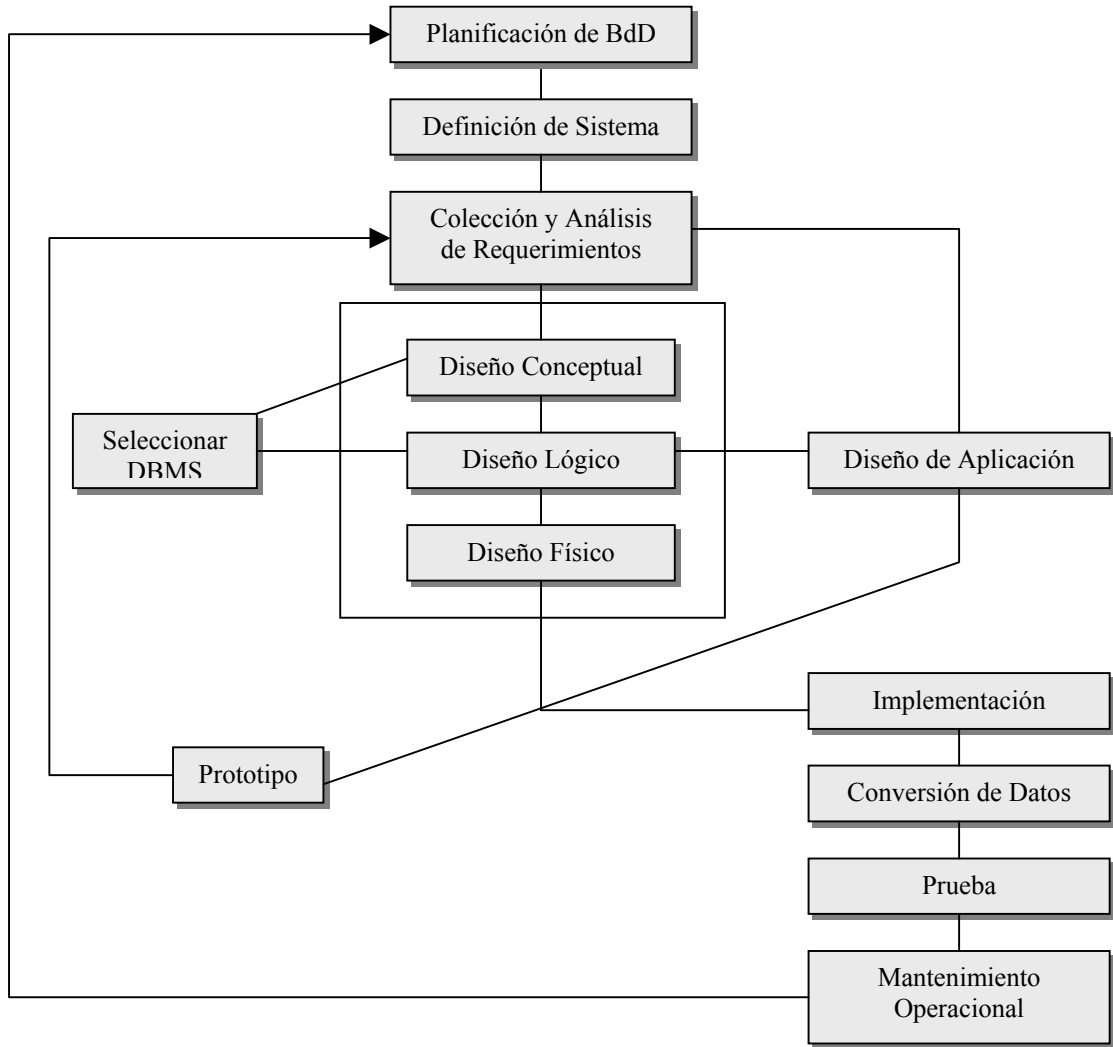
Esta etapa no será considerada debido a que el módulo responde a una necesidad de la empresa que no cuenta con un sistema previo, que requiera ser reemplazado por esta aplicación.

#### **4.10 Prueba**

Tiene por objetivo encontrar errores en la aplicación al ejecutar el programa.

#### **4.11 Mantenimiento Operacional**

Implica monitorear y mantener el sistema después de la instalación. Cabe destacar que el mantenimiento Operacional estará a cargo de la empresa AKVAsmart Chile S.A. debido a una determinación interna de la empresa.



**Figura N° 4: Ciclo de vida de una Base de Datos**

## **5. RECURSOS**

### **5.1 Hardware**

El hardware a utilizar para llevar a cabo el presente Proyecto es el siguiente:

- ✓ Computador con procesador Pentium III de 700 Mhz, 128 MB de memoria RAM, Disco Duro 20GB, 8 MB de memoria de Video.

Requerimientos mínimos para los equipos donde se implementará el sistema:

- ✓ Computador con procesador Pentium II de 600 Mhz, 64 MB de memoria RAM, espacio en Disco Duro de 70 MB.

### **5.2 Software**

Para materializar el presente proyecto se eligió como herramienta de desarrollo el software Visual Basic 6.0 y como motor de Base de Datos Microsoft Access 2000.

Además se utilizará adicionalmente el siguiente software:

- ✓ RoboHELP Office : herramienta que permite el desarrollo de los archivos de ayuda.

- ✓ Sheridan Data Widgets: posee herramientas que permiten la creación de grillas de una manera fácil. Trabaja en conjunto con Visual Basic 6.0.
- ✓ Crystal Report 6.0 : herramienta que permite generar informes.
- ✓ Wise Install Builder 7.04 : herramienta que permite la creación de archivos de instalación.

### **5.3 Arquitectura**

Antes de ver lo que es la Arquitectura del sistema en sí es importante recordar los puntos que se deben tener claros para diseñar una base datos.

- ✓ Es importante conocer exactamente para qué se quiere usar la base de datos, qué datos son los que interesan de los que existen en la realidad y qué información se necesitará extraer.
- ✓ Una vez que esto esté claro, se definen las Tablas que compondrán la base de datos. Dentro de cada tabla, se piensa qué campos serán necesarios. Conviene detenerse y definir correctamente la base de datos, ya que un mal diseño hará que el sistema sea lento y los resultados no sean los esperados.

Las tres cosas básicas que debe permitir un gestor de base de datos son: introducir datos, almacenarlos y recuperarlos. Cualquier gestor debe permitir: ordenar los datos, realizar búsquedas, mostrar distintas vistas de los datos, realizar cálculos, resumirlos, generar informes, importarlos y exportarlos.

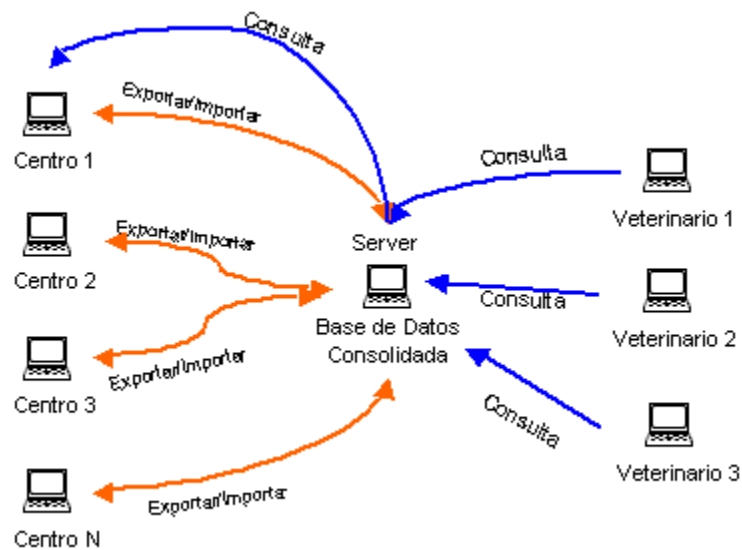
Al mismo tiempo permiten otra serie de funciones que hacen de ellos herramientas incomparablemente superiores a los métodos tradicionales de almacenamiento de datos: archivadores, carpetas, etc.

Microsoft Access es un potente sistema de **administración de bases de datos relacionales**. Las bases de datos en Access son documentos combinados donde se divide la información por parejas de *objetos* especializados. Así, por ejemplo, como elemento primario de información se encuentran las tablas. Aunque las tablas se crean de manera independiente, pueden crearse otros elementos relacionados a las tablas como formularios, consultas, informes, entre otros.

Fjord Seafood cuenta en la actualidad con tecnología de punta en cuanto a Arquitectura de Red, teniendo implementado un servidor con Metaframe lo cual permite un control total de los computadores mediante la instalación del cliente que se adquiere con el producto Metaframe. La ventaja de este servicio es que se pueden ejecutar de remotamente aplicaciones que se encuentran

geográficamente alejadas de manera que el usuario puede realizar todo lo que desee como si estuviera frente al computador.

Para aprovechar las características de Access y del esquema de red de la empresa, en el desarrollo del presente sistema se utilizará el siguiente esquema: (ver Figura N° 5)



**Figura N° 5 : Arquitectura del Módulo de Tratamientos**

Los usuarios del sistema (veterinarios) trabajan directamente en el sistema de control de Producción Superior Control y en el Módulo de Tratamientos en el computador del centro de manera local. Para actualizar la base de datos consolidada que se encontrará en uno de los servidores de la empresa, el usuario creará una exportación de los datos del sistema (eligiendo uno de las dos formas que el sistema permite) y por intermedio de la red de área local de la empresa se realizará la transferencia de los datos. Los usuarios que se encuentren en la oficina central podrán conectarse mediante la definición de unidades de red al Módulo de Tratamientos para consultar la información que necesiten. También podrán conectarse a los centros remotamente mediante cliente Metaframe (ICA) y consultar la información que ellos tengan registrada localmente. La información disponible en el servidor solo servirá para consultas y puede ser accesada por todos los usuarios que lo requieran, las modificaciones o ingresos de datos deberán realizarse en los centros.



## **6. PLANIFICACIÓN DE BASE DE DATOS**

Antes de comenzar con un proyecto es necesario determinar la factibilidad del proyecto. El Estudio de Factibilidad es el análisis de una empresa para determinar si el proyecto que se propone será bueno o malo, y en cuales condiciones se debe desarrollar para que sea exitoso. Factibilidad es el grado en que lograr algo es posible o las posibilidades que tiene de lograrse.

Iniciar un proyecto o fortalecerlo significa invertir recursos como :

- ✓ Tiempo,
- ✓ Dinero,
- ✓ Materia prima y equipos.

Como los recursos siempre son limitados, es necesario tomar una decisión; las buenas decisiones sólo pueden ser tomadas sobre la base de evidencias y cálculos correctos, de manera que se tenga seguridad de que el proyecto se desempeñará correctamente y que producirá beneficios.

El presente proyecto dispone de un estudio previo de Factibilidad realizado por la empresa Fjord Seafood, por lo que sólo se mencionarán los puntos más importantes que se consideran al momento de realizar la Factibilidad de Proyecto.

## **6.1 Tiempo**

El proyecto a desarrollar por el equipo de trabajo consiste en diseñar, programar e implementar el sistema que dará solución a los requerimientos expresados por los usuarios del departamento de salud de la empresa Fjord Seafood.

Luego de intensas reuniones se determinaron los plazos de entrega e implementación final del sistema los cuales se definieron en 10 semanas, quedando estipulados en un contrato realizado entre ambas empresas. A pesar de que los plazos fueron especificados, el tiempo que tome este proyecto también estará determinado por la eficacia en la toma de requerimientos y en su correcta interpretación por el desarrollador, teniendo en claro que durante el proceso se puedan determinar otros requerimientos o cambios en la estructura del sistema.

## **6.2 Costos**

Los Costos asociados a este proyecto, corresponde a un factor muy importante ya que determinará si el proyecto es viable y si producirá ganancias en el tiempo. Este punto fue detenidamente evaluado por personal especializado de la empresa Fjord Seafood, determinando que es un proyecto absolutamente viable y necesario, ya que no se dispone de soluciones ni alternativas para agilizar y optimizar el tema salud en los centros de cultivo.

Un punto importante dentro de este análisis que se debe mencionar es que para abaratar algunos costos, la empresa determinó que el sistema deberá adaptarse a la infraestructura con que cuentan actualmente los centros, es decir, no habrá actualización de computadores ya que esto fue realizado hace poco tiempo. Esto permitirá reducir los costos de implementación ya que gran parte de los recursos, como por ejemplo computadores, se encuentran disponibles.

También se debe mencionar que la viabilidad del proyecto estuvo determinada por el tiempo de retorno de la inversión, ya que se estima que el ahorro en tiempo, recursos humanos y materia prima permitirá recuperar la inversión en unos 12 meses, considerando que ya no existirán muchas horas de digitación ni creación de complicadas planillas electrónicas, el tiempo por lo

tanto se reducirá a la habilidad y conocimiento que tenga el veterinario en ingresar la información. Los datos para obtener la “Declaración de Garantía” se encontraran ingresados en el sistema y podrán ser impresos inmediatamente, sin perder tiempo en buscar la información. Las cosechas se podrán estimar con mayor precisión ya que al determinar el término de los períodos de carencia de los grupos que han sido sometidos a tratamientos se podrá planificar y determinar la producción con tiempo lo cual se traducirá en mayores beneficios económicos para la empresa.

### **6.3 Materia prima y recursos**

La materia prima y los recursos son un factor determinante para llevar a cabo el presente proyecto. Dentro de materia prima podemos incluir el software de desarrollo y todos los programas y aplicaciones que se utilizarán para llevar a cabo la implementación del sistema.

Los recursos disponibles pueden interpretarse como el elemento que permitirá mediante la utilización de materia prima, llevar a cabo el proyecto.

El recurso grupo de trabajo fue determinado mediante reuniones entre la empresa Fjord Seafood y la empresa encargada de llevar a cabo el proyecto, AKVAsmart Chile S.A., quedando conformado de la siguiente manera:

- ✓ Un conjunto de veterinarios pertenecientes a la empresa Fjord Seafood,
- ✓ Tres Consultores en Acuicultura pertenecientes al departamento IT&C de la empresa AKVAsmart Chile S.A., los cuales entregarán las bases médicas y biológicas al sistema.
- ✓ Alumno tesista, quien estará a cargo del desarrollo de la parte informática del proyecto, el cual pertenece al departamento IT&C de la empresa AKVAsmart Chile S.A.

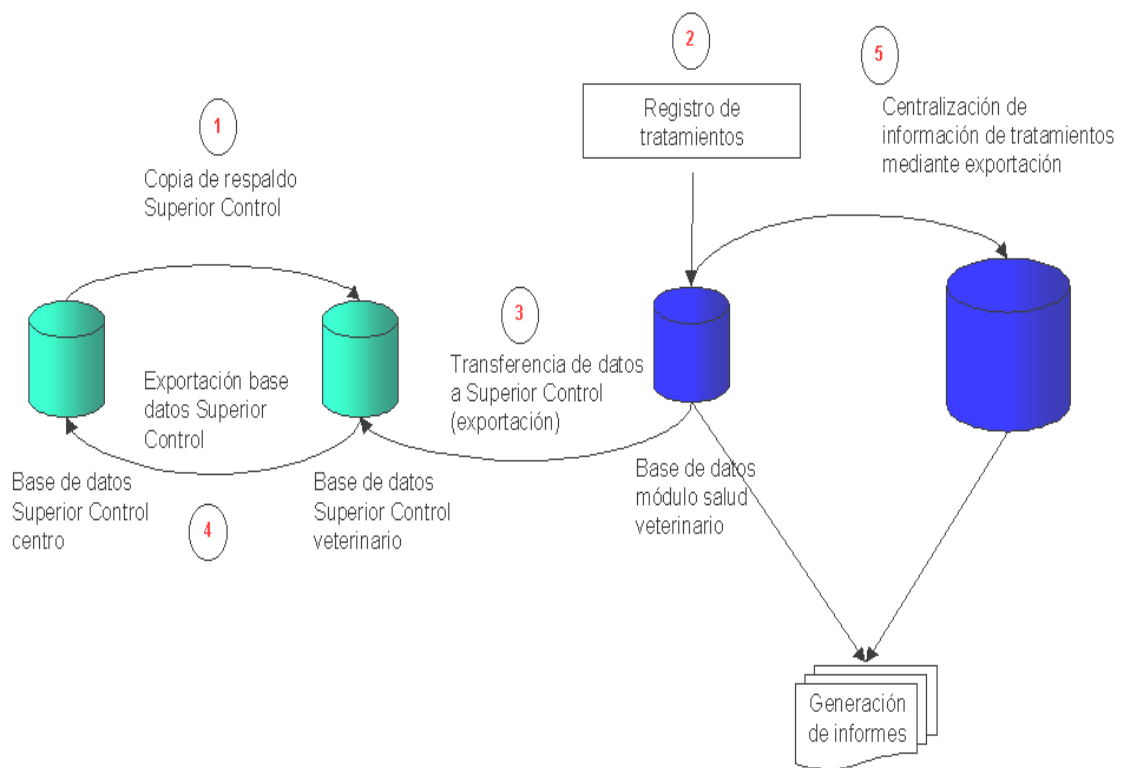
## **7. DEFINICIÓN DE SISTEMA**

Para comenzar con un proyecto es necesario definir los límites que tendrá el sistema y la manera en la cual se comunicará con las demás fuentes de información de la empresa.

La sección con la cual el sistema se comunicará será principalmente el Departamento de Salud de la empresa.

El departamento de salud de la empresa Fjord Seafood, no dispone en estos momentos de un sistema que permita registrar y obtener información de los tratamientos suministrados a los distintos centros de cultivo, por lo que una herramienta de esta naturaleza sería un real aporte al trabajo que se realiza en los centros y luego en las oficinas, al permitir ahorrar tiempo y recursos en trabajo manual.

La siguiente figura esquematiza el funcionamiento que deberá tener el sistema:



**Figura N° 6 : funcionamiento del Módulo de Tratamientos**

Su funcionamiento será el siguiente:

Existen datos ingresados en Superior Control de los centros ( jaulas, número de peces, grupos, subgrupos, biomasa, pesos promedios, entre otros), los cuales serán respaldados mediante una copia de respaldo de este sistema por el encargado del centro para mantener un resguardo ante cualquier eventualidad. Posteriormente, si es necesario, se importará desde el Superior Control local del veterinario algunos datos y parámetros, facilitando de esta manera el manejo de información evitando la pérdida de tiempo que conlleva digitar nuevamente datos que ya se encuentran ingresados. Este paso es posible debido a que Superior Control permite ingresar información de tratamientos y vacunas pero no a un mayor nivel de detalle. Si se trata de ingresar tratamientos que no se encuentran detallados en Superior Control, se puede obviar esta opción y partir digitando información desde cero.

El proceso de exportar datos corresponde a la función que tiene el sistema para poder extraer datos desde las tablas de Superior Control y guardarlas en un archivo con formato comprimido para que luego el sistema tome dicho archivo e incorpore los datos en otro computador que disponga de Superior Control. La acción de incorporar información al sistema corresponde



al proceso de “importación” de datos. El Módulo de Tratamientos incorpora ambas funcionalidades.

Continuando con la descripción, luego el veterinario digita información en el Módulo de Tratamientos la cual será exportada hacia la base de datos consolidada del Módulo que se encontrará en las oficinas centrales de Fjord Seafood. Este consolidado de datos corresponde a una recopilación de toda la información digitada en los centros referente a los tratamientos, la cual es almacenada en las oficinas centrales para que pueda ser analizada por las personas encargadas de obtener estadísticas y datos referentes a medicinas, tratamientos, estado de los centros, estado de los peces, etc. De esta manera, se encontrará disponible en todo momento la información, permitiendo la obtención de documentos y reportes indispensables para el correcto funcionamiento de los centros de cultivo de la empresa y para la toma de decisiones.

De igual manera, será posible importar directamente los datos digitados en el Módulo de Tratamientos hacia Superior Control para que se incorporen en las tablas donde se maneja información referente a tratamientos. Estas tablas de Superior solo permiten observar en el sistema que jaulas se encuentran con tratamientos y carencias asociadas, sin entrar en detalles. Este proceso se realizará para mantener cierta consistencia con el sistema de producción que es

desde donde se extrae casi la totalidad de los datos y reportes referentes a la producción misma (cosechas, mortalidades, porcentajes de crecimiento, alimentación, saldos de bodega, etc.)

Una parte importante a considerar es la implementación de los procesos que serán calculados por el sistema. El proceso de crear nuevos tratamientos implica programar complejas formulas matemáticas las cuales determinaran con precisión las dosis de medicamentos, el crecimiento de los peces, las tazas de alimentación, distribución del alimento, entre otros. Estos procesos son muy importantes ya que son los que permitirán calcular y ejecutar acciones relevantes para el correcto funcionamiento del sistema

Algunas de las fórmulas que se utilizarán se muestran a continuación:

$$a) Wf = Wi * e^{(SGR * \Delta t) / 100}$$

Esta fórmula corresponde al método para calcular el peso final de los peces ( $Wf$ ) a partir de un peso inicial ( $Wi$ ) multiplicado por el factor ( $e^{(SGR * \Delta t) / 100}$ ) el cual está compuesto por el SGR (Specific Growth Rate, Tasa de Crecimiento Específico) y un  $\Delta t$  que corresponde al período de tiempo o días considerados para calcular el peso final de los peces.

Esta fórmula se utilizará para hacer crecer los peces día a día en el sistema, corresponde además al crecimiento real de los peces durante el período de duración del tratamiento. Esta fórmula se utiliza para calcular el crecimiento de los peces en los centros de cultivo.

$$b) SGR = \frac{LN\left(\frac{Wf}{Wi}\right) * 100}{\Delta t}$$

Esta fórmula corresponde al SGR (Specific Growth Rate, Tasa de Crecimiento Específico) que permite observar la tasa de crecimiento de los peces en un período de tiempo. Está compuesta por el Logaritmo Natural ( $LN$ ) del resultado de la división del peso final ( $Wf$ ) por el peso inicial ( $Wi$ ), lo cual se

multiplica por la constante 100 y se divide por un delta de tiempo ( $\Delta t$ ) que corresponde al número de días que se desea calcular la tasa de crecimiento.

$$c) FCR = \frac{a\ limento}{IncrementoBiomasa}$$

Esta fórmula corresponde al Factor de Conversión el cual se refiere a la relación entre el alimento (*a limento*) entregado a los peces y el Incremento de la Biomasa (*IncrementoBiomasa*). El incremento de la biomasa se obtiene a partir de la resta entre la biomasa final y la biomasa inicial.

Si se determina que el FCR es 1, entonces por cada 1 kilo de alimento que se le entregue a los peces estos crecerán 1 kilo.

Con esta fórmula también se puede determinar la cantidad de alimento que se le debe suministrar a los peces para alcanzar la biomasa determinada. Por ejemplo, si el FCR es 0,8 y se desea alcanzar una biomasa de 1500 Kgr., entonces:

$$0,8 = \frac{a\ limento}{1500}$$

La cantidad de alimento que se debe suministrar es de 1200 Kgr.

$$d) \%PESOCUERPO = FCR * SGR * 100$$

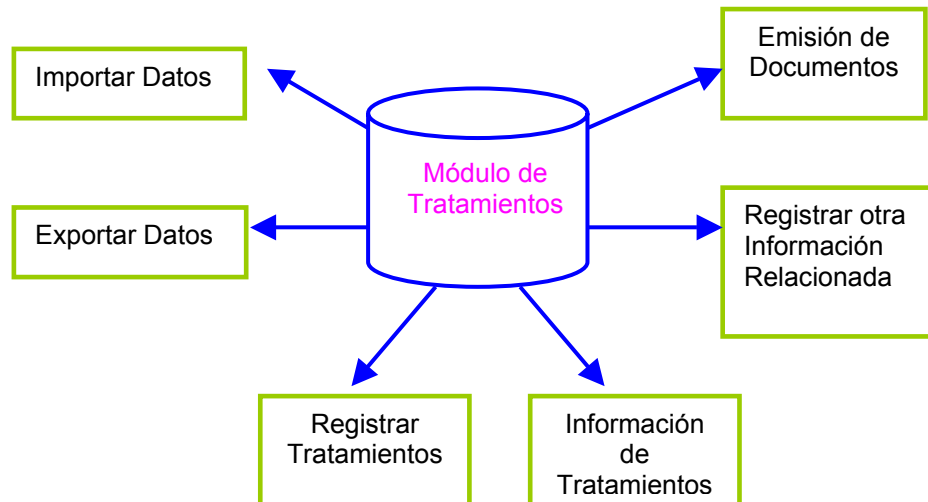
Para obtener el Porcentaje de Peso Cuerpo (*%PESOCUERPO*) se debe obtener primero el FCR y el SGR. En el módulo de Tratamientos el Porcentaje Peso Cuerpo se obtendrá de los últimos siete días reales alimentados, es decir, los siete días en los que realmente se entregó alimento

$$e) \%Crecimiento = \frac{Wf - Wi}{Wf}$$

Esta fórmula permite obtener el porcentaje de crecimiento de los peces. Este porcentaje se calcula en base al peso promedio inicial y final de los peces no de la biomasa.

En el anexo A sección A.1.4 se describe el código para determinar los cálculos de algunas de las fórmulas utilizadas.

Otro de los puntos importantes para la definición del sistema es determinar las funciones o acciones que se incluirán en él. La figura N° 7 esquematiza las funciones incluidas en el sistema.



**Figura N° 7: Funciones que incluirá el Módulo de Tratamientos**

El sistema estará compuesto principalmente por las siguientes funciones:

- a) **Importar datos:** esta función permitirá extraer datos desde el sistema de producción de peces Superior Control hacia el Módulo de Tratamientos, de esta manera la información que ya se encuentra ingresada en el sistema de producción podrá ser aprovechada por el nuevo Módulo de Tratamientos. Ver anexo A sección A.1.2.

- b) **Exportar datos:** esta funcionalidad permitirá “enviar” datos o extraerlos a un archivo de base de datos Access para guardarlo como respaldo del sistema. Permitirá además enviar información hacia la base de datos de Superior Control de manera que el sistema de producción cuente con información actualizada de los tratamientos para determinar con precisión los grupos y unidades que pueden ser cosechados. Ver anexo A sección A.1.1 y sección A.1.3.
  
- c) **Registrar Tratamientos:** esta funcionalidad permitirá crear e ingresar tratamientos al sistema de manera que pueda existir un registro de información que luego permitirá su utilización en otras funciones del sistema por ejemplo: importación de datos, exportación de datos, emisión de documentos, etc.
  
- d) **Información de Tratamientos:** esta funcionalidad permitirá visualizar información referente a detalles de los tratamientos, por ejemplo: Tratamientos registrados, Declaraciones de Garantía registradas, tipos de tratamientos, centros, veterinarios, alimentos, etc.
  
- e) **Registrar otra información relacionada:** esta funcionalidad permitirá ingresar al sistema otra información también relacionada con los tratamientos, por ejemplo: crear nuevas empresas, agregar veterinarios

al sistema, crear nuevas dietas o alimentos, agregar centros, agregar tipos y proveedores de tratamientos, entre otros.

- f) **Emisión de documentos:** esta función será diseñada para permitir la salida de documentos legales exigidos por SERNAPESCA, por ejemplo: Declaración de Garantía, Prescripción Veterinaria.

Otra de las funcionalidades que se incorporarán al sistema son los archivos de ayuda, los cuales serán desarrollados con la herramienta RoboHelp Office 9.0. Estos archivos se desarrollan como una aplicación totalmente independiente y posteriormente dentro del código del sistema se integran realizando una llamada a la aplicación.

También se considera la creación de los archivos de instalación del sistema. Esta funcionalidad genera los archivos necesarios para lograr el funcionamiento del sistema en cualquiera de los ambientes que se requiera, por ejemplo : Windows 98, Windows 2000, entre otros, agregando todos los archivos relacionados con la aplicación, generando de esta manera la instalación del Módulo de Tratamientos. El software a utilizar es Wise Install Builder 7.04.



Es determinante considerar todos los aspectos que involucra un sistema para lograr alcanzar los objetivos planteados por los usuarios en los requerimientos. El presente sistema involucra diseñar, desarrollar e implementar el Módulo de Tratamientos para lo cual ya se definieron las funciones más importantes que serán incluidas, a continuación se detallan los requerimientos especificados por los usuarios y posteriormente se detalla todo el proceso que involucró el diseño y creación de la Base de Datos, lo cual es parte importante dentro del sistema ya que deberá interactuar en forma directa con una base de datos externa (Sistema de Producción Superior Control 5.11) para lo cual se debe diseñar e implementar entidades y relaciones que permitan dicha interacción.

## **8. ANALISIS Y RECOLECCION DE REQUERIMIENTOS**

Uno de los problemas que se presenta durante el desarrollo de un proyecto es la captura de los requisitos del usuario. Para empezar, se necesita “recoger los requisitos de los usuarios de una manera sistemática y organizada. Para ello se precisa de unas directrices o líneas guía, ya que en general los usuarios expresan los requerimientos de la aplicación de forma muy variable, tanto en la forma como en el contenido, por lo tanto es necesario sistematizar la captura, con el fin de hacer los requisitos manejables y analizables”. [García1996]

Una vez conseguidos los requisitos, continúa la fase de análisis de requerimientos. En ella, se analizan los requisitos obtenidos de los usuarios con el fin de comprenderlos, y a partir de ellos desarrollar una especificación de la aplicación, que deberá ser completa y consistente. En este proceso, aparecen habitualmente gran cantidad de problemas en los requisitos, áreas no especificadas, requisitos contradictorios, y afirmaciones (aparentemente) vagas e irrelevantes. Esto significa volver a los usuarios con el fin de mejorar la calidad de los requisitos: pero se deben abordar sabiendo lo que se quiere

conseguir, qué aspectos de los requisitos obtenidos inicialmente interesa aclarar, y el porqué.

Después de obtener una buena lista de requisitos, se comienza con el análisis y el diseño de la aplicación.

Las técnicas que se utilizaron en la recolección de requerimientos son:

- ✓ Entrevistas al personal especializado que participó en el proyecto o que va a utilizar la aplicación
- ✓ Examinar documentos que se utilicen actualmente

A continuación se muestran las técnicas de recolección de requerimientos que se utilizaron para el diseño del Módulo Tratamientos.

## 8.1 Entrevistas

Esta técnica es una de las más utilizadas en la recolección de requerimientos, ya que es posible obtener opiniones muy personales de los usuarios, y por lo tanto, se logra determinar y definir cuales requerimientos de los usuarios deben ser satisfechos por la aplicación.

En cada entrevista aparecen nuevos requerimientos, por lo tanto, se debe realizar un proceso iterativo que permita obtener todos los requerimientos necesarios.

Para el desarrollo del presente proyecto, se realizarán dos tipos de entrevistas:

✓ **Entrevistas:** se entrevistó a personal especializado de la empresa Fjord Seafood, en este caso a distintos veterinarios, los cuales aportaron los requerimientos específicos del área salud.

✓ **Reuniones:** se realizaron reuniones con personal de la empresa Fjord Seafood del área informática y con personal del departamento IT&C de AKVAsmart Chile S.A. del área consultoría en acuicultura e informática, para determinar los requerimientos y el esquema de trabajo.

## **8.2 Examinar documentos**

Al observar el procedimiento manual con que se encontraban registrando la información en los centros, se puede obtener importante información acerca de la necesidad de disponer de un método que proporcione mayor confiabilidad y comodidad.

Los documentos que se analizaron corresponden a planillas generadas en Microsoft Excel, las cuales mediante complejas fórmulas logran realizar los cálculos necesarios que permitirán manejar y registrar el historial de tratamientos de los centros. El problema surge cuando se necesita obtener información que ya fue registrada en los documentos y que debe ser buscada en los archivos y folios donde se almacenan.

Por otro lado se analizaron otros documentos relacionados con cálculos que se realizan antes de suministrar un tratamiento, los cuales fueron creados por los médicos veterinarios de la empresa. En el anexo B del capítulo 14 se presenta el formato de los documentos analizados.

El diagnosticar la situación actual, significa el estudio y comprensión de los procedimientos manuales que maneja la empresa, tanto en relación al trabajo mismo de los veterinarios como generales respecto del área Informática, es decir la situación ideal, para así poder definir alternativas de solución, según las cuales se podrá avanzar desde lo que se posee, hacia el punto que se pretende llegar.

### 8.3 Requerimientos

Una vez terminada la tarea de colección y análisis de requerimientos, se puede concluir que los requerimientos a considerar son los siguientes:

- 1) Importar información de los tratamientos desde Superior Control.
- 2) Exportar información a una base de datos.
- 3) Exportación automática hacia Superior Control de la información de los nuevos tratamientos.
- 4) Crear nuevos tratamientos tanto para centros de agua mar y dulce.
- 5) Permitir el registro de tratamientos por vía oral y baño.
- 6) Obtener información en todo momento de los tratamientos en el centro.
- 7) Debe permitir el ingreso de proveedores de medicamentos y tratamientos.
- 8) Permitir el ingreso de medicamentos por proveedor.
- 9) Permitir un registro de veterinarios en el centro.
- 10) Agregar y crear nuevas dietas medicadas en el centro.
- 11) Emisión de documentos legales, ejemplo: la Declaración de Garantía.
- 12) Obtener información económica involucrada en el tratamiento (costos asociados).
- 13) Obtener información de dosificación de medicamento.

- 14) Obtener información de la alimentación y su distribución durante los días de tratamiento de las jaulas o unidades involucradas.
- 15) Obtener información relevante de los peces durante el período de tratamiento: N° inicial y final de peces, porcentaje de mortalidad, FCR, biomasa, etc.
- 16) Consolidar Información.
- 17) Soporte para múltiples empresas y múltiples centros. Soporte multilinguaje.
- 18) Entorno amigable.
- 19) Archivos de ayuda y soporte.



## **9. DISEÑO DE LA BASE DE DATOS**

Las bases de datos antiguamente eran el resultado de una compleja programación y complicados mecanismos de almacenamiento. Con el crecimiento y desarrollo de la informática, la aparición de aplicaciones específicas también trajo con ella la disponibilidad de herramientas de gestión de datos, que terminaron desembocando en los denominados sistemas de gestión de bases de datos (Data Base Management Systems, DBMS).

Para que fuese posible, y para asegurar a los usuarios cierta seguridad en el intercambio de datos entre diferentes sistemas, y en el diseño de bases de datos, fue necesario normalizar los esquemas que guiaban la creación de bases de datos. De esta manera surgen diversas metodologías y procedimientos, tendientes a definir un enfoque estructurado de los pasos a seguir en el diseño de una base de datos.

La Metodología que se va a utilizar en este proyecto es la que lleva por título “Ciclo de Vida de Base de Datos” del autor Thomas Connolly [Connolly1999].

## **9.1 Diseño de Base de Datos Conceptual**

El diseño de base de datos conceptual puede definirse como una descripción abstracta y general de la parte o sector del universo real que el contenido de la base de datos va a representar [Connolly1999]. En este nivel de análisis se está tratando con una descripción de la realidad, no con datos, y contiene un detalle de los tipos de entidades, de las relaciones existentes entre esas entidades y de las restricciones de integridad que se aplican sobre ellas.

### **9.1.1 Identificación de entidades**

Aunque al principio resulta complicado definir las entidades involucradas en un sistema, es a través de un correcto y detallado análisis de los requerimientos que se logra obtener el esquema que conformará el sistema. A continuación la Tabla N° 1 identifica las entidades presentes en el Módulo de Tratamientos.

**Tabla N° 1 Identificación de Entidades**

<b>Entidad</b>	<b>Descripción</b>	<b>Alias</b>	<b>Ocurrencia</b>
<b>Company</b>	Entidad que registra la información de la empresa		Existe una empresa con muchos centros
<b>Sites</b>	Describe los centros presentes en la base de datos de Superior Control.		Una empresa puede tener muchos centros.
<b>Treatments</b>	Información de todos los Tratamientos que son usados en los centros de cultivo por la empresa.		Un tratamiento puede ser suministrado a una o más jaulas de un centro.
<b>Dtreatments</b>	Entidad que registra el detalle del tratamiento por jaula tratada. Registra la información del proveedor de medicamentos, inicio y fin del tratamiento, dosis, mortalidad, etc.		Existen tantos registros como tratamientos se administren en el centro.
<b>Treatment_Supplier</b>	Proveedores de Tratamientos o medicamentos.		Un proveedor puede proveer muchos tratamientos.
<b>Feed_Supplier</b>	Información del proveedor de alimento medicado y el código asociado a este.		Pueden existir muchos proveedores de alimentos o dietas en un centro

<b>Entidad</b>	<b>Descripción</b>	<b>Alias</b>	<b>Ocurrencia</b>
<b>Veterinary</b>	Entidad que registra la identificación de los profesionales (veterinarios) a cargo de la salud de los centros de cultivo.		En una empresa pueden existir muchos veterinarios.
<b>TreaBySupplier</b>	Entidad que registra a un proveedor y sus tratamientos.		Un proveedor puede tener muchos tratamientos.
<b>Type_Treatments</b>	Entidad que registra los tipos de tratamientos y sus principales detalles: UTA, días de carencia, dosis, etc.		En un centro se pueden registrar muchos tratamientos distintos
<b>Ltreatment</b>	Descripción de los tratamientos en distintos idiomas. Esta tabla está directamente relacionada con la tabla de parámetros de Superior Control		Un tratamiento puede tener muchas descripciones (en distintos idiomas)
<b>FeedTypes</b>	Entidad que almacena la información referente a los tipos de dietas medicadas que se utilizaran en los tratamientos: proveedor de alimento, proveedor de medicamento, pigmento de la dieta, calibre, entre otros.		Existen muchos tipos de dietas, con proveedores, concentraciones de medicamento, calibres y pigmentos distintas.

Entidad	Descripción	Alias	Ocurrencia
<b>FeedSize</b>	Registra el calibre de una dieta o alimento.		Una misma dieta o alimento puede tener muchas unidades de medida dependiendo del proveedor de alimentos.
<b>Diets</b>	Tipos de dietas o alimentos, descritas según su nombre comercial por proveedor.		Pueden existir muchas dietas de distintos proveedores.
<b>Dfeeding</b>	Entidad que registra en los tratamientos orales, el detalle de la alimentación por día y por jaula		Una jaula puede tener muchos días de alimentación especial.

### **9.1.2 Identificación de relaciones**

En este paso se identifican las relaciones existentes entre las entidades presentes en el sistema.

**Tabla N° 2 Identificación de relaciones**

<b>Entidad</b>	<b>Relación</b>	<b>Descripción</b>	<b>Entidad</b>	<b>Ocurrencia</b>	<b>Participación</b>
<b>Company</b>	Tiene	Describe la pertenencia de los centros a una empresa determinada	<b>Sites</b>	<b>1 : N</b>	<b>M : M</b>
<b>Sites</b>	Registra	Describe la relación entre los centros de cultivo y su registro de tratamientos	<b>Treatments</b>	<b>1 : N</b>	<b>M : M</b>
<b>Treatments</b>	Tiene	Describe el detalle del registro de los tratamientos.	<b>Dtreatments</b>	<b>1 : N</b>	<b>M : M</b>
	Es prescrito	Relación del profesional de la salud del centro y los tratamientos que prescribe	<b>Veterinary</b>	<b>N : 1</b>	<b>M : M</b>

Entidad	Relación	Descripción	Entidad	Ocurrencia	Participación
<b>Treatment</b> (viene de la página anterior)	Tiene	Relación que indica que muchos tratamientos pueden provenir de un proveedor de Tratamiento puede	<b>TreatBySupplier</b>	<b>N : 1</b>	<b>M : M</b>
<b>Dtreatment</b>	Se asocia	Describe la relación entre el detalle del tratamiento y la alimentación día a día de cada jaula.	<b>Dfeeding</b>	<b>1 : N</b>	<b>M : M</b>
	Se asocia	Relación entre el detalle del tratamiento y los tipos de dietas que se utilizarán	<b>FeedTypes</b>	<b>1 : 1</b>	<b>M : O</b>
<b>FeedSupplier</b>	Tiene	Describe la relación entre el proveedor y la dieta o alimento	<b>Diets</b>	<b>1 : N</b>	<b>M : M</b>
	Tiene	Relación existente entre el proveedor y la unidad de medida que describe la dieta	<b>FeedSize</b>	<b>1 : N</b>	<b>M : M</b>



Entidad	Relación	Descripción	Entidad	Ocurrencia	Participación
<b>Diets</b>	Tiene	Describe la relación entre el nombre comercial de un alimento y su tipo	<b>FeedTypes</b>	<b>1 : N</b>	<b>M : M</b>
<b>FeedTypes</b>	Tiene	Toda dieta perteneciente a un proveedor tiene una unidad de medida o calibre distinto ,	<b>FeedSize</b>	<b>N : 1</b>	<b>M : M</b>
	Tiene	Relación entre un proveedor determinado y un tipo de tratamiento	<b>Type_Treatment</b>	<b>N : 1</b>	<b>M : M</b>
<b>Treatment_Supplier</b>	Tiene	Cada proveedor de tratamientos puede tener más de un tratamiento asociado	<b>TreatBySupplier</b>	<b>1 : N</b>	<b>M : M</b>
<b>Type_Treatment</b>	Tiene	Cada tipo de tratamiento tiene asociado su descripción en distintos idiomas (inglés, español).	<b>Ltreatment</b>	<b>1 : N</b>	<b>M : M</b>

Entidad	Relación	Descripción	Entidad	Ocurrencia	Participación
<b>Type_Treatment</b>  (viene de la página anterior)	Tiene	Cada tipo de tratamiento puede tener asociado más de un código por tipo de proveedor.	<b>TreatBySupplier</b>	<b>1 : N</b>	<b>M : M</b>

### **9.1.3 Identificación y asociación de atributos con tipos de entidades y relaciones**

Siguiendo el esquema propuesto por la metodología, se identifican y determinan los atributos de las entidades y relaciones, las cuales van a ser representadas en la base de datos final.

#### **Nomenclatura:**

**R** : Restricción

**VD** : Valor por Defecto

**VN** : Valor Nulo

**D** : Derivado

**M** : Multivalorado

**C** : Compuesto

**Tabla N° 3 Identificación y asociación de atributos con tipos de entidades  
y relaciones**

<b>Entidad/ Relación</b>	<b>Atributos</b>	<b>Descripción</b>	<b>Tipo de Dato Tamaño</b>	<b>R</b>	<b>VD</b>	<b>VN</b>	<b>D</b>	<b>M</b>	<b>C</b>
<b>Company</b>	Code_Company	Código que identifica a la empresa	Texto	SI	NO	NO	NO	NO	NO
	Name_company	Nombre de la empresa	Texto	NO	NO	SI	NO	NO	NO
	Phone	Teléfono de la empresa	Numérico	0-9	0	SI	NO	NO	NO
	Fax	Fax de la empresa	Numérico	0-9	0	SI	NO	NO	NO
<b>Sites</b>	Code_Company	Código que identifica a la empresa	Texto	SI	NO	NO	NO	NO	NO
	Id_Concession	Código asignado a la concesión marítima	Texto	SI	NO	NO	NO	NO	NO
	Code_Site	Código que identifica al centro	Texto	SI	NO	NO	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Sites</b> (viene de la página anterior)	Description	Descripción del tratamiento	Texto	NO	NO	SI	NO	NO	NO
	TypeSite	Tipo de centro (agua mar o dulce)	Numérico	0-9	0	SI	NO	NO	NO
<b>Treatments</b>	N_Identity	Código que identifica al médico veterinario.	Numérico	0-9	0	NO	NO	NO	NO
	YearReg	Año del tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	CorrReg	Correlativo del registro de tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	Code_Company	Código que identifica a la empresa	Texto			SI	NO	NO	NO
	Id_Concession	Código asignado a la concesión marítima	Texto			SI	NO	NO	NO
	Code_Site	Código que identifica al centro	Texto			SI	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Treatment</b> (Viene de la Página Anterior)	Dates	Fecha del tratamiento	Fecha/hora			SI	NO	NO	NO
	Code_TSite	Código tipo de centro	Numérico	0-9	0	SI	NO	NO	NO
	Code_TUnit	Código tipo de unidad o jaula	Numérico	0-9	0	SI	NO	NO	NO
	Code_WTreat	Forma de suministrar el tratamiento (oral o baño)	Numérico	0-9	0	SI	NO	NO	NO
	Code_TDiag	Tipo de diagnóstico	Numérico	0-9	0	SI	NO	NO	NO
	Code_Treatment	Código del tipo de tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	Code_Supplier	Código del proveedor de medicamento	Numérico	0-9	0	SI	NO	NO	NO
	N_Days_Treatment	Número de días de duración del tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	CorrPC	Corrección de peso cuerpo	Numérico	0-9	0	SI	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
Treatment (Viene de la Página Anterior)	FCR	Factor de conversión	Numérico	0-9	0	SI	NO	NO	NO
	PretDose	Dosis predeterminada de medicamento	Numérico	0-9	0	SI	NO	NO	NO
	Concent	Concentración de medicamento (si es al 50% o 80%)	Numérico	0-9	0	SI	NO	NO	NO
	ProdCom	Nombre comercial del medicamento	Numérico	0-9	0	SI	NO	NO	NO
	ValNeto	Valor Neto del producto	Numérico	0-9	0	SI	NO	NO	NO
	ValNetoT	Valor Neto Total del producto	Numérico	0-9	0	SI	NO	NO	NO
	ValMG	Valorización del medicamento	Numérico	0-9	0	SI	NO	NO	NO
	CostKg	Costo del medicamento por kilo	Numérico	0-9	0	SI	NO	NO	NO

<b>Entidad/ Relación</b>	<b>Atributos</b>	<b>Descripción</b>	<b>Tipo de Dato Tamaño</b>	<b>R</b>	<b>VD</b>	<b>VN</b>	<b>D</b>	<b>M</b>	<b>C</b>
<b>Treatment (Viene de la Página Anterior)</b>	CostDay	Costo del medicamento por día	Numérico	0-9	0	SI	NO	NO	NO
	PerIngAct	Porcentaje de ingrediente activo (presente en el medicamento)	Numérico	0-9	0	SI	NO	NO	NO
	PerProdCom	Porcentaje del producto comercial	Numérico	0-9	0	SI	NO	NO	NO
	KgIngAct	Kilogramos de ingrediente activo	Numérico	0-9	0	SI	NO	NO	NO
	KgProdCom	Porcentaje producto comercial	Numérico	0-9	0	SI	NO	NO	NO
<b>Veterinary</b>	N_Id	Código que identifica al médico veterinario.	Numérico	0-9	0	NO	NO	NO	NO
	Name_Veterinary	Nombre del médico veterinario	Texto			SI	NO	NO	NO



Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Veterinary</b> (viene de la página anterior)	Position	Cargo que ocupa en la empresa (ejemplo: jefe de salud)	Texto			SI	NO	NO	NO
	N_VetCollege	Número de registro en el colegio médico veterinario	Numérico	0-9	O	SI	NO	NO	NO
	E_mail	Email del veterinario	Texto			SI	NO	NO	NO
	Paswd	Password del veterinario para ingresar al sistema	Texto			SI	NO	NO	NO
<b>Diets</b>	IdSup	Identificación del proveedor de dietas	Numérico	0-9	0	NO	NO	NO	NO
	IdDiet	Código de identificación interna de la dieta en el sistema	Numérico	0-9	0	NO	NO	NO	NO
	NameDiet	Nombre de la dieta o alimento	Texto			SI	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Diets</b> (viene de la página anterior)	Activ	Código que identifica si la dieta está disponible para ser utilizada	Numérico	0-9	0	SI	NO	NO	NO
<b>FeedSupplier</b>	IdSup	Identificación del proveedor de dietas	Numérico	0-9	0	NO	NO	NO	NO
	NameSup	Nombre del proveedor de alimento	Testo			SI	NO	NO	NO
	Activ	Código que identifica si la dieta está disponible para ser utilizada	Numérico	0-9	0	SI	NO	NO	NO
<b>FeedSize</b>	IdSup	Identificación del proveedor de dietas	Numérico	0-9	0	NO	NO	NO	NO
	IdSize	Código que identifica el tamaño o calibre de la dieta.	Numérico	0-9	0	NO	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>FeedSize</b> (viene de la página anterior)	DesSize	Descripción del calibre o tamaño de la dieta	Texto			SI	NO	NO	NO
<b>FeedTypes</b>	IdSup	Identificación del proveedor de dietas	Numérico	0-9	0	NO	NO	NO	NO
	IdDiet	Código que identifica la dieta o alimento	Numérico	0-9	0	NO	NO	NO	NO
	IdSize	Código que identifica el tamaño o calibre de la dieta. Distintos proveedores pueden tener distintos calibres	Numérico	0-9	0	NO	NO	NO	NO
	IdPig	Identificador del pigmento que compone la dieta.	Numérico	0-9	0	NO	NO	NO	NO
	IdTPig	Código que identifica el tipo de pigmento	Numérico	0-9	0	NO	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>FeedTypes</b> (viene de la página anterior)	IdTreat1	Un alimento puede estar mezclado hasta con dos tratamientos	Numérico	0-9	0	NO	NO	NO	NO
	IdTreat2	Un alimento puede estar mezclado hasta con dos tratamientos	Numérico	0-9	0	NO	NO	NO	NO
	FullCode	Código compuesto de distintos valores de la tabla.	Texto			SI	NO	NO	NO
<b>Treatment_Supplier</b>	Code_Supplier	Código del proveedor de alimentos o dietas	Numérico	0-9	0	NO	NO	NO	NO
	Name_Supplier	Nombre del proveedor de alimento.	Texto			SI	NO	NO	NO
	Phone	Teléfono del proveedor de alimento	Numérico	0-9	0	SI	NO	NO	NO

<b>Entidad/ Relación</b>	<b>Atributos</b>	<b>Descripción</b>	<b>Tipo de Dato Tamaño</b>	<b>R</b>	<b>VD</b>	<b>VN</b>	<b>D</b>	<b>M</b>	<b>C</b>
<b>Treatment_ Supplier</b> (viene de la Página anterior)	Fax	Fax del proveedor de alimento	Numérico	0-9	0	SI	NO	NO	NO
	Contac	Contacto entre la empresa y el proveedor de alimentos	Texto			SI	NO	NO	NO
<b>Ltreatment</b>	Code_Treatment	Código del tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	Code_Lang	Código del idioma del parámetro	Numérico	0-9	0	NO	NO	NO	NO
	Name_Treatment	Nombre del tratamiento	Texto			SI	NO	NO	NO
<b>Type_Treatment</b>	Code_Treatment	Código del tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	Cod_Med	Código del medicamento	Texto			SI	NO	NO	NO
	ATU	Unidades Térmicas Acumuladas	Numérico	0-9	0	SI	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Type_Treatment</b> (viene de la página anterior)	QDays	Días predeterminados de duración de la carencia	Numérico	0-9	0	SI	NO	NO	NO
	Rec_Dose	Dosis recomendada de medicamento	Numérico	0-9	0	SI	NO	NO	NO
	MaxCon	Concentración máxima de medicamento a utilizar	Numérico	0-9	0	SI	NO	NO	NO
	Remark	Código que indica que tipo de tratamiento se está utilizando	Memo			SI	NO	NO	NO
<b>Dfeeding</b>	N_Identity	Código que identifica al médico veterinario.	Numérico	0-9	0	NO	NO	NO	NO
	YearReg	Año en que se registra el tratamiento	Numérico	0-9	0	NO	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Dfeeding</b> (viene de la página anterior)	CorrReg	Correlativo del registro del tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	Id_Cage	Identificación de la unidad o jaula	Texto			NO	NO	NO	NO
	NDay	Número de días de duración del tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	AmountFeed	Cantidad de alimento a utilizar durante el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
<b>Dtreatments</b>	N_Identity	Código que identifica al médico veterinario.	Numérico	0-9	0	NO	NO	NO	NO
	YearReg	Año en que se registra el tratamiento	Numérico	0-9	0	NO	NO	NO	NO
	CorrReg	Correlativo que identifica el registro	Numérico	0-9	0	NO	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Dtreatments</b> (viene de la página anterior)	Code_Company	Código que identifica a la empresa	Numérico	0-9	0	SI	NO	NO	NO
	Id_Concession	Código asignado a la concesión marítima	Texto			SI	NO	NO	NO
	Code_Site	Código que identifica al centro	Numérico	0-9	0	SI	NO	NO	NO
	Code_Group	Código que identifica el grupo de peces.	Texto			SI	NO	NO	NO
	Correlative	Correlativo que identifica al subgrupo de peces	Texto			SI	NO	NO	NO
	Id_Cage	Identificador de la jaula o unidad	Texto			NO	NO	NO	NO
	Dates_Reg	Fecha de registro del tratamiento	Fecha/hora			SI	NO	NO	NO



Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Dtreatments</b> (viene de la página anterior)	Code_Specie	Código de la especie a la que pertenecen los peces.	Numérico	0-9	0	SI	NO	NO	NO
	Name_Group	Nombre que identifica al grupo de peces	Texto			SI	NO	NO	NO
	NFishIni	Número de peces al iniciar el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	AWIni	Peso promedio al iniciar el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	Biolni	Biomasa al iniciar el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	RealPC	Peso cuerpo real calculado de los últimos 7 días alimentados	Numérico	0-9	0	SI	NO	NO	NO
	PCCalc	Peso real por el porcentaje de corrección (CorrPC x RealPC)	Numérico	0-9	0	SI	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Dtreatments</b> (viene de la página anterior)	PerMort	Porcentaje de mortalidad, valor ingresado en porcentaje	Numérico	0-9	0	SI	NO	NO	NO
	NFishEnd	Número de peces al finalizar el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	AWEnd	Peso promedio al finalizar el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	BioEnd	Biomasa al finalizar el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	FullCode	Código compuesto de diversos valores de campos de esta tabla.	Texto			SI	NO	NO	NO
	FeedxDay	Alimentación calculada por día por unidad	Numérico	0-9	0	SI	NO	NO	NO
	FeedTot	Alimentación total durante el tratamiento	Numérico	0-9	0	SI	NO	NO	NO

Entidad/ Relación	Atributos	Descripción	Tipo de Dato Tamaño	R	VD	VN	D	M	C
<b>Dtreatments</b> (viene de la página anterior)	Start_Date	Fecha inicial de alimentación	Fecha/hora			SI	NO	NO	NO
	End_Date	Fecha final de alimentación	Fecha/hora			SI	NO	NO	NO
	Date_Quarantine	Fecha en que termina la cuarentena	Fecha/hora			SI	NO	NO	NO
	Dose	Dosis de medicamento para ser utilizado en el tratamiento	Numérico	0-9	0	SI	NO	NO	NO
<b>TreatBySupplier</b>	Code_Treatment	Código del tratamiento	Numérico	0-9	0	SI	NO	NO	NO
	Code_Supplier	Código del proveedor de alimentos o dietas	Numérico	0-9	0	SI	NO	NO	NO

#### **9.1.4 Determinación de dominios de atributos**

Para los atributos definidos en el Modelo, se determinan los dominios de las entidades, de esta manera se pueden conocer cuales son los valores que puede adoptar un atributo.

**Tabla N° 4 Determinación de dominios de atributos**

<b>Atributo</b>	<b>Característica del Atributo</b>	<b>Ejemplos</b>
Code_company	50 Caracteres Alfanuméricos	775098902
Name_company	50 Caracteres Alfabético	AKVASMART CHILE S.A.
Phone	Caracter Numérico (entero largo)	257502
Fax	Caracter Numérico (entero largo)	257508
ID_Concession	50 Caracteres Alfanuméricos	75098902
Code_Site	Caracter Numérico (entero largo)	006
Description	50 Caracter Alfabético	MN Huelden
Typesite	Caracter Numérico (Entero largo)	77301
N_Identity	Caracter Numérico (entero largo)	1
YearReg	Caracter Numérico (entero largo)	2002
CorrReg	Caracter Numérico (entero largo)	2
Dates	Fecha/Hora	11/09/2001
Code_TSite	Caracter Numérico (entero largo)	77301
Code_TUnit	Caracter Numérico (entero largo)	77101

<b>Atributo</b>	<b>Característica del Atributo</b>	<b>Ejemplos</b>
Code_WTreat	Caracter Numérico (entero largo)	77202
Code_TDiag	Caracter Numérico (entero largo)	77502
Code_Treatment	Caracter Numérico (Entero largo)	30
Code_Supplier	Caracter Numérico (Entero largo)	3
N_Days_Treatment	Caracter Numérico (entero largo)	7
CorrPC	Caracter Numérico (doble)	1500
FCR	Caracter Numérico (doble)	1
PretDose	Caracter Numérico (doble)	90
Concent	Caracter Numérico (doble)	50
ProdCom	Caracter Numérico (doble)	50
ValNeto	Caracter Numérico (doble)	1500
ValNetoT	Caracter Numérico (doble)	15750
ValMG	Caracter Numérico (doble)	0.003
CostKg	Caracter Numérico (doble)	1.89
CostDay	Caracter Numérico (doble)	11905
PerIngAct	Caracter Numérico (doble)	2.5
PerProdCom	Caracter Numérico (doble)	5
KgIngAct	Caracter Numérico (doble)	25
KgProdCom	Caracter Numérico (doble)	50
Code_group	04 Caracter Alfanumérico	0032
Correlative	03 Caracter Alfanumérico	022
Id_Cage	04 Caracter Alfanumérico	0201

<b>Atributo</b>	<b>Característica del Atributo</b>	<b>Ejemplos</b>
Dates_Reg	Fecha/Hora	11/09/2001
Code_especie	Caracter Numérico (entero largo)	1
Name_Group	50 Caracter Alfabético	SLAILANO00
NFishIni	Caracter Numérico (entero largo)	24589
AWIni	Caracter Numérico (doble)	1424.4644775
BioIni	Caracter Numérico (doble)	35469.165491
RealPC	Caracter Numérico (doble)	0.3534906193
PCCalc	Caracter Numérico (doble)	0.3534906193
PerGrowth	Caracter Numérico (doble)	2.44407116287145
PerMort	Caracter Numérico (doble)	81
NFishEnd	Caracter Numérico (entero largo)	12107
AWEnd	Caracter Numérico (doble)	4667.1785398
BioEnd	Caracter Numérico (doble)	56505.530581
Fullcode	100 Caracter Alfanumérico	1551677601300
FeedxDay	Caracter Numérico (doble)	200
FeedTot	Caracter Numérico (doble)	1400
Start_Date	Fecha/Hora	08/09/2001
End_Date	Fecha/Hora	14/09/2001
Date_Quarantine	Fecha/Hora	13/12/2001
Dose	Caracter Numérico (doble)	12150
N_Id	Caracter Numérico (Entero largo)	1
Name_Veterinary	50 Caracter Alfabético	Ana María Moris
Position	50 Carácter Alfabético	Jefe Salud

<b>Atributo</b>	<b>Característica del Atributo</b>	<b>Ejemplos</b>
N_VetCollege	Caracter Numérico (Entero largo)	456456
E_mail	50 Carácter Alfanumérico	<a href="mailto:amoris@jefesalud.cl">amoris@jefesalud.cl</a>
Passwd	50 Carácter Alfanumérico	Amoris2002
Id_sup	Caracter Numérico (entero largo)	1
IdDiet	Caracter Numérico (entero largo)	1
NameDiet	50 Carácter Alfanumérico	Vextra Alpha
Activ	Caracter Numérico (entero)	-1
IdSize	Caracter Numérico (Entero largo)	5
DesSize	50 Caracter Alfanumérico	XS, 3.0, .....
NameSup	50 Carácter Alfabético	Salmofood
IdPig	Caracter Numérico (Entero largo)	1
Pigment	Caracter Numérico (Entero largo)	20
IdTPig	Caracter Numérico (Entero largo)	77605
IdTreat1	Caracter Numérico (Entero largo)	60
IdTreat2	Caracter Numérico (Entero largo)	20
Code_Supplier	Caracter Numérico (Entero largo)	1
Name_Supplier	50 Carácter Alfabético	Animal Service Latina



<b>Atributo</b>	<b>Característica del Atributo</b>	<b>Ejemplos</b>
Contac	50 Carácter Alfabético	Ana María Moris
Code_Treatment	Caracter Numérico (Entero largo)	10
Code_lang	Caracter Numérico (Entero largo)	2000
Name_treatment	100 Carácter Alfabético	Oxitetraciclina
Cod_med	04 Carácter Alfanumérico	EMA
ATU	Caracter Numérico (Entero largo)	150
QDays	Caracter Numérico (Entero)	30
Rec_Dose	Caracter Numérico (Entero largo)	30
IActive	Caracter Numérico (Entero largo)	0.2
MaxCon	Caracter Numérico (Entero largo)	0.0063
Remark	Memo	Requiere prescripción
Nday	Caracter Numérico (entero largo)	7
AmountFeed	Caracter Numérico (entero doble)	200
Id	Caracter Numérico (Entero largo)	11000
Idiom	Caracter Numérico (Entero largo)	2000
Des	255 Carácter Alfabético	&Language

<b>Atributo</b>	<b>Característica del Atributo</b>	<b>Ejemplos</b>
IsParam	Caracter Numérico (Entero largo)	-1, 0
CodeSup	Caracter Numérico (Entero largo)	29581

### **9.1.5 Identificación de claves candidatas y elección de claves primarias para las entidades**

Se deben determinar todas las posibles claves candidatas y primarias para cada entidad.

Un enfoque para la elección de una clave primaria desde un conjunto de claves candidatas puede consistir en:

- ✓ La clave candidata con el mínimo conjunto de atributos
- ✓ La clave candidata que tiene menor probabilidad de cambiar
- ✓ La clave candidata que tiene menor probabilidad de perder la cualidad única en el futuro
- ✓ La clave candidata que tiene el menor número de caracteres (atributos textuales)

Se debe identificar si existe una entidad fuerte o débil. Si se puede asignar una clave es fuerte, si no, es débil. La entidad débil sólo se identifica por su dueña a través de una clave foránea.

A continuación se muestran las claves primarias y candidatas para el Módulo de Tratamientos.

**Tabla N° 5 Identificación de claves candidatas y elección de claves primarias para las entidades**

<b>Entidad</b>	<b>Claves Alternas</b>	<b>Claves Primarias</b>
Company		Code_company
Sites		Code_Company Id_Concession Code_Site
Treatments		N_Identity YearReg CorrReg
Dtreatments		N_Identity YearReg CorrReg Id_Cage
Dfeeding		N_Identity YearReg CorrReg Id_Cage NDay
Veterinary		N_Id
FeedSupplier		IdSup
Diets		IdSup IdDiet
FeedSize		IdSup IdSize

Entidad	Claves Alternas	Claves Primarias
FeedTypes		IdSup IdDiet IdSize IdPig IdTPig IdTreat1 IdTreat2
Treatment_Supplier		Code_Supplier
Ltreatment		Code_Treatment Code_Lang
TreatBySupplier		Code_Treatment Code_Supplier
Type_Treatment		Code_Treatment

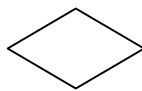
### 9.1.6 Modelo conceptual E-R

A continuación el Modelo Entidad Relación del modelo conceptual.

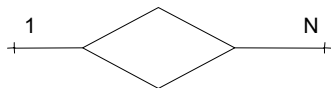
#### Nomenclatura:



= Entidad Fuerte



= Relación



= Relación uno a muchos

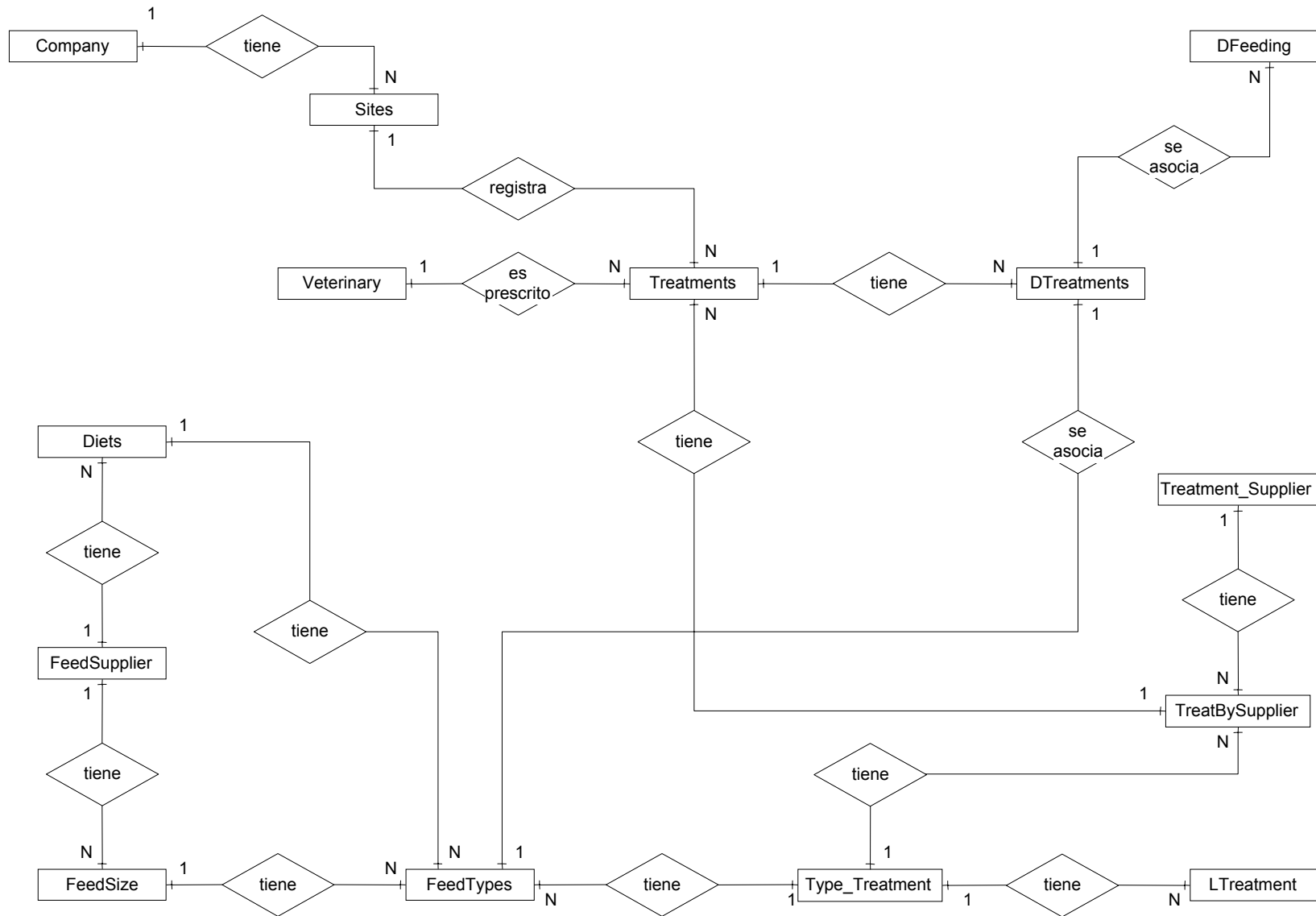


Figura N° 8 :Diagrama E-R del Modelo Conceptual del Módulo de Tratamientos

## 9.2 Diseño de Base de Datos Lógico para el Modelo Relacional.

Es el proceso de construcción de un modelo de información usado en una organización basado en las especificaciones del modelo de datos, pero independientemente de un DBMS en particular y otras consideraciones físicas. [Connolly1999].

La finalidad de esta etapa es transformar el modelo de datos conceptual al lógico, para ello se deben llevar a cabo las siguientes tareas:

- ✓ Eliminación de relaciones Muchos-a-Muchos
- ✓ Eliminación de relaciones complejas
- ✓ Eliminación de relaciones recursivas
- ✓ Eliminación de relaciones con atributos
- ✓ Eliminación de atributos multivalóricos
- ✓ Derivación de relaciones del modelo de datos lógico
- ✓ Validación del modelo utilizando normalización
- ✓ Validación del modelo contra las transacciones de usuario

Como paso final de esta etapa se confecciona el diagrama E-R final para el sistema.



## **9.2.1 Transformación del Modelo Conceptual al Modelo Lógico**

En este paso se refina el modelo conceptual y se transforma en el modelo lógico.

### **9.2.1.1 Eliminación de relaciones Muchos-a-Muchos**

Para un conjunto binario de relaciones  $R$  entre los conjuntos de entidades  $A$  y  $B$ , se dice que existe una relación Muchos-a-Muchos cuando una entidad en  $A$  está asociada con un número cualquiera de entidades en  $B$ , y una entidad en  $B$  está asociada a un número cualquiera de entidades en  $A$ . [Connolly1999].

En el diseño del presente sistema no existen relaciones Muchos-a-Muchos.

### **9.2.1.2 Eliminación de relaciones con atributos**

No existen relaciones con atributos

### **9.2.1.3 Eliminación de atributos multivalóricos**

No existen atributos multivalóricos

### 9.2.2 Derivación de relaciones del Modelo de Datos Lógico

En este paso se describen las relaciones utilizando el Lenguaje de Definición de Base de Datos (Database Definition Language, DDL).

Se define la relación entre entidades mediante **Clave Primaria / Clave Foránea**. A continuación se describen las entidades Padre e Hijo mediante el Lenguaje de Definición de Base de Datos:

a) **Company** (Code\_Company, Name\_Comapny, Phone, Fax)

**Primary Key** (Code\_Company)

b) **Sites** (Code\_Company, Id\_Concession, Code\_Site, Description, Type\_Site)

**Primary Key** (Code\_Company , Id\_Concession , Code\_Site)

**Foreing Key** (Code\_company) **references** Company (Code\_Company)

c) **Treatments** (N\_Identity, YearReg, CorrReg, Code\_Company, Id\_Concession, Code\_Site, Dates, Code\_TSite, Code\_TUnit, Code\_WTreatment, Code\_TDiag, Code\_Treatment, Code\_Supplier, N\_Days\_Treatment, CorrPC, FCR, PretDose, Concent, ProdCom, ValNeto, ValNetoT, ValMG, CostKg, CostDay, PerIngAct, PerProdCom, KgIngAct, KgProdCom)

**Primary Key** (N\_Identity , YearReg , CorrReg)

**Foreing Key** (Code\_Company , Id\_Concession , Code\_Site) **references** Sites (Code\_Company , Id\_Concession , Code\_Site)

**Foreing Key** (N\_Identity) **references** Veterinary (N\_Id)

d) **Veterinary** (N\_Id, Name\_Veterinary, Position, N\_VetCollege, E\_mail, Passwd)

**Primary Key** (N\_Id)

e) **DTreatments** (N\_Identity, YearReg, CorrReg, Code\_Company, Id\_Concession, Code\_Site, Code\_Group, Correlative, Id\_Cage, Dates\_Reg, Code\_Specie, Name\_Group, NFishIni, AWIni, BioIni, RealPC, PCCalc, PerGrowth, PerMort, NFishEnd, AWEnd, BioEnd, FullCode, FeedxDay, FeedTot, Start\_Date, End\_Date, Date\_Quarantine, Dose)

**Primary Key** (N\_Identity , YearReg , CorrReg , Id\_Cage)

**Foreing Key** (N\_Identity , YearReg , CorrReg) **references** Treatments (N\_Identity , YearReg , CorrReg)

f) **DFeeding** (N\_Identity, YearReg, CorrReg, Id\_Cage, Nday, AmountFeed)

**Primary Key** (N\_Identity , YearReg , CorrReg , Id\_Cage , Nday)

**Foreing Key** (N\_Identity ,YearReg , CorrReg, Id\_Cage) **references** Treatments (N\_Identity ,YearReg , CorrReg, Id\_Cage)

g) **FeedSupplier** (Id\_Sup, NameSup, Activ)

**Primary Key** (Id\_Sup)

h) **Diets** (IdSup, IdDiet, NameDiet, Activ)

**Primary Key** (IdSup , IdDiet)

**Foreing Key** (IdSup) **references** FeedSupplier (IdSup)

i) **FeedSize** (IdSup, IdSize, DesZise)

**Primary Key** (IdSup , IdSize)

**Foreing Key** (IdSup) **references** FeedSupplier (IdSup)

j) **FeedTypes** (IdSup, IdDiet, IdSize, IdPig, IdTPig, IdTreat1, IdTreat2,  
FullCode)

**Primary Key** (IdSup , IdDiet , IdSize , IdPig , IdTPig , IdTreat1 , IdTreat2)

**Foreing Key** (IdSize , IdSup) **references** FeedSize (IdSize , IdSup)

**Foreing Key** (IdDiet , IdSup) **references** Diets (IdDiet, IdSup)

**Foreing Key** (IdTreat1) **references** Type\_Treatment ( Code\_Treatment)

k) **Treatment\_Supplier** (Code\_Supplier, Name\_Supplier, Phone, Fax,  
Contact)

**Primary Key** (Code\_Supplier)

l) **TreatBySupplier** (Code\_Treatment, Code\_Supplier)

**Primary Key** (Code\_Treatment , Code\_Supplier)

**Foreing Key** (Code\_Treatment) **references** Type\_Treatment  
(Code\_Treatment)

**Foreing Key** (Code\_Supplier) **references** Treatment\_Supplier  
(Code\_Supplier)

m) **Type\_Treatment** (Code\_Treatment, Code\_Med, ATU, QDays,  
Rec\_Dose, IActive, MaxCon, Remark)

**Primary Key** (Code\_Treatment)

n) **Ltreatment** (Code\_Treatment, Code\_Lang, Name\_Treatment)

**Primary Key** (Code\_Treatment , Code\_Lang)

**Foreing Key** (Code\_Treatment) **references** Type\_Treatment  
(Code\_Treatment)

### 9.2.3 Validación del modelo utilizando Normalización

Un buen diseño para generar una base de datos relacional debe considerar el objetivo de almacenar información sin redundancia innecesaria, pero que a la vez permita recuperar información fácilmente. Una técnica consiste en diseñar esquemas que tengan una forma normal adecuada.

Los problemas que pueden surgir a partir de un mal diseño de base de datos son:

- ✓ Repetición de información
- ✓ Incapacidad para representar cierta información
- ✓ Pérdida de información

“Normalización es el proceso de decidir cuales atributos deben estar juntos en un tipo de entidad, asegurando que el modelo resultante es consistente, con redundancia mínima y máxima estabilidad.” [Connolly1999].

A continuación se presenta el proceso de normalización para el Módulo de Tratamientos.



### **9.2.3.1 Primera Forma Normal (1FN)**

“Una tabla se dice en primera forma normal (1FN) sí solo sí todas las columnas tienen valores atómicos; esto es, no existen grupos repitentes (columnas) dentro de la fila.” [Connolly1999].

En las entidades descritas dentro del presente sistema no se encuentran atributos multivalorados, por lo que es posible asegurar que el sistema se encuentra en primera forma normal , ya que todos los valores están representados únicamente en sus filas.

### **9.2.3.2 Segunda Forma Normal (2FN)**

Para que una relación se encuentre en segunda forma normal, ésta debe poseer las siguientes propiedades:

- ✓ Debe estar en primera forma normal (1FN)
- ✓ Debe tener una clave compuesta.

“Una tabla se dice en segunda forma normal (2FN) si solo si está en 1FN y cada atributo no-clave es completamente dependiente de la clave primaria.” [Connolly1999].

a) **Company** (Code\_Company, Name\_Company, Phone, Fax)

**Primary Key** (Code\_Company)

Entonces: Name\_Company, Phone, Fax. Todos estos atributos son completamente dependientes de la clave primaria (Code\_Company) y no de una parte de esta.

b) **Sites** (Code\_Company, Id\_Concession, Code\_Site, Description, Type\_Site)

**Primary Key** (Code\_Company , Id\_Concession , Code\_Site)

Entonces: Description, Type\_Site. Todos estos atributos son completamente dependientes de la clave primaria (Code\_Company , Id\_Concession , Code\_Site) y no de una parte de esta.

c) **Treatments** (N\_Identity, YearReg, CorrReg, Code\_Company, Id\_Concession, Code\_Site, Dates, Code\_TSite, Code\_TUnit, Code\_WTreatment, Code\_TDiag, Code\_Treatment, Code\_Supplier, N\_Days\_Treatment, CorrPC, FCR, PretDose, Concent, ProdCom,

ValNeto, ValNetoT, ValMG, CostKg, CostDay, PerIngAct, PerProdCom, KgIngAct, KgProdCom)

**Primary Key** (N\_Identity , YearReg , CorrReg)

Entonces: Code\_Company, Id\_Concession, Code\_Site, Dates, Code\_TSite, Code\_TUnit, Code\_WTreatment, Code\_TDiag, Code\_Treatment, Code\_Supplier, N\_Days\_Treatment, CorrPC, FCR, PretDose, Concent, ProdCom, ValNeto, ValNetoT, ValMG, CostKg, CostDay, PerIngAct, PerProdCom, KgIngAct, KgProdCom. Todos estos atributos son completamente dependientes de la clave primaria (N\_Identity , YearReg , CorrReg) y no de una parte de esta.

d) **Veterinary** (N\_Id, Name\_Veterinary, Position, N\_VetCollege, E\_mail, Passwd)

**Primary Key** (N\_Id)

Entonces : Name\_Veterinary, Position, N\_VetCollege, E\_mail, Passwd. Todos estos atributos son completamente dependientes de la clave primaria (N\_Id) y no de una parte de esta.

e) **DTreatments** (N\_Identity, YearReg, CorrReg, Code\_Company, Id\_Concession, Code\_Site, Code\_Group, Correlative, Id\_Cage, Dates\_Reg, Code\_Specie, Name\_Group, NFishIni, AWIni, Biolni, RealPC, PCCalc, PerGrowth, PerMort, NFishEnd, AWEnd, BioEnd,

FullCode, FeedxDay, FeedTot, Start\_Date, End\_Date, Date\_Quarantine, Dose)

**Primary Key** (N\_Identity , YearReg , CorrReg , Id\_Cage)

Entonces: Code\_Company, Id\_Concession, Code\_Site, Code\_Group, Correlative, Id\_Cage, Dates\_Reg, Code\_Specie, Name\_Group, NFishIni, AWIni, BioIni, RealPC, PCCalc, PerGrowth, PerMort, NFishEnd, AWEnd, BioEnd, FullCode, FeedxDay, FeedTot, Start\_Date, End\_Date, Date\_Quarantine, Dose. Todos estos atributos son completamente dependientes de la clave primaria (N\_Identity , YearReg , CorrReg , Id\_Cage) y no de una parte de esta.

f) **DFeeding** (N\_Identity, YearReg, CorrReg, Id\_Cage, Nday, AmountFeed)

**Primary Key** (N\_Identity , YearReg , CorrReg , Id\_Cage , Nday)

Entonces: Id\_Cage, Nday, AmountFeed. Todos estos atributos son completamente dependientes de la clave primaria (N\_Identity , YearReg , CorrReg , Id\_Cage , Nday) y no de una parte de esta.

g) **FeedSupplier** (Id\_Sup, NameSup, Activ)

**Primary Key** (Id\_Sup)

Entonces: NameSup, Activ. Todos estos atributos son completamente dependientes de la clave primaria (Id\_Sup) y no de una parte de esta.

h) **Diets** (IdSup, IdDiet, NameDiet, Activ)

**Primary Key** (IdSup , IdDiet)

Entonces: NameDiet, Activ. Todos estos atributos son completamente dependientes de la clave primaria (IdSup , IdDiet) y no de una parte de esta.

i) **FeedSize** (IdSup, IdSize, DesZise)

**Primary Key** (IdSup , IdSize)

Entonces : DesZise. Este atributo es completamente dependiente de la clave primaria (IdSup , IdDiet) y no de una parte de esta.

j) **FeedTypes** (IdSup, IdDiet, IdSize, IdPig, IdTPig, IdTreat1, IdTreat2, FullCode)

**Primary Key** (IdSup , IdDiet , IdSize , IdPig , IdTPig , IdTreat1 , IdTreat2)

Entonces: FullCode. Este atributo es completamente dependiente de la clave primaria (IdSup , IdDiet , IdSize , IdPig , IdTPig , IdTreat1 , IdTreat2) y no de una parte de esta.

k) **Treatment\_Supplier** (Code\_Supplier, Name\_Supplier, Phone, Fax, Contact)

**Primary Key** (Code\_Supplier)

Entonces: Name\_Supplier, Phone, Fax, Contact. Estos atributos son completamente dependientes de la clave primaria (Code\_Supplier) y no de una parte de esta.

l) **TreatBySupplier** (Code\_Treatment, Code\_Supplier)

**Primary Key** (Code\_Treatment , Code\_Supplier)

Entonces: Esta tabla se encuentra en 2FN porque solo consta de dos atributos los cuales constituyen la clave primaria.

m) **Type\_Treatment** (Code\_Treatment, Code\_Med, ATU, QDays, Rec\_Dose, IActive, MaxCon, Remark)

**Primary Key** (Code\_Treatment)

Entonces: Code\_Med, ATU, QDays, Rec\_Dose, IActive, MaxCon, Remark. Todos estos atributos son completamente dependientes de la clave primaria (Code\_Treatment) y no de una parte de esta.

n) **Ltreatment** (Code\_Treatment, Code\_Lang, Name\_Treatment)

**Primary Key** (Code\_Treatment , Code\_Lang)

Entonces: Name\_Treatment. Este atributo es completamente dependiente de la clave primaria (Code\_Treatment , Code\_Lang) y no de una parte de esta.

### 9.2.3.3 Tercera Forma Normal (3FN)

“Una tabla está en tercera forma normal (3FN) sí solo sí está en 2FN y todos los atributos no clave dependen de manera no transitiva de la clave primaria. (Ningún atributo sin clave puede depender aún en forma indirecta en otro atributo sin clave).” [Connolly1999].

Se debe analizar la dependencia de los atributos, eliminando las dependencias transitivas.

#### a) **Company**

Code\_company -----> Name\_Company

Code\_Company -----> Phone

Code\_Company -----> Fax

#### b) **Sites**

Code\_company, Id\_Concession, Code\_Site -----> Description

Code\_company, Id\_Concession, Code\_Site -----> TypeSite

**c) Treatments**

N\_Identity, YearReg, CorrReg -----> Code\_Company  
N\_Identity, YearReg, CorrReg -----> Id\_Concession  
N\_Identity, YearReg, CorrReg -----> Code\_Site  
N\_Identity, YearReg, CorrReg -----> Dates  
N\_Identity, YearReg, CorrReg -----> Code\_TSite  
N\_Identity, YearReg, CorrReg -----> Code\_TUnit  
N\_Identity, YearReg, CorrReg -----> Code\_WTreat  
N\_Identity, YearReg, CorrReg ----->Code\_TDiag  
N\_Identity, YearReg, CorrReg -----> Code\_Treatment  
N\_Identity, YearReg, CorrReg -----> Code\_Supplier  
N\_Identity, YearReg, CorrReg -----> N\_Days\_Treatment  
N\_Identity, YearReg, CorrReg -----> CorrPC  
N\_Identity, YearReg, CorrReg -----> FCR  
N\_Identity, YearReg, CorrReg -----> PretDose  
N\_Identity, YearReg, CorrReg -----> Concent  
N\_Identity, YearReg, CorrReg -----> ProdCom  
N\_Identity, YearReg, CorrReg -----> ValNeto  
N\_Identity, YearReg, CorrReg -----> ValNetoT  
N\_Identity, YearReg, CorrReg -----> ValMG  
N\_Identity, YearReg, CorrReg -----> CostKg  
N\_Identity, YearReg, CorrReg -----> CostDay



N\_Identity, YearReg, CorrReg ----->PerIngAct  
 N\_Identity, YearReg, CorrReg -----> PerProdCom  
 N\_Identity, YearReg, CorrReg -----> KgIngAct  
 N\_Identity, YearReg, CorrReg -----> KgProdCom

**d) Dtreatments**

N\_Identity, YearReg, CorrReg, Id\_Cage -----> Code\_Company  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Id\_Concession  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Code\_Site  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Code\_Group  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Correlative  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Dates\_Reg  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Code\_Specie  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Name\_Group  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> NFishIni  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> AWIni  
 NFishIni, AWIni -----> Biolni  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> RealPC  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> PCCalc  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> PerGrowth  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> PerMort  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> NFishEnd

N\_Identity, YearReg, CorrReg, Id\_Cage -----> AWEnd  
 NFishEnd, AWEnd -----> BioEnd  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> FullCode  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> FeedxDay  
 FeedxDay, Start\_Date, End\_Date -----> FeedTot  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Start\_Date  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> End\_Date  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Date\_Quarantine  
 N\_Identity, YearReg, CorrReg, Id\_Cage -----> Dose

DTreatment no se encuentra en 3FN porque el atributo BioIni depende funcionalmente de los atributos NFishIni, AWIni. El atributo BioEnd depende funcionalmente de los atributos NFishEnd, AWEnd. El atributo FeedTot depende funcionalmente de los atributos FeedxDay, Start\_Date, End\_Date. Esta tabla se mantendrá de-normalizada debido a que a pesar de que los campos antes mencionados dependen de otros campos no claves, para lograr obtener estos valores se necesita una serie de cálculos y valores que se determinan a partir de otros valores y porcentajes, por lo que es necesario mantenerlos creados dentro de esta entidad por su complejidad para ser obtenidos.

**e) Veterinary**

N\_Id -----> Name\_Veterinary

N\_Id -----> Position

N\_Id -----> N\_VetCollege

N\_Id -----> E\_mail

N\_Id -----> Passwd

**f) Dfeeding**

N\_Identity, YearReg, CorrReg, Id\_Cage, NDay -----> AmountFeed

**g) FeedSupplier**

IdSup -----> NameSup

IdSup -----> Activ

**h) Diets**

IdSup, IdDiet -----> NameDiet

IdSup, IdDiet -----> Activ

**i) FeedSize**

IdSup, IdSize -----> DesSize

**j) FeedTypes**

IdSup, IdDiet, IdSize, IdPig, IdTPig, IdTreat1, IdTreat2 -----> Fullcode

**k) TreatBySupplier**

Solo clave primaria

**l) Treatment\_Supplier**

Code\_Supplier -----> Name\_Supplier

Code\_Supplier -----> Phone

Code\_Supplier -----> Fax

Code\_Supplier -----> Contact

**m) Type\_Treatment**

Code\_Treatment -----> Cod\_med

Code\_Treatment -----> ATU

Code\_Treatment -----> QDays

Code\_Treatment -----> Rec\_Dose

Code\_Treatment -----> IActive

Code\_Treatment -----> MaxCon

Code\_Treatment -----> Remark

**n) Ltreatments**

Code\_Treatment, Code\_Lang -----> Name\_Treatment













#### **9.2.4 Validación del modelo contra las transacciones de usuario**

En esta etapa se debe asegurar que el modelo de datos lógico local apoye las transacciones que son requeridas por los usuarios.

Utilizando el diagrama ER, las claves primarias y foráneas se puede verificar cada transacción de forma manual. De acuerdo a las transacciones que se presenten (las transacciones corresponden a los requerimientos presentados por los usuarios), se crea un mapa transaccional. Este mapa mostrará la ruta a seguir dentro del modelo lógico que permita suplir cada requerimiento presentado por los usuarios. Si las transacciones no se pudieran ser efectuadas se debe modificar el modelo de datos.

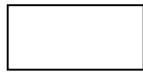
A continuación se muestran los mapas transaccionales que satisfacen los requerimientos de los usuarios, los cuales fueron identificados en el capítulo N°8 sección 8.3 del presente documento.

**Tabla N° 6: lista de colores para las trazas que satisfacen los requerimientos de usuario del Módulo Tratamiento.**

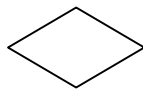
Color	Transacción
	T(1)
	T(2), T(3)
	T(4), T(5), T(6)
	T(7), T(8)
	T(9)
	T(10)
	T(11)
	T(12)
	T(13)
	T(14)
	T(15)
	T(16)

A continuación se muestran los mapas transaccionales pertenecientes al sistema.

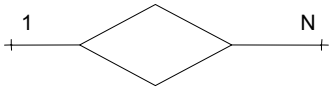
**Nomenclatura:**



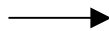
= Entidad Fuerte



= Relación



= Relación uno a muchos



= Flujo de Datos

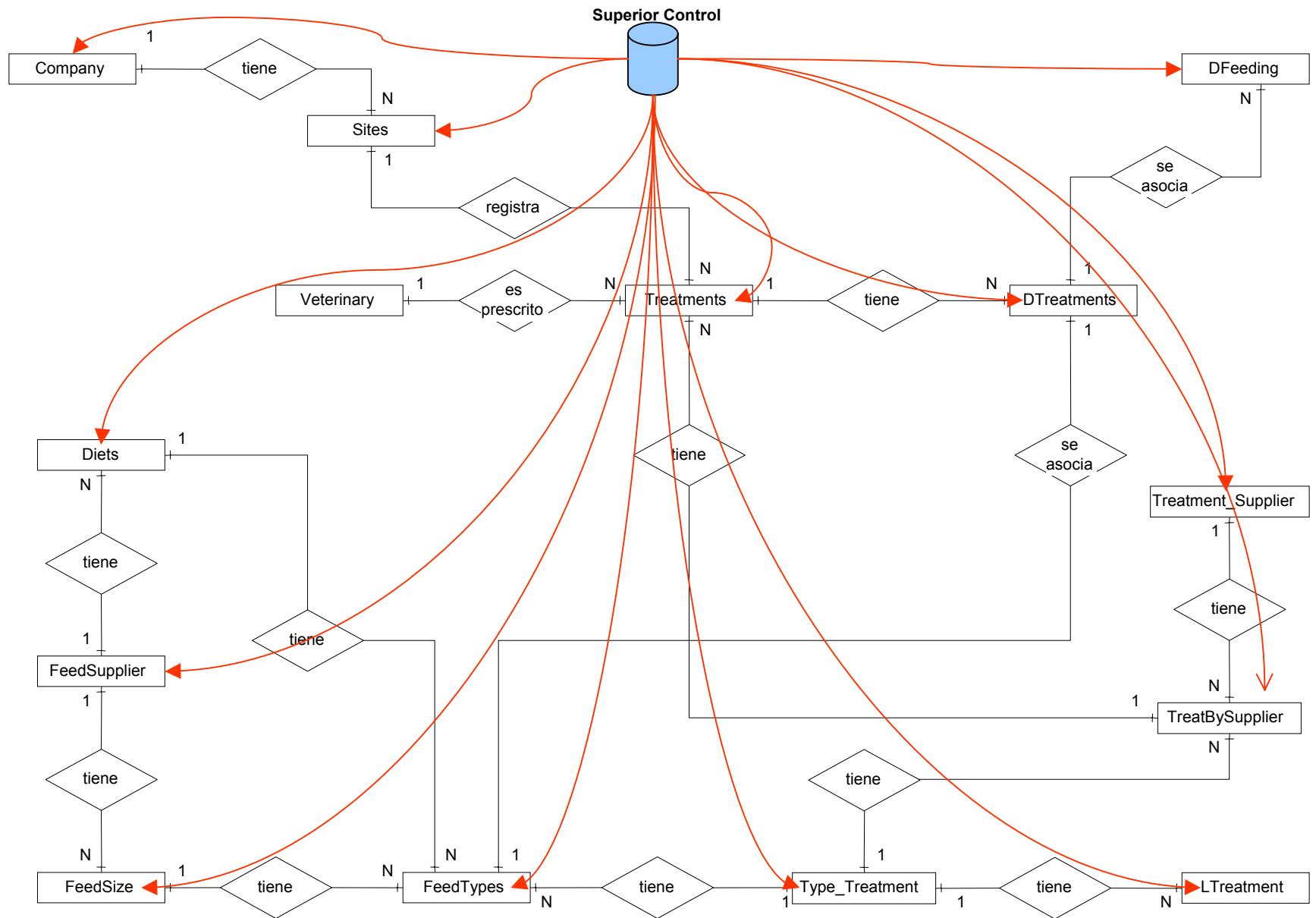


Figura N° 9 : Mapa transaccional que satisface el requerimiento T (1)



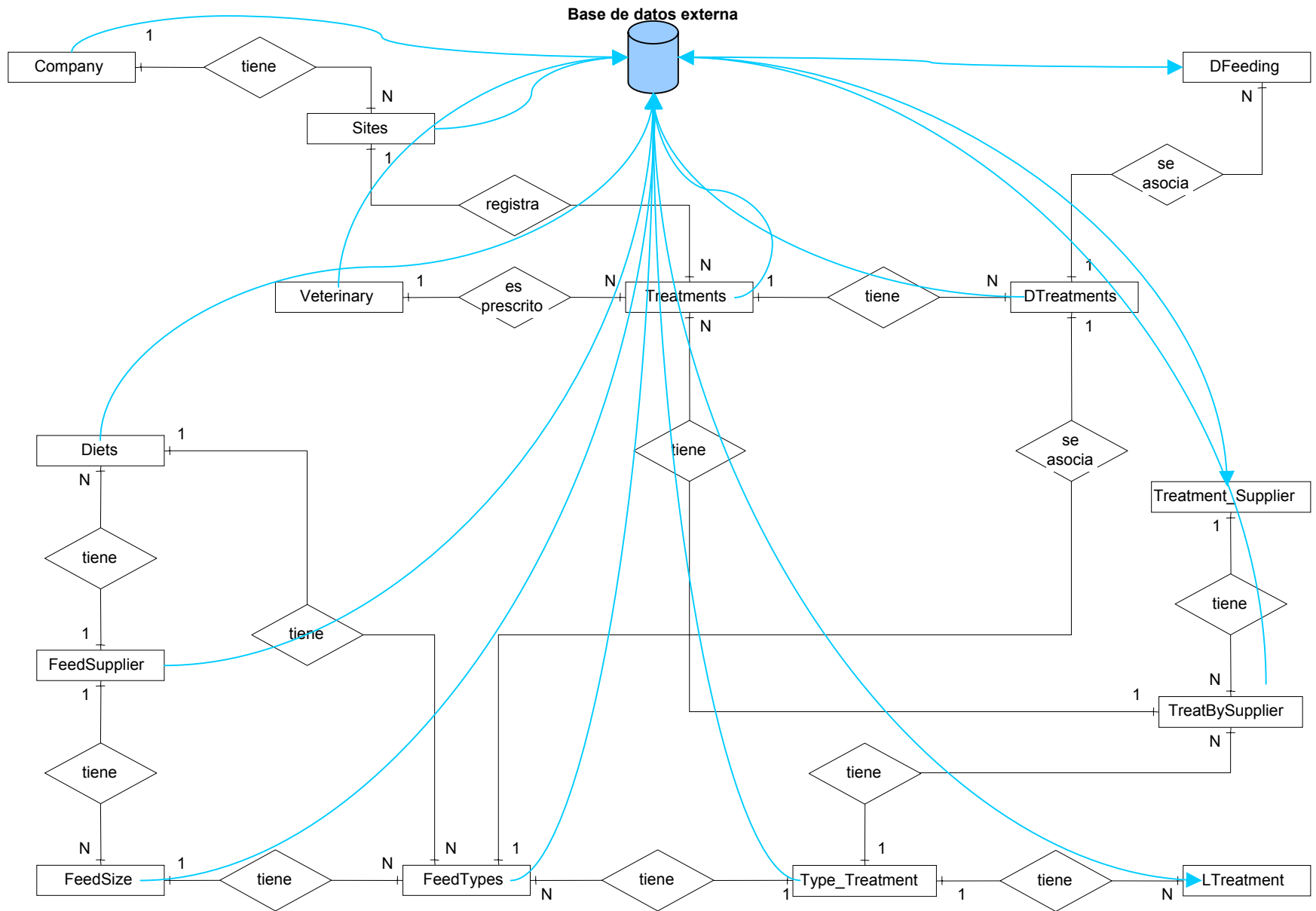


Figura N° 10 : Mapa transaccional que satisface los requerimientos T(2), T(3)

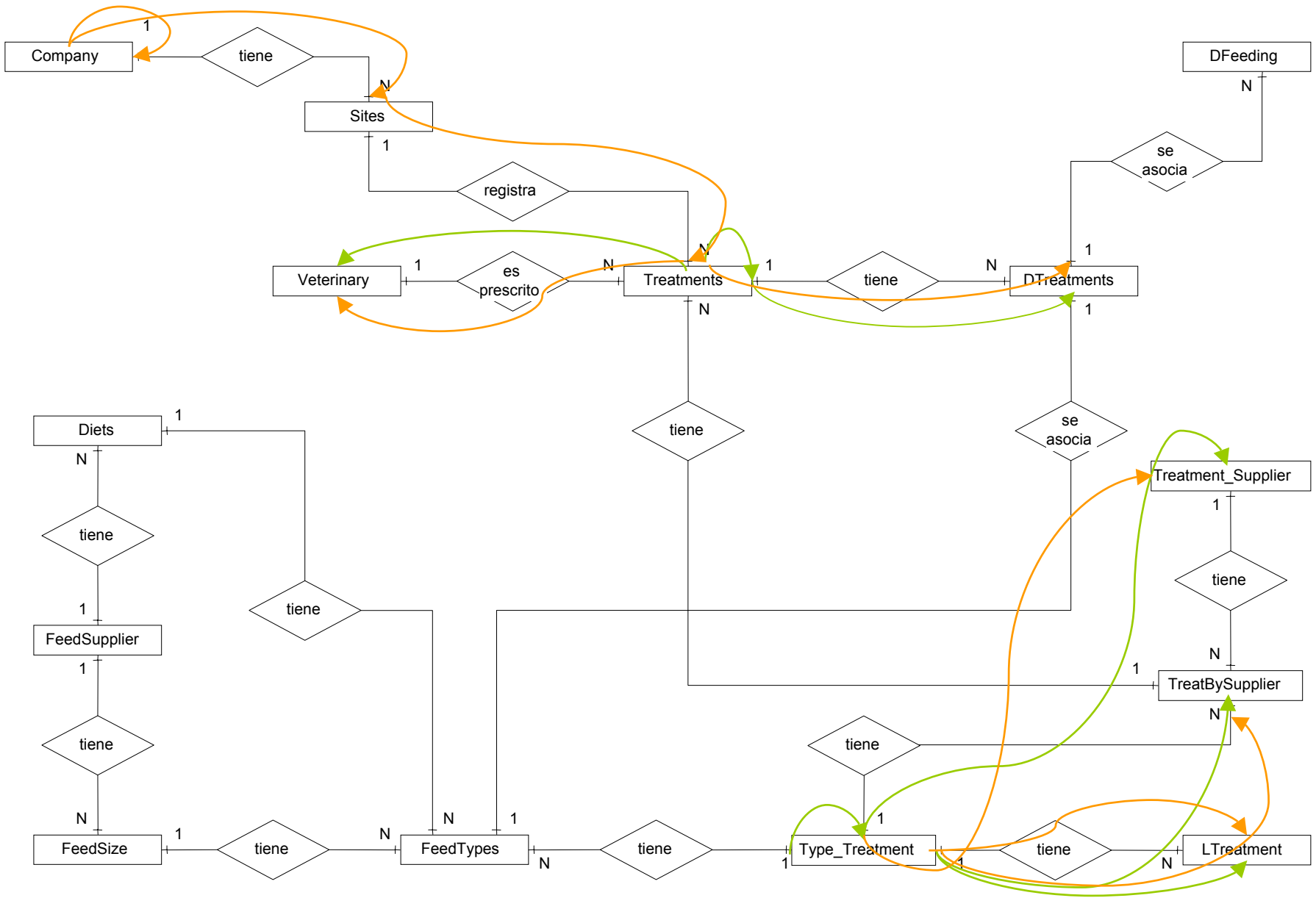


Figura N° 11 : Mapa transaccional que satisface los requerimientos T(4), T(5), T(6),T(11)



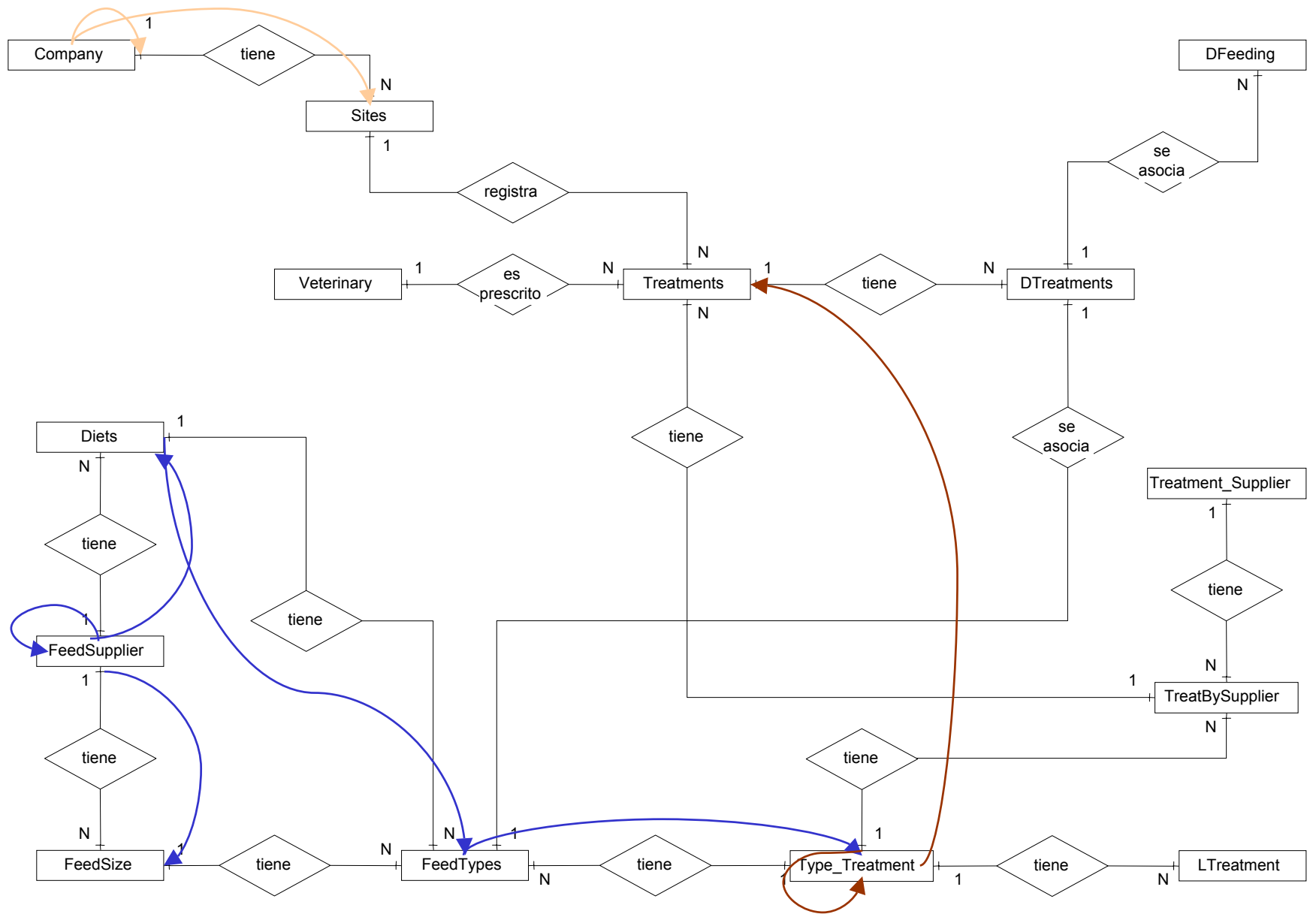
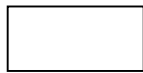


Figura N° 13 : Mapa transaccional que satisface los requerimientos T(10), T(13), T(16)

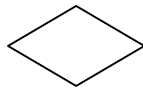
### 9.2.5 Diagrama ER del Modelo Lógico

A continuación se muestra el diagrama E-R correspondiente al Diseño Lógico del Módulo de Tratamiento.

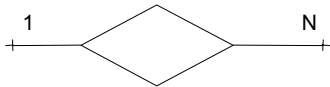
#### Nomenclatura:



= Entidad Fuerte



= Relación



= Relación uno a muchos

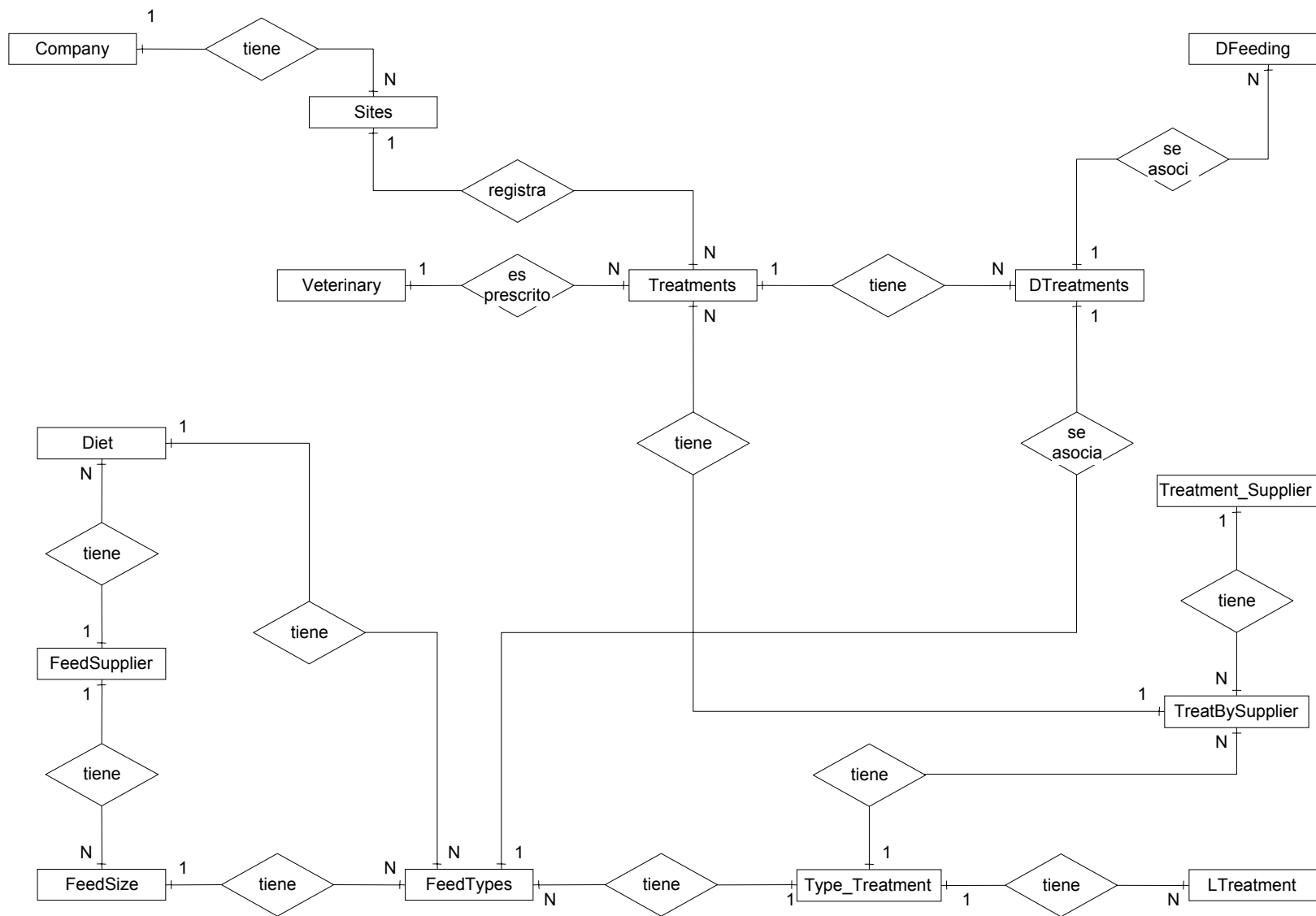


Figura N°14 :Diagrama E-R del Modelo Lógico del Módulo de Tratamientos

## **9.2.6 Definición de restricciones de integridad**

En este paso se definen las restricciones de integridad de la organización. Estas restricciones de integridad se utilizan para proteger la base de datos contra inconsistencias. No se analiza si el DMBS permite o no integridad referencial, solo debe asegurarse que el diseño lo incorpore.

Existen cinco tipos de restricciones de integridad:

### **9.2.6.1 Datos requeridos**

Algunos de los atributos siempre deben tener un valor por defecto válido, esto significa que no pueden tener valores nulos (NULL). En la tabla N° 3 sección 9.1.3 del presente capítulo se encuentra documentadas estas restricciones.

### **9.2.6.2 Restricciones de dominios de atributos**

Cada atributo que se encuentra presente en las entidades mantiene un dominio, que son un conjunto de valores legales que se visualiza como un conjunto de valores válidos para el atributo.

La tabla N° 4 sección 9.1.4 del presente informe muestra los dominios para los atributos presente en el sistema

### **9.2.6.3 Integridad de entidades**

Esta etapa establece que la clave primaria de cada entidad no puede ser nula. La Tabla N° 5 de la sección 9.1.5 del presente capítulo considera esta restricción.

### **9.2.6.4 Integridad referencial**

Integridad referencial significa que si una clave foránea tiene un valor, ésta debe referirse a una instancia específica de la entidad padre. Esto permite asegurar que un valor que aparece en una relación para un conjunto de atributos dado, también aparece para un cierto conjunto de atributos en otra relación.

Se debe considerar que el diseño asegure la integridad referencial. Para asegurar esto se utilizan restricciones de existencia, que especifican bajo que condiciones se puede insertar, actualizar o eliminar claves foráneas. Se debe analizar cada caso y ver si es afectado o no.



La integridad referencial para el Módulo de Tratamientos implica las opciones:

**Tabla N° 7 Integridad referencial del Módulo Tratamientos**

<b>Entidad</b>	<b>Claves Foráneas</b>	<b>On delete</b>	<b>On update</b>
Sites	Code_Company	Cascade	
Treatments	Code_Company N_Identity Id_Concession Code_Site Code_Treatment Code_Supplier	Cascade No action Cascade Cascade Cascade Cascade	
Dtreatment	N_Identity YearReg CorrReg	Cascade Cascade Cascade	
Dfeeding	N_Identity YearReg CorrReg Id_Cage	Cascade Cascade Cascade Cascade	

Entidad	Claves Foráneas	On delete	On update
Diets	IdSup	No action	
FeedSize	IdSup	No action	
FeedTypes	IdSup IdSize IdDiet IdTreat	No action No action No action Cascade	
TreatBySupplier	Code_Supplier Code_Treatment	Cascade Cascade	
Ltreatment	Code_Treatment	Cascade	

Las restricciones utilizadas son:

**NO ACTION:** Prevenir que se elimine de la relación padre si existe un hijo.

**CASCADE:** Cuando se elimina el padre, automáticamente se elimina cualquiera referencia del hijo. En el caso que el hijo también actúe como padre para otra relación, se utiliza la misma regla y así eliminando en cascada. Eliminaciones del padre se transfieren a la relación hijo.

#### **9.2.6.5 Restricciones de la empresa**

Esta se pueden referir como reglas del negocio. Actualizaciones pueden ser restringidas basadas en reglas que gobiernan las transacciones de la empresa.

No existen restricciones de la empresa en este sistema.

### **9.3 Diseño de Base de Datos Físico para el Modelo Relacional**

La Metodología de diseño de Base de Datos se divide en tres etapas: conceptual, lógico y físico.

La fase previa al diseño físico, denominada diseño lógico de base de datos es en gran parte independiente de los detalles de implementación, pero dependiente del modelo de datos de datos específico. La salida de este proceso es un modelo de datos lógico global y documentación que describe este modelo semejante al diccionario de datos del esquema relacional. Todos estos representan los orígenes de información para el proceso de diseño físico.

La etapa del diseño lógico se preocupa del “Qué”, el diseño físico de base de datos se preocupa del “Cómo”.

El diseño físico es el proceso de producción y descripción de la implementación de la base de datos en el almacenamiento secundario, describe las estructuras de almacenamiento y métodos de acceso utilizados para lograr un acceso eficiente a los datos.

### **9.3.1 Transformación del modelo de datos lógico global para un DBMS específico**

El objetivo de esta tarea es producir un esquema relacional del modelo de datos lógico global.

Los pasos a seguir son:

- ✓ Coordinar la información obtenida durante el modelamiento de datos lógico y aquella documentada en el diccionario de datos.
  
- ✓ Se utiliza esta información para diseñar las relaciones base. Es necesario conocer la funcionalidad del sistema administrador de base de datos elegido.

### 9.3.1.1 Diseño de relaciones bases para un DBMS específico

Es necesario decidir cómo representar las relaciones base que se han identificado en el modelo de datos lógico para el sistema administrador de base de datos específico

Para cada relación se tiene:

- ✓ El nombre de la relación
- ✓ Una lista de atributos simples
- ✓ Clave primaria donde sean apropiadas claves alternas y claves foráneas

El diccionario de datos para cada atributo tiene:

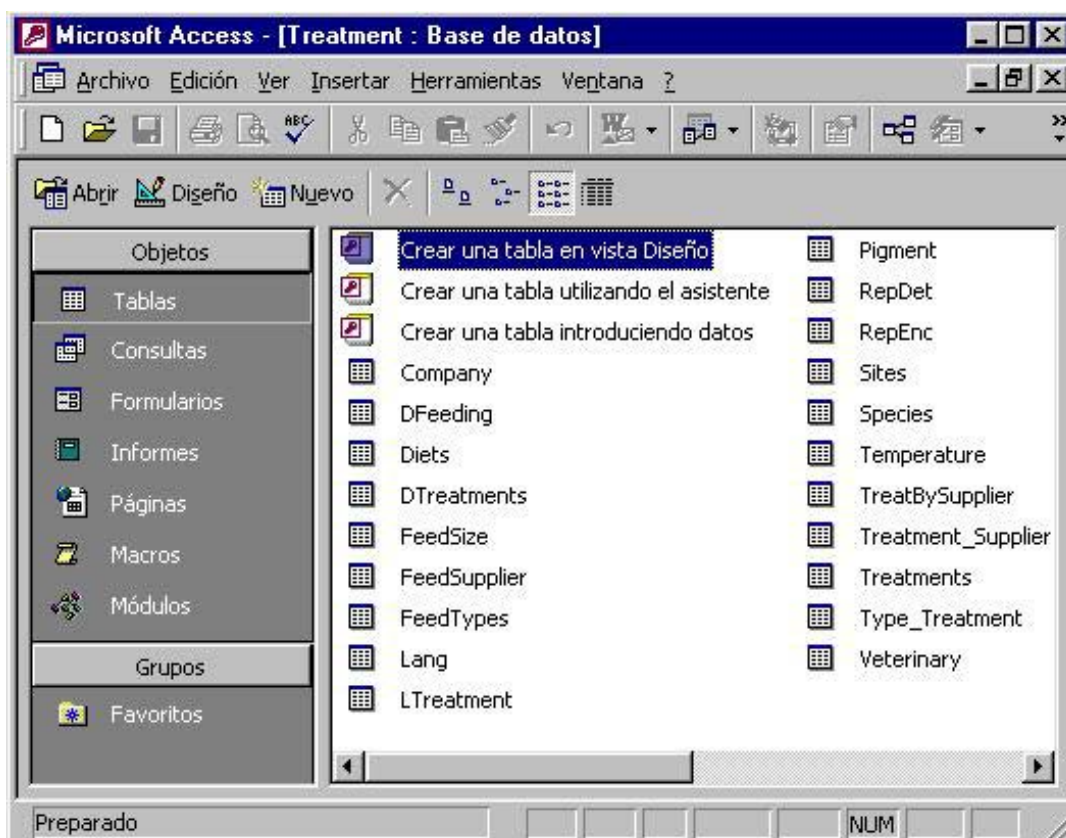
- ✓ Su dominio, tipo de datos, largo y restricciones del dominio
- ✓ Valor por defecto, opcional
- ✓ Si es posible que almacene Nulls
- ✓ Si es derivado, cómo se calcula

Relaciones base, traducción del modelo E-R a modelo relacional:

Para definir las relaciones se utiliza el lenguaje de definición de base de datos. En este punto es necesario explicar que Microsoft Access 2000 no dispone de un lenguaje de definición que pueda ser utilizado por el desarrollador, todo el diseño de las tablas se realiza mediante opciones gráficas, es decir para crear una tabla y definir sus campos se puede seleccionar una de las siguientes opciones (ver Figura N° 15)

- ✓ Crear una tabla en vista de diseño
- ✓ Crear una tabla utilizando el asistente
- ✓ Crear una tabla introduciendo datos

Luego se crean los campos, las restricciones de integridad, índices primarios y secundarios (ver Figura N° 16), claves, etc. Lo que debe estar bien definido es el diseño de las tablas y campos que se van a utilizar, las relaciones entre entidades, claves primarias y secundarias, para que al momento de empezar a crear las tablas y relaciones el modelo no presente inconsistencias o no represente los requerimientos de los usuarios.



**Figura N° 15 : Creación de tablas en Microsoft Access 2000**



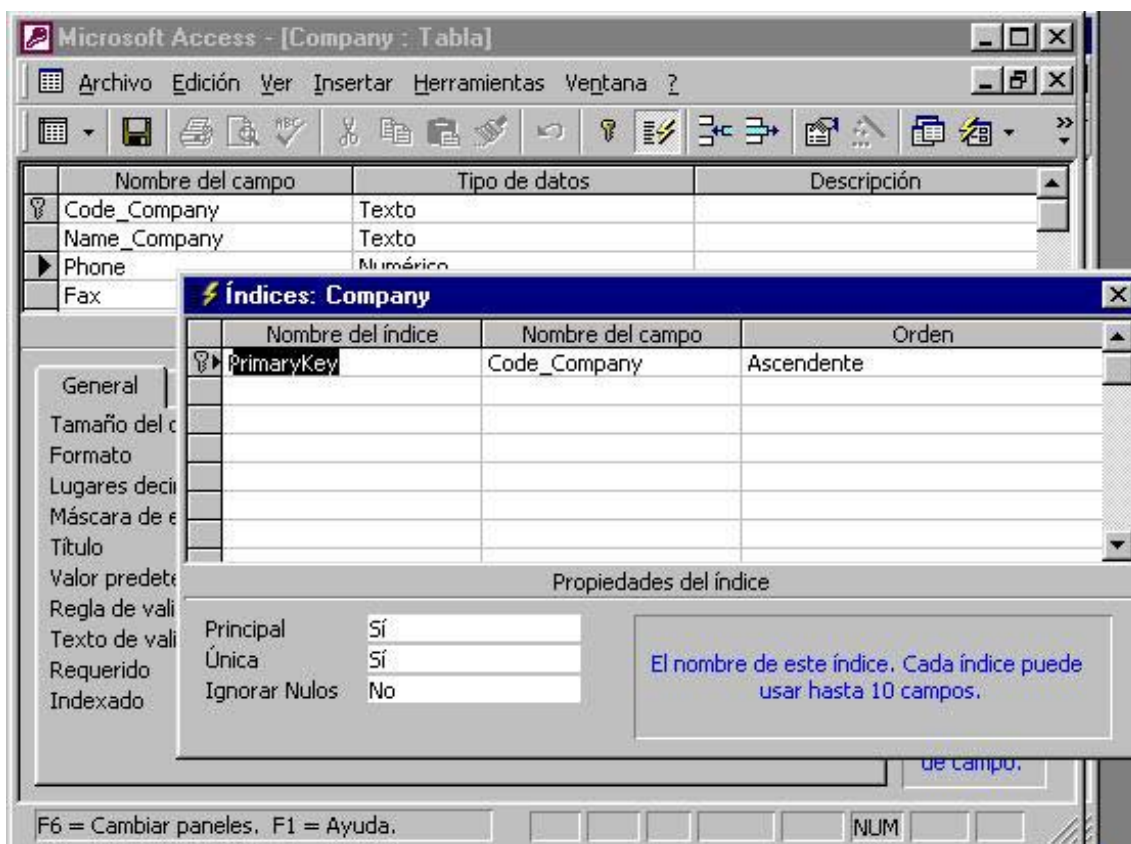


Figura N° 16: creación de claves primarias e índices en Microsoft Access

2000

### **9.3.1.2 Diseño de restricciones de la empresa para un DBMS específico**

La forma de controlar la consistencia y creación de restricciones de acceso a la base de datos es a través de trigger (gatillador), esto es una relación asociada con un evento que causa cambios en el contenido de una relación.

Existen tres eventos que pueden activar un trigger, estos son:

- ✓ Insertar
- ✓ Modificar
- ✓ Eliminar

La consistencia de la base de datos es controlada por las restricciones que se encuentran en la tabla N° 7 Integridad Referencial del Módulo de Tratamientos.

La integridad referencial se debe definir y controlar al momento de crear las relaciones entre las entidades.

Otro punto importante es la creación de índices, que son un mecanismo de acceso que hace más rápida la búsqueda de datos en los registros de las tablas. Microsoft Access 2000 permite la creación de índices seleccionando dentro de la tabla el campo que se desea sirva de índice y seleccionando la opción que lo identifica como índice primario (clave) o secundario.

### **9.3.2 Diseño de representación física**

El objetivo de este punto es determinar la organización de los archivos y métodos de acceso que se utilizarán para almacenar las relaciones base en forma eficiente; el camino en el cual las relaciones y tuplas serán mantenidas en el almacenamiento secundario.

#### **9.3.2.1 Análisis de transacciones**

El objetivo de este punto es determinar la funcionalidad de las transacciones que se manejan en la base de datos y analizar las transacciones importantes.

Para cada transacción se debe determinar la frecuencia esperada de cada transacción y los atributos y relaciones asociados. La tabla que se presenta a continuación muestra el análisis de transacciones, a través de los promedios de acceso, periodicidad y número de ejecuciones por período. Cabe destacar que estos cálculos son estimativos.

**Tabla N° 8 : Frecuencia de acceso a las tablas del Módulo de Tratamientos**

<b>Transacción</b>	<b>Promedio de acceso</b>	<b>Periodicidad</b>	<b>Número de ejecuciones por período</b>	<b>Atributos</b>
T(1)	Medio	Semanal	1	Todos
T(2)	Medio	Semanal	1	Todos
T(3)	Medio	semanal	1	Todos
T(4)	Medio	Semanal	1	Todos
T(5)	Medio	Semanal	1	Todos
T(6)	Alto	Diario	N	Todos
T(7)	Medio	Semanal	1	Todos
T(8)	Medio	Semanal	1	Todos
T(9)	Bajo	Mensual	1	N_Id

<b>Transacción</b>	<b>Promedio de acceso</b>	<b>Periodicidad</b>	<b>Número de ejecuciones por período</b>	<b>Atributos</b>
				Name_veterinary Position N_VetCollege E_mail Passwd
T(10)	Medio	Semanal	1	Todos
T(11)	Medio	Diario	N	Todos
T(12)	Bajo	Mensual	1	Todos
T(13)	Medio	Diario	1	Todos
T(14)	Alto	Diario	N	Todos
T(15)	Alto	Diario	N	Todo
T(16)	Bajo	Mensual	1	Todos

### **9.3.2.2 Elección de organización de archivos**

El objetivo de este paso es seleccionar una organización de archivos que sea eficiente para cada relación base dentro del sistema.

Microsoft Access 2000 no permite especificar el tipo de organización que utilizará cada tabla de la base de datos

### **9.3.2.3 Elección de índices secundarios**

La generación de índices secundarios es un paso muy importante debido a que en algunos casos es necesario la utilización de atributos que no son claves primarias para la búsqueda de información. Microsoft Access 2000 genera de manera automática los índices primarios los campos que se definan como claves primarias. Para identificar índices secundarios se selecciona la opción “crear índices secundarios” y luego el campo que se desea sirva como índice secundario. El Módulo de Tratamientos no tiene definidos índices secundarios y no posee claves alternas que son las que generan dichos índices.

#### **9.3.2.4 Considerar la introducción de redundancia controlada (Denormalización)**

La denormalización es el proceso inverso a la normalización, es decir, se realizan cambios a la estructura de las tablas base, dejando la tabla en una forma normal inferior. Se refiere a un refinamiento del esquema relacional y se utiliza cuando se combinan dos tablas en una nueva. Esta tabla nueva tiene una forma normal inferior a las tablas originales.

El objetivo de la denormalización es mejorar el rendimiento de la base de datos para las consultas, no para las actualizaciones.

En el diseño del Módulo de Tratamientos no se utiliza el proceso de denormalización.

#### **9.3.2.5 Estimación de espacio de requerimiento en disco**

El propósito de esta tarea es estimar la cantidad de espacio en disco que es requerido para almacenar la base de datos. En general, la estimación está basada en el tamaño de cada tupla y el número de tuplas de la relación.

Considerando la capacidad de disco con que cuentan los equipos donde será implementado el sistema se puede afirmar que pueden soportar sin problemas la cantidad de información que se manejará en el sistema.

### **9.3.3 Diseño de mecanismos de seguridad**

Una base de datos es considerada una inversión y un importante recurso para la empresa por lo que el concepto de seguridad cobra real importancia.

Los requerimientos de seguridad deben ser implementados durante el diseño lógico de base de datos. El objetivo de este paso es decidir cómo serán realizadas esas medidas de seguridad.

#### **9.3.3.1 Diseño de vistas de usuario**

El objetivo es diseñar las vistas de usuario para la base de datos, basadas en el modelo de datos lógico local.

Se generará sólo una vista de usuario la cual permitirá el acceso total a la base de datos. No se implementarán otras vistas de usuario debido a que las únicas personas que tendrán instalada la aplicación serán los veterinarios de



los centros y deben tener acceso total al sistema para poder registrar la información.

### **9.3.3.2 Diseño de reglas de acceso**

La principal medida de seguridad con la que cuentan las bases de datos es a través de un Id. de usuario y una contraseña, los cuales entregan la autorización para el ingreso a la base de datos.

De acuerdo a qué nivel de autorización tenga cada usuario, este tendrá permiso para insertar, eliminar o actualizar información de una tabla.

En el presente sistema se define un tipo de usuario el cual posee acceso total a la base de datos, es decir podrá realizar actualizaciones y modificaciones a los datos en todas las tablas.

## **10. DISEÑO DE APLICACIÓN**

Los dos elementos claves en esta etapa son:

- ✓ Diseño Transaccional
- ✓ Diseño de interfaces de usuario

### **10.1 Diseño Transaccional**

Una transacción es una acción o serie de estas realizadas a través de un usuario o programa que accede o cambia el contenido de la base de datos

Existen tres tipos de transacciones:

- ✓ Transacciones de búsqueda: se utiliza para despliegue de datos
- ✓ Transacciones de actualización : se utilizan para insertar nuevos registros, eliminar o modificar registros existentes.
- ✓ Transacciones mixtas: se utilizan las dos operaciones mencionadas anteriormente.

La siguiente tabla muestra las tablas bases utilizadas para las transacciones del Módulo de Tratamientos

**Tabla N° 9 : Tablas bases utilizadas en las transacciones**

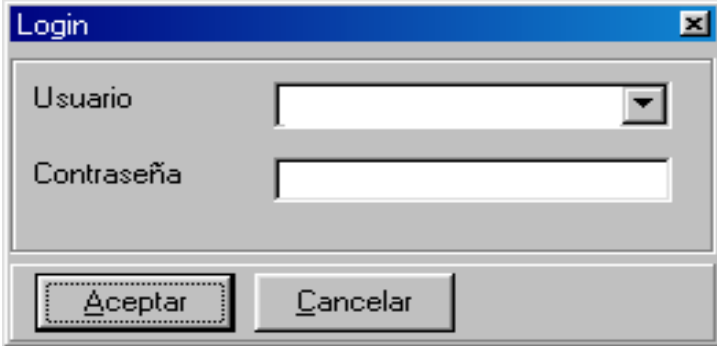
<b>T()</b>	<b>Tablas utilizadas</b>
T(1)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment, Company, Sites, FeedSupplier, Diets, FeedSize, FeedTypes, DFeeding
T(2)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment, Company, Sites, FeedSupplier, Diets, FeedSize, FeedTypes, DFeeding
T(3)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment, Company, Sites, FeedSupplier, Diets, FeedSize, FeedTypes, DFeeding
T(4)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment
T(5)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment
T(6)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment
T(7)	Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment
T(8)	Type_Treatment, TreatBySupplier, Treatment_Supplier,

T( )	Tablas utilizadas
	Ltreatment
T(9)	Veterinary
T(10)	FeedSupplier, Diets, FeedSize, FeedTypes, Type_Treatment
T(11)	Treatment, Dtreatments, Veterinary, Type_Treatment, TreatBySupplier, Treatment_Supplier, Ltreatment, Company, Sites, Veterinary
T(12)	Treatment, Dtreatment
T(13)	Type_Treatment, Treatment
T(14)	Dfeeding, Dtreatments
T(15)	Treatment, Dtreatments, Dfeeding
T(16)	Company, Sites

Como segundo punto a abarcar dentro del diseño de aplicación se encuentra el diseño de las interfaces de usuario.

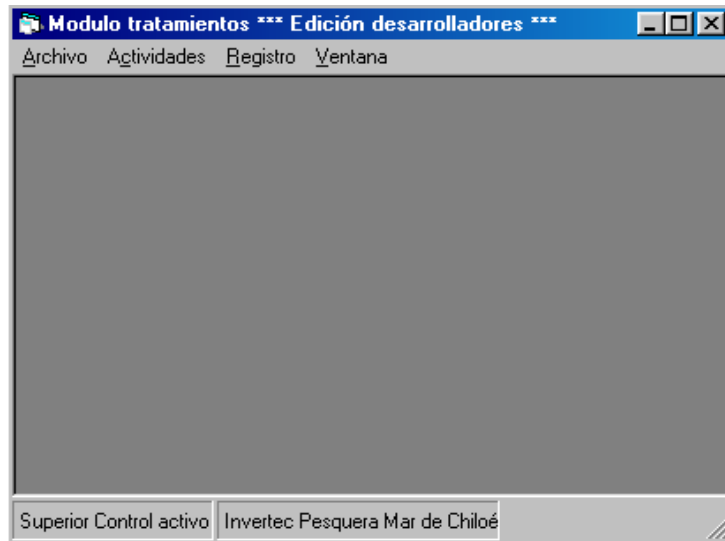
## 10.2 Diseño de interfaces de usuario

A continuación se presentan las pantallas correspondientes al Módulo de Tratamientos.

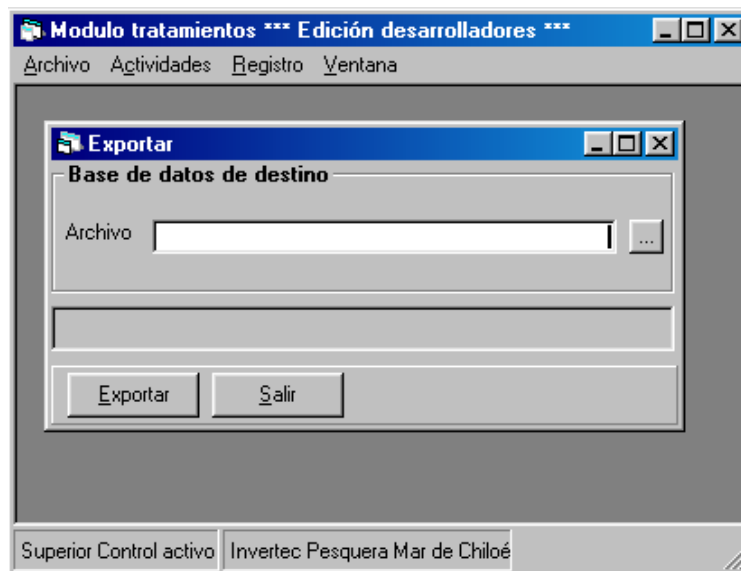


The image shows a standard Windows-style dialog box titled "Login". It features a blue title bar with a close button (X) in the top right corner. The main area is light gray and contains two input fields: "Usuario" (User) and "Contraseña" (Password). The "Usuario" field is a text box with a small downward arrow on the right side, indicating it might be a dropdown menu. The "Contraseña" field is a standard text box. At the bottom of the dialog, there are two buttons: "Aceptar" (Accept) and "Cancelar" (Cancel). The "Aceptar" button has a dotted border, while the "Cancelar" button has a solid border.

**Figura N° 17: Pantalla Ingreso al Módulo de Tratamientos**



**Figura N° 18: Pantalla Menú principal**



**Figura N° 19: Pantalla Exportar Datos a una Base de Datos externa**

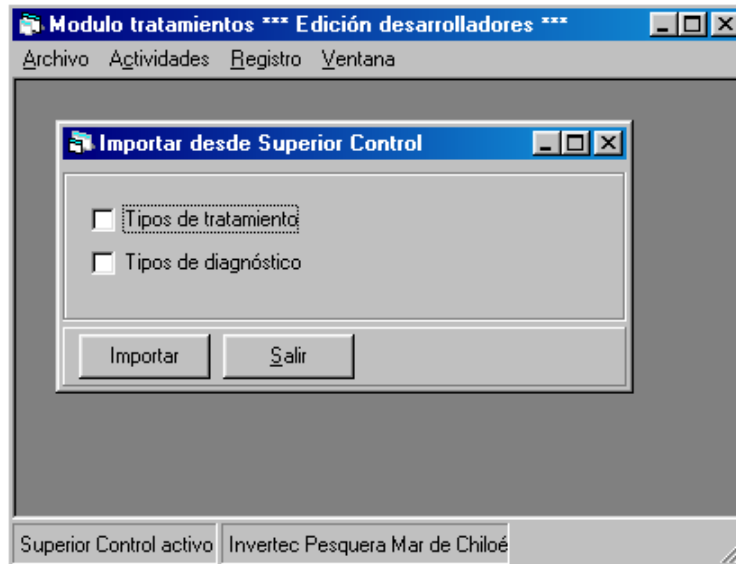


Figura N° 20 : Pantalla Importar Datos desde Superior Control

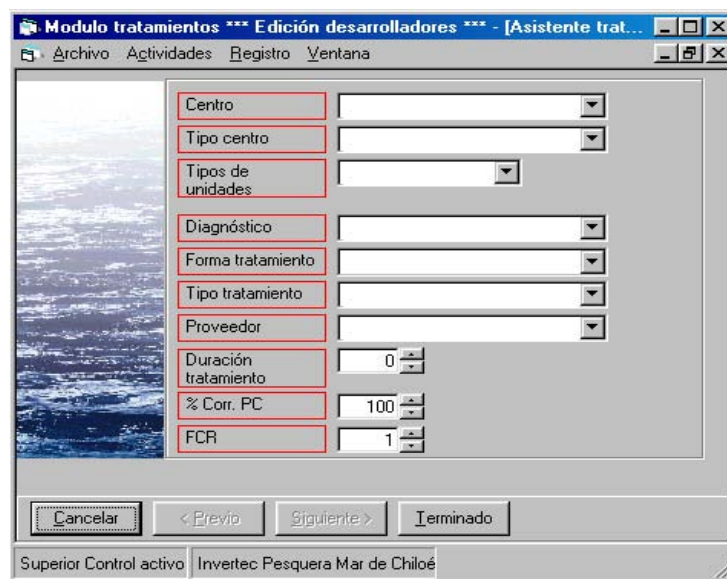


Figura N° 21 : Pantalla Crear nuevos Tratamientos

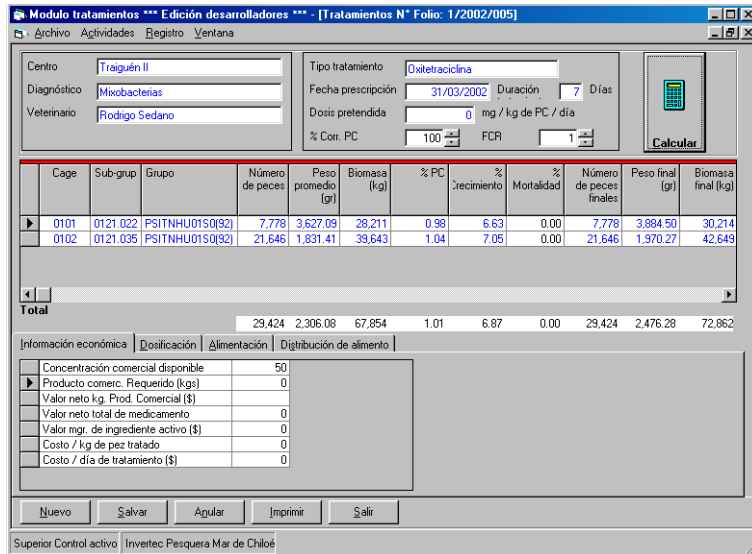


Figura N° 22: Pantalla Panorámica de un Tratamiento Ingresado

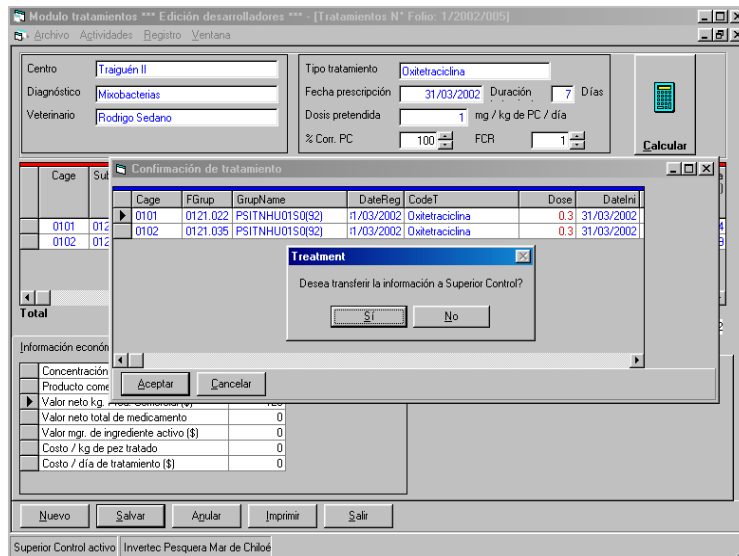


Figura N° 23: Pantalla Exportar Información de los Tratamientos directamente a Superior Control



Modulo tratamientos \*\*\* Edición desarrolladores \*\*\* - [Declaración de Garantía]

Archivo Actividades Registro Ventana

Centro:  Fecha:

Especie:

Jaula/Estanque:  N° de peces cosechados:

**DOCUMENTACION DE RESPALDO: incluir solo los que correspondan**

Documento	N° Folio	Fecha
*		

NOMBRE:

RUT:

PROFESION:

DIRECCION:

TELEFONO/FAX:

HACCP 1A:

N° OT:

Guardar Imprimir Salir

Superior Control activo Invertec Pesquera Mar de Chiloé

Figura N° 24: Pantalla Declaración de Garantía

Panorámica empresas

	Company #	Nombre	Teléfono	Fax #
▶	775098902	Fjord Seafood Chile SA	0	0
	797979902	Invertec Pesquera Mar de	0	0
*				

Agregar Salir

Figura N° 25: Pantalla Agregar nueva empresa



Figura N° 26: Pantalla Panorámica de los centros

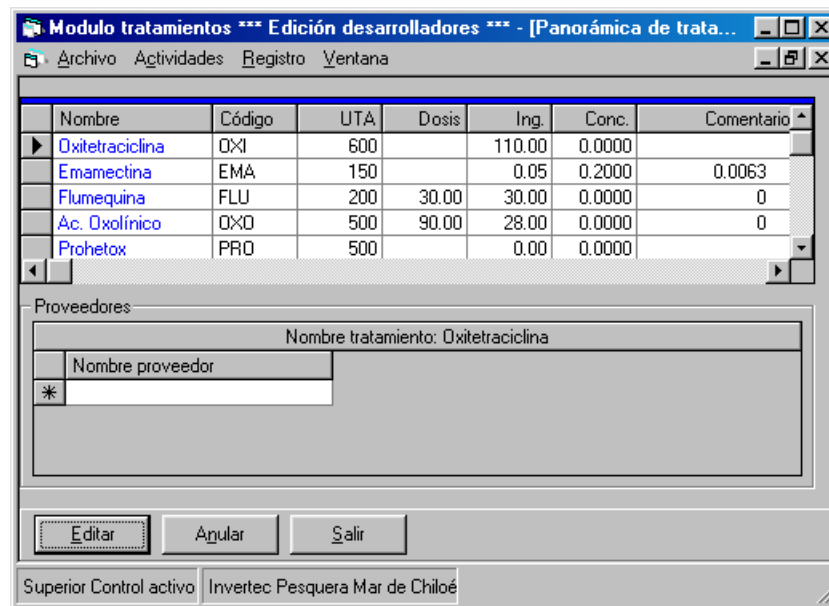


Figura N° 27: Pantalla Agregar Nuevas Medicinas para los Tratamientos



Figura N° 28: Pantalla Agregar Nuevos proveedores de Tratamientos

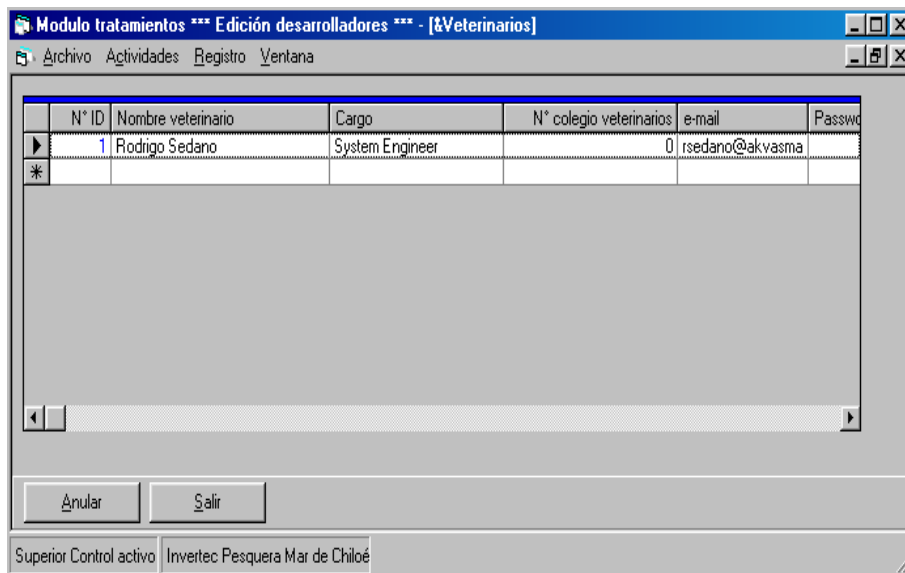



Figura N° 29: Pantalla Agregar Nuevos Veterinarios

1 of 1 75% Total:2 100% 2 of 2

 (N° FOLIO) 1/2002/002

### PRESCRIPCIÓN VETERINARIA

Fecha: 8 de Septiembre de 2001  
 Centro: MN Huelden      Area: Mar      Tipo Unidad: Jauja  
 Responsable:

#### Identificación de unidades tratadas

Jauja	Especie	Grupo	Número de peces	Peso promedio (gr)	Biomasa (kg)	% PC	Alimento por día (kg)	Alimento total (kg)	Ing. activo requerido
0304	Salmon Atlántico	AFANPUS0304P	12.107	4.193	50.124	0,35	200	1.400	35,30
0307	Salmon Atlántico	AFANPUS0307P	5.937	4.467	26.739	0,33	100	700	17,30
			18.104	4.325	76.863	0,34	300	2.100	52,60

Diagnóstico BKD  
 Medicinas Ac. Cristalino      Proveedor Veterinaria  
 Vía tratamiento Oral  
 Dosis: 1)      52,50 mg / kg / día - ppm / día  
 Fecha inicio de tratamiento (aprox.) 8 de Septiembre de 2001      Duración 7 Días  
 Fecha final de tratamiento (aprox.) 14 de Septiembre de 2001      Carenza 90 Días  
 Fecha final de carencia (aprox.) 12 de Diciembre de 2001

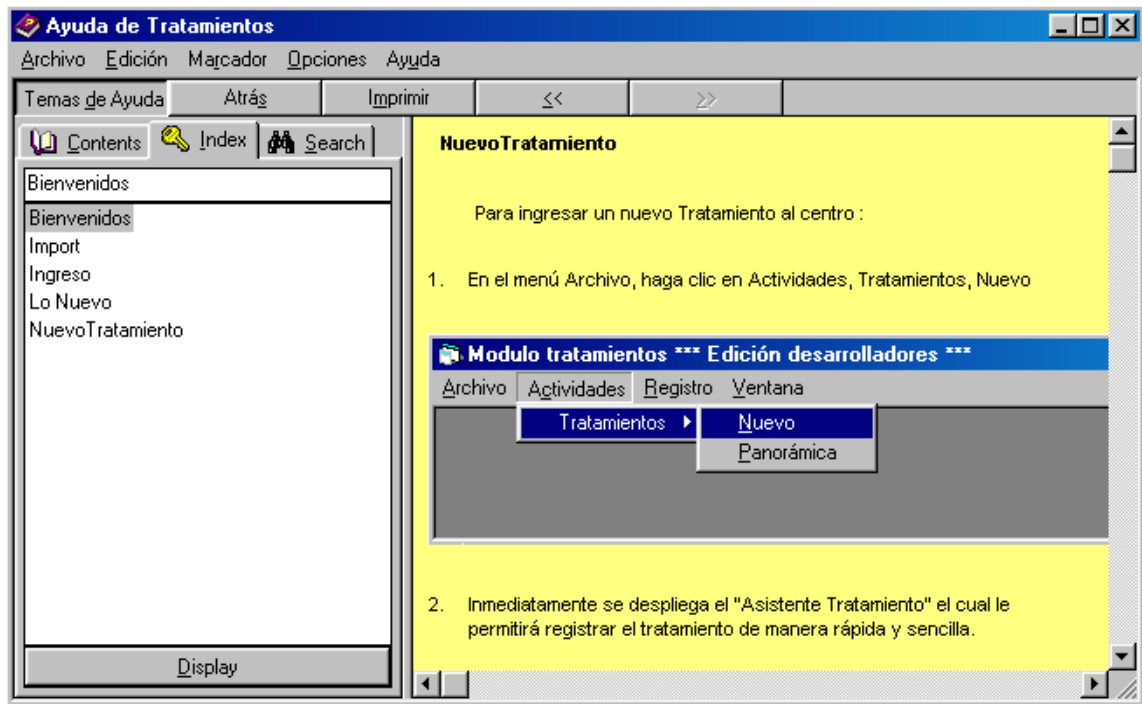
#### Concentración de medicina en el alimento o en el baño

LA: Medicina      25,00 gr LA / feed or      50,00 gr. Prod.      5,00 % por/kg de alimento

VETERINARIO A CARGO: Rodrigo Sedano  
 POSICION: System Engineer  
 N° IDENTIFICACION: 0  
 e-mail: rsedano@akwasmart.cl

**NOTAS:**  
 1) El grupo(s) de peces indicado, puede ser solamente cosechado y comercializado para el consumo humano en el mercado, después de completar la fecha final de carencia indicada.  
 2) La fecha final de carencia indica no se considera si el grupo de peces recibe un nuevo tratamiento después del actual.

**Figura N° 30: Pantalla Panorámica de los reportes que emite el Módulo de Tratamientos**



**Figura N° 31: Pantalla del Archivo de Ayuda desarrollado para el Módulo de Tratamientos**

En el anexo A sección A.1.5 se muestra un resumen de algunas características del software utilizado en la creación de los Archivos de Ayuda del sistema.

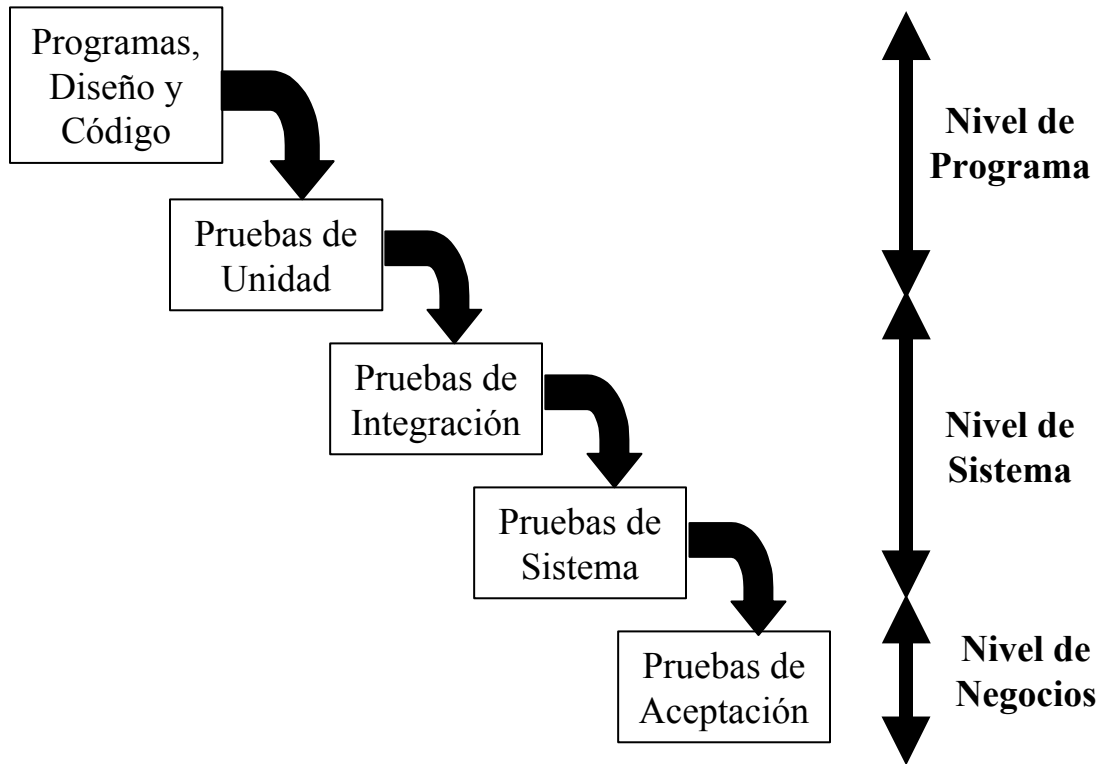
## **11. PRUEBA**

“La filosofía detrás de la prueba de sistema es la de encontrar errores. Los casos de prueba se diseñan con este propósito en mente. Un “caso de prueba” es un conjunto de datos o actividades que el sistema debe procesará como entrada normal. Sin embargo, los datos se crean con la intención expresa de determinar si el sistema los procesará correctamente. Cada caso se diseña con la intención de encontrar errores en la forma en que el sistema los procesará.” [Connolly1999]

### **11.1. Plataforma de cuatro pruebas para el desarrollo de sistemas.**

Durante el ciclo de desarrollo de los sistemas, las pruebas de gran cantidad de aplicaciones involucran cuatro distintos niveles de pruebas.

La secuencia en la cual la mayoría de las aplicaciones realizan sus pruebas con respecto al nivel en el cual se desarrollan se muestra a continuación.



**Figura N° 32 : Secuencia de Pruebas de un Sistema**

### **11.1.1 Pruebas de Unidad**

El objetivo de este tipo de pruebas es verificar que las pequeñas unidades lógicas del código de la aplicación trabajen según las especificaciones de diseño del software. La prueba de unidad es un proceso altamente iterativo el cual comienza con los primeros módulos y programas de aplicación codificados por los desarrolladores. Las especificaciones de los programas son usadas como principal fuente de información para derivar en los procedimientos de prueba y los casos de pruebas.

#### **11.1.1.1 Caja Negra**

Esta prueba se centrará en el análisis de las Interfaces de Usuario. Se analizará los datos de entrada para visualizar si producen la salida esperada.

Dos son los casos a verificar en este punto:

- ✓ Movimiento de datos
- ✓ Consultas

En el caso de ingreso de datos se analiza si los datos de entrada se encuentran reflejados en la base de datos.



Para la modificación de un registro se verifica que la actualización del dato se encuentre actualizado en la base de datos.

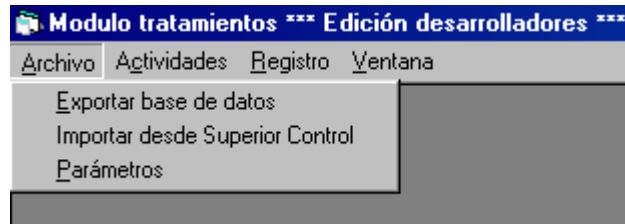
El caso de consultas de registros se verifica si se entrega los resultados esperados.

El Módulo de Tratamientos está dividido en tres menús :

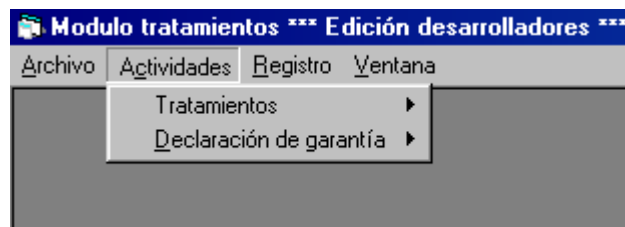
**Tabla N° 10 : Menús de usuario para el Módulo de Tratamientos**

<b>Nombre del Menú</b>	<b>Denominación en el sistema</b>
Archivos	MnuFile
Actividades	MnuAc
Registro	MnuRegister

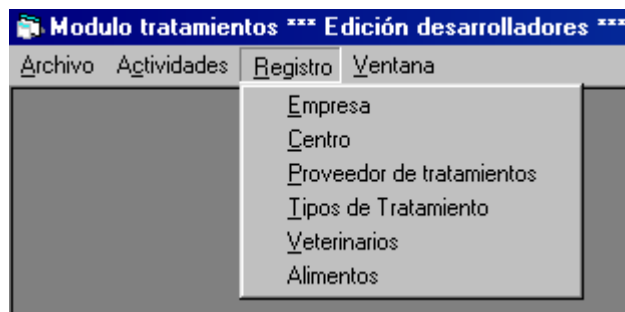
Las figuras mostradas a continuación visualizan la interfaz que representa cada Menú de Usuario en el Módulo de Tratamientos.



**Figura N° 33: Menú Archivo**



**Figura N° 34: Menú Actividades**



**Figura N° 35: Menú Registro**

A continuación se muestra las pruebas a los menús correspondientes al Módulo de Tratamientos especificando el resultado de la prueba.

Una prueba tendrá el valor de satisfactorio si:

- ✓ La consulta entrega los valores esperados
- ✓ El movimiento de datos se ve reflejado en la Base de Datos

En el caso de que la prueba para una cierta acción retorne un resultado no satisfactorio se procederá al análisis del problema y posterior etapa de solución.

### **Nomenclatura**

**C** : Consulta

**M**: Movimiento sobre Base de Datos

**Tabla N° 11: Resultados Prueba de Caja Negra**

<b>Menú</b>	<b>Tipo de Acción</b>	<b>Descripción</b>	<b>Resultado Satisfactorio</b>	<b>Resultado No Satisfactorio</b>
<b>Menú Archivo</b>				
Exportar Base de Datos	M	Exporta datos a una Base de Datos externa en formato .mdb	X	
Importar desde Superior Control	M	Importa datos desde Superior Control y los ingresa en las tablas del Módulo de Tratamientos	X	
Parámetros	C	Muestra los parámetros del sistema en diversos idiomas	X	
<b>Menú Actividades</b>			X	
Tratamientos			X	
Nuevo	M	Ingresa nuevos tratamientos al Módulo	X	
Panorámica	C	Muestra los tratamientos que se encuentran registrados	X	
Declaración de Garantía			X	
Nueva	M	Ingresa información para emitir la Declaración de Garantía	X	
Panorámica	C	Muestra las Declaraciones de Garantía emitidas	X	
<b>Menú Registro</b>			X	
Empresa	M	Permite agregar una nueva empresa	X	
Centro	C	Muestra los centros presentes en el Módulo	X	
Proveedor de Tratamiento	M	Permite agregar un nuevo proveedor de medicamentos	X	
Tipos de Tratamiento	M	Permite agregar un nuevo tipo de Tratamiento	X	
Veterinarios	M	Permite agregar información de nuevos veterinarios	X	
Alimentos	M	Permite agregar nuevas dietas medicadas	X	

### **11.1.2 Pruebas de Integración.**

El objetivo de estas pruebas es verificar que múltiples módulos del software o programas pueden trabajar según las especificaciones, cuando son gradualmente combinados unos con otros.

Las pruebas de integración fueron consideradas en el punto anterior ya que el Módulo interactúa con Superior Control. Todas las pruebas que incluyeron accesos a la Base de Datos de Superior y viceversa fueron aprobadas satisfactoriamente.

### **11.1.3 Prueba de Validación**

Esta prueba estará orientada a verificar que cada transacción de usuario sea satisfecha por el sistema.

La descripción de las transacciones se encuentra en el capítulo 8 en la sección 8.3 del presente documento y que corresponden a los Requerimiento de Usuario.

**Tabla N° 12: Validación de transacciones v/s formularios (procesos)**

<b>Transacción</b>	<b>Formulario Asociado (Menú/Formulario)</b>	<b>Resultado Satisfactorio</b>	<b>Resultado No Satisfactorio</b>
T(1)	Archivo/Importar desde Superior Control	X	
T(2)	Archivo/Exportar Base de Datos	X	
T(3)	Tratamientos/ Nuevos	X	
T(4)	Actividades/Tratamientos/ Nuevos	X	
T(5)	Actividades/Tratamientos/ Nuevos	X	
T(6)	Actividades/Tratamientos/ Panorámica	X	
T(7)	Registro/Tipos de Tratamientos	X	
T(8)	Registro/Proveedor de Tratamiento	X	
T(9)	Registro/Veterinarios	X	
T(10)	Registro/Alimentos	X	
T(11)	Actividades/Tratamientos/Declaración de Garantía	X	
T(12)	Actividades/Tratamientos/ Panorámica	X	
T(13)	Actividades/Tratamientos/ Panorámica	X	
T(14)	Actividades/Tratamientos/ Panorámica	X	
T(15)	Actividades/Tratamientos/ Panorámica	X	
T(16)	Registro/Empresa	X	
	Registro/Centro	X	

#### **11.1.4 Pruebas de Sistema.**

Estas pruebas comienzan después de las pruebas de integración y finaliza antes de las pruebas de aceptación. Su principal objetivo es verificar que el sistema por completo realice exitosamente todas las funciones del proyecto que han sido documentados en los requerimientos de sistema solicitados.

Algunos de las pruebas de este tipo son:

- ✓ Pruebas de respaldo/recuperación.
- ✓ Pruebas de seguridad.
- ✓ Pruebas de resistencia
- ✓ Pruebas de rendimiento.

##### **11.1.4.1 Prueba de recuperación**

Verifican que la aplicación pueda recuperarse fácilmente ante una falla en el Hardware, Software y en la red. Los casos de prueba deben ser contruidos con la intención de causar fallas en el sistema y luego verificar que la aplicación pueda ser recuperada para especificar a priori los eventos que las producen.

Algunos de los casos a realizar son:

- ✓ Respaldo completo de la Base de Datos o archivos planos.
- ✓ Recuperación completa de la Base de Datos o archivos planos.

El Módulo de Tratamientos mediante la opción “Exportar Base de Datos” permite realizar una respaldo completo de todas las tablas de la base de datos en formato \*.mdb (Datos de Microsoft Access), lo cual se puede realizar diariamente dependiendo de la política de respaldo que maneje el administrador de sistemas de la empresa. Fjord Seafood determinó que el papel de administrador del sistema corresponderá a uno de los Ingenieros de Sistema que pertenecen a la empresa el cual tendrá un papel clave en lo que se refiere a respaldos de la Base de Datos ya que deberá asegurarse de resguardar los datos en un medio y lugar seguros para disponer de ellos cuando ocurra alguna eventualidad.

Las pruebas realizadas en este punto concluyeron de manera satisfactoria.



#### **11.1.4.2 Prueba de seguridad**

Verifican que la aplicación es inmune a los intentos de accesos no autorizados a ella. Los casos son diseñados con la firme intención de quebrar la aplicación sobrepasando las características de seguridad.

Algunos de los casos que son importantes de analizar son:

- ✓ Procedimientos de ingreso de la clave de acceso a la aplicación.
- ✓ Características de la seguridad para controlar el acceso a ventanas específicas de la aplicación.
- ✓ Características de seguridad para controlar el acceso a ítems de menús específicos u objetos de control que son desplegados en la ventana.
- ✓ Características de seguridad para controlar el acceso a las Bases de Datos o archivos planos.
- ✓ Maneras potenciales de acceso a los servidores de Bases de Datos vía otras formas que los propios programas de la aplicación.
- ✓ Procedimientos de cambio de las claves de acceso.

Los usuarios del Módulo de Tratamiento son veterinarios de la empresa los cuales poseen una cuenta de usuario y una clave o password. Cualquier otro usuario que no se encuentre previamente definido como usuario no podrá

tener acceso a las interfaces del sistema. Además, en la base de datos consolidada que se encuentra en el Servidor Principal de la empresa se encuentran definidos los permisos y privilegios de acceso a la carpeta que contiene los datos por lo que un acceso a la aplicación de un usuario no autorizado por un medio externo al sistema es imposible.

#### **11.1.4.3 Prueba de resistencia**

La prueba de resistencia consiste en saturar al sistema mediante el acceso múltiple de usuarios. La idea es ver cuanta sobrecarga puede ser capaz de sobrellevar el sistema.

El Módulo de Tratamiento está diseñado para que un usuario a la vez pueda ingresar y modificar datos, por lo que no existirá una sobrecarga de acceso a la base de datos. El proceso de actualizar el consolidado de datos en la oficina central lo llevará a cabo el administrador del sistema por lo que tampoco existirá una sobrecarga de acceso y actualizaciones de la base de datos.

#### **11.1.4.4 Prueba de rendimiento**

Verifica que tan bien responde la aplicación ante los distintos niveles de requerimientos de servicios, como por ejemplo el tiempo de respuesta de las transacciones bajo condiciones operativas normales, pero con variaciones en la carga de los datos.

Para el caso del Módulo de Tratamientos, el registro de información se realizará diariamente ya que se debe ingresar alimentación y mortalidad relacionada con los peces en tratamiento, lo cual generará flujo de información desde y hacia la Base de Datos, la cual tendrá un tamaño variable. La base de datos consolidada contendrá los datos de todos los centros, 36 en total, lo cual se traducirá en una exportación de datos semanal. Es difícil estimar el tamaño de las Bases de Datos ya que dependerá de la cantidad de datos que se registren.

En las pruebas realizadas en el Módulo de Tratamiento no existen variaciones significativas en cuanto al rendimiento de la aplicación.

### **11.1.5 Pruebas de Aceptación del Usuario.**

Estas pruebas validan la funcionalidad completa del sistema, incluyendo los manuales de procedimientos. Esto asegura que el sistema responde a todos los requerimientos del proyecto que los usuarios identificaron para el sistema.

Estas pruebas cubren todos los elementos del sistema incluyendo el software, hardware, documentación y los componentes de la interfaz gráfica del usuario. Siempre que sea posible hacer, el sistema debería ser probado en un ambiente que emule cercanamente el ambiente de producción particular.

Este tipo de pruebas son diseñadas, construidas, y ejecutadas por los propios usuarios, en este caso fueron cuatro personas. Adicionalmente, ciertos componentes del sistema pueden también ser probados por el grupo operacional que será responsable de apoyar la aplicación después que sea transferida al ambiente productivo.

Luego de chequear que todos los requerimientos de los usuarios han sido cubiertos por el sistema, incluyendo un entorno amigable y archivos de ayuda, se procedió a la instalación del sistema para evaluar un período de marcha blanca dentro de la empresa.

## **12. CONCLUSIONES Y/O RECOMENDACIONES**

Luego de finalizado el proyecto se puede concluir que los objetivos planteados al inicio del proyecto se cumplieron de manera satisfactoria.

Las tareas involucradas en el registro de Tratamientos se encuentran automatizadas de manera que los usuarios pueden disponer de una herramienta que les permite agilizar los procesos y obtener información fiable y oportuna.

Es importante destacar los factores que influyeron para lograr un resultado satisfactorio: la disponibilidad de herramientas de desarrollo, el trabajo coordinado y participativo de las distintas personas involucradas en el proyecto, los cuales sumados a la correcta elección de la Metodología permitieron desarrollar una herramienta que cumple los objetivos y requerimientos planteados y que además servirá de real apoyo a la gestión de la empresa en el Area Salud.

Se recomienda que la empresa Fjord Seafood designe sólo personal especializado, por ejemplo veterinarios, para el proceso de ingresar información al Módulo ya que de esta manera los resultados serán el reflejo de la situación actual del centro de cultivo, de igual manera se hace la misma recomendación

para el proceso de análisis de la información ya que los datos que se obtienen no pueden ser interpretados por personas que no tengan conocimientos previos en la materia.

Así mismo se recomienda una política de respaldo de la información del Módulo de Tratamientos tanto de los centros como de la Base de Datos consolidada, para prevenir cualquier eventualidad producto de fallas eléctricas o de hardware. De igual manera se sugiere un Mantenimiento Operacional para cubrir posibles sugerencias o nuevos requerimientos de los usuarios del Sistema.

### 13. BIBLIOGRAFIA

- [Barría2002] Barría Vargas María. Sistema Control Docente, Evaluación e Implantación de Sistemas Intranet Instituto de Informática. Universidad Austral de Chile. 2002.
- [Connolly1999] Connolly Thomas – Begg Carolyn – Strachan Anne. Database System. Addison – Wesley Longmann Limited. Segunda Edición. 1999.
- [García1996] García Alcázar Enrique. Análisis de requerimientos y trazabilidad en el modelo de objetos. Grupo de trabajo de Integral Object. II Jornadas de OO de Madrid. 1996.
- [Rivas2002] Rivas Yonekura Kenyiro. Análisis y diseño de Sistemas.  
Disponibile en:  
<http://www.monografias.com/trabajos7/resi/resi.shtml>.  
2002.

- [Mansilla2001] Mansilla Mendoza Jorge. Diseño de un Sistema de Apoyo al Control del Mantenimiento de Equipos, Maquinaria y Vehículos. Universidad Austral de Chile. 2001.
- [Pressman1993] Pressman Roger. Ingeniería de Software: Un enfoque Práctico. Mc Graw-Hill. Tercera Edición. 1993.
- [Unav2002] Universidad de Navarra – Centro de Tecnología Informática. Microsoft Access 2000.  
Disponible en:  
<http://www.unav.es/cti/manuales/access97/#12>.  
2002.



## 14. ANEXOS

### A. Procesos

#### A.1 Proceso de Importación/ exportación de Datos

El proceso de Importar/Exportar datos desde y hacia el Sistema de Producción “Superior Control” involucra una serie de llamadas a procedimientos y la utilización de sentencias de código específicas, debido a que este sistema fue desarrollado utilizando el motor de Base de Datos Btrieve de Pervasive.

La opción “Exportar” permite generar archivos con datos del sistema los cuales más tarde pueden ser recuperados en la base central. El sistema permite exportar a una Base de Datos Access o directamente a las tablas de Superior Control.

La opción “Importar” permite obtener datos desde Superior Control e incorporarlos al Módulo de Tratamientos.

Para realizar una de estas acciones se accede al menú “Archivo” y luego al menú “Exportar Datos a una Base de Datos externa” o “Importar Datos desde Superior Control”. La opción “Exportar datos a Superior Control” se despliega

cuando se ha creado un nuevo tratamiento en el centro y se confirma la información.

A continuación se muestra parte del código que permite realizar estos procesos:

### **A.1.1 Exportar datos a Superior Control**

```
Private Sub BtnOk_Click()  
    MsgResp = MsgBox(GetDescrip(100013), vbYesNo)  
    Select Case MsgResp  
        Case vbYes  
            Set T_SpkHElse = CreateObject(SpkSrv & "SPKHElse")  
            T_SpkHElse.Konstruer  
            T_SpkHElse.sti = GetFolder(0)  
            T_SpkHElse.modus = 0  
  
            SC_SpkHElse = T_SpkHElse.Open  
            If SC_SpkHElse <> 0 Then  
                MsgBox "Error: " & SC_SpkHElse & " When opening SpkHElse.dta"  
            End If  
  
            Set T_SpkEnhet = CreateObject(SpkSrv & "SPKENhet")  
            T_SpkEnhet.Konstruer  
            T_SpkEnhet.sti = GetFolder(0)  
            T_SpkEnhet.modus = 0  
  
            SC_SpkEnhet = T_SpkEnhet.Open  
            If SC_SpkEnhet <> 0 Then
```

```

MsgBox "Error: " & SC_SpkEnhet & " When opening SpkEnhet.dta"
End If
BMark = GridConfirm.Bookmark
GridConfirm.Redraw = False
GridConfirm.MoveFirst
For RR = 0 To GridConfirm.Rows - 1
    'Abre tabla tratamiento para guardar la información

    T_SpkHElse.Indeks = 0
    T_SpkHElse.NullstillNokkel

    T_SpkHElse.SettNokkel 0, GridConfirm.Columns(0).Value
    T_SpkHElse.SettNokkel 1, PrmRecType1
    T_SpkHElse.SettNokkel 2, GridConfirm.Columns(2).Value
    T_SpkHElse.SettNokkel 3, ConvertD_Str(GridConfirm.Columns(6).Value)
    T_SpkHElse.SettNokkel 4, Val(GridConfirm.Columns(7).Value)
    SC_SpkHElse = T_SpkHElse.GetFirst
    If SC_SpkHElse <> 0 Then
        T_SpkHElse.Kons = GridConfirm.Columns(0).Value
        T_SpkHElse.RecType = PrmRecType1
        T_SpkHElse.FGruppe = GridConfirm.Columns(2).Value
        T_SpkHElse.Obs_Dato = ConvertD_Str(GridConfirm.Columns(6).Value)
        T_SpkHElse.Kode = Val(GridConfirm.Columns(7).Value)
        T_SpkHElse.Enhet = GridConfirm.Columns(3).Value
        T_SpkHElse.Vaks_Maate = 0
        T_SpkHElse.Undersok = 0
        T_SpkHElse.Beh_start = ConvertD_Str(GridConfirm.Columns(9).Value)
        T_SpkHElse.Beh_slutt = ConvertD_Str(GridConfirm.Columns(10).Value)
        T_SpkHElse.Kar_Slutt = ConvertD_Str(GridConfirm.Columns(11).Value)
        T_SpkHElse.Aarsak = 0
        T_SpkHElse.Antall = 0
        T_SpkHElse.Vekt = 0
        T_SpkHElse.Uttaker = 0
        T_SpkHElse.Mottaker = 0
    End If
Next RR

```

```

    T_SpkHElse.Mengde = GridConfirm.Columns(8).Value
    T_SpkHElse.Dosering = 0
    T_SpkHElse.Holdbarhetsdato = 0
    T_SpkHElse.Merknad = 0
    T_SpkHElse.Ledig = ""
    SC_SpkHElse = T_SpkHElse.InsertRec
    If SC_SpkHElse <> 0 Then
        MsgBox "Error saving on SpkHElse"
    End If

    T_SpkEnhet.Indeks = 0
    T_SpkEnhet.NullstillNokkel
    T_SpkEnhet.SettNokkel 0, GridConfirm.Columns(0).Value
    T_SpkEnhet.SettNokkel 1, GridConfirm.Columns(3).Value
    SC_SpkEnhet = T_SpkEnhet.GetFirst

    If SC_SpkEnhet = 0 Then
        T_SpkEnhet.Karan_Dato = ConvertD_Str(GridConfirm.Columns(11).Value)
        SC_SpkEnhet = T_SpkEnhet.UpdateRec
        If SC_SpkEnhet <> 0 Then
            MsgBox "Error saving on SpkEnhet"
        End If
    End If
End If
GridConfirm.MoveNext
Next
GridConfirm.Redraw = True
GridConfirm.Bookmark = BMark

SC_SpkEnhet = T_SpkEnhet.Close
If SC_SpkEnhet <> 0 Then
    MsgBox "Error: " & SC_SpkEnhet & " When Closing SpkEnhet.dta"
End If
T_SpkEnhet.Destruer

```

```

Set T_SpkEnhet = Nothing

SC_SpkHElse = T_SpkHElse.Close
If SC_SpkHElse <> 0 Then
MsgBox "Error: " & SC_SpkHElse & " When Closing SpkHElse.dta"
End If
T_SpkHElse.Destruer
Set T_SpkHElse = Nothing

Sql_Query = " Select * From Sites " & _
           " Where Code_Company=" & FrmNewTreat.CmbSites.Columns(4).Value & ""
AND " & _
           " Id_Concession=" & Mid(FrmNewTreat.CmbSites.Columns(0).Value, 1, 8) & ""
AND " & _
           " Code_Site=" & Mid(FrmNewTreat.CmbSites.Columns(0).Value, 9, 11) & ""

Set Rec_NewSite = New ADODB.Recordset
Rec_NewSite.CursorLocation = adUseClient
Rec_NewSite.CursorType = adOpenDynamic
Rec_NewSite.LockType = adLockOptimistic
Rec_NewSite.Open Sql_Query, Con
If Rec_NewSite.EOF And Rec_NewSite.BOF Then
    Rec_NewSite.AddNew
        Rec_NewSite("Code_Company") = FrmNewTreat.CmbSites.Columns(4).Value
        Rec_NewSite("Id_Concession") = Mid(FrmNewTreat.CmbSites.Columns(0).Value,
1, 8)
        Rec_NewSite("Code_Site") = Mid(FrmNewTreat.CmbSites.Columns(0).Value, 9,
11)
        Rec_NewSite("Description") = FrmNewTreat.CmbSites.Columns(1).Value
        Rec_NewSite("TypeSite") = FrmNewTreat.CmbTypeSite.Columns(0).Value
    Rec_NewSite.Update
End If
Rec_NewSite.Close
Set Rec_NewSite = Nothing

```

### **'Guarda datos de Tratamientos**

```
Select Case FrmNewTreat.CmbWTreat.Columns(0).Value
    Case TreatOral
        Call SaveTreatOral
    Case TreatBath
        Call SaveTreatBath
End Select

MsgBox GetDescrip(100014), vbInformation
Case Else
End Select
End Sub
```

### **A.1.2 Importar Datos desde Superior Control**

```
Private Sub BtnImp_Click()
Screen.MousePointer = 11
DoEvents
If ChkOpt(0).Value = 1 Then
    Call SpkParam(PrmSupTreat)
End If
If ChkOpt(1).Value = 1 Then
    Call SpkParam(PrmSupDiag)
End If
Screen.MousePointer = 0
If ChkOpt(0).Value = 1 Or ChkOpt(1).Value = 1 Then
    MsgBox GetDescrip(100016), vbInformation + vbOKOnly
End If
End Sub
```

### A.1.3 Exportar Datos a una Base de Datos externa

```
Private Sub BtnExport_Click()  
If Trim(TxtFile.Text) = "" Then  
    MsgBox GetDescrip(100025), vbCritical + vbOKOnly  
    Exit Sub  
Else  
    If Dir(Trim(TxtFile.Text), vbArchive) = "" Then  
        MsgBox GetDescrip(100026), vbCritical + vbOKOnly  
        Exit Sub  
    End If  
    If Nom_DBase <> Mid(Trim(TxtFile.Text), Len(Trim(TxtFile.Text)) - 12, 13) Then  
        MsgBox GetDescrip(100027), vbCritical + vbOKOnly  
        Exit Sub  
    End If  
End If
```

```
Dim Rec_Origen As ADODB.Recordset  
Dim Rec_Destino As ADODB.Recordset  
Set Rec_Origen = New ADODB.Recordset  
Set Rec_Destino = New ADODB.Recordset
```

```
TotTab = 17  
NTab = 0  
Call OpenDatabaseDes(TxtFile.Text)  
PBar.Visible = True  
DoEvents
```

```
*****
```

```
** Tratamientos (solo esta tabla como ejemplo del procedimiento)
```

```
*****
```

```

NTab = NTab + 1
Sql_Query = "Select * From Treatments "
Set Rec_Origen = New ADODB.Recordset
Rec_Origen.CursorLocation = adUseClient
Rec_Origen.CursorType = adOpenForwardOnly
Rec_Origen.LockType = adLockOptimistic
Rec_Origen.Open Sql_Query, Con
Do While Not Rec_Origen.EOF
    Sql_Query = " Select * From Treatments " & _
        " Where N_Identity=" & Rec_Origen("N_Identity") & " AND " & _
        " YearReg=" & Rec_Origen("YearReg") & " AND " & _
        " CorrReg=" & Rec_Origen("CorrReg")

    Set Rec_Destino = New ADODB.Recordset
    Rec_Destino.CursorLocation = adUseClient
    Rec_Destino.CursorType = adOpenDynamic
    Rec_Destino.LockType = adLockOptimistic
    Rec_Destino.Open Sql_Query, ConDes
    If Rec_Destino.EOF And Rec_Destino.BOF Then
        Rec_Destino.AddNew
        Rec_Destino("N_Identity") = Rec_Origen("N_Identity")
        Rec_Destino("YearReg") = Rec_Origen("YearReg")
        Rec_Destino("CorrReg") = Rec_Origen("CorrReg")
        Rec_Destino("Code_Company") = Rec_Origen("Code_Company")
        Rec_Destino("Id_Concession") = Rec_Origen("Id_Concession")
        Rec_Destino("Code_Site") = Rec_Origen("Code_Site")
        Rec_Destino("Dates") = Rec_Origen("Dates")
        Rec_Destino("Code_TSite") = Rec_Origen("Code_TSite")
        Rec_Destino("Code_TUnit") = Rec_Origen("Code_TUnit")
        Rec_Destino("Code_WTreat") = Rec_Origen("Code_WTreat")
        Rec_Destino("Code_TDiag") = Rec_Origen("Code_TDiag")
        Rec_Destino("Code_Treatment") = Rec_Origen("Code_Treatment")
        Rec_Destino("Code_Supplier") = Rec_Origen("Code_Supplier")
        Rec_Destino("N_Days_Treatment") = Rec_Origen("N_Days_Treatment")
    End If
Loop

```



```

Rec_Destino("CorrPC") = Rec_Origen("CorrPC")
Rec_Destino("FCR") = Rec_Origen("FCR")
Rec_Destino("PretDose") = Rec_Origen("PretDose")
Rec_Destino("Concent") = Rec_Origen("Concent")
Rec_Destino("ProdCom") = Rec_Origen("ProdCom")
Rec_Destino("ValNeto") = Rec_Origen("ValNeto")
Rec_Destino("ValNetoT") = Rec_Origen("ValNetoT")
Rec_Destino("ValMG") = Rec_Origen("ValMG")
Rec_Destino("CostKG") = Rec_Origen("CostKG")
Rec_Destino("CostDay") = Rec_Origen("CostDay")
Rec_Destino("PerIngAct") = Rec_Origen("PerIngAct")
Rec_Destino("PerProdCom") = Rec_Origen("PerProdCom")
Rec_Destino("KgIngAct") = Rec_Origen("KgIngAct")
Rec_Destino("KgProdCom") = Rec_Origen("KgProdCom")
Rec_Destino("NomResp") = Rec_Origen("NomResp")
Rec_Destino.Update
End If
Rec_Destino.Close
Rec_Origen.MoveNext
Loop
Rec_Origen.Close
*****
PBar.Value = NTab / TotTab * 100
DoEvents

Set Rec_Origen = Nothing
Set Rec_Destino = Nothing
ConDes.Close

MsgBox GetDescrip(100024), vbInformation + vbOKOnly
PBar.Value = 0
PBar.Visible = False
End Sub

```

#### A.1.4 Fórmulas utilizadas en el Módulo de Tratamientos

Para lograr determinar de manera exacta los distintos cálculos que involucró el sistema se utilizaron una serie de fórmulas que han sido determinadas por profesionales del área, ya sean matemáticos, estadísticos o biólogos marinos. Muchas de estas formulas no son divulgadas por las empresas ya que implica publicar resultados que producen beneficios productivos y económicos.

Como ejemplo se muestra el código para alguna de las fórmulas utilizadas en el desarrollo del sistema:

$$a) \quad \%CRECIMIENTO = \frac{Wf - Wi}{Wf}$$

#### Código para el % Crecimiento

```
Private Sub BtnSave_Click()  
    If Trim(TxtNomTab.Text) <> Empty Then  
        Sql_Query = " Select * From HTabPC " & _  
            " Where NameTab=" & Trim(TxtNomTab.Text) & ""  
        Set Rec_TabPC = New ADODB.Recordset  
        Rec_TabPC.CursorLocation = adUseClient  
        Rec_TabPC.CursorType = adOpenDynamic  
        Rec_TabPC.LockType = adLockOptimistic
```

```

Rec_TabPC.Open Sql_Query, Con
If Rec_TabPC.EOF And Rec_TabPC.BOF Then
    If ChkDefault.Value <> 0 Then
        Con.Execute "Update HTabPC Set HTabPC.IsDef=0"
    End If
    Rec_TabPC.AddNew
    Rec_TabPC("NameTab") = Trim(TxtNomTab.Text)
    Rec_TabPC("IsDef") = ChkDefault.Value * -1
    Rec_TabPC.Update
    Rec_TabPC.Close

    Sql_Query = " SELECT * From HTabPC " & _
        " Order by NameTab"
    Set Rec_TabPC = New ADODB.Recordset
    Rec_TabPC.CursorLocation = adUseClient
    Rec_TabPC.CursorType = adOpenDynamic
    Rec_TabPC.LockType = adLockOptimistic
    Rec_TabPC.Open Sql_Query, Con
    If Not Rec_TabPC.EOF And Not Rec_TabPC.BOF Then
        ChkDefault.Value = Abs(Rec_TabPC("IsDef"))
        GridTabPC.AllowAddNew = True
        TxtNomTab.Visible = False
        CmbNomTab.Visible = True
        Call GetDataCMB(Rec_TabPC, Sql_Query, "NameTab", "NameTab", "NameTab",
FrmTabPC.CmbNomTab)
        Call GetDataGrid(Rec_TabPC("NameTab"))
        CmbNomTab.Visible = True
    Else
        TxtNomTab.Visible = True
    End If
Else
    If ChkDefault.Value <> 0 Then
        Con.Execute "Update HTabPC Set HTabPC.IsDef=0"
    End If

```

```

        Rec_TabPC("IsDef") = ChkDefault.Value * -1
    Rec_TabPC.Update
    Rec_TabPC.Close
End If
End If
End Sub

```

b) 
$$FCR = \frac{a \text{ limento}}{\text{IncrementoBiomasa}}$$

$$W_f = W_i * e^{(SGR * \Delta t) / 100}$$

$$\% \text{PesoCuerpo} = FCR * SGR * 100$$

### **Código para el FCR, Wf y % Peso Cuerpo**

```

For RR = 0 To GridRTreat.Rows - 1

```

```

    Call ProcCrecPC(GridRTreat.Columns(9).Value, _
        GridRTreat.Columns(10).Value, _
        GridRTreat.Columns(13).Value, _
        CDbI(TxtFCR.Text), _
        GridRTreat.Columns(15).Value, _
        25)

```

```

    TotIniNFish = TotIniNFish + GridRTreat.Columns(9).Value

```

```

    TotIniBio = TotIniBio + GridRTreat.Columns(11).Value

```

```

    PromPC = PromPC + GridRTreat.Columns(13).Value

```

```

    TotEndNFish = TotEndNFish + GridRTreat.Columns(16).Value

```

```

    TotEndBio = TotEndBio + GridRTreat.Columns(18).Value

```

```

TotFeedDay = TotFeedDay + GridRTreat.Columns(20).Value
TotFeed = TotFeed + GridRTreat.Columns(21).Value
TotDose = TotDose + GridRTreat.Columns(22).Value

'Feed distribution
If GridRTreat.Columns(19).Value <> "" Then

    If Rec_DataFeedDis.RecordCount > 0 Then
        Rec_DataFeedDis.MoveFirst
    End If
    ExisteFeed = False
    Do While Not Rec_DataFeedDis.EOF
        If Trim(Rec_DataFeedDis("FullCode")) = Trim(GridRTreat.Columns(19).Value)
Then
            Rec_DataFeedDis("Amount") = Rec_DataFeedDis("Amount").Value +
GridRTreat.Columns(21).Value
            Rec_DataFeedDis.Update
            ExisteFeed = True
        Exit Do
    End If
    Rec_DataFeedDis.MoveNext
Loop
If Not ExisteFeed Then
    Rec_DataFeedDis.AddNew
    Rec_DataFeedDis("FullCode") = Trim(GridRTreat.Columns(19).Value)
    Rec_DataFeedDis("Amount") = GridRTreat.Columns(21).Value
    Rec_DataFeedDis.Update
End If
End If

GridRTreat.MoveNext
Next
GridTot.Columns(9).Value = TotIniNFish
GridTot.Columns(10).Value = TotIniBio / TotIniNFish * 1000

```

```

GridTot.Columns(11).Value = TotIniBio
GridTot.Columns(13).Value = PromPC / GridRTreat.Rows
GridTot.Columns(16).Value = TotEndNFish
If TotEndNFish <> 0 Then
    GridTot.Columns(17).Value = TotEndBio / TotEndNFish * 1000
Else
    GridTot.Columns(17).Value = 0
End If
GridTot.Columns(18).Value = TotEndBio
GridTot.Columns(20).Value = TotFeedDay
GridTot.Columns(21).Value = TotFeed
GridTot.Columns(22).Value = TotDose
If GridTot.Columns(17).Value <> 0 Then
    GridTot.Columns(14).Value = (GridTot.Columns(17).Value -
GridTot.Columns(10).Value) / GridTot.Columns(17).Value * 100
Else
    GridTot.Columns(14).Value = 0
End If
GridTot.Columns(15).Value = (TotIniNFish - TotEndNFish) / TotIniNFish * 100

GridRTreat.Bookmark = BMark
GridFeeding.Bookmark = BMark
GridRTreat.Redraw = True
GridFeeding.Redraw = True
Call CalEco

Set GridFeedDis.DataSource = Rec_DataFeedDis
GridFeedDis.MoveFirst

```

### A.1.5 Software de creación de Archivos de Ayuda

Los archivos de ayuda para el Módulo de Tratamientos fueron desarrollados en RoboHelp Office 9.0, software que permite crear aplicaciones de ayuda que luego pueden ser utilizadas como procesos dentro de otras aplicaciones o como aplicaciones independientes.

A continuación se describen algunas características de esta herramienta y su utilización en la creación de los Archivos de Ayuda.

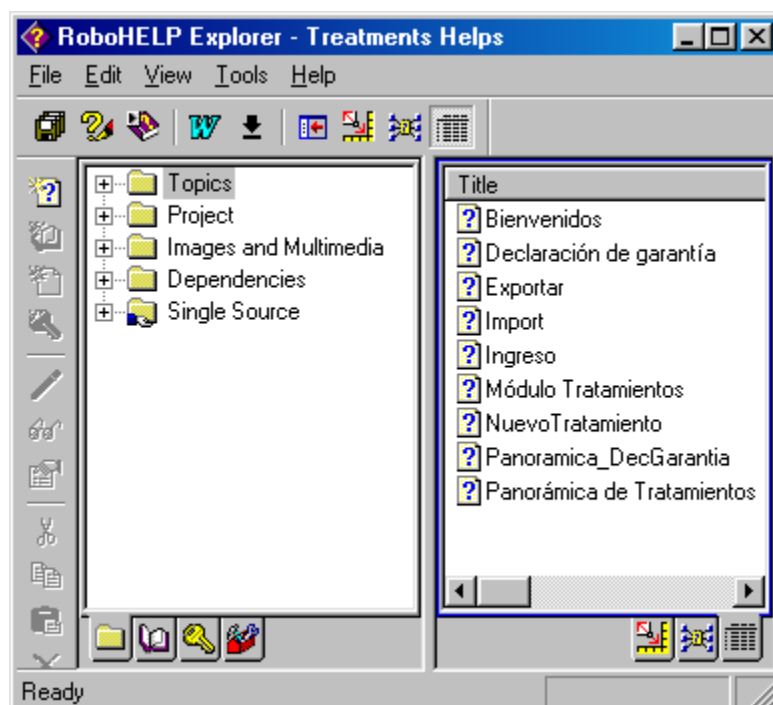
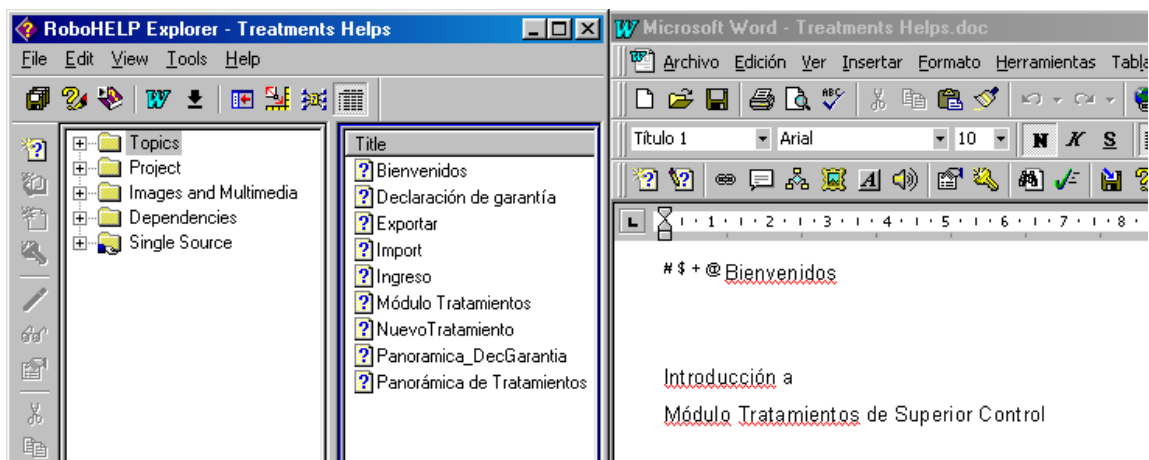


Figura N° 36 : Vista del panel de trabajo en RoboHelp Office 9.0

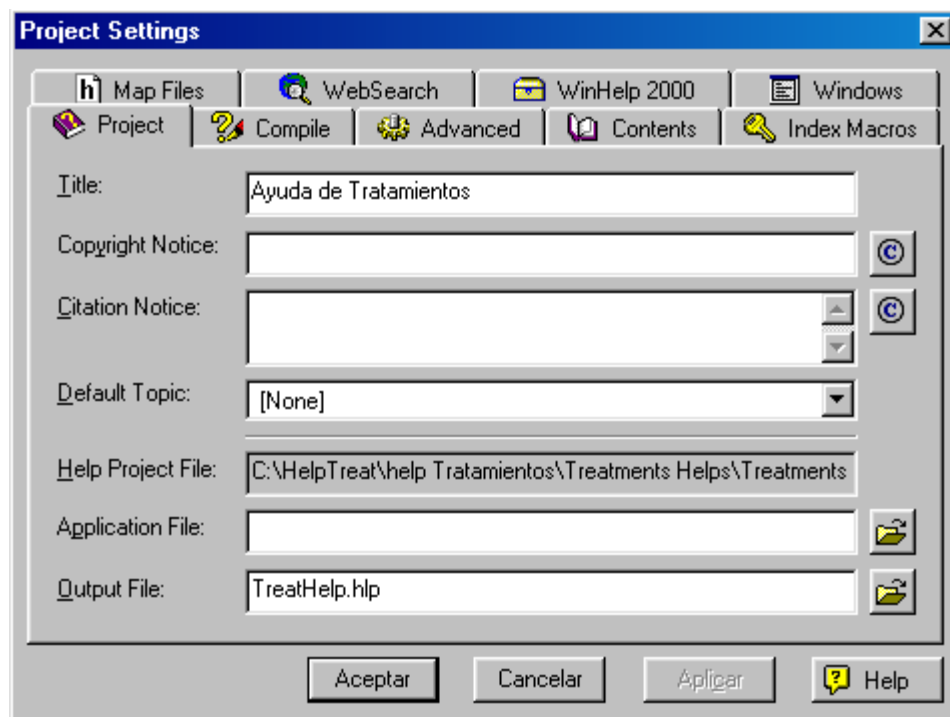
Para crear las pantallas de ayuda, RoboHelp utiliza Microsoft Word como editor de texto, de esta manera los textos incluidos se crean de manera sencilla de la misma forma en que se digita un texto cualquiera. Las páginas creadas en la aplicación se denominan tópicos y se pueden crear cuantos tópicos se estime necesario. La herramienta permite incorporar imágenes, sonidos y archivos de multimedia se así se desea.



**Figura N° 37: Interacción de RoboHelp Office 9.0 con Microsoft Word 6.0**

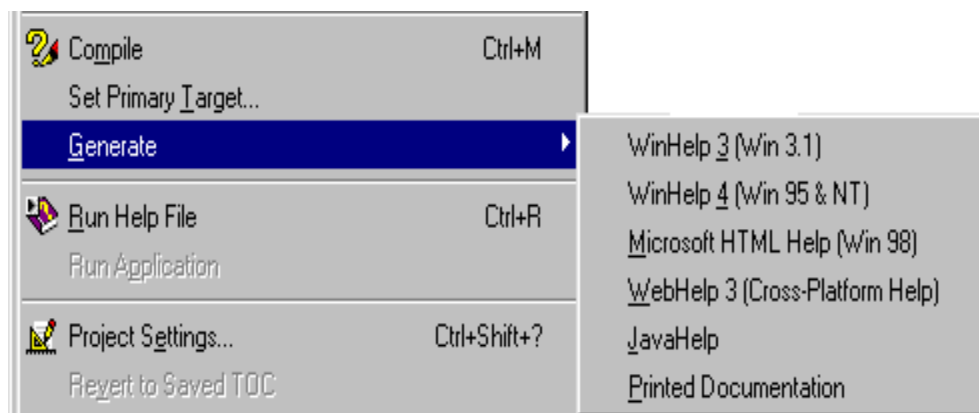


Una vez que se tiene creada la aplicación se definen algunas restricciones de búsqueda y la organización de la información mediante la creación de claves de búsqueda e índices. En el menú de “Archivo” en la opción “Project Settings” se configura cada una de las restricciones, así como el orden de búsqueda de la información, contenido, etc., además de definir el formato de compilación de la aplicación.



**Figura N° 38: Pantalla de configuración del proyecto “Project Setting”**

La aplicación una vez creada se puede compilar como documentos para ser utilizados en páginas Web o como archivos HLP que son los archivos de ayuda más comunes. También se pueden compilar en otros formatos existiendo las opciones para hacerlo.



**Figura N° 39: Pantalla para generar archivos de ayuda con distintas extensiones**

## **B. Documentos**

Se presentan a continuación los documentos analizados en el desarrollo del Módulo de Tratamientos.

- ✓ Veterinarian Prescription (Prescripción Veterinaria)
- ✓ Planilla Prescripción y cálculo de baños en jaulas
- ✓ Planilla Cálculo de alimento para tratamiento oral
- ✓ Planilla Cálculo de alimento para Tratamiento oral con Emamectina
- ✓ Planilla Tratamientos disponibles y tiempos de carencia requeridos
- ✓ Declaración de Garantía