

**UNIVERSIDAD POLITÉCNICA
SALESIANA**

FACULTAD DE INGENIERÍAS

SEDE QUITO-CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN INFORMÁTICA PARA LA GESTIÓN

**ANÁLISIS, DISEÑO Y DESARROLLO DE LOS MÓDULOS DE
CONTROL REPRODUCTIVO Y CONTROL SANITARIO PARA LA
FINCA "SAN ANTONIO".**

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

ALTAMIRANO PLAZARTE DIEGO MARCELO

YUNGA ILBAY FOAT SALVADOR

DIRECTOR: INGENIERO RENÉ ARÉVALO

Quito, Abril del 2011

DECLARACIÓN

Nosotros, Altamirano Plazarte Diego Marcelo y Yunga Ilbay Foat Salvador, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normatividad institucional vigente.

Altamirano P. Diego M.

Yunga I. F. Salvador

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por el Sr. Altamirano Plazarte Diego Marcelo y el Sr. Yunga Ilbay Foat Salvador, bajo mi dirección.

Ing. René Arévalo
Director de Tesis

AGRADECIMIENTOS

Diego:

Mi agradecimiento a Dios quien bendice mi vida y mi hogar, a mis padres quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento, a mi esposa, compañera inseparable de cada jornada. Ella representó gran esfuerzo y tesón en momentos de decline y cansancio, a mi familia que con su apoyo me sentí respaldado en lo emprendido.

Al Ing. René Arévalo quien más que un tutor, ha sido un amigo.

A la finca San Antonio, quienes confiaron en mis capacidades y honestidad.

A mi amigo Patricio, a mis profesores y a mi compañero de tesis que ayudaron a conseguir este anhelado propósito.

AGRADECIMIENTOS

Salvador:

Agradezco en primer lugar a Dios por haberme dado la vida necesaria para poder culminar este sueño, por guiarme en el camino de la superación y de la vida; a toda mi linda familia, a mis padres, a mi esposa y a mis hijos, que han sido mi apoyo permanente, a mis amigos y compañeros de aula, que me brindaron su ayuda desinteresada, a todos mis profesores que compartieron sus conocimientos y experiencias de vida, a mi amigo Diego, y su familia, con el que hemos compartido largas jornadas de trabajo; a mis amigos Luis Paredes y Pato Porras que me brindaron su amistad y conocimiento; a nuestro Director de Tesis, el Ing. René Arévalo que siempre nos dio su guía y aliento para culminar con éxito esta tarea.

DEDICATORIA

Diego:

Este trabajo lo dedico a mis padres, a mi amada esposa, a mi hermano, a mis sobrinos y abuelitos.

Su apoyo incondicional contribuyó al desarrollo y fiel cumplimiento de esta tarea que refleja el esfuerzo y dedicación puestos de manifiesto en el mismo.

DEDICATORIA

Salvador:

Dedico este trabajo, a toda mi familia, principalmente a mi esposa e hijos, pues el tiempo que debía compartir con ellos, lo dediqué a mis estudios. También dedico este trabajo a todas las personas que me apoyaron de uno u otro modo para que pueda culminar mis estudios. En resumen a todos los que creyeron en mi....."Mil gracias".

INDICE DEL CONTENIDO

CAPITULO 1		Páginas
1	Marco teórico	
1.1	Finca "San Antonio"	
1.1.1	Historia	1
1.1.1.1	Inicios	1
1.1.1.2	Visión	2
1.1.1.3	Misión	2
1.1.2	Puestos de trabajo	2
1.1.2.1	Descripción de los puestos de trabajo	2
1.1.2.2	Áreas de la finca "San Antonio"	3
1.2	Conceptos Generales	4
1.2.1	Base de Datos	4
1.2.1.1	Sistema Manejador de Base de Datos	4
1.2.1.2	Esquema, asignaciones e instancias de Base de datos	10
1.2.1.3	Problemas de Seguridad	11
1.2.1.4	Problemas de integridad	13
1.2.2	Postgresql	13
1.2.2.1	Introducción a postgresql	14
1.2.2.2	Conceptos de bases de datos relacionales	15
1.2.2.3	Tablas	16
1.2.2.4	Arquitectura de base de datos	17

1.2.3	C#	18
1.2.3.1	Conceptos de C#	19
1.2.3.2	Cualidades de C#	20
1.2.3.3	Rendimiento	22
1.2.3.4	Integración de base de datos	23
1.2.3.5	Bibliotecas incorporadas	23
1.2.3.6	Compatibilidad en el enfoque orientado a objetos	25

CAPITULO 2

2.	Planeamiento y Análisis del Sistema		
2.1	Determinar el ámbito del proyecto	27
2.2	Análisis de las necesidades	35
2.3	Requisitos de software y hardware	33
2.4	Incorporación de comentarios a las especificaciones	35

CAPITULO 3

3.	Modelo de Diseño		
3.1	Diseño del sistema	39
3.2	Diseño de objetos	40

3.3	Diseño y especificación de las clases	40
3.4	Diseño de algoritmos	40
3.5	Diseño de la arquitectura del sistema	41
3.6	Diseño de interfaces	41
3.7	Diagrama entidad relación	43
3.8	Diseño de menús y mapas de navegación	45
3.9	Diseño de ayudas	46
3.10	Métricas aplicadas al diseño	46

CAPITULO 4

PAG.

4	Desarrollo	
4.1	Revisión de especificaciones de funcionamiento	49
4.2	Desarrollo de la arquitectura del sistema	50
4.3	Desarrollo del código fuente	59
4.4	Pruebas	60
4.5	Manual Técnico	64
4.5.1	Descripción general	64
4.5.1.1	Alcance	64
4.5.2	Ingeniería del proyecto	64
4.5.2.1	Modelo de requisitos	64

4.5.2.1.1	Diagramas de casos de uso	65
4.5.2.1.2	Modelo de interfaces	78
4.5.2.1.3	Modelo de Dominio	78
4.5.2.1.4	Diccionario de datos	79
4.5.2.2	Modelo de análisis	92
4.5.2.2.1	Arquitectura de clases	93
4.5.2.2.2	Diagramas de secuencia	94
4.5.2.2.3	Diccionario de clases según módulos	96
4.5.2.3	Modelo de diseño	97
4.5.2.3.1	Diseño de objetos	97
4.5.2.3.2	Diseño y especificación de clases	98
4.5.2.3.3	Diseño de algoritmos	99
4.5.2.3.4	Diseño de interfaces	103
4.5.2.3.5	Diseño del sistema	105
4.5.2.3.6	Archivo o base de datos	106
4.5.2.3.7	Diseño de menús y mapa de navegación	108
4.5.2.3.8	Diseño de ayudas	109
4.5.2.3.9	Diseño de la arquitectura del sistema	110
4.5.2.4	Codificación y programación	113
4.5.3	Diagrama global del sistema	113
4.5.3.1	Arquitectura del sistema	113
4.5.3.2	Diagrama de componentes	116
4.5.3.3	Diagramas de distribución	117

4.5.4	Instrucciones de instalación, ejecución , reinicio y desinstalación	118
4.5.4.1	Requisitos de hardware y software	119
4.5.5	Seguridad y control	119
4.5.5.1	Seguridad en el acceso a la información	119
4.5.5.2	Creación de usuarios	120
4.5.5.3	Seguridad de acceso	121
4.5.5.4	Seguridad de usuario	121
4.5.5.5	Posibles usuarios del sistema	121
4.5.5.6	Copias de seguridad	122
4.5.5.7	Lista de mensajes de error	124
4.5.6	Programas fuente	127
4.5.7	Pruebas	128
4.6	Manual de instalación	137
4.7	Manual del usuario	153

CAPITULO 5

PAG.

5	Pruebas e implementación del sistema		
5.1	Identificar anomalías	208

5.2	Modificación de código	208
5.3	Prueba de código modificado	209
5.4	Entrega de software	212
5.5	Instalación del software	212

CAPITULO 6

6.	Conclusiones y Recomendaciones		
6.1	Conclusiones	232
6.2	Recomendaciones	233
	Bibliografía	234
	Anexos	237

INDICE DE GRÁFICOS

CAPITULO 1		Páginas
N°	Nombre	
1	Sistema de base de datos	10
2	Espacio de nombres de la BCL	24
 CAPÍTULO 3		
3	Pirámide del Diseño Orientado a Objetos	38
4	Transformación de un modelo de análisis OO a un modelo de diseño OO	39
5	Diagrama entidad-relación del módulo Seguridad	43
6	Diagrama entidad-relación del módulo de Unidades Bovinas e Insumos	44
7	Menú principal del Sistema SISA	45
 CAPÍTULO 4		
8	Capa de presentación	51
9	Interfaces de la carpeta "Insumo".	51
10	Interfaces de la carpeta "Principal"	52
11	Interfaces de la carpeta "Reportes".	53
12	Interfaces de la carpeta "Resources".	54
13	Interfaces de la carpeta "Seguridad".	54

14	Interfaces de la carpeta "UBovina"	55
15	Capa de Negocios	56
16	Clases de la carpeta "NInsumo"	56
17	Interfaces de la carpeta "Reportes".	56
18	Clases de la carpeta "NSeguridad".	57
19	Clases de la carpeta "NUbovina".	57
20	Capa de Acceso a Datos.	58
21	Clases de la carpeta "ADInsumo"	58
22	Clases de la carpeta "ADManager"	58
23	Clases de la carpeta "ADSeguridad".	59
24	Clases de la carpeta "ADUbovina".	59

Manual Técnico

25	Diagrama de casos de uso para ingreso al sistema	65
26	Diagrama de casos de uso del módulo de Unidades Bovinas.	66
27	Diagrama de casos de uso del módulo de Insumos	67
28	Diagrama de casos de uso del módulo de Seguridad y Usuarios.	68
29	Diagrama de casos de uso del módulo de Reportes	69
30	Diagrama de casos de uso del módulo de Ayuda	70
31	Diagrama de estados de un Usuario	73

32	Diagrama de estados de un Propietario	73
33	Diagrama de estados de una Unidad Bovina	74
34	Diagrama de estados de un Insumo.	74
35	Diagrama de estados de una Enfermedad	75
36	Diagrama de estados de un Reporte	75
37	Diagrama de objetos del Módulo de Usuarios y Seguridades	76
38	Diagrama de objetos de Crear una Historia Clínica.	76
39	Diagrama de objetos de Crear un registro	77
40	Diagrama de objetos para el Control de suplementos	77
41	Diagrama de objetos de Control Insumo	78
42	Modelo de dominio.	78
43	Modelo de análisis	93
44	Diagramas de Secuencia para Ingresar al Sistema.	94
45	Diagramas de Secuencia para Crear permisos y Perfiles.	95
46	Diagramas de Secuencia para Crear Usuarios	95
47	Diagramas de Secuencia para Editar y Eliminar usuarios	96
48	Módulos principales del Sistema SISA	96
49	Tarjeta que nos proporciona información de una	98

	clase		
50	Interfaz cuyo objetivo es recibir datos del usuario y su clave	103
51	Interfaz que es el menú principal del Sistema SISA	104
52	Interfaz principal del Módulo Insumos	104
53	Interfaz para Registro y actualización de registros	105
54	Interfaz usada como filtro de información	105
55	Diagrama de base de datos Módulo Seguridad y Usuarios	106
56	Base de Datos Módulos Unidades Bovinas e Insumos	107
57	Menú principal del Sistema SISA	108
58	Mapa navegacional	109
59	Menú principal del sistema donde se ve la opción "Ayuda" y el submenú "Sistema	110
60	Gráfico explicativo del proceso aplicado a un sistema, donde las métricas son parte del mismo	112
61	Capa de Presentación del sistema SISA	114
62	Capa de Negocios del sistema SISA	115
63	Capa de Datos del sistema SISA	116
64	Diagrama de Distribución	118
65	Secuencia de interfaces para la creación de un Usuario del sistema	121

66	Pantalla principal del pgAdmin III.	122
67	Pantalla principal del pgAdmin III.	123
68	Pantalla principal del pgAdmin III.	123
69	Pantalla para realizar Backup de la Base de Datos	124
70	Mensaje de error	124
71	Mensaje de error	124
72	Mensaje de error	125
73	Mensaje de error	125
74	Mensaje de error	125
75	Mensaje de error	125
76	Mensaje de error	126
77	Mensaje de error	126
78	Mensaje de error	126
79	Mensaje de error	126
80	Mensaje de error	127
81	Mensaje de error	127
82	Mensaje de error	127
83	Código a evaluar para Prueba de caja Blanca	129
84	Grafo de flujo	130
85	Grafo de caminos	131
86	Interfaz de ingreso al sistema	131

87	Mensaje error	132
88	Interfaz principal para la Administración de Unidades Bovinas	132
89	Interfaz principal para la Administración de Unidades Bovinas conteniendo un datagrid	133
90	Interfaz para editar un registro	133
91	Interfaz principal para administrar Usuarios	134
92	Interfaz para crear Usuarios	134

Manual de Instalación

93	Interfaz con datos para crear Usuarios	135
94	Mensaje informativo de confirmación	135
95	Mensaje informativo	135
96	Pantalla informativa de instalación de Visual Studio 2008	138
97	Pantalla de instalación de Visual Studio 2008	138
98	Pantalla de instalación de Visual Studio 2008	138
99	Pantalla de instalación de Visual Studio 2008	139
100	Pantalla de instalación de Visual Studio 2008	139
101	Pantalla de instalación de Visual Studio 2008	139
102	Pantalla de instalación de Visual Studio 2008	140
103	Pantalla de instalación de postgresQL	140
104	Pantalla de instalación de postgresQL	141

105	Pantalla de instalación de postgresQL	141
106	Pantalla de instalación de postgresQL	141
107	Pantalla de instalación de postgresQL	142
108	Pantalla de instalación de postgresQL	142
109	Pantalla de instalación de postgresQL	142
110	Pantalla final de instalación de postgresQL	143
111	Pantalla de instalación de Stack Builder	143
112	Pantalla de instalación de Stack Builder	144
113	Pantalla de instalación de Stack Builder	144
114	Pantalla de descarga	144
115	Pantalla informativa de Stack Builder	145
116	Pantalla de instalación de los paquetes que contienen el psqLODBC	145
117	Pantalla de instalación de los paquetes que contienen el psqLODBC	146
118	Pantalla de instalación de los paquetes que contienen el psqLODBC	146
119	Pantalla de instalación de los paquetes que contienen el psqLODBC	147
120	Pantalla principal del pgAdmin III	147
121	Pantalla de instalación del ODBC	148
122	Pantalla de instalación del ODBC	148
123	Pantalla de instalación del ODBC	148

124	Pantalla de instalación del ODBC	149
125	Pantalla de instalación del ODBC	149
126	Pantalla del Panel Control	150
127	Pantalla del Panel Control	150
128	Pantalla del Administrador de orígenes de datos ODBC	151
129	Pantalla para crear un nuevo origen de datos	151
130	Pantalla para crear un nuevo origen de datos	151
131	Pantalla informativa	152
132	Pantalla de Ingreso al sistema SISA	153
133	Pantalla principal del Sistema SISA	154
134	Pantalla principal en la que se muestra los submenús de la opción Inicio	156
135	Pantallas descriptivas de las opciones de "Gestión Unidad Bovina"	157
136	Pantalla de la Opción Administrar UB.	158
137	Pantalla o formulario para crear Unidad Bovina	158
138	Pantalla para filtrar información referente a Unidades bovinas	159
139	Pantalla que nos muestra en una tabla todas las Unidades Bovinas	159
140	Pantalla para editar los datos de una Unidad Bovina	160
141	Pantalla para administrar las Historias Clínicas de	161

las Unidades Bovinas

142	Pantalla para crear una nueva Historia Clínica	161
143	Pantalla para administrar la vacunación	162
144	Pantalla para crear un nuevo registro de vacunación	162
145	Pantalla de la opción Control Reproductivo	163
146	Pantalla Filtro para búsqueda por evento reproductivo	163
147	Pantalla conteniendo la tabla con las Unidades Bovinas a las que se puede registrar un evento reproductivo	164
148	Pantalla para crear el registro reproductivo de una UB	164
149	Pantalla para administrar los complementos nutricionales	165
150	Pantalla para crear un complemento nutricional	165
151	Pantalla para control del uso de los complementos nutricionales	166
152	Pantalla para crear registros del uso de complementos nutricionales	166
153	Pantalla para administrar enfermedades	167
154	Pantalla para crear el registro de una enfermedad	168
155	Pantalla conteniendo una tabla con las enfermedades registradas	168
156	Pantalla para editar datos de una enfermedad	169

157	Pantalla para administrar tratamientos para las enfermedades de las Unidades Bovinas	169
158	Pantalla para crear un tratamiento	170
159	Pantalla conteniendo una tabla con los tratamientos registrados	170
160	Pantalla para editar datos de un tratamiento	171
161	Pantalla para administrar eventos reproductivos de las Unidades Bovinas	171
162	Pantalla para crear un evento reproductivo	172
163	Pantalla conteniendo una tabla con los eventos reproductivos registrados	172
164	Pantalla para editar datos de un evento reproductivo	173
165	Pantalla para administrar tipos de madurez de las Unidades Bovinas	173
166	Pantalla para crear un tipo de madurez	174
167	Pantalla conteniendo una tabla con los tipos de madurez registrados	174
168	Pantalla para editar datos de un tipo de madurez	175
169	Pantalla para administrar el inventario de insumos	176
170	Pantalla filtro para elegir el tipo de insumo a mostrar	176
171	Pantalla que contiene una tabla donde se despliega información de los insumos	177

	inventariados		
172	Pantalla para editar la presentación, la mínima y la máxima cantidad de un insumo	177
173	Pantalla para administrar las presentaciones que pueden tener los insumos	178
174	Pantalla para crear una presentación.	178
175	Pantalla que contiene una tabla donde se despliega información de las presentaciones existentes	179
176	Pantalla para editar datos de una presentación	179
177	Pantalla para administrar los insumos	180
178	Pantalla para crear insumos	180
179	Pantalla con el registro de un insumo tipo “MEDICINAL”	181
180	Pantalla para complementar otros datos de un insumo tipo “MEDICINAL”	181
181	Pantalla filtro para elegir el tipo de insumo a mostrar	182
182	Pantalla que contiene una tabla donde se despliega información de los insumos tipo “MEDICINAL”	182
183	Pantalla para editar un insumo	183
184	Pantalla para administrar procesos	183
185	Pantalla para crear un proceso	184
186	Pantalla que contiene una tabla donde se	184

	despliega información de los procesos existentes		
187	Pantalla para editar un proceso	185
188	Pantalla para administrar Proveedores	185
189	Pantalla para crear un proveedor	186
190	Pantalla que contiene una tabla donde se despliega información de todos los proveedores	186
191	Pantalla para editar un proveedor	187
192	Pantalla para administrar las compras	187
193	Pantalla para crear una compra	188
194	Pantalla que contiene una tabla donde se despliega información de las compras registradas	188
195	Pantalla para editar una compra	189
196	Pantalla para administrar los egresos de los insumos	189
197	Pantalla para crear un egreso	190
198	Pantalla que contiene una tabla donde se despliega información de los egresos registrados	190
199	Pantalla para editar un egreso	191
200	Pantalla para administrar los insumos medicinales o medicamentos	191
201	Pantalla que contiene una tabla donde se despliega información de los medicamentos existentes	192
202	Pantalla para editar un medicamento	192

203	Pantalla para administrar suplementos	193
204	Pantalla que contiene una tabla donde se despliega información de los suplementos existentes	193
205	Pantalla para editar un suplemento	194
206	Pantalla para administrar usuarios	195
207	Pantalla para crear un nuevo usuario	195
208	Pantalla que contiene una tabla donde se despliega información de los usuarios existentes	196
209	Pantalla para editar los datos de un usuario	196
210	Pantalla informativa de los datos de un usuario	197
211	Pantalla para administrar perfiles	197
212	Pantalla para crear un perfil	198
213	Pantalla que contiene una tabla donde se despliega información de los perfiles existentes	198
214	Pantalla para editar los datos de un perfil	199
215	Pantalla para Administrar Permisos	199
216	Pantalla para crear permisos	200
217	Pantalla que contiene una tabla donde se despliega información de los suplementos existentes	200
218	Pantalla para editar permisos	201
219	Pantalla para administrar Propietarios	201
220	Pantalla para crear un Propietario	202

221	Pantalla que contiene una tabla donde se despliega información de los Propietarios	202
222	Pantalla para editar datos de un Propietario	203
223	Pantalla informativa de los datos de un Propietario	203
224	Pantalla para Crear Claves	204
225	Pantalla para Administrar el catálogo	205
226	Pantalla que muestra los registros que se tiene en el Catálogo	205
227	Pantalla para editar el Catálogo	206
228	Pantalla informativa del catálogo	206

CAPÍTULO 5

229	Interfaz para crear una Historia Clínica, anterior a la modificación del código	209
230	Interfaz de la Historia Clínica, después de modificar el código	210
231	Interfaz donde se muestra detalles del tratamiento recibido	210
232	Interfaz del módulo Reportes para Eventos Reproductivos	211
233	Interfaz que muestra un reporte de "TODOS" los eventos reproductivos dentro de un rango de tiempo, definido por el usuario	211

234	Interfaz que muestra el reporte de un evento reproductivo "CELO" dentro de un rango de tiempo, definido por el usuario	212
235	Pantalla guía previa a la creación del instalador del sistema "SISA"	213
236	Pantalla para iniciar el proceso de creación del instalador del sistema "SISA"	214
237	Pantalla del proceso de creación del instalador del sistema "SISA"	214
238	Pantalla del proceso de creación del instalador del sistema "SISA"	215
239	Pantalla del proceso de creación del instalador del sistema "SISA"	215
240	Pantalla del proceso de creación del instalador del sistema "SISA"	216
241	Pantalla del proceso de creación del instalador del sistema "SISA"	216
242	Pantalla del proceso de creación del instalador del sistema "SISA"	217
243	Pantalla del proceso de creación del instalador del sistema "SISA"	217
244	Pantalla del proceso de creación del instalador del sistema "SISA"	218
245	Pantalla del proceso de creación del instalador del sistema "SISA"	218
246	Pantalla del proceso de creación del instalador	219

	del sistema "SISA"		
247	Pantalla del proceso de creación del instalador del sistema "SISA"	220
248	Pantalla del proceso de creación del instalador del sistema "SISA".	220
249	Pantalla del proceso de creación del instalador del sistema "SISA"	221
250	Pantalla del proceso de creación del instalador del sistema "SISA"	222
251	Pantalla del proceso de creación del instalador del sistema "SISA"	223
252	Pantalla del proceso de creación del instalador del sistema "SISA"	224
253	Pantalla del proceso de creación del instalador del sistema "SISA"	224
254	Pantalla del proceso de creación del instalador del sistema "SISA"	225
255	Pantalla del proceso de creación del instalador del sistema "SISA"	226
256	Pantalla que muestra el contenido de la carpeta "Release".	227
257	Pantalla de instalación del sistema SISA	228
258	Pantalla de instalación del sistema SISA	228
259	Pantalla de instalación del sistema SISA	229
260	Pantalla de instalación del sistema SISA	229

261	Pantalla de instalación del sistema SISA	230
262	Pantalla de instalación del sistema SISA	230
263	Pantalla de instalación del sistema SISA	231

RESUMEN

El manejo adecuado de la información, permite planificar actividades, reducir gastos y mejorar procesos. Es por esto que el presente trabajo, analiza, diseña y desarrolla módulos de control reproductivo y control sanitario para la finca "San Antonio"; direccionado al área ganadera, es una herramienta tecnológica que pretende ser un aporte en el desarrollo de esta actividad económica y busca que las personas involucradas en el manejo o administración de pequeñas fincas ganaderas, valoren los beneficios que les brinda el uso adecuado de sistemas informáticos. Basados en los resultados de una encuesta realizada a personal involucrado en el tema, y mediante la observación directa en fincas ganaderas de los cantones Mejía, Cayambe y específicamente en el sector de Tandapi, se pudo determinar que existe la necesidad de disponer de un sistema informático para esta área de la economía, que direccionado a nuestra realidad pueda ser aplicado a nivel externo en economías semejantes a la nuestra.

Lo relevante de la investigación, para el desarrollo e implementación del sistema, se traduce en reducir el desperdicio de insumos, aumentar la producción de leche, mejorar el control sanitario y reproductivo de las unidades bovinas. Y para lograrlo, el sistema tiene una variedad de interfaces amigables con el usuario, sin importar el nivel de conocimiento que tenga en esta materia, además tiene un alto nivel de seguridad, para el manejo de datos y acceso al sistema.

PRESENTACIÓN

La globalización de la economía y la estructuración de bloques de mercado en la mayoría de los países, ha hecho que los diversos sectores económicos tengan que cambiar de manera acelerada hacia esquemas mucho más competitivos en los cuales se ha intensificado la competencia. Esto ha traído como consecuencia la reforma de los procesos agropecuarios en muchos países de Latinoamérica, basados en el principio de que la competencia presiona la mayor eficiencia y la búsqueda de la calidad de la materia prima y los productos, como elementos fundamentales de supervivencia de las pequeñas, medianas y grandes empresas.

La finca "San Antonio" es una pequeña empresa, sin embargo de aquello no deja de ser una unidad compleja conformada por varias áreas, entre las que tenemos la administrativa, operativa y logística, que tiene sus inicios allá por los años 60.

Actualmente se evidencia que los datos que generan las diferentes actividades y procesos de la finca, no se guardan de una manera adecuada al tiempo y a la tecnología de que disponemos en la actualidad, esto es atribuible principalmente al factor económico, ya que el costo de un sistema informático de este tipo es bastante alto, sin mencionar que la mayoría de los que se encuentra en el mercado son producidos en otros países, y a decir de los que conocen del tema, son muy complejos e inmensos para el medio nuestro, no queriendo decir que son malos ni tampoco inútiles sus prestaciones.

El sistema informático "SISA" surge de la necesidad de cubrir un vacío existente en esta área de la economía y ofrece sus prestaciones para el registro de los datos de todas las unidades bovinas, para disponer de un inventario actualizado de las mismas; además brinda un método moderno de control sanitario a través de las historias clínicas individuales, donde podemos guardar información de todas las enfermedades que tuvo la

unidad bovina, con fechas y tratamientos bien detallados, sumado a esto el sistema también permite llevar un control de los estados reproductivos, celo, preñez, partos o abortos por los que pasa cada unidad bovina en edad reproductiva, mediante el registro de los mismos con fecha de inicio y fecha fin. Siendo muy importante lo antes mencionado, aun queda algo que el sistema ofrece, y es un módulo para administrar y controlar los insumos que se utilizan en la finca, clasificándolos de acuerdo a su uso: nutricionales, medicinales y de uso general, lo que nos permite saber que hay y planificar las compras a futuro. Para complementar las bondades del sistema tenemos un módulo para generar reportes parametrizados de la información residente en la base de datos.

CAPÍTULO I

MARCO TEÓRICO

1.1 FINCA SAN ANTONIO

1.1.1 HISTORIA

1.1.1.1 INICIOS

La finca “San Antonio” tiene sus inicios en los años 60, cuando un grupo de personas abandonando sus tierras de origen fueron en busca de un mejor futuro, adentrándose en lo que hoy se conoce como la parroquia Cornejo Astorga, Cantón Mejía, Provincia de Pichincha.

La mayoría de los fundadores eran de Machachi, y muchos de ellos tenían parentescos familiares, así por ejemplo eran hermanos, primos, padres e hijos, etc. En ese tiempo se parcelaron esas tierras y fueron repartidas por un organismo gubernamental, llamado IERAC, el cual ya no existe.

Con el transcurrir de los años, y debido a las duras condiciones en las que vivían, por la falta casi absoluta de caminos de acceso y las duras condiciones de vida, muchos de los fundadores abandonaron estas tierras, vendiéndolas, en algunos casos, a sus vecinos.

El Sr. Segundo Antonio Chicaiza Jaguaco, nacido en la ciudad de Machachi, padre de 6 hijos, fue uno de los fundadores de la parroquia, el cual con su trabajo tesonero y su visión ayudó al progreso del sector. A lo largo de su vida compró algunas tierras, las cuales en la actualidad forman parte de la finca que en la actualidad lleva su nombre.

Hace 11 años falleció el Sr. Antonio Chicaiza, y han sido sus hijos en unión con la esposa de Don Antonio, quienes han sabido conservar este patrimonio familiar, llevando a cabo varias mejoras en los métodos de administración y mantenimiento de la finca, lo cual ha permitido elevar los estándares de reproducción del ganado de leche, de mantenimiento de pastizales, de conservación del bosque nativo que posee la finca y mejorar la calidad de vida del personal que labora en la finca.

1.1.1.2 VISIÓN

La finca “San Antonio”, guiada por los principios éticos y morales de sus propietarios aspira a constituirse en una finca referente en su medio, manejando sus recursos de manera sustentable, haciendo buen uso de la tecnología y respetando el medio ambiente.

1.1.1.3 MISIÓN

Proveer materia prima de excelente calidad, a costos razonables y precios competitivos. Liderar el proceso de conservación de los bosques húmedos para mantener la provisión de agua tanto para uso propio, así como para la comunidad.

1.1.2 PUESTOS DE TRABAJO

La finca “San Antonio” tiene como personal de planta.

- Un administrador
- Un Veterinario
- Un cuidador
- Un ordeñador
- Un vaquero
- Tres personas que hacen mantenimiento de los pastizales .

1.1.2.1 DESCRIPCIÓN DE LOS PUESTOS DE TRABAJO

Administrador.- Persona que se encarga de coordinar las diferentes actividades relacionadas con la finca, gestionar la compra de insumos y la venta de los productos (leche y/o carne).

Veterinario.- Es el profesional encargado de la planificación del control sanitario y medicación del ganado cuando tiene alguna enfermedad, controla y supervisa el ganado en estado de gestación.

Cuidador.- Persona que se encarga de cuidar las instalaciones físicas (casa destinada a la vivienda, bodega de insumos, etc.).

Ordeñador.- Es la persona encargada de juntar al ganado de leche, separar a los terneros antes del ordeño, mantener limpios los recipientes para almacenar y entregar la leche, ordeñar a las vacas, medir y registrar la cantidad de leche que se produce de manera diaria, entregar el producto, recoger y limpiar nuevamente los recipientes, soltar a los terneros para que se junten con sus madres y de esta manera puedan tomar el resto de leche que queda en las ubres y por último soltar el ganado en el pastizal que corresponda.

Vaquero.- Es el encargado de cuidar el ganado de carne que posee la finca, debiendo cambiarlo periódicamente de pastizal, colaborar junto con el administrador y el Médico veterinario en las tareas de desparasitación, control sanitario y vacunas.

Trabajador de pastizales.- Se dedica enteramente al mantenimiento de los pastizales, debiendo realizar todas las tareas relacionadas con ellos, tales como: limpieza de malezas, siembra de hierba y cercado de pastizales.

1.1.2.2 ÁREAS DE LA FINCA SAN ANTONIO

La finca específicamente tiene un área administrativa, una operativa y una logística.

Área administrativa.- Esta es un área muy importante para cumplir con los objetivos que se proponen alcanzar los dueños de la finca. La lidera su administrador, el cual planifica, coordina y dirige todas las actividades relacionadas con la finca, y en algunos casos también ejecuta, junto con los trabajadores, algunas tareas.

Área operativa.- Área dedicada a implementar todas las tareas que se llevan a cabo en la finca. A ella pertenecen el vaquero, los trabajadores de mantenimiento de pastizales, el veterinario, el personal que realiza las compras de insumos y medicamentos, aunque se puede incluir en esta área al administrador cuando realiza tareas de compra de ganado, medicamentos o insumos.

1.2 CONCEPTOS GENERALES

1.2.1 BASE DE DATOS

“Es una colección compartida de datos lógicamente relacionados, junto con una descripción de estos datos, que están diseñados para satisfacer las necesidades de información de una organización.”

(Connolly & Begg)

1.2.1.1 SISTEMA MANEJADOR DE BASE DE DATOS (DBMS)

Un Sistema de Gestión de Base de Datos (**SGBD**) o en inglés Database Management System (**DBMS**), consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada Base de datos, contiene información relevante para una empresa.

Su principal objetivo, es proporcionar una forma de almacenar grandes o pequeñas cantidades de información y recuperarla de una base de datos de manera que sea tanto practica, segura, integral y eficiente.

Los **SGBD** cuentan con estructuras de almacenamiento de la información y de mecanismos que ayudan a la manipulación y fiabilidad de la información, a pesar de caídas del sistema o intentos de acceso sin autorización.

Normalmente los **SGBD** proporcionan las siguientes funcionalidades.

Permite definir la base de datos mediante un lenguaje de definición de datos (**DDL, Data Definition Language**). El DDL permite a los usuarios especificar las estructuras y tipos de datos y las restricciones aplicables a los datos que hay que almacenar en la base de datos.

Permite a los usuarios insertar, actualizar, borrar y extraer datos de la base de datos, usualmente mediante un lenguaje de manipulación de datos (**DML, Data Manipulation Language**). Al disponer de un repositorio centralizado para todos los datos y las descripciones de los datos, el lenguaje DML puede proporcionar un mecanismo general de consulta de esos datos, denominado lenguaje de consulta.

La existencia de un lenguaje de consulta resuelve el problema de los sistemas basados en archivos en los que el usuario tenía que trabajar con un conjunto fijo de consultas, o bien en los que existía una proliferación de programas que provocaban graves problemas de gestión del software. El lenguaje de consulta más común es el lenguaje SQL (**Structured Query Language**, lenguaje estructurado de consulta), que es ahora tanto el estándar formal como el estándar de facto para los SGBD relacionales.

Proporciona un acceso controlado a la base de datos entre los cuales se tiene:

- **Seguridad**, que evita que los usuarios no autorizados accedan a la base de datos.
- **Integridad**, que mantiene la coherencia de los datos almacenados.
- **Control de concurrencia** que permite el acceso compartido a la base de datos.
- **Control de recuperación**, que restaura la base de datos a un estado previo coherente después de cada fallo hardware o software.
- **Catálogo accesible por el usuario**, que contiene descripciones de los datos que están almacenados en la base de datos.

Ventajas y Desventajas de los SGBD

Los sistemas de gestión de bases de datos presentan enormes ventajas potenciales, Desafortunadamente, también tiene algunas desventajas.

Ventajas

- **Control de la redundancia de los datos**, para evitar como los sistemas basados en archivos el desperdicio de espacio al almacenar la misma información en más de un archivo.
- **Coherencia de los datos**, al eliminar o controlar la redundancia, reducimos el riesgo de que se produzcan incoherencias. Si un elemento de datos sólo se almacena una vez en la base de datos, las actualizaciones

de su valor solo tienen que llevarse a cabo una vez y el nuevo valor estará disponible de forma inmediata para todos los usuarios. Si almacenamos un elemento más de una vez y el sistema es consciente de esto, el sistema podrá garantizar que todas las copias del elemento sean coherentes.

- **Más información a partir de la misma cantidad de datos**, al integrar los datos operacionales, la información puede deducir información adicional a partir del conjunto de datos existente.
- **Compartición de los datos**, normalmente, los archivos son propiedad de las personas o departamentos que los usan. Por otro lado, la base de datos pertenece a toda la organización y debe ser compartida por todos los usuarios autorizados. De este modo, un número mayor de usuarios puede compartir una mayor cantidad de datos.
- **Mayor integridad en los datos**, el concepto de integridad de la base de datos hace referencia a la validez y coherencia de los datos almacenados. La integridad se suelen expresar en términos de restricciones, que son reglas de coherencia que no se permite que la base de datos viole.
- **Mayor seguridad**, la seguridad de la base de datos es la protección de los datos frente a su uso por personas no autorizadas. Sin unas medidas de seguridad adecuadas, la integración hace que los datos sean más vulnerables que en los sistemas basados en archivos.
- **Imposición de estándares**, de nuevo, la integración permite al DBA definir e imponer los estándares necesarios. Puede tratarse de estándares departamentales, de la organización, nacionales o internacionales referidos a cosas tales como los formatos de datos necesarios para facilitar el intercambio de datos entre sistemas, los convenios de denominación, los estándares de documentación, los procedimientos de actualización y las reglas de acceso.
- **Economía de escala**, al combinar todos los datos operacionales de una organización en una única base de datos y crear un conjunto de aplicaciones que funcionan con esta fuente centralizada de datos, puede reducirse enormemente los costes.
- **Equilibrio entre los requisitos conflictivos**, cada usuario de departamentos tiene necesidades que pueden entrar en conflicto con las

de otros usuarios. Puesto que la base de datos está bajo control del DBA, este puede tomar decisiones acerca del diseño y la utilización operacional de la base de datos que proporcionen el mejor uso de los recursos, considerando a la organización como un todo.

- **Mejor accesibilidad de los datos y mayor capacidad de respuesta**, de nuevo, como el resultado de la integración, los datos que atraviesan las fronteras departamentales son accesibles de modo directo por los usuarios finales.
- **Mayor productividad**, el SGBD proporciona muchas de las funciones estándar que el programador tendría normalmente que incluir dentro de su aplicación basada en archivos. A un nivel básico, el SGBD proporciona todas las rutinas de bajo nivel para gestión de archivos que se utilizan normalmente en los programas de aplicación.
- **Mantenimiento simplificado gracias a la independencia de datos**, en los sistemas basados en archivos, las descripciones de los datos y la lógica para acceder a los datos están integradas en cada programa de aplicación, haciendo que los programas sean dependientes de los datos.
- **Mayor nivel de concurrencia**, en algunos sistemas basados en archivos, si se permite a dos o más usuarios acceder al mismo archivo simultáneamente, es posible que los accesos se interfieran entre sí, provocando una pérdida de información o incluso de la integridad de los datos. Muchos SGBD se encargan de gestionar el acceso concurrente a la base de datos y garantizan que dichos problemas no puedan presentarse.
- **Servicios mejorados de copia de seguridad y recuperación**, muchos sistemas basados en archivos asignan a usuarios la responsabilidad de proporcionar medidas para proteger los datos frente a fallas de programas o de sistemas de aplicación. Esto puede requerir realizar copias de seguridad de los datos diario, durante las horas nocturnas. En caso de que se produzca un fallo al día siguiente, se restaura la copia de seguridad y el trabajo que se haya realizado desde que se hizo la copia de seguridad se pierde, los SGBD modernos proporcionan facilidades para minimizar la cantidad de procesamiento que se pierde debido a un fallo.

Desventajas

- **Complejidad**, para que un buen SGBD pueda proporcionar la funcionalidad esperada, el SGBD tiene que ser un programa software de gran complejidad. Los desarrolladores y diseñadores de base de datos, los administradores de datos y de base de datos y los usuarios finales deben ser capaces de comprender esta funcionalidad. Si no se comprende adecuadamente el sistema, pueden tomarse decisiones de diseño incorrectas, lo que podría tener serias consecuencias para la organización.
- **Tamaño**, la complejidad y el amplio rango de funcionalidades hacen que el SGBD sea un programa software de gran tamaño que ocupa muchos megabytes de espacio de disco y requiere una cantidad de memoria importante para poder ejecutarse de manera eficiente.
- **Coste del SGBD**, varía significativamente, dependiendo del entorno y de la funcionalidad proporcionada, dependerá del tamaño de la organización y si el SGBD es monousuario o multiusuario.
- **Coste del hardware adicional**, los requisitos de almacenamiento en disco para SGBD y de la base de datos pueden imponer la compra de espacio de almacenamiento adicional.
- **Costes de conversión**, en algunas situaciones, el coste del SGBD y del hardware adicional pueden ser insignificantes y lo comparamos con el coste de convertir las aplicaciones existentes para que se ejecuten sobre el nuevo SGBD y sobre la nueva plataforma hardware, este coste también incluye el coste de formar al personal en la utilización de los nuevos sistemas y, posiblemente, la contratación de personal especializado con ayuda durante la conversión y la operación del sistema.
- **Mayor impacto de los fallos**, la centralización de los recursos implementa la vulnerabilidad del sistema puesto que todos los usuarios y aplicaciones dependen de la disponibilidad del SGBD, el fallo de ciertos componentes pueden hacer que se detengan todas las operaciones.

Clasificación de los SGBD

Esta clasificación está basada en el modelo de datos en que está basado el SGBD.

Los modelos de datos más habituales son:

- **Relacional (SGBDR).**- Representa a la base de datos como una colección de tablas. Estas bases de datos suelen utilizar SQL como lenguaje de consultas de alto nivel.
- **Orientado a objetos.**- Define a la base de datos en términos de objetos, sus propiedades y sus operaciones. Todos los objetos que tienen la misma estructura y comportamiento pertenecen a una clase y las clases se organizan en jerarquías.
- **Objeto-relacional o relacional extendido.**- Son los sistemas relacionales con características de los orientado a objetos.
- **Jerárquico.**- Representa los datos como estructuras jerárquicas de árbol.
- **En red o CODASYL DBTG.**

Un SGBD también puede clasificarse por el número de usuario a los que da servicio:

- **Mono usuario**
- **Multiusuario**

También puede clasificarse según el número de sitios en los que está distribuida la base de datos:

- **Centralizado.**- La base de datos y el software SGBD están almacenados en un solo sitio (una sola computadora).
- **Distribuido (SGBDD).**- La base de datos y el software SGBD pueden estar distribuidos en múltiples sitios conectados por una red.

Estructura de un DBMS

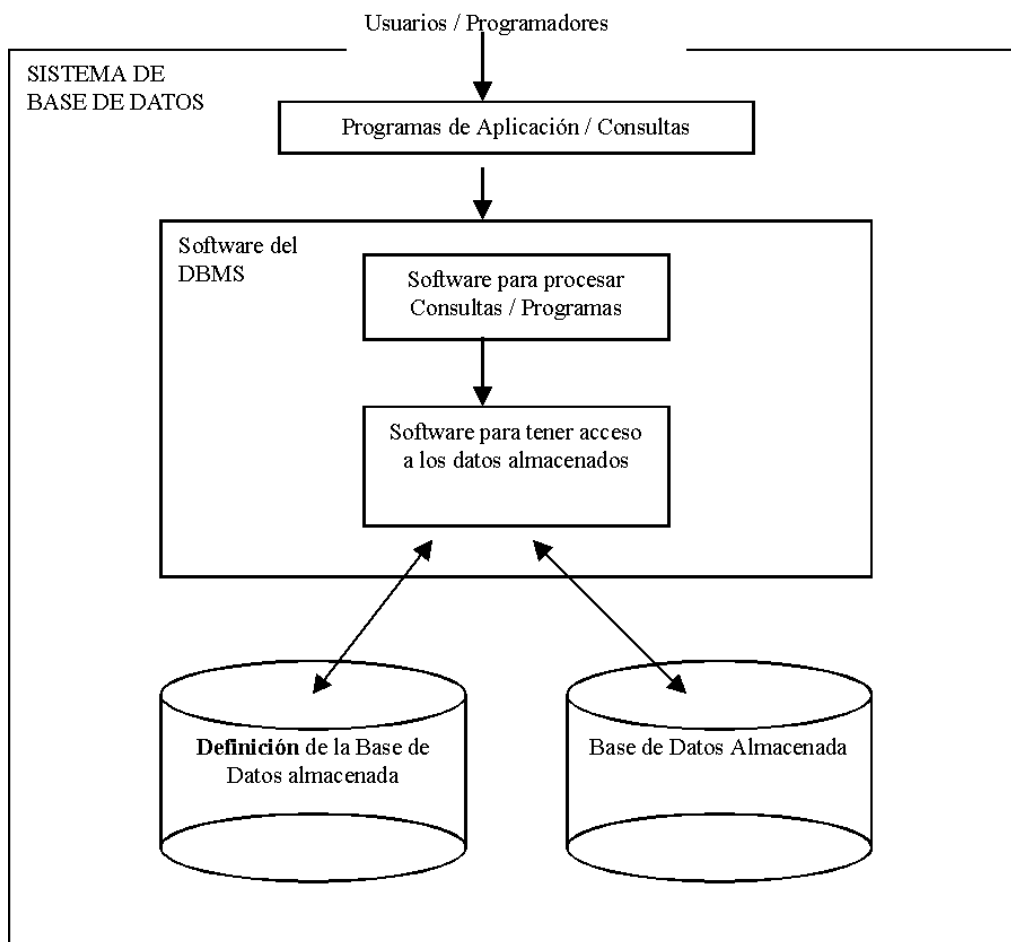


Figura 1. Sistema de Bases de Datos

Fuente: <http://www.unalmed.edu.co/~mstabare/DBMS.gif>

1.2.1.2 ESQUEMA, ASIGNACIONES E INSTANCIAS DE BASE DE DATOS

La descripción global de la base de datos se denomina esquema de la base de datos. Existen tres tipos diferentes de esquemas en la base de datos y dichos tipos se clasifican de acuerdo con los niveles de abstracción de la arquitectura en tres niveles. En el nivel más alto, disponemos de múltiples **niveles externos**, también, denominados sub esquemas, que corresponden con las diferentes vistas de los datos. En el **nivel conceptual**, se tiene el esquema conceptual que describe todas las entidades, atributos y relaciones, junto con las restricciones de integridad aplicables. En el nivel más bajo de abstracción, se tiene el **esquema interno** que es una descripción completa del modelo interno en la que se incluye

las definiciones de los registros almacenados, los métodos de representación, los centros de los datos y de los índices y estructuras de almacenamiento utilizados. Sólo existe un esquema conceptual y un esquema interno en cada base de datos.

El SGBD es responsable de establecer la correspondencia entre estos tres tipos de esquema. También debe comprobarse la coherencia de los esquemas, del SGBD, por lo tanto este se encarga de comprobar que cada esquema externo pueda derivarse a partir del esquema conceptual, y debe emplear la información del esquema conceptual para establecer la correspondencia entre cada esquema externo y el esquema interno. El esquema conceptual se relaciona con el esquema interno mediante una correspondencia conceptual/interno. Esto permite al SGBD localizar el registro o combinación de registros dentro del almacenamiento físico que constituye un registro lógico en el esquema conceptual, junto con cualquier restricción que haya que aplicar a las operaciones que se realicen sobre dicho registro lógico. También permite resolver las diferencias que existan entre los nombres de entidad, nombres de atributo, orden de los atributos, tipos de datos, etc. Finalmente cada esquema externo se relaciona con el esquema conceptual mediante la asignación externo/conceptual. Esto permite al SGBD establecer la correspondencia entre los nombres de la vista de los usuarios y las partes correspondientes del esquema conceptual.

1.2.1.3 PROBLEMAS DE SEGURIDAD

La información que es almacenada en una base de datos, es muy valiosa y debe ser tratada con la mayor responsabilidad por ende no todos los usuarios de un sistema de base de datos deberían poder acceder a todos los datos. Por ejemplo, en el sistema al vaquero no podrá hacer cambios en el control de vacunas. Como los programas de aplicación se añaden al sistema de una forma ad hoc, es difícil garantizar tales restricciones de seguridad.

El SGBD facilita normalmente mecanismos para prevenir los fallos (subsistema de control), para detectarlos una vez que se han producido (subsistema de detección) y para corregirlos después de haber sido detectados (subsistema de recuperación).

Aspectos fundamentales de la seguridad:

- **Confidencialidad.-** No desvelar datos a usuarios no autorizados. Comprende también la privacidad (protección de datos personales). Se cumple cuando solo las personas autorizadas (en su sentido amplio podríamos referirnos también a sistemas) pueden conocer los datos o la información correspondiente.
- **Accesibilidad.-** La información debe estar disponible.
- **Integridad.-** Permite asegurar que los datos no han sido falseados. Consiste en que sólo las personas autorizadas puedan variar (modificar o borrar) los datos. Además deben quedar pistas para control posterior y para auditoría.

La seguridad en las bases de datos abarca varios temas:

- Cuestiones éticas y legales relativas al derecho a tener acceso a cierta información.
- Cuestiones de política en el nivel gubernamental, institucional o corporativo relacionadas con la información que no debe estar disponible para el público.
- Cuestiones relacionadas con el sistema.
- Necesidad en algunas organizaciones de identificar múltiples niveles de seguridad y de clasificar los datos y los usuarios según estos niveles.

Mecanismos de seguridad:

- **Discrecionales**, se usan para otorgar privilegios a los usuarios.
- **Obligatorios**, sirven para imponer seguridad de múltiples niveles clasificando los datos y los usuarios en varias clases de seguridad e implementando después la política de seguridad apropiada de la organización.
- **Cifrado de datos**, que sirve para proteger datos confidenciales que se transmiten por satélite o algún tipo de red de comunicaciones. Asimismo el

cifrado puede proveer protección adicional a secciones confidenciales de una base de datos. Los datos se codifican mediante algún algoritmo de codificación. Un usuario no autorizado tendrá problemas para descifrar los datos codificados, pero un usuario autorizado contará con algoritmos para descifrarlos.

1.2.1.4 PROBLEMAS DE INTEGRIDAD

Los valores de los datos almacenados en la base de datos deben satisfacer ciertos tipos de restricciones de consistencia.

Las restricciones de integridad también contribuyen a mantener un sistema de base de datos seguro, impidiendo que los datos lleguen a ser inválidos y que puedan conducir a resultados erróneos o susceptibles de ser mal interpretados.

Por ejemplo, la vacuna de una res no puede ser antes de su fecha de nacimiento. Se debe hacer cumplir la restricción en el sistema añadiendo el código apropiado en los diversos programas de aplicación. Sin embargo, cuando se añaden nuevas restricciones, es difícil cambiar los programas para hacer que se cumplan. El problema es complicado cuando las restricciones implican diferentes elementos de datos de diferentes archivos.

1.2.2 POSTGRESQL

Es un sistema de gestión de base de datos relacional orientada a objetos, con más de 15 años de vida, con una larga evolución, la cual se inicia en 1982, en la Universidad de Berkeley con el proyecto Ingres liderado por Michael Stonebraker. Este proyecto fue el primer intento por implementar un motor de base de datos relacional luego de lo cual se lo uso como base para otro proyecto.

En el año de 1986 este proyecto es llamado **post-ingres** o simplemente **Postgres**, por lo que podríamos decir que la implementación de POSTGRES DBMS empieza en 1986, cabe señalar que Postgres usó muchas ideas de Ingres pero no su código. Los conceptos iniciales para el sistema fueron presentados en “*El diseño de Postgres*”, y la definición del modelo de datos inicial apareció en “*El*

modelo de datos POSTGRES". El diseño de las reglas del sistema fue descrito en "*El diseño de las reglas del sistema POSTGRES*". Por último lo relacional y la arquitectura del administrador de almacenaje fueron detallados en "*El diseño del sistema de almacenaje de POSTGRES*".

Desde entonces Postgres ha ido avanzando periódicamente y es así que el primer *demo* operacional da inicio en el año de 1987 y al año siguiente aparece la versión utilizable, la cual podríamos decir que es la Versión 1; la Versión 2 aparece en Junio de 1990 con un nuevo sistema de reglas. La Versión 3 apareció en 1991 y añadió soporte para múltiples administradores de almacenamiento, y mejoró el ejecutor de peticiones.

En la actualidad existe la Versión 4.2.

En el año de 1994, Andrew Yu y Jolly Chen añadieron el lenguaje interprete SQL a POSTGRES. Cambiando en 1995 su nombre a Postgres95, que luego derivaría en PostgreSQL. El código de Postgres95 fue completamente ANSI C y reducido en un 25%, lo que hizo que fuera entre un 30 y un 50 % más rápido. En el año de 1996 PostgreSQL entra a formar parte del Open Source, iniciando con la versión 6.0.

En el 2000, se comienza a implementar el soporte para Ipv6. La versión 8.0 en el año 2004 es considerada en los Estados Unidos de Norteamérica como la 5º DBMS más popular, siendo al año siguiente sometida al test de Coverty Inspected, teniendo como resultado que se le detectó únicamente 20 errores en 775,000 líneas de código. Al año siguiente apareció la Versión 8.1.4

1.2.2.1 INTRODUCCIÓN A POSTGRESQL

PostgreSQL está bajo licencia BSD (Berkeley Software Distribution) con lo cual tiene menos restricciones en comparación con otras licencias tal como la GPL. Al contrario que la GPL, BSD permite el uso del código fuente en software no libre, pero además puedo usar PostgreSQL para fines comerciales de cualquier tipo, puedo revenderlo, cambiarle el nombre, es decir hacer cualquier cosa.

Entre las características principales se tiene las siguientes:

Atomicidad.- Significa que una operación se realiza o no se realiza, por lo que ante un fallo del sistema la operación no se queda a medias.

Consistencia.- Esta es otra propiedad importante y nos asegura que solo se empieza aquello que se puede terminar, realizando operaciones que no van a romper las reglas y directrices de integridad de la base de datos.

Aislamiento.- Esta propiedad nos asegura que una operación no puede afectar a otras, es decir que dos transacciones sobre la misma información de la base de datos nunca generarán ningún tipo de error.

Durabilidad.- Consiste en que una vez que realizamos una operación, ésta se mantendrá y no se podrá deshacer aunque falle el sistema.

También se puede decir que PostgreSQL corre en casi todos los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc.

Su documentación está muy bien organizada, es pública y libre, con comentarios de los propios usuarios.

Tiene comunidades muy activas y algunas en español y es altamente adaptable a las necesidades del cliente.

Soporte nativo para los lenguajes más populares del medio: PHP, C, C+, Perl, Python, etc.

Drivers: Odbc, Jdbc, .Net, etc.

Tiene soporte para todas las características de una base de datos profesional, tales como: triggers, store procedures, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, etc.

1.2.2.2 CONCEPTOS DE BASES DE DATOS RELACIONALES

Considero importante, en primer lugar, definir lo que es una base de datos en términos generales. Una base de datos es “un conjunto de datos estructurados, que son fiables coherentes y compartidos por usuarios con necesidades de información diferentes”, la base datos constituye la imagen de una organización.

Una base de datos relacional es un sistema de almacenamiento de datos visibles al usuario, basado en un conjunto de tablas unidas mediante relaciones.

Los datos se almacenan en un depósito común, independientemente de los programas que deben tratarlos.

1.2.2.3 TABLAS

Anteriormente al definir la base de datos y posteriormente la base de datos relacional, se mencionó las tablas, ahora se describe las tablas de manera más detallada.

Hay que empezar diciendo que, una tabla es un objeto de la base de datos que almacena datos en filas y columnas.

El modelo relacional ofrece una manera única de representar los datos, esta es mediante una tabla bidimensional que se denomina *relación*.

Atributos

Los atributos de una relación sirven para dar nombres a las columnas de la relación, que por lo regular describen el significado de las entradas de la columna situada debajo de ella.

Esquemas

El nombre de una relación (tabla) y el conjunto de sus atributos reciben el nombre de *esquema* de la relación. El *esquema* de la relación se denota con el nombre de ésta seguido de una lista de sus atributos entre paréntesis. Por ejemplo:

Unidad_Bovina(código, número, nombre, sexo, foto).

Tuplas

A los renglones de una relación, si no son el renglón del encabezado que contiene los atributos, se les da el nombre de *tuplas*. Una tupla tiene un componente para cada uno de los atributos de la relación.

Las relaciones son conjuntos de tuplas y no es posible que una tupla aparezca más de una vez en la relación. Por lo tanto si se quiere que una relación represente una clase de objetos, hay que asegurarse que tenga un conjunto suficiente de atributos, de manera que dos objetos no puedan tener los mismos valores en todos los atributos de la relación.

Dominios

El modelo relacional exige que los componentes de una tupla sean atómicos, o sea que pertenezcan a algún tipo elemental tal como enteros o cadenas de caracteres. No se admite que un valor sea una estructura de registro, un conjunto, una lista, un arreglo o cualquier otro tipo de dato que pueda tener sus valores divididos en componentes más pequeños.

Representaciones equivalentes de una relación.

Anteriormente se pudo ver que tanto el esquema como las tuplas de una relación son conjuntos, no listas, razón por la cual no importa el orden en que sean representadas. Es más, se puede reordenar los atributos de la relación como se quiera, sin cambiar con ello la relación, lo que supone que si se cambia el orden de los atributos, también se cambia el orden de sus columnas.

Instancias de relación

Las relaciones cambian con el tiempo, lo que incluye aumento o disminución de tuplas, o cambios en la tuplas actuales.

Es menos frecuente que cambie el esquema de la relación. Pero pueden darse esas situaciones donde tal vez se quiera agregar o eliminar atributos. Los cambios de esquema, aunque posibles en los sistemas comerciales de bases de datos, son muy costosos porque es necesario reescribir cada una de los quizá millones de tuplas para agregar o suprimir componentes.

1.2.2.4 ARQUITECTURA DE BASE DE DATOS

La arquitectura de la base de datos permite que los usuarios, a pesar de que no estén familiarizados con los sistemas informáticos ni con las computadoras,

puedan interactuar de manera sencilla con los diferentes tipos de datos que maneje un sistema. Para ello, las personas encargadas de desarrollarlos, en cierta forma, ocultan la complejidad que tienen las bases de datos, mediante aplicaciones a través de las cuales van a poder manipular los datos.

La arquitectura en sí, tiene tres niveles:

- Físico o interno
- Lógico o conceptual
- Externo o de vistas

Nivel Físico.- En realidad se refiere a donde y como están almacenados los datos. A este nivel se diseñan los archivos que contienen la información, su ubicación y su organización, además se define las rutas de acceso para la base de datos.

Nivel lógico o conceptual.- Se llama nivel lógico porque trabaja con elementos de este tipo tales como: entidades, atributos y relaciones. Llamado también nivel conceptual porque describe la estructura de la base de datos ocultando los detalles físicos de almacenamiento.

Nivel externo o de vistas.- Por medio de este nivel, el usuario puede tener acceso a los datos de acuerdo a la aplicación que esté usando, mientras que el resto de la información está oculta.

Esta arquitectura de base de datos de 3 niveles fue aprobado por la ANSI-SPARC¹ en el año de 1975.

Fuente: <http://www.devjoker.com/contenidos/catss/125/Introduccion-a-C.aspx>

1.2.3 C#

C# es un lenguaje de programación de propósito general que fue diseñado por Microsoft para incorporarse en su plataforma de .NET, esta herramienta de programación ha sido desarrollada de tal forma que la programación sea más

¹ American National Standard Institute-Standards Planning and Requirements Committee

sencilla e intuitiva ya que C# carece de elementos heredados innecesariamente en .NET entre los principales creadores de esta herramienta se tiene a Scott Wiltamuth y Anders Hejlsberg.

C# toma las mejores características de los lenguajes de programación como Visual Basic, Java, C++, por lo que la sintaxis y estructuración de C# es similar a la de C++ o Java, esto se debe a que Microsoft quería facilitar a los desarrolladores, el migrar el código escritos en estos lenguajes y crear aplicaciones de forma más sencilla y de un alto nivel de productividad.

1.2.3.1 CONCEPTOS DE C#

Al hablar de C# se debe empezar por conocer .NET Framework que es el corazón de la tecnología .NET. Es el marco de trabajo y ejecución común a toda la tecnología .NET. Es por lo tanto un elemento indispensable dentro de la tecnología .NET.

El .NET Framework es un componente de software que se instala en el sistema operativo.

Dentro del .NET framework están integrados los lenguajes .NET (C#, VB.NET, J#), el CLR, el CLS, ADO.NET, ASP.NET.

.NET soporta varios lenguajes de programación, siendo los más populares:

- C#
- VB.NET
- C++ controlado
- J#
- Delphi

Todos ellos trabajan perfectamente integrados con el .NET framework, siendo C# el lenguaje principal, ya que es el único que ha sido desarrollado específicamente para .NET. Gran parte de la biblioteca de clases que componen el framework están escritas en C#.

C# es un lenguaje de programación visual, controlado por eventos, en el cual se crean programas mediante el uso de un Entorno de Desarrollo Integrado (IDE).

C# está orientado a objetos y contiene una poderosa biblioteca de clases, que consta de componentes pre construidos que permite desarrollar aplicaciones con rapidez.

El lenguaje de programación C# original se estandarizó a través de Ecma International en diciembre del 2002, como Estándar ECMA-334: Especificación del lenguaje C#. Desde entonces, Microsoft propuso varias extensiones del lenguaje que se han adoptado como parte del estándar Ecma C# revisado. Microsoft hace referencia al lenguaje C# completo incluyendo las extensiones adoptadas como C# 2.0.

Fuente: <http://msdn.microsoft.com/library/aa287997.aspx>

1.2.3.2 CUALIDADES DE C#

- C# es un eficaz lenguaje de programación orientado a objetos y con seguridad de tipos.
- Compilador de C# completamente compatible con ANSI que incorpora mejoras de lenguaje a la versión 3.0 de C#.
- Entorno de desarrollo integrado (IDE) con IntelliSense™ que facilita la creación y generación de sus aplicaciones.
- Depurador eficaz para buscar y corregir errores.
- Compatibilidad completa con formularios Windows Forms y controles que facilitan la creación de interfaces de usuario completas para las aplicaciones.
- Compatibilidad para crear aplicaciones de Windows Presentation Foundation (WPF).
- Compatibilidad con bases de datos para aplicaciones de datos locales.
- Compatibilidad con Language-Integrated Query (LINQ) que habilita funciones de consulta de datos.
- Capacidad de consumir servicios Windows Communication Foundation (WCF).

- Implementación ClickOnce para simplificar el proceso de compartir el trabajo finalizado.

Características de C#

Entre las principales características se tiene

- **Sencillez:** C# elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET
- **Modernidad:** C# incorpora en el propio lenguaje elementos que a lo largo de los años ha ido demostrándose son muy útiles para el desarrollo de aplicaciones por ejemplo las variables decimal, string y bool.
- **Orientación a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas.
- **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura.
- **Extensibilidad de tipos básicos:** C# permite definir, a través de **estructuras**, tipos de datos para los que se apliquen las mismas optimizaciones que para los tipos de datos básicos.
- **Versionable:** C# incluye una **política de versionado** que permite crear nuevas versiones de tipos sin temor a que la introducción de nuevos miembros provoquen errores difíciles de detectar en tipos hijos previamente desarrollados y ya extendidos con miembros de igual nombre a los recién introducidos.
- **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo, y a diferencia de Java, en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros.

1.2.3.3 RENDIMIENTO

El término rendimiento suele hacer referencia a la velocidad de ejecución de un programa. En ocasiones, es posible incrementar la velocidad de ejecución si se siguen algunas reglas básicas en el código fuente. En algunos programas, es importante examinar el código con detenimiento y usar generadores de perfiles para asegurarse de que se está ejecutando a la mayor velocidad posible. En otros programas no es necesario realizar esta optimización, ya que el código se ejecuta a una velocidad aceptable según se escribe.

Al medir y optimizar el rendimiento, se deben seguir estas instrucciones generales:

Se comienza por establecer objetivos de rendimiento y medir el rendimiento del programa a fin de determinar en qué casos no cumple dichos objetivos el código.

Se debe escribir el código de forma que sea correcta. Realice optimizaciones más adelante solamente si determina que éste no cumple los objetivos de rendimiento. El código optimizado para un mejor rendimiento suele ser más difícil de leer y mantener. En general, es mejor escribir código que sea legible, sólido y fácil de mantener aunque sea algo más lento que el código más optimizado que podría escribir.

Al optimizar, se debe empezar por las partes más lentas del programa. Si se determina que un programa no cumple los objetivos de rendimiento, hay que identificar las ubicaciones específicas en las que éste se puede mejorar y los problemas de rendimiento que son las causas principales del error. No suele tener sentido optimizar un método al que se llama en pocas ocasiones, ni optimizar un método "A" para que se ejecute en menos de 50 milisegundos si el programa en su totalidad tiene que esperar 30 segundos para que el método "B" se ejecute en otro subproceso.

Fuente: [http://msdn.microsoft.com/es-es/library/ms173196\(VS.90\).aspx](http://msdn.microsoft.com/es-es/library/ms173196(VS.90).aspx)

1.2.3.4 INTEGRACIÓN DE BASE DE DATOS

Un lenguaje de programación se conecta e interactúa con una base de datos relacionada a través de una interfaz de base de datos: un software que facilita la comunicación entre un sistema de administración de base de datos y un programa. Los programas en C# se comunican con las bases de datos y manipulan sus datos a través de ADO.NET.

C# dispone de herramientas que permiten trabajar con la parte visual del IDE, que simplifican el proceso de conectarse y de manipular una base de datos.

Otra forma para integrar una base de datos a la aplicación es la utilización de un ODBC (Open Data Base Connectivity Standard, Estándar de conectividad abierta de Base de datos) que permite establecer una conexión a cualquier base de datos que suministre un controlador de ODBC. Entre ellas, se incluyen los productos de Microsoft, y de productos de software libre como por ejemplo Mysql y Postgres entre otros.

1.2.3.5 BIBLIOTECAS INCORPORADAS

La Librería de Clase Base (BCL) es una librería incluida en el .NET Framework formada por cientos de tipos de datos que permiten acceder a los servicios ofrecidos por el CLR (Common Language Runtime) es el núcleo de la plataforma .NET. Es el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad.

Con las clases prefabricadas el programador puede crear nuevas clases que mediante herencia extiendan su funcionalidad y se integren a la perfección con el resto de clases de la BCL. Por ejemplo, implementando ciertas interfaces se puede crear nuevos tipos de colecciones que serán tratadas exactamente igual que cualquiera de las colecciones incluidas en la BCL.

Esta librería está escrita en MSIL, por lo que puede usarse desde cualquier lenguaje cuyo compilador genere MSIL. A través de las clases suministradas en ella es posible desarrollar cualquier tipo de aplicación, desde las tradicionales

aplicaciones de ventanas, consola o servicio de Windows NT hasta los novedosos servicios Web y páginas ASP.NET.

C# solo se basa en el BCL ya que es tal la riqueza de servicios que ofrece que incluso es posible crear lenguajes que carezcan de librería de clases propia.

Dada la amplitud de la BCL, ha sido necesario organizar las clases en ella incluida en **espacios de nombres** que agrupen clases con funcionalidades similares. Por ejemplo, los espacios de nombres más usados son:

Espacio de nombres	Utilidad de los tipos de datos que contiene
System	Tipos muy frecuentemente usados, como los tipos básicos, tablas, excepciones, fechas, números aleatorios, recolector de basura, entrada/salida en consola, etc.
System.Collections	Colecciones de datos de uso común como pilas, colas, listas, diccionarios, etc.
System.Data	Manipulación de bases de datos. Forman la denominada arquitectura ADO.NET .
System.IO	Manipulación de ficheros y otros flujos de datos.
System.Net	Realización de comunicaciones en red.
System.Reflection	Acceso a los metadatos que acompañan a los módulos de código.
System.Runtime.Remoting	Acceso a objetos remotos.
System.Security	Acceso a la política de seguridad en que se basa el CLR.
System.Threading	Manipulación de hilos.
System.Web.UI.WebControls	Creación de interfaces de usuario basadas en ventanas para aplicaciones Web.
System.Windows.Forms	Creación de interfaces de usuario basadas en ventanas para aplicaciones estándar.
System.XML	Acceso a datos en formato XML.

Figura 2: Espacios de nombres de la BCL más usados

Fuente: <http://www.devjoker.com/contenidos/Conceptos-generales-NET/93/Librer%c3%ada-de-clases.aspx>

1.2.3.6 COMPATIBILIDAD EN EL ENFOQUE ORIENTADO A OBJETOS

C# es un lenguaje orientado a objetos puros, lo que significa que todo con lo que se trabajará en este lenguaje son objetos. Un objeto es un agregado de datos y de métodos que permiten manipular dichos datos, y un programa en C# no es más que un conjunto de objetos que interaccionan unos con otros a través de sus métodos.

La programación orientada a objetos (POO) nos permite escribir código menos propenso a fallos además de permitirnos la reutilización de código de forma más conveniente.

LOS PILARES DE LA POO

Las principales características de un lenguaje orientado a objetos, es decir, se considera que un lenguaje está totalmente orientado a objetos si es capaz de proveer estas tres características:

- **Encapsulación**
 - **Herencia**
 - **Polimorfismo**
- **ENCAPSULACIÓN** es la cualidad de unificar los datos y la forma de manipularlos, de esta forma se puede ocultar el funcionamiento de una clase y exponer solo los datos que manipula (mediante propiedades), así como proveer de medios para poder manipular dichos datos (mediante métodos). De esta forma solo se expone al mundo exterior la información y la forma de manipularla, ocultando los detalles usados para manejar esos datos y, lo que es más importante, evitando que nadie manipule de una forma no controlada dicha información.
- **HERENCIA** es la cualidad de poder crear nuevas clases (o tipos) basadas en otras clases, de forma que la nueva clase obtenga todas las características de la clase que ha heredado, tanto los datos que contiene como la forma de manipularlos, pudiendo añadir nuevas características e incluso cambiar el comportamiento de algunas de las incluidas en la clase base, (siempre que así se haya previsto). Mediante la herencia se puede crear de forma fácil una jerarquía

de clases que comparten un mismo comportamiento básico pero que cada nueva generación puede tener (y de hecho tiene) un nuevo comportamiento.

- **POLIMORFISMO** es la cualidad de implementar de forma particular algunas de las características que tienen las clases, de forma que cuando se necesite usarlas no preocupe la implementación interna que cada una tenga, lo que realmente interesa o debe importar es que se pueda usar esas características e incluso se pueda acceder a ellas de forma anónima... o casi.

En .NET siempre se usa una clase para escribir cualquier tipo de código. Por tanto, cualquier cosa que se haga en .NET Framework, hay que hacerlo dentro de una clase. Esto no quiere decir que siempre se deba usar las características de la POO, ya que si simplemente se desea hacer una aplicación que muestre un mensaje en la consola, el código no tiene porqué usar la herencia, el polimorfismo o la encapsulación, simplemente se escribe el código que muestre el mensaje y asunto arreglado, pero lo que si se podría hacer es usar algunas de las "otras" ventajas que nos aporta la programación orientada a objetos.

Fuente: http://www.elguille.info/NET/dotnet/POO_VB_NET_tp6.htm

CAPÍTULO II

PLANEAMIENTO Y ANÁLISIS DEL SISTEMA

Objetivo General.

Analizar, diseñar y desarrollar un sistema de información que permita un adecuado manejo de los datos generados en la finca ganadera “San Antonio”, para controlar y optimizar la información relacionada con los procesos de registro reproductivo y sanitario.

Objetivos específicos.

- Automatización de la información que se genera de manera frecuente y periódica relacionada con el registro reproductivo y sanitario de la finca “San Antonio”.
- Diseñar y construir la base de datos, la cual almacenará la información generada en la finca.
- Desarrollar interfaces por medio de las cuales se permita registrar, actualizar y mostrar la información de la base de datos.
- Generar reportes
 - Informativos
 - De Control

2.1 DETERMINAR EL ÁMBITO DEL PROYECTO

La finca “San Antonio” es una propiedad dedicada a la crianza de ganado vacuno para la producción de leche, la misma que se entrega diariamente a una persona, que la recoge de varias fincas del sector, para luego entregarla a la fábrica Nestle.

En la finca generalmente laboran en promedio diez personas, en tres áreas bien definidas, dos personas fijas en la administrativa, 5 o 6 en la operativa y 3 en la logística.

El área administrativa por ejemplo es la encargada de registrar, actualizar y eliminar, en cartulinas individuales, los datos de las Unidades Bovinas. Los datos de los terneros, por ejemplo, no son registrados sino hasta que tienen cierta edad, solo ahí se les pone una identificación numérica... una tarjeta plástica en la oreja. Para identificar a quien pertenece la Unidad Bovina, en algunos casos, se le hace unos cortes en las orejas y en otros se le pone otro tipo de marca.

En lo que es el área operativa, los trabajadores, por ejemplo, realizan tareas tales como: suministrar medicamentos y suplementos alimenticios, realizar curaciones, identificar a las Unidades Bovinas enfermas, vacunar, hacer pruebas de mastitis, etc.

Todos los datos referentes a estas actividades no tienen un registro bien detallado, solo de manera global se cuantifica y se confía en la memoria de las personas inmersas en las tareas para recordar, por ejemplo, cual Unidad Bovina estuvo enferma, cuanta medicación se le administró, que medicación, cuál tuvo mastitis, etc.

El área logística realiza compras de insumos para la finca: medicamentos, instrumental veterinario, suplementos nutricionales, material de construcción y mantenimiento para las cercas de los pastizales, etc., y aunque se lleva un registro de todas estas actividades, este no es el mejor; se puede hacerlo de manera más precisa y detallada.

Para ello se desarrollará algunos módulos, los cuales se detallan a continuación:

1. Módulo de ingreso de datos

Este módulo permitirá, mediante varios tipos de interfaces, registrar y actualizar datos referentes a:

- a. Identificación del ganado.
- b. Eventos del proceso reproductivo
- c. Nuevos usuarios.
- d. Enfermedades.
- e. Medicamentos
- f. Vacunas.

- g. Insumos utilizados en la finca para los diferentes eventos o procesos que en ella se dá.

2. Módulo de Control Reproductivo

Este es uno de los más importantes, ya que brindará una ayuda permanente en el registro y planificación de las actividades relacionadas con la reproducción del ganado, permitiendo anticipar los eventos reproductivos de las vacas (partos, controles de parto, vacas en celo, inseminaciones o monta, vacas secas, etc.) y su registro de forma detallada.

En este módulo el sistema controlará y dará mensajes sobre eventos relacionados con la reproducción de cada una de las Unidades Bovinas hembras en estado reproductivo.

3. Módulo de Control Sanitario

Por medio de este módulo se administra información de:

- a. Enfermedades
- b. Historias clínicas individuales de las UB
- c. Medicamentos
- d. Tratamientos para las enfermedades.

Además permitirá llevar un control sobre los eventos que tienen relación con la parte sanitaria de las UB, entre ellos están los siguientes:

- a. Vacunación periódica de las UB.
- b. Desparasitación periódica.
- c. Exámenes tipo laboratorio, por ejemplo periódicamente se hace un control de la mastitis a todas las UB hembras, por medio de recoger muestras de leche de cada una de las tetas en un recipiente propio para hacer el examen, al cual se añade el reactivo.
- d. Suministro de suplementos.
- e. Seguimiento de los tratamientos.

4. Módulo de Seguridad y Usuarios.

Este módulo lo manejará directamente el administrador del sistema, o el usuario del sistema que tenga este perfil, ya que es información sensible.

Este módulo permitirá controlar la seguridad del sistema así como su información, además de la administración de usuarios.

5. Módulo de Consulta y Reportes.

Este módulo permitirá hacer uso de los datos contenidos en la Base de Datos, organizándolos de una manera adecuada y fácil de entender, para luego mostrar la información requerida y definida por el usuario, en interfaces adecuadas para ello.

La información desplegada será de todos los módulos existentes en el sistema:

- a. Módulo de Unidades Bovina
- b. Módulo e Control sanitario
- c. Módulo Reproductivo
- d. Módulo de Seguridad
- e. Módulo de compra de insumos e inventario.

Reportes generados desde el módulo de Unidades Bovinas

Reporte Individual

Este reporte tendrá información del ganado, en forma individual de su identificación general, como código, nombre, raza, peso, procedencia, sexo, fecha de nacimiento, edad, padres, propietario, etc.

Reporte general

Este reporte mostrará información básica de todo el ganado que existe en la finca, así por ejemplo contendrá la siguiente información:

- Número
- Nombre
- Sexo

- Estado de madurez o estado de desarrollo, y
- Propietario

Reporte por sexo de las UB.

Este permite conocer cuantas UB son machos y hembras.

Reporte de Bajas

Muestra que ganado ya no existe en la finca y el motivo para tal hecho, pudiendo ser por venta o muerte y sus causas.

Reportes generados desde el módulo de Control Sanitario

Reporte Individual Sanitario

Este reporte está enfocado al campo medico. Viene a ser una historia clínica de cada UB. Se mostrará información identificativa básica pero se detallará de manera completa las enfermedades que ha tenido así como los tratamientos recibidos junto con las fechas de los mismos.

Reporte por enfermedad.

Este reporte permite conocer que unidades bovinas padecieron de una determinada enfermedad, así como la fecha.

Reporte de Vacunas

Este reporte muestra qué grupo de ganado recibió, o no, las diferentes vacunas existentes y en qué fecha la recibieron. Este reporte además incluirá identificación básica del animal.

Reporte de los medicamentos.

Es una especie de inventario de los medicamentos de que se dispone, o no.

Reportes generados desde el módulo de Control Reproductivo

Reporte para Control Preñes

Por medio de este reporte se puede conocer la identificación de las vacas que se encuentren preñadas, el tipo de servicio (monta o inseminación), número de partos, el tiempo de preñes y la fecha del próximo parto.

Reporte de crecimiento

Este reporte permite conocer cuántos animales corresponden a cada categoría:

- a. Terneros
- b. Terneras
- c. Vaconas media
- d. Vaconas fierro
- e. Vaconas de vientre
- f. Vacas en producción
- g. Vacas secas
- h. Animales para descarte
- i. Toretes
- j. Toros reproductores.

Reporte de eventos

Este reporte permite conocer que tipos de eventos, relacionados con las actividades de la finca, existen. Esta información es relevante al momento de hacer uso de los insumos, pues permitirá justificar el uso de los insumos en los diferentes eventos.

Reportes generados desde el módulo de Insumos

Reporte general de insumos.

Este reporte es un inventario general y permitirá conocer todos los insumos que hay en stock. Entre los insumos se incluyen los medicamentos, suplementos vitamínicos y los diferentes tipos de utensilios que se usan y necesitan en la finca. Brindará además una información de referencia para conocer cuando un insumo está en su valor mínimo o máximo, esto es para reabastecer a tiempo o para no tener en exceso, ya que existe una columna en una tabla de la base de datos donde se tiene un valor mínimo y un máximo para cada uno de ellos.

Reportes generados desde el módulo de Seguridades y Usuarios.

Reporte general de Usuarios.

Proporciona información general o individual de cada uno de los usuarios del sistema, activos o inactivos. Lo de activos o inactivos se debe a que por algún motivo un usuario deja de tener acceso al sistema, este no elimina sus datos sino que lo pone en estado inactivo.

Reporte de perfiles y Opciones.

Permite saber que perfiles pueden tener los usuarios así como las opciones que estos manejan.

Auditoria de uso del sistema.

Es un reporte muy importante ya que informa quien modificó los datos más sensibles de la base de datos y en qué fecha.

2.2 ANÁLISIS DE LAS NECESIDADES

El registro de los procesos que se llevan a cabo en la finca se lo hace de la manera tradicional; esto quiere decir que los datos de diferente índole que maneja la finca se los guarda en hojas y/o cartulinas, y en el mejor de los casos se usa hojas de Excel.

El área administrativa necesita manejar los datos de manera segura, rápida y confiable, para evitar pérdidas de información, hacer un control más minucioso de los recursos, para reducir gastos y generar reportes, los cuales servirán de apoyo en la toma de decisiones del área.

El área operativa, de igual manera, necesita llevar un control adecuado de todas las actividades de mantenimiento, prevención y correctivas, que realiza, tales como: vacunación, control de preñez, control nutricional y control de enfermedades, con el fin de planificar sus acciones inmediatas y futuras.

El área logística, en cambio, necesita tener un inventario detallado de las UB existentes, de los medicamentos y de los insumos, con el fin de controlar, planificar, adquirir y suministrar los recursos necesarios a las áreas administrativa y operativa.

2.3 REQUISITOS DE SOFTWARE Y HARDWARE

A continuación se describe los requisitos que se deben cumplir para poder instalar el sistema.

PostgreSQL 8.4

Microsoft Visual C# Express Edition

Un ordenador PC con:

- 256 Mb en Ram
- Disco Duro de 40 GB
- Procesador Pentium IV
- S.O Windows 98 o superiores

2.4 INCORPORACIÓN DE COMENTARIOS A LAS ESPECIFICACIONES.

Las reuniones con el administrador y los usuarios del sistema, han servido para determinar sus diferentes necesidades, respecto a la automatización de la información que se genera en los diferentes procesos de la finca, pues a pesar que se lleva un control básico, este es insuficiente, la información es centralizada e inaccesible para muchos de los involucrados en los procesos, los reportes y consultas toman demasiado tiempo y se lo hace mediante la consulta manual de los escritos existentes, y la calidad de los mismos no es la óptima.

Con la implementación del sistema habrá un cambio sustancial, ya que es un sistema amigable para los usuarios, con información accesible para todos los involucrados, confiable y seguro. Para lograrlo, se trabaja de manera consciente.

CAPÍTULO III

MODELO DE DISEÑO

Es necesario e importante desarrollar de manera breve pero consistente en su contenido, este tema.

El proceso de diseño comprende dos fases importantes: la diversificación y la convergencia. Donde la diversificación es la adquisición de una variedad de alternativas, de material primitivo de diseño, tales como: componentes, soluciones de componentes y conocimiento, todo esto dentro de libros de texto, catálogos, consultas y en la mente misma. En cambio en la convergencia, el diseñador debe elegir y combinar los elementos adecuados del repertorio del que dispone para satisfacer los objetivos del diseño y los requisitos, todo esto en concordancia con lo acordado con el cliente.

Otra fase, luego de las dos anteriores, es la eliminación gradual de cualquier configuración de componentes, dejando una sola en particular, a partir de la cual se procede a la creación del producto final.

Pero definamos

¿Qué es el diseño? Es una representación significativa o importante de ingeniería de algo que se va a construir, donde se puede hacer un seguimiento basándose en los requisitos del cliente, y al mismo tiempo evaluar su calidad, tomando como referencia un conjunto de criterios predefinidos, por entidades encargadas de esa área.

¿Quién lo hace? Lo hace el ingeniero del software, pero los conocimientos que se requieren en cada nivel de diseño funcionan de diferente manera. Así en el nivel de datos y arquitectura, el diseño se centra en los patrones definidos y que se aplicarán al sistema. En el nivel de interfaz, es la ergonomía humana la que dicta el enfoque de diseño. Y en el nivel de componentes o módulos, será un enfoque de programación la que conduzca a diseños de datos y procedimientos eficaces.

¿Por qué es importante? Porque evita cometer errores y riesgos, es como hacer una casa sin un plano, solo la intuición no es suficiente, sería un desastre.

¿Cuáles son los pasos? El diseño comienza con el modelo de requisitos y se trabaja para transformarlo y obtener cuatro niveles de detalle de diseño: la estructura de datos, la arquitectura del sistema, la interfaz y el detalle de los componentes.

¿Cuál es el producto obtenido? Un modelo de diseño que describen los datos, arquitectura, interfaces y componentes.

¿Cómo estar seguro de que lo hecho sea correcto? En cada etapa se revisan los productos del diseño del software en cuanto a: claridad, corrección, finalización y consistencia, comparándolos con los requisitos.

El diseño orientado a objetos transforma el modelo de análisis, en un modelo de diseño que sirve como anteproyecto para la construcción del software.

El diseño de software orientado a objetos es difícil, y el diseño de software reusable es aún más difícil, pues debe identificar los objetos pertinentes, clasificarlos dentro de las clases, definir interfaces de clases y jerarquías de herencia y establecer relaciones clave entre ellos. El diseño debe ser específico al problema que se tiene que enfrentar o resolver, pero suficientemente general para adaptarse a problemas y requerimientos futuros. Dicho en otras palabras debe tener una modularidad global.

Este sistema tiene un diseño orientado a objetos, no solo por el uso de un lenguaje, C#, totalmente orientado a objetos, sino por la modularidad del mismo así como una arquitectura de software multicapa.

El diseño Orientado a Objetos se divide en dos grandes actividades:

- Diseño del sistema
- Diseño de objetos.

En el siguiente gráfico se ve la pirámide para sistemas orientados a objetos.

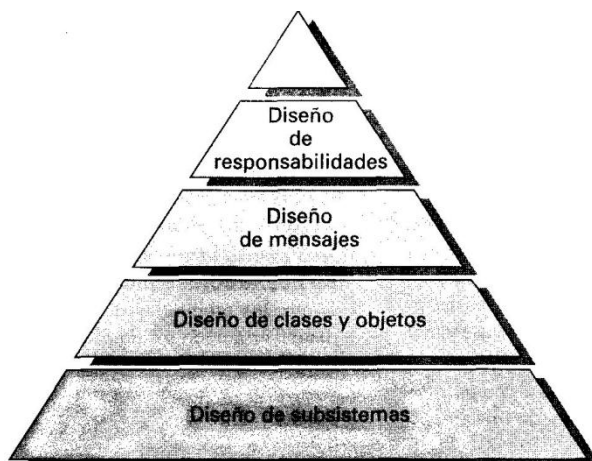


Fig. 3: Pirámide del Diseño Orientado a Objetos.

Fuente: Libro "Ingeniería del software" pag.380

La capa subsistema.- Contiene una representación de cada uno de los subsistemas, para permitir al software conseguir requisitos definidos por el cliente e implementar la infraestructura que soporte los requerimientos del cliente.

La capa de clases y objetos.- Contiene la jerarquía de clases, que permiten al sistema ser creado.

La capa de mensajes.- Contiene detalles de diseño, que permite a cada objeto comunicarse con sus colaboradores.

La capa de responsabilidades.- Contiene estructuras de datos y diseños algorítmicos, para todos los atributos y operaciones de cada objeto.

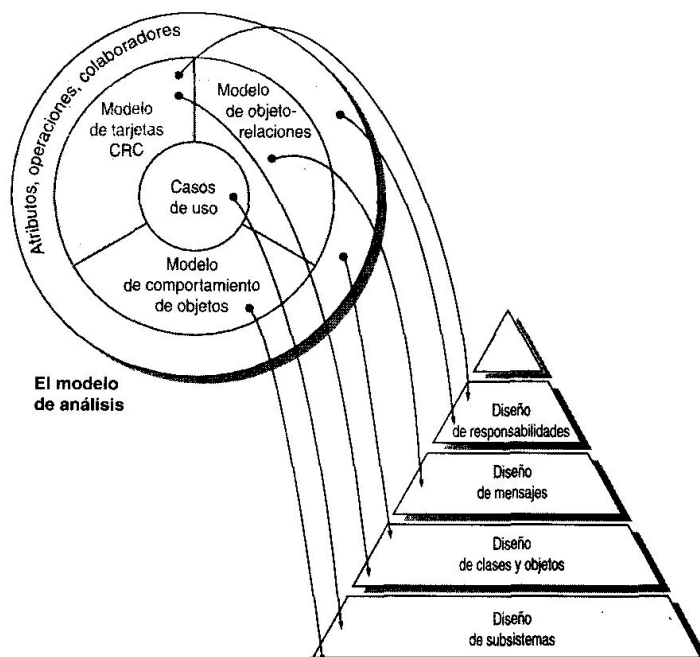


Fig. 4: Transformación de un modelo de análisis OO a un modelo de diseño OO.

Fuente: Libro "Ingeniería del software" pag.380

3.1 DISEÑO DEL SISTEMA

El objetivo del proceso de Diseño del Sistema de Información es en si la definición de la arquitectura del sistema o producto, definiendo una serie de capas que cumplen funciones específicas del sistema e identifica las clases que son encapsuladas por los subsistemas que residen en cada capa. También define el entorno tecnológico que le va a dar soporte, junto con la especificación detallada de los componentes del sistema de información.

A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la descripción técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando proceda.

3.2 DISEÑO DE OBJETOS

Tiene que ver con la manera de representar e identificar un objeto, de hacerlo diferente, único, tomando en cuenta sus propios datos, propiedades e interacciones con otros objetos. Esto requiere de un adecuado diseño de las clases, pues será el soporte para el diseño del sistema.

3.3 DISEÑO Y ESPECIFICACIÓN DE LAS CLASES

Consiste principalmente en transformar el modelo de clases lógico, en un modelo de clases de diseño. Este modelo recoge la especificación detallada de cada una de las clases, es decir, atributos, operaciones, métodos y el diseño preciso de las relaciones establecidas entre ellas, tales como agregación, asociación o jerarquía.

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

Los diagramas de Clases por definición son estáticos, esto es, representan que partes interactúan entre sí, más no lo que ocurre, ni cuándo.

Más adelante, dentro de lo que son los diagramas UML, se muestra el diagrama de clases.

3.4 DISEÑO DE ALGORITMOS

En términos generales, se crea un algoritmo para implementar las especificaciones para cada operación. Muchas veces el algoritmo es una simple secuencia computacional o procedural. Sin embargo, si la especificación de la operación es compleja, será necesario modularizar la operación.

Existen muchos tipos diferentes de operaciones, normalmente se pueden agrupar en tres grandes categorías:

- Operaciones que manipulan los datos de alguna manera:
 - Agregando

- Eliminando
 - Reformateando
 - Seleccionando
- Operaciones que ejecutan cálculos
 - Operaciones que monitorizan o supervisan al objeto para la ocurrencia de un suceso controlado.

3.5 DISEÑO DE LA ARQUITECTURA DEL SISTEMA.

Es un proceso, que mediante el análisis y estudio define una solución estructural para solventar los diferentes tipos de requisitos de un sistema.

Existen dos tipos de requisitos:

- Funcionales
- No funcionales

Requisitos Funcionales.- Definen una característica requerida del sistema, la cual va a ser satisfecha por medio de un subsistema o bloque de código en el software, expresa una capacidad de acción, una funcionalidad. Generalmente se expresan de una forma verbal.

Requisitos No Funcionales.- Son aquellos que no son objeto de codificación, aunque podrían llegar a afectarla. Son cualidades que se imponen al proyecto, por ejemplo el lenguaje de programación o la plataforma tecnológica, que se deben cumplir.

La aplicación de esta teoría a este sistema se hará más adelante en el Manual Técnico, capítulo 4.5.2.3.9

3.6 DISEÑO DE INTERFACES.

Con el tiempo esta tarea se ha convertido en algo realmente importante y tal es así que podría determinar en gran parte el éxito o el fracaso en el desarrollo de un

sistema. Razón por demás para tener en cuenta los siguientes aspectos al momento de diseñarlas.

- Utilizar títulos que sean significativos, que identifiquen sin ambigüedad el propósito del informe o formulario.
- Dar instrucciones breves y fáciles de comprender.
- Agrupar y secuenciar los campos de forma lógica.
- Hacer que el aspecto del informe o formulario sea atractivo a la vista.
- Utilizar nombres familiares para etiquetar los campos.
- Utilizar terminología y abreviaturas consistentes.
- Hacer un uso razonable y consistente de los colores.
- Dejar un espacio visible para los datos de entrada y delimitarlos.
- Permitir un uso sencillo y adecuado del cursor.
- Permitir la corrección carácter a carácter y de campos completos.
- Dar mensajes de error para los valores que son ingresados de manera incorrecta
- Marcar los campos que sean opcionales.
- Dar mensajes a nivel de campo para explicar su significado.
- Dar una señal que indique cuándo el informe o formulario está completo.

Este tema se desarrollará de una manera más amplia en el Manual del Sistema o Manual Técnico.

3.7 DIAGRAMA ENTIDAD RELACIÓN.

El diagrama de entidad-relación permite especificar los objetos de datos que entran y salen de un sistema, los atributos que definen las propiedades de estos objetos y las relaciones entre ellos.

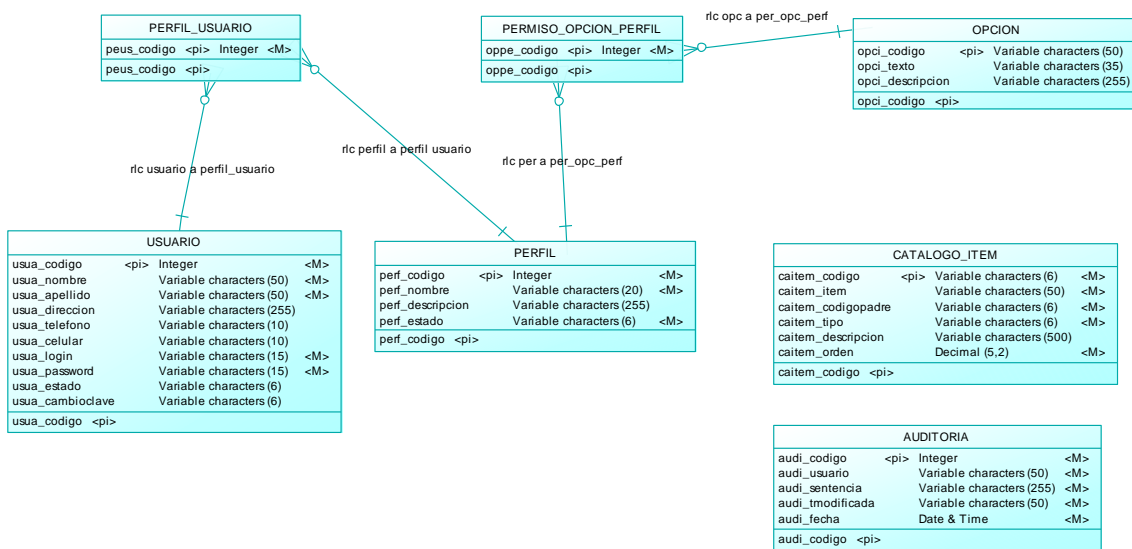


Fig. 5: Diagrama del módulo Seguridad

Realizado por: Los Autores de la Tesis

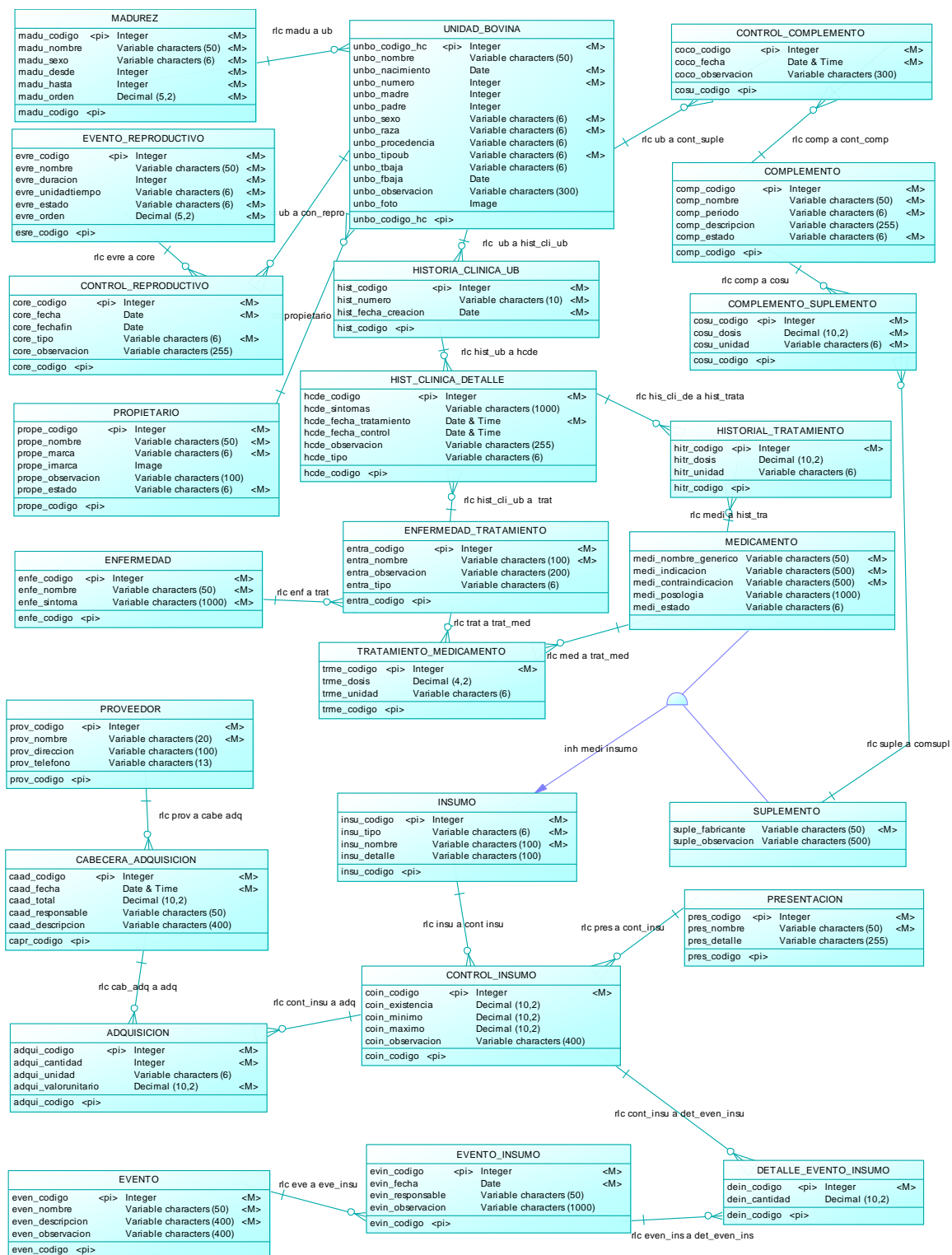


Fig. 6: Diagrama del módulo de Unidades Bovinas e Insumos.

Realizado por: Los Autores de la Tesis

3.8 DISEÑO DE MENÚ Y MAPAS DE NAVEGACIÓN.

El sistema SISA tiene implementado un menú principal para el manejo de los procesos de la Finca San Antonio, Fig. 7, este menú es al que primero se accede cuando un usuario se le permite ingresar. Como se ve está compuesto por las siguientes opciones:

- Inicio
- Ver
- Gestión Unidad Bovina
- Gestión Insumos
- Gestión Seguridad
- Reportes
- Sistema
- Ventana
- Ayuda

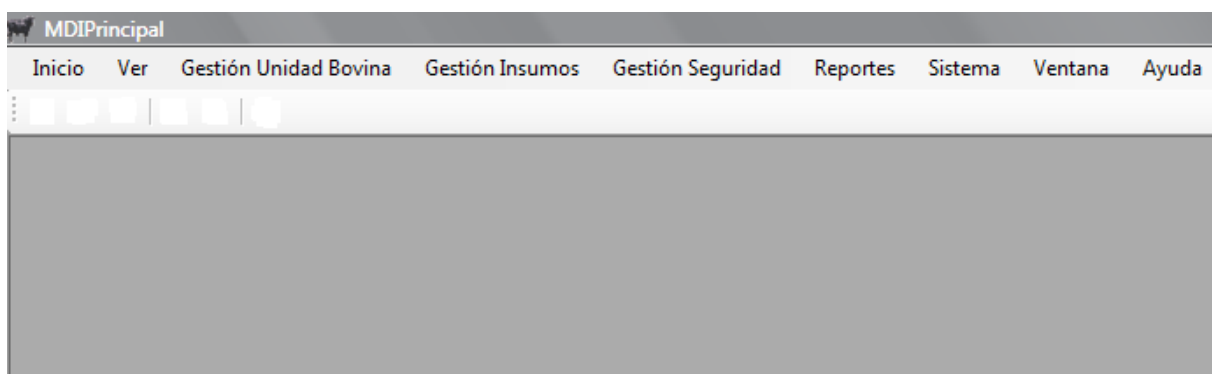


Fig. 7: Menú principal del Sistema SISA.

Realizado por: Los Autores de la Tesis

Una descripción más detallada de este tema se encuentra en el Manual Técnico o Manual del Sistema, en el capítulo 4 literal 4.6.

3.9 DISEÑO DE AYUDAS

En los diferentes formularios o pantallas que tiene el sistema, se ha implementado enlaces de ayuda, de manera que los usuarios puedan despejar sus dudas respecto a cualquier tema.

El menú principal tiene la opción "Ayuda", que brinda información más general respecto al sistema, al hacer clic en esta opción se despliega otras tres opciones que son:

- Sistema
- La Finca , y
- Acerca De...

Cada una de estas se anlaza a información relevante al tema.

En el capítulo 4.5.2.3.8 en lo que es el Manual Técnico o Manual del Sistema se amplía este tema.

3.10 MÉTRICAS APLICADAS AL DISEÑO.

En cualquier campo de la ingeniería las métricas y las mediciones son muy importantes, por lo tanto el desarrollo de software no puede estar exento de ellas, por desgracia a decir de los expertos el uso de métricas para sistemas ha progresado mucho más despacio que el uso de otros métodos Orientados a Objetos. Ed Berard menciona, diríamos, con ironía lo siguiente:

“La gente involucrada en el software parece tener una relación amor-odio con las métricas. Menosprecian y desconfían de cualquier cosa que parezca o suene a medición. Son rápidos, bastante rápidos para señalar las “imperfecciones”, en los argumentos de cualquiera que hable acerca de las mediciones a los productos de software, procesos de software y especialmente personas involucradas en software.”

Las métricas sirven para evaluar la calidad del diseño, a nivel arquitectónico y de componentes, donde la clase es la piedra fundamental de la arquitectura Orientada a Objetos.

Los objetivos primarios de las métricas orientadas a objetos son:

- Entender mejor la calidad del producto
- Evaluar la efectividad del proceso
- Mejorar la calidad del trabajo llevado a cabo al nivel del proyecto.

Cinco características regulan las métricas especializadas: Localización, encapsulación, ocultamiento de información, herencia y técnicas de abstracción de objetos.

Localización

Es una característica del software que indica la manera en que la información se concentra en un programa. Por ejemplo los métodos manejados por datos localizan la información en torno a estructuras de datos específicas.

Puesto que la clase es la unidad básica de un Sistema Orientado a Objetos, la localización se basa en los objetos. Por esta razón, las métricas deben aplicarse a la clase (objeto) como una entidad completa.

Encapsulación

La encapsulación engloba las responsabilidades de una clase, incluyendo sus atributos y operaciones, y los estados de la clase, definidos por valores de atributos específicos.

Ocultación de información

La ocultación de información suprime u oculta los detalles operacionales de un componente de programa. Solo se proporciona la información necesaria para acceder al componente a aquellos otros componentes que deseen acceder.

Herencia

La herencia es un mecanismo que habilita las responsabilidades de un objeto, para propagarse a otros objetos. La herencia ocurre a través de todos los niveles de jerarquía de clases.

Abstracción

La abstracción es un mecanismo que permite al diseñador concentrarse en los detalles esenciales de un componente de programa, ya sean datos o procesos, prestando poca atención a los detalles de bajo nivel.

Ya que una clase es una abstracción, que puede visualizarse a diferentes niveles de detalle, de diferentes maneras (por ejemplo, como una lista de operaciones, como una secuencia de estados, como una serie de colaboraciones), las métricas Orientadas a Objetos representan abstracciones en términos de mediciones de una clase (por ejemplo, número de instancias por clase por aplicación, número o clases parametrizadas por aplicación, y proporción de clases parametrizadas con clases no parametrizadas).

CAPÍTULO IV

DESARROLLO.

4.1 REVISIÓN DE ESPECIFICACIONES DE FUNCIONAMIENTO.

Para que el sistema funcione se necesita instalar :

- Microsoft Visual Studio 2008
- Postgresql
- Power designer

Microsoft Visual Studio 2008

Los requerimientos mínimos son: tener Windows XP, Windows Server 2003 o Windows Vista, una Pentium III 1.6 GhZ y 1 gb de Ram mínimo. Esto sería lo básico, aunque siempre estos requerimientos permiten ejecutar el entorno y para desarrollos más avanzados requiera hardware más potente, aun mas si lo vas a instalar sobre Windows Vista.

Arquitecturas compatibles:

- x86
- x64

Sistemas operativos compatibles:

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Windows Vista
- Microsoft Windows Server 2008
- Windows 7

Requisitos de hardware:

- Requisitos mínimos: CPU a 1,6 GHz, 384 MB de RAM, pantalla de 800 x 600 y disco duro de 4200 rpm

- Requisitos recomendados: CPU a 2,2 GHZ, 1024 MB de RAM, pantalla de 1280 x 1024 y disco duro de 7200 rpm
- En Windows Vista: CPU a 2,4 GHz, 768 MB de RAM
- 1,6 GB de espacio disponible en disco duro en la unidad de instalación para la instalación mínima
- 2,5 GB de espacio disponible en disco duro en la unidad de instalación para la instalación completa.

Postgresql.

Para poder instalar Postgresql, en Windows, se requiere:

- Tener Windows 2000 en adelante.
- Tener una versión, de Postgresql, Superior a 8.2
- Tener instalado el Framework 3.5

Además de lo anteriormente descrito, es necesario instalar el ODBC para Postgresql, el cual debe ser instalado en una computadora con sistema operativo de 32 bits, que es único disponible a la fecha. Lo cual descarta a las computadoras más actuales que tienen en su gran mayoría, por no decir todas, sistemas operativos de 64 bits.

4.2 DESARROLLO DE LA ARQUITECTURA DEL SISTEMA.

Anteriormente se hizo una descripción general de la arquitectura del sistema, a continuación se describe el contenido de la capa de presentación o de usuario, la capa de negocio y la capa de acceso a datos.

Capa de presentación o usuario.

Esta capa proporciona la interfaz visual que los usuarios del sistema utilizarán para ver la información y los datos. En esta capa los componentes son responsables de solicitar y recibir servicios de otros componentes del mismo nivel o de la capa de negocio.

En el siguiente gráfico se puede ver el contenido esta capa.

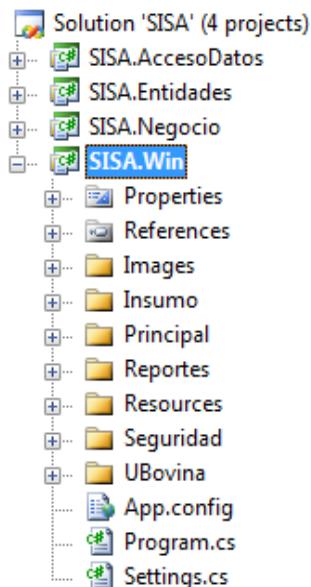


Fig. 8: Capa de presentación

Realizado por: Los Autores de la Tesis

La carpeta “Images” contiene archivos gráficos.

La carpeta “Insumo” contiene las interfaces del módulo de Insumos, y son las siguientes:

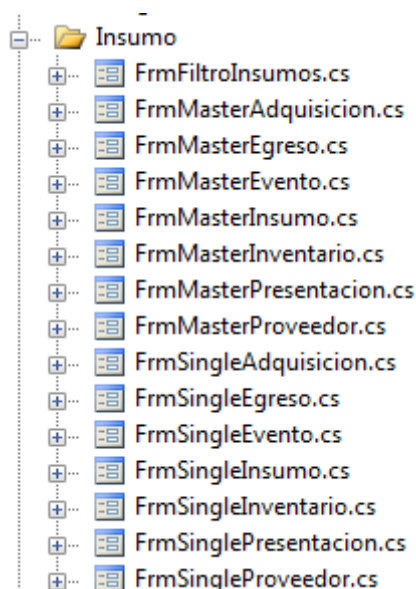


Fig. 9: Interfaces de la carpeta "Insumo".

Realizado por: Los Autores de la Tesis

La carpeta “Principal” contiene interfaces que sirven para que otras interfaces hereden atributos de ellas. Además contiene clases específicas que se utilizan en las diferentes interfaces, por ejemplo la clase ClsAsignarWidth.cs que permite definir tamaños de columnas. Y como esta clase existen otras que cumplen otras funciones diferentes.

El contenido de esta carpeta es el siguiente:

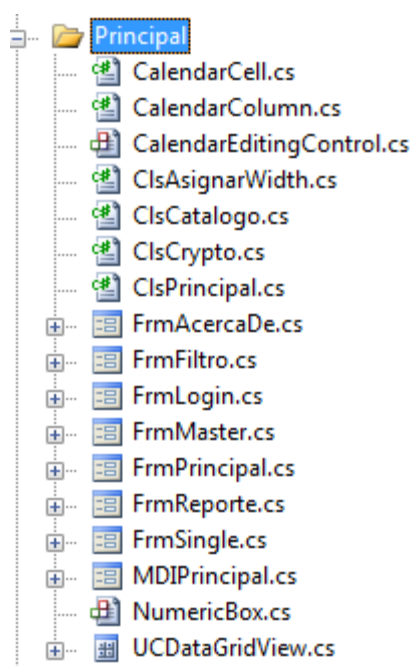


Fig. 10: Interfaces de la carpeta "Principal".

Realizado por: Los Autores de la Tesis

La carpeta “Reportes” contiene 2 carpetas más en su interior: “Diseño” y “Formularios”.

La carpeta “Diseño” contiene los reportes diseñados con Crystal Report, y la carpeta “Formularios” en cambio contiene las interfaces donde estarán contenidos los diferentes reportes.

En el siguiente gráfico se puede ver todo este conjunto.

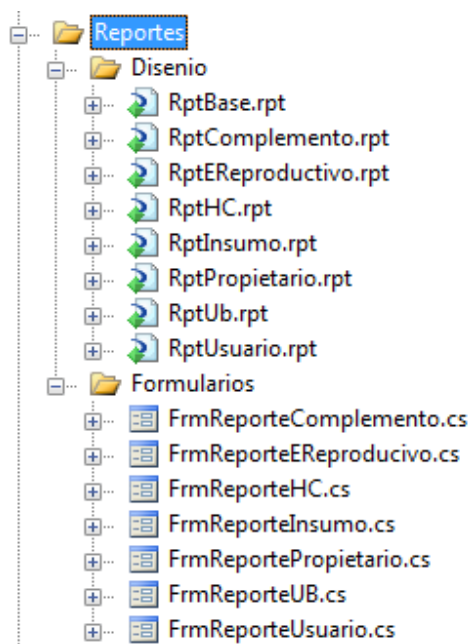


Fig. 11: Interfaces de la carpeta "Reportes".

Realizado por: Los Autores de la Tesis

La carpeta "Resources" contiene archivos con extensión .gif, .png y .jpg , los mismos que son usados en las diferentes interfaces.

Su contenido se puede ver en el siguiente gráfico.

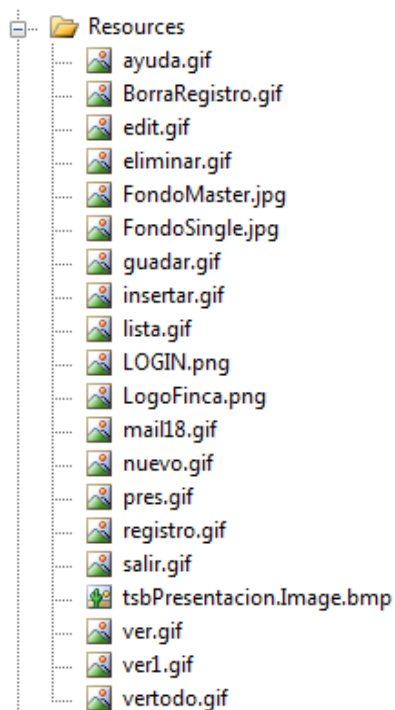


Fig. 12: Archivos de la carpeta "Resources".

Realizado por: Los Autores de la Tesis

La carpeta "Seguridad" contiene las interfaces del módulo de Seguridad, su contenido se puede ver en el siguiente gráfico.

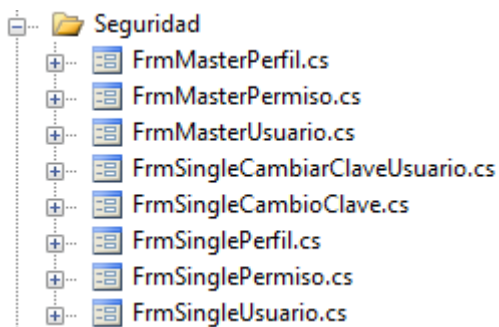


Fig. 13: Interfaces de la carpeta "Seguridad".

Realizado por: Los Autores de la Tesis

La carpeta "UBovina" contiene los formularios del módulo Unidades Bovinas, y su contenido es el siguiente.

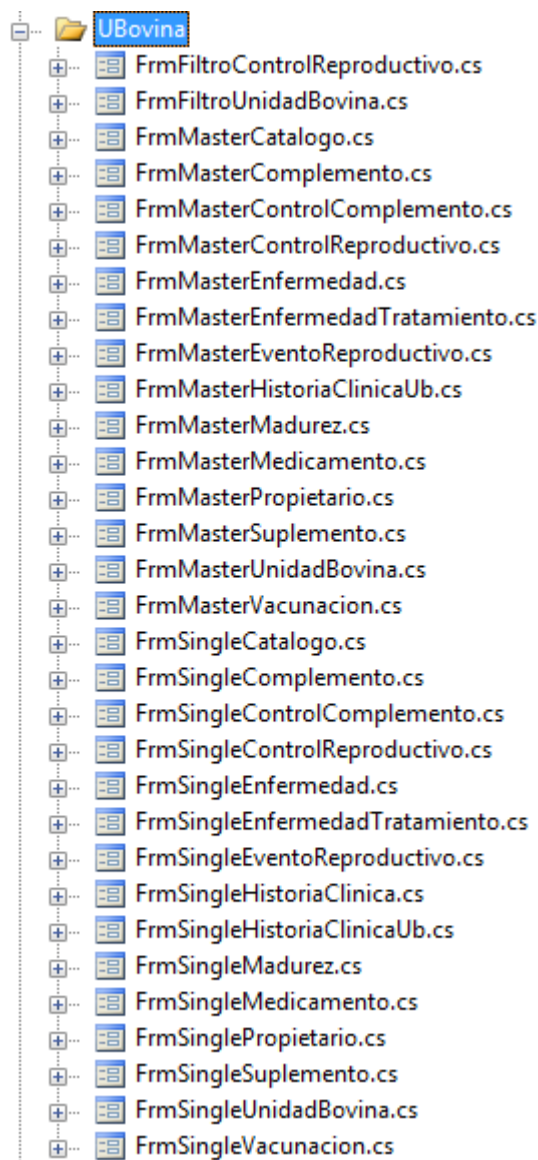


Fig. 14: Interfaces de la carpeta "UBovina".

Realizado por: Los Autores de la Tesis

Capa de Negocios

Esta capa sirve de puente entre la capa de interfaces y la capa de datos. Los objetos de esta capa proporcionan servicios que completan las tareas de negocio tales como verificar los datos enviados por el cliente antes de llevar a cabo una transacción en la Base de Datos.

Esta capa brinda seguridad ya que el usuario no tiene acceso directo a la Base de Datos, lo que se traduce en una mayor seguridad en su integridad.

El contenido de esta capa se puede ver en el siguiente gráfico.

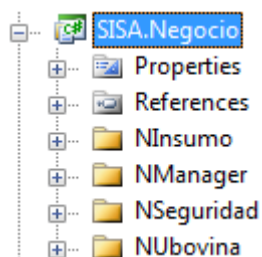


Fig. 15: Capa de Negocios.

Realizado por: Los Autores de la Tesis

La carpeta "NInsumo" contiene varias clases, todas ellas relacionadas con el módulo de Insumos, como se puede ver en el gráfico siguiente:

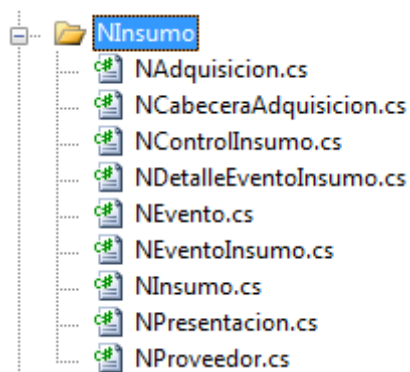


Fig. 16: Clases de la carpeta "NInsumo".

Realizado por: Los Autores de la Tesis

La carpeta "NManager" contiene una única clase llamada NGeneral.cs.

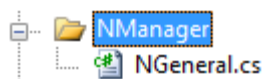


Fig. 17: Interfaces de la carpeta "Reportes".

Realizado por: Los Autores de la Tesis

La carpeta "NSeguridad" contiene varias clases, todas ellas relacionadas con el módulo de Seguridad, como se puede ver en el gráfico siguiente:

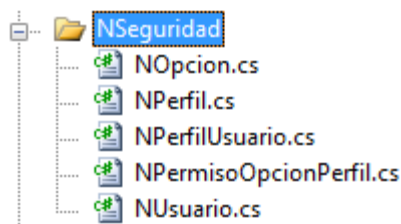


Fig. 18: Clases de la carpeta "NSeguridad".

Realizado por: Los Autores de la Tesis

La carpeta "NUbovina" contiene varias clases, todas ellas relacionadas con el módulo de Unidades Bovinas, como se puede ver en el gráfico siguiente:

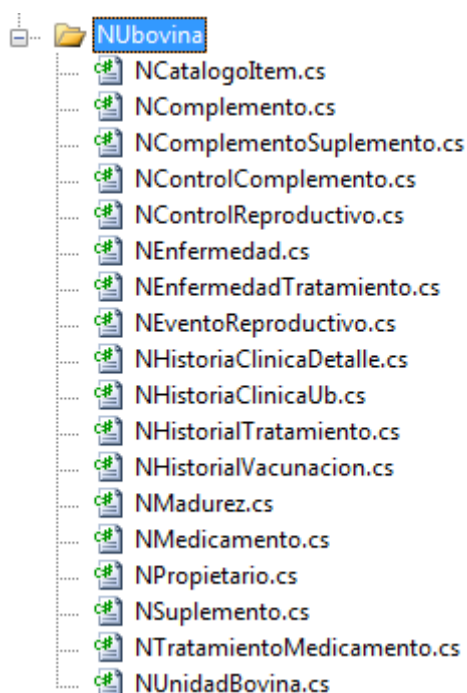


Fig. 19: Clases de la carpeta "NUbovina".

Realizado por: Los Autores de la Tesis

Capa de Acceso a Datos.

Esta capa es la encargada de realizar las tareas relacionadas con los datos: Inserción, Modificación, Consulta u Borrado.

En el sistema esta capa tiene el siguiente contenido:

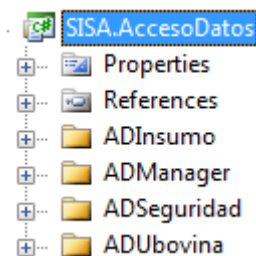


Fig. 20: Capa de Acceso a Datos.

Realizado por: Los Autores de la Tesis

La carpeta "ADInsumo" contiene varias clases, todas ellas relacionadas con el módulo de Insumos, como se puede ver en el gráfico siguiente:

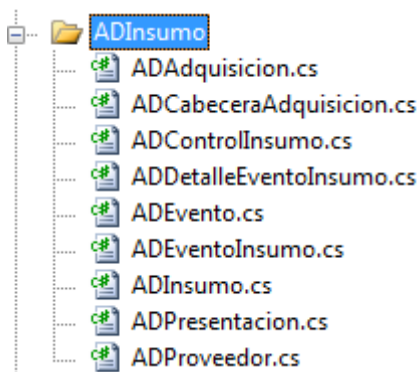


Fig. 21: Clases de la carpeta "ADInsumo".

Realizado por: Los Autores de la Tesis

La carpeta "ADManager" contiene 2 clases como se puede ver en el siguiente gráfico:

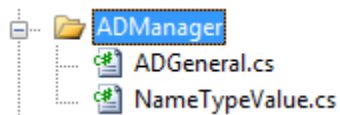


Fig. 22: Clases de la carpeta "ADManager".

La carpeta "ADSeguridad" contiene varias clases, todas ellas relacionadas con el módulo de Seguridad, como se puede ver en el gráfico siguiente:

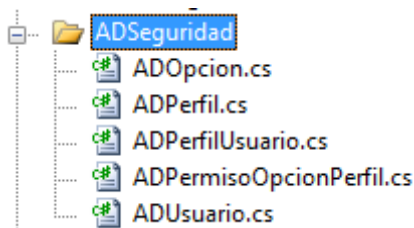


Fig. 23: Clases de la carpeta "ADSeguridad".

Realizado por: Los Autores de la Tesis

La carpeta "ADUbovina" contiene varias clases, todas ellas relacionadas con el módulo de Unidades Bovinas, como se puede ver en el gráfico siguiente:

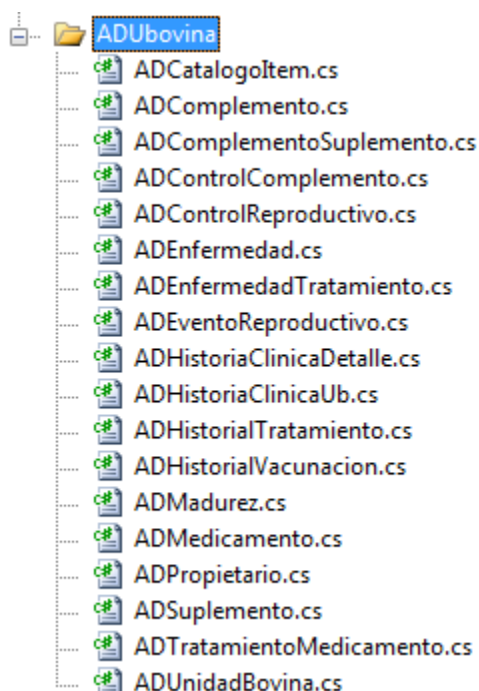


Fig. 24: Clases de la carpeta "ADUbovina".

Realizado por: Los Autores de la Tesis

4.3 DESARROLLO DEL CÓDIGO FUENTE.

Para que las clases de la capa de interfaces accedan a la Base de Datos, para llevar, traer o llevar y traer datos, se necesita de la capa de Negocios, la cual maneja la transaccionalidad y sirve de nexo con la capa de Acceso a Datos la

cual contiene la carpeta "ADManager", dentro de esta carpeta está la clase "ClsGeneral" que contiene tres métodos de conexión que son los siguientes:

- EjecutarConsulta
- EjecutarInstrucción
- EjecutarInsert

El código de los métodos anteriormente enumerados se detalla en el apartado de "Anexos".

Para Insertar , actualizar o borrar registros en la base de datos se tiene varios métodos, los cuales son diferentes de acuerdo al módulo y están implementados en la capa de acceso a datos.

Para ejemplo, el código de los métodos InsertarInsumo, ActualizarInsumo, SeleccionarInsumo, SeleccionarInsumoTodos, SeleccionarInsumoFiltro y EliminarInsumo, está en el apartado de "Anexos".

4.4 PRUEBAS.

Tema importante en cualquier campo de la ingeniería, razón suficiente para desarrollar el tema; en primer lugar, de manera general, y luego enfocado al sistema.

A continuación unas ideas generales pero importantes sobre las pruebas.

Objetivo de las pruebas.- El propósito principal de las pruebas es validar el funcionamiento correcto del producto que se está probando. Las pruebas exitosas encuentran defectos.

Se pueden validar todos los artefactos. Se pueden probar todos los artefactos no solamente el código fuente. Como mínimo se puede revisar los modelos y documentos y por lo tanto encontrar y corregir los defectos antes que lleguen al código.

Probar frecuentemente y temprano. El potencial para el costo del cambio de aumentar exponencialmente lo motiva a probar tan temprano como sea.

Las pruebas construyen confianza. En la obra "Extreme Programming Explained" Kent Beck realiza una observación interesante de que cuando se tiene un suite completa de pruebas (una suite de pruebas es una colección de pruebas), y se la ejecuta tan frecuentemente como sea posible, entonces eso le da el coraje para avanzar. Muchas personas temen realizar cambios al código porque temen dañarlo, pero como una suite completa de pruebas si se daña algo, se sabe que se detectará y entonces se corrige.

Probar contra el riesgo de un artefacto. Entre más riesgos, más es necesario que se revise y se pruebe. En otras palabras, se debe probar los grandes productos o sistemas o algo tan simple como una aplicación "Hello World" .

Una prueba vale mil opiniones. Alguien puede decirnos que su aplicación funciona, pero mientras no muestre los resultados de las pruebas, no se puede creer.

Hay diferentes tipos de pruebas, no solo se puede probar el código, también los modelos y los documentos. A continuación se muestra un cuadro con algunas de las diferentes pruebas.

	TÉCNICA	DESCRIPCIÓN
1	Pruebas de caja blanca	Se realiza sobre las funciones internas de un módulo o clase.
2	Prueba de Valores-Frontera	Es la prueba de situaciones extremas que el ítem debe ser capaz de manejar.
3	Prueba de Clases	Es el acto de asegurar que una clase y todas sus instancias cumplen con el comportamiento definido

4	Prueba de Componente	Es el acto de validar que un componente funciona tal como está definido.
5	Prueba de Cubrimiento	Es el acto de asegurar que toda línea de código se ejercita al menos una vez.
6	Prueba de Regresión de Herencia	Es el acto de ejecutar casos de prueba de las súper clases, tanto de forma directa como indirecta, en una subclase específica.
7	Prueba de Integración	Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntas.
8	Prueba de Método	Consiste en realizar pruebas para verificar que un método (función miembro) funciona tal como está definido.
9	Prueba de Caminos	Es el acto de asegurar que todos los caminos lógicos en el código se ejercitan al menos una vez.
10	Prueba de Regresión	El acto de asegurar que los comportamientos previamente probados todavía trabajan como se espera luego que se han realizado cambios a la aplicación.
11	Prueba de Stress	El acto de asegurar que el sistema funciona como se espera bajo grandes volúmenes de transacciones, usuarios, carga y demás.
12	Revisión Técnica	Una técnica de aseguramiento de la calidad en la cual el diseño de la aplicación es revisado de forma exhaustiva por un grupo de compañeros. Una revisión típicamente se enfoca en la precisión, calidad, facilidad de

		uso y completitud. A este proceso usualmente se le llama recorrido, inspección, o revisión de compañeros.
13	Prueba de Interfaz de Usuario	Consiste en probar la interfaz de usuario para garantizar que cumple los estándares y requerimientos definidos. Usualmente se refiere a la prueba de interfaz de usuario gráfica.
14	Prueba de Caja-Blanca	Consiste en realizar pruebas para verificar que líneas específicas de código funcionan tal como esta definido. También se le conoce como prueba de caja-transparente.

Cabe señalar también que "La Prueba Exhaustiva es Imposible", aunque lo ideal sería exponer al sistema a todas las situaciones posibles, pues así se encuentra todos las fallas.

Esto es imposible desde todos los puntos de vista: humano, económico e incluso matemático. Es por esto que hay que saber elegir las pruebas más adecuadas y económicamente aceptables para encontrar los errores.

En el capítulo 4.5.7 del Manual Técnico se desarrolla este tema, enfocado específicamente al sistema.

4.5 MANUAL TÉCNICO

INTRODUCCIÓN

4.5.1 DESCRIPCIÓN GENERAL

La finca “San Antonio” desea implementar un sistema informático administrador de los recursos y bienes que posee. El administrador de la finca ha definido que este sistema esté siempre disponible.

El sistema desarrollado e implementado permite registrar los datos de todas las Unidades Bovinas, tener Historias Clínicas, llevar un Control Reproductivo de las Unidades Bovinas y también tener un pequeño inventario de los insumos que maneja la finca.

4.5.1.1 ALCANCE

El sistema informático SISA implementará los siguientes módulos:

Modulo de Gestión Unidades Bovinas

Módulo de Gestión Insumos

Módulo de Gestión Seguridad

Módulo de Reportes

4.5.2 INGENIERÍA DEL PROYECTO

4.5.2.1 MODELO DE REQUISITOS

Tiene como objetivo delimitar el sistema y capturar la funcionalidad que ofrecerá desde la perspectiva del usuario y tiene como fundamento el modelo de casos de uso.

4.5.2.1 Diagrama de casos de uso

Módulo de Gestión de Unidades Bovinas

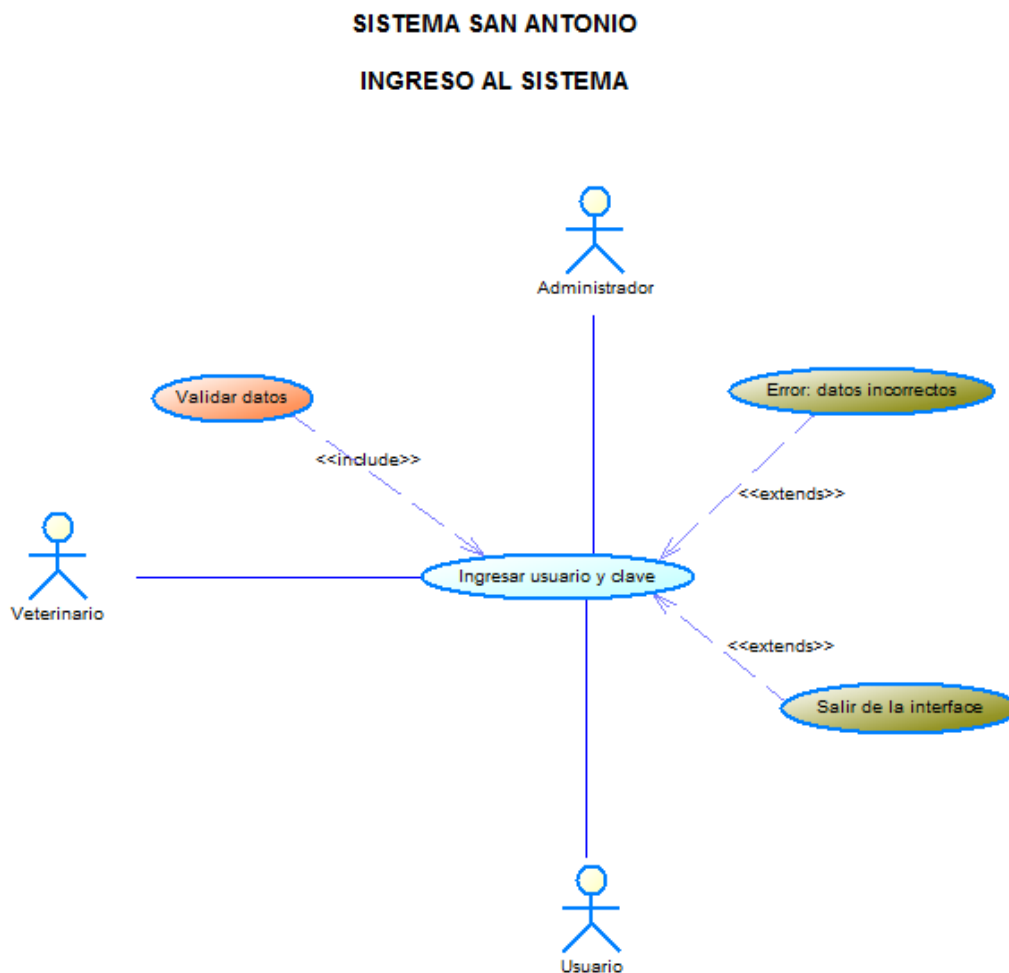


Fig. 25: Diagrama de casos de uso para ingreso al sistema.

Realizado por: Los Autores de la Tesis

Módulo Unidades Bovinas



Fig. 26: Diagrama de casos de uso del módulo de Unidades Bovinas.

Realizado por: Los Autores de la Tesis

Módulo Insumos

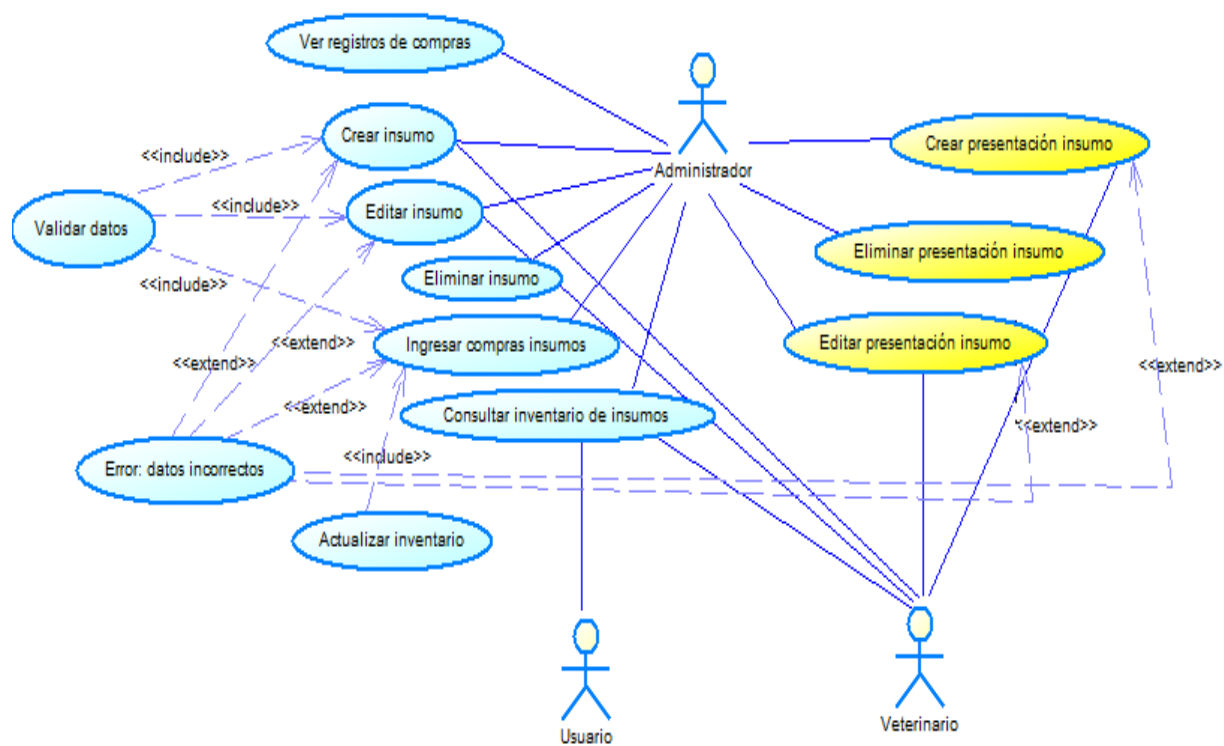


Fig. 27: Diagrama de casos de uso del módulo de Insumos.

Realizado por: Los Autores de la Tesis

Módulo Seguridades y Usuarios

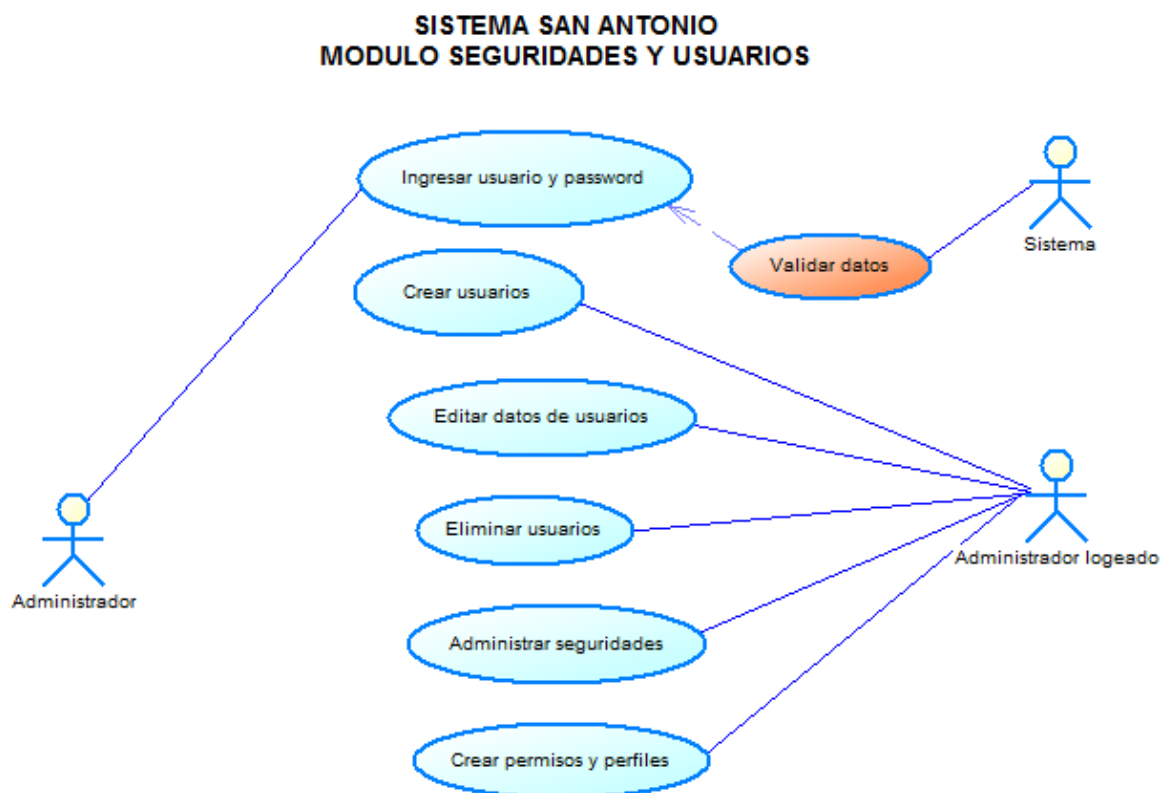


Fig. 28: Diagrama de casos de uso del módulo de Seguridad y Usuarios.

Realizado por: Los Autores de la Tesis

Módulo Reportes



Fig. 29: Diagrama de casos de uso del módulo de Reportes.

Realizado por: Los Autores de la Tesis

Módulo Ayuda

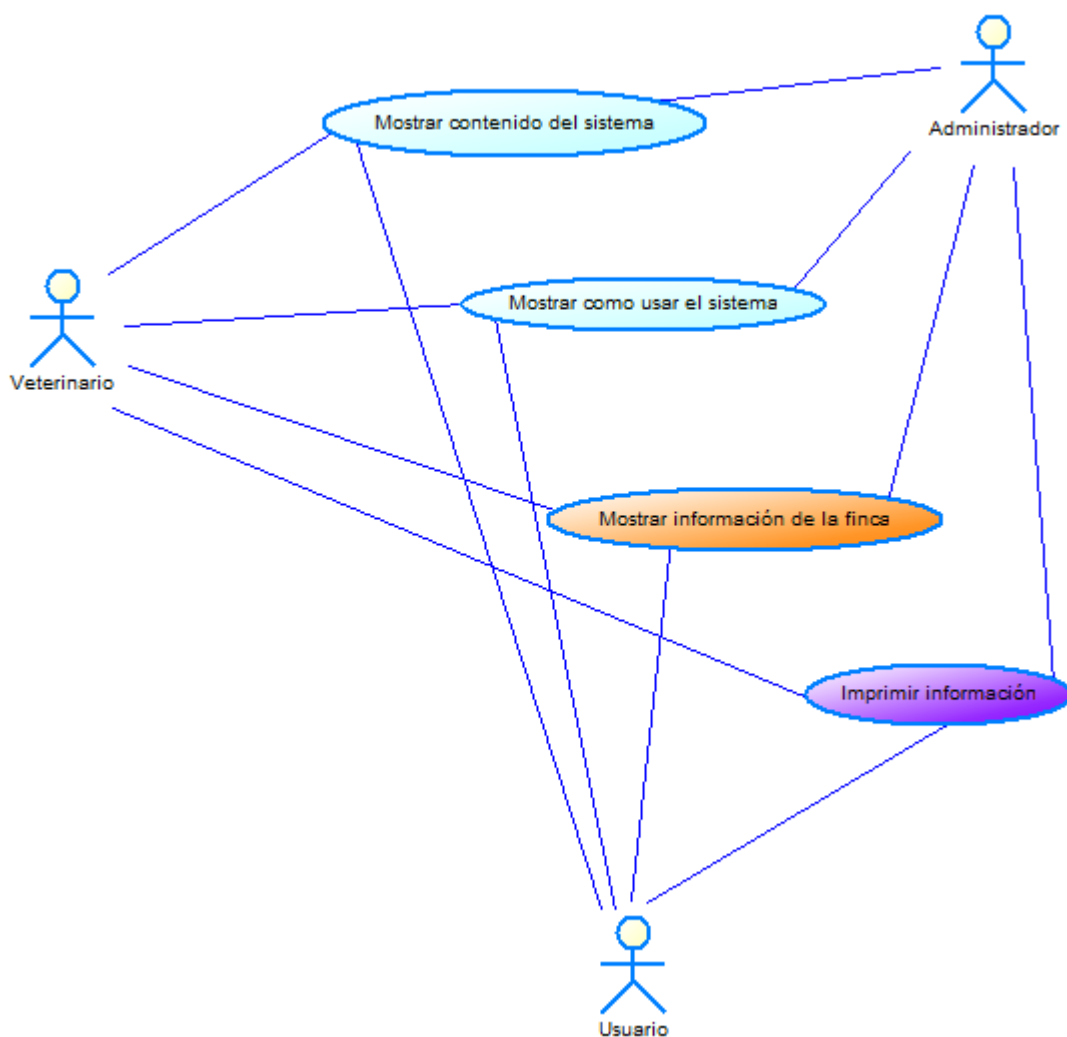


Fig. 30: Diagrama de casos de uso del módulo de Ayuda.

Realizado por: Los Autores de la Tesis

Diagramas UML

A continuación y en las siguientes páginas se describe de manera literal y gráfica los diferentes diagramas UML aplicados al sistema.

Cabe señalar que UML viene a ser como un conjunto de herramientas que permite modelar (analizar y diseñar) sistemas orientados a objetos.

Gracias a UML se puede comunicar y compartir el conocimiento de una arquitectura, gracias a que se combina de manera simultanea cinco perspectivas:

1. Definir.- Fijar, determinar, decidir, explicar un concepto a través de sus atributos distintivos. Señalar sus límites y dar una idea exacta de lo que es esencial y de lo que es circunstancial.
2. Organizar.- Establecer unos recursos, disponer un orden de responsabilidades y formalizar unas reglas de relación y actuación; todo ello orientado a conseguir un propósito.
3. Visualizar.- Representar mediante imágenes y/o símbolos el contenido y la organización de los conceptos que configuran un sistema. Hacer visible su naturaleza y su complejidad.
4. Actuar.- Pensar y tomar decisiones de manera ágil y sistemática, siguiendo un método; éste a su vez, define el modo de actuar en base a la relación de un conjunto de actores, actividades, entregables y certificaciones posibles en un escenario concreto.
5. Certificar.- Comprobar de manera fehaciente que un entregable es completo, coherente y usable para el propósito que ha sido creado.

El resultado, es una mayor comprensión y claridad sobre la naturaleza de los objetos, eventos y hechos que tienen consecuencias dentro de un dominio.

Según Joaquín García, especialista en el tema, **"El 80% de los problemas se pueden resolver usando tan solo el 20% de UML"**

Anteriormente se mencionó el conjunto de herramientas, ahora se las enlista.

- Diagrama de casos de uso
- Diagrama de clases
- Diagrama de estados
- Diagrama de secuencias
- Diagrama de actividades
- Diagrama de colaboraciones
- Diagrama de componentes
- Diagrama de distribución

¿Qué vamos a usar de UML?

Hay que tener en cuenta que es muy importante saber qué y cuando usar estas herramientas. Para este sistema dado el tipo y tamaño, se utiliza las siguientes herramientas UML:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de estados.
- Diagrama de secuencias.
- Diagrama de actividades.
- Diagrama de objetos.

Diagrama de estados

Este tipo de diagramas muestra la secuencia de estados por los que pasa un caso de uso, también muestra el comportamiento de un objeto, es decir, el conjunto de estados por los cuales pasa un objeto durante su vida, junto con los cambios que permiten pasar de un estado a otro.

Los diagramas de estados se dibujan para una sola clase, mostrando el comportamiento del objeto durante su ciclo de vida.

Para ello es importante definir de manera previa el conjunto finito de estados en los que puede estar el objeto y luego definir los estímulos que pueden cambiar dichos estados.

El estado de un objeto se puede caracterizar por el valor de uno o varios de los atributos de su clase, además el estado de un objeto también se puede caracterizar por la existencia de un enlace con otro objeto.

Diagrama de estados de un Usuario

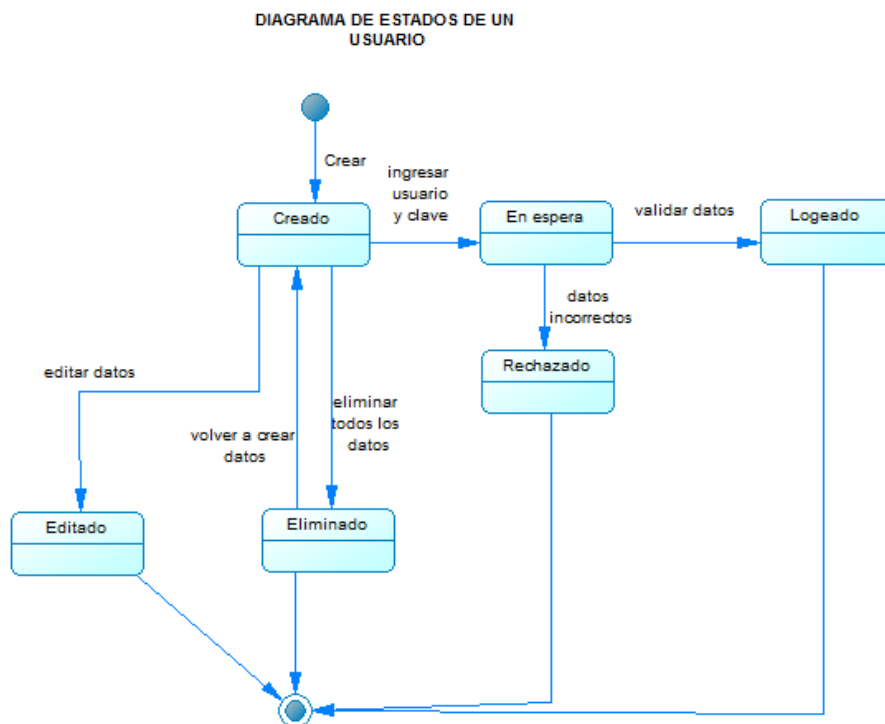


Fig. 31: Diagrama de estados de un Usuario.

Realizado por: Los Autores de la Tesis

Diagrama de estados de un Propietario

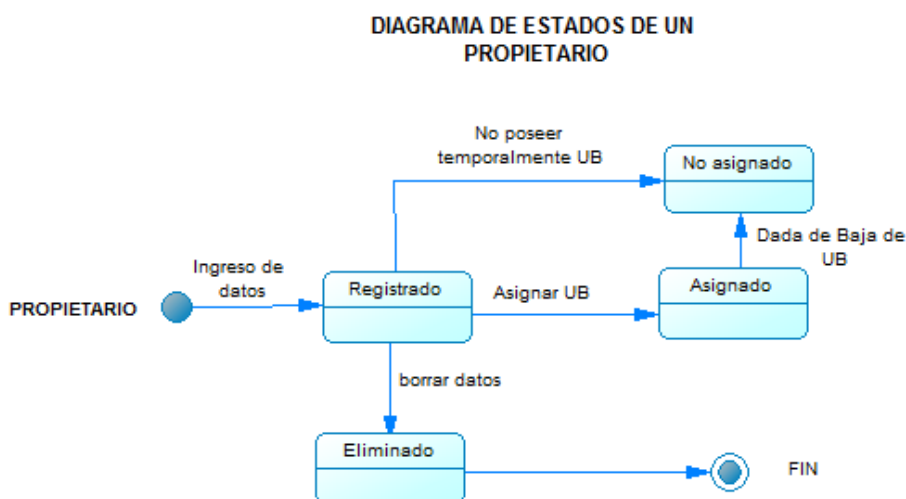


Fig. 32: Diagrama de estados de un Propietario.

Realizado por: Los Autores de la Tesis

Diagrama de estados de una UB

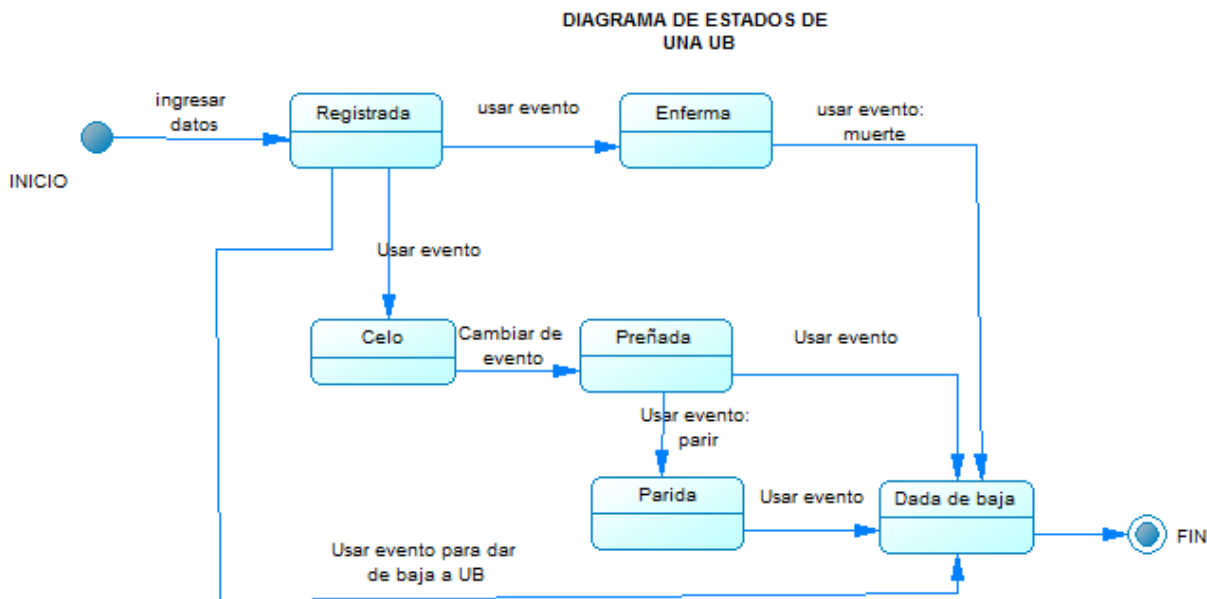


Fig. 33: Diagrama de estados de una Unidad Bovina.

Realizado por: Los Autores de la Tesis

Diagrama de estados de un Insumo

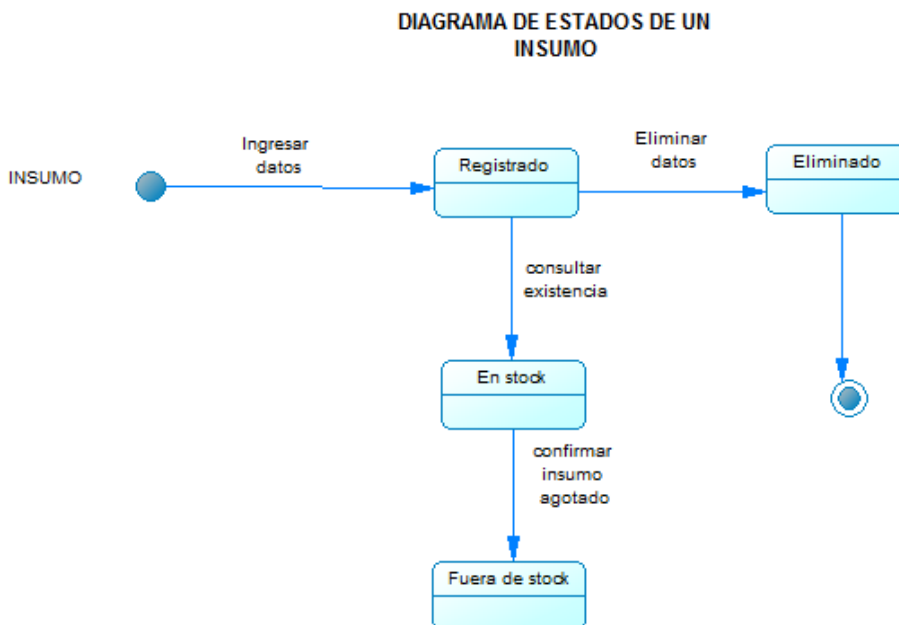


Fig. 34: Diagrama de estados de un Insumo.

Realizado por: Los Autores de la Tesis

Diagrama de estados de una enfermedad

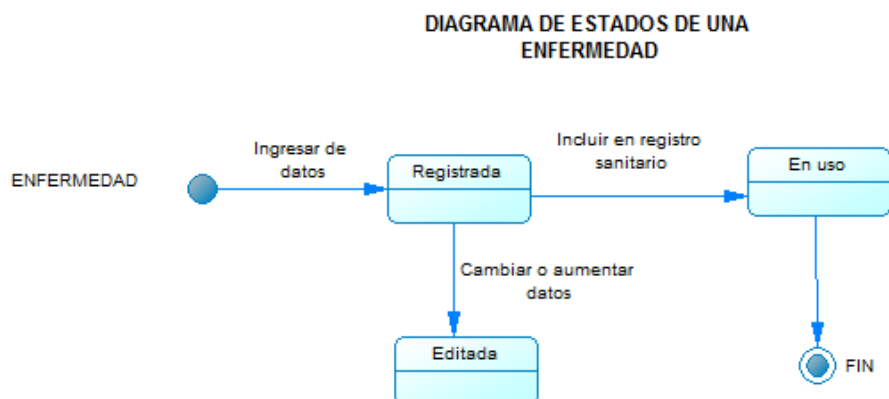


Fig. 35: Diagrama de estados de una Enfermedad.

Realizado por: Los Autores de la Tesis

Diagrama de estados de un Reporte

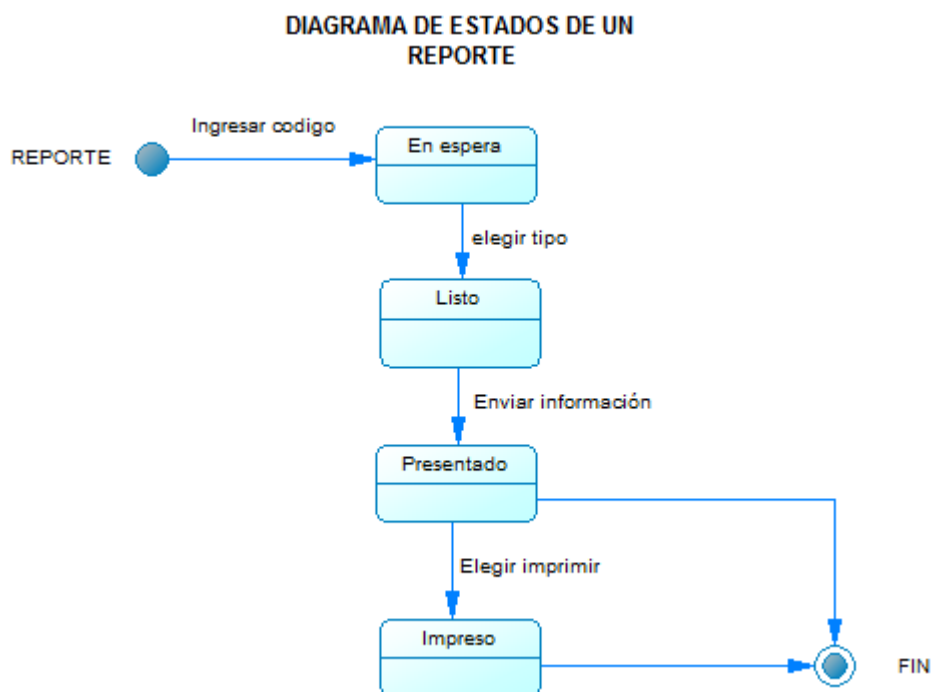


Fig. 36: Diagrama de estados de un Reporte.

Realizado por: Los Autores de la Tesis

Diagrama de objetos

Esta clase de diagrama pertenece a los diagramas de estructura y permite modelar las instancias de las clases del diagrama de clases. Muestra los objetos y sus relaciones, además de sus enlaces, pero en un momento concreto del sistema. Lo que significa que primero se debe decidir que situación se quiere representar del sistema.

Diagrama de objetos del módulo Usuarios y Seguridades

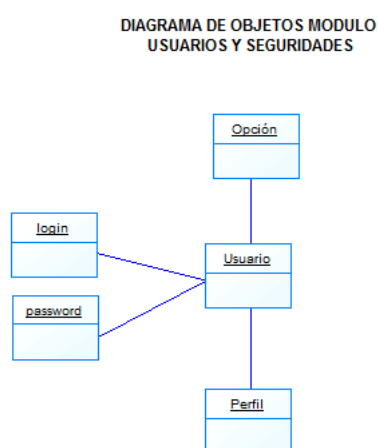


Fig. 37: Diagrama de objetos del Módulo de Usuarios y Seguridades.

Realizado por: Los Autores de la Tesis

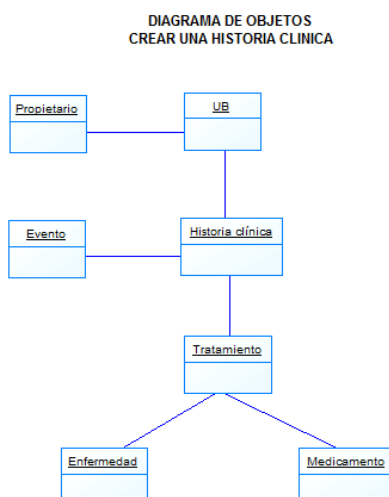


Fig. 38: Diagrama de objetos de Crear una Historia Clínica.

Realizado por: Los Autores de la Tesis

DIAGRAMA DE OBJETOS
CREAR UN REGISTRO DE CONTROL REPRODUCTIVO

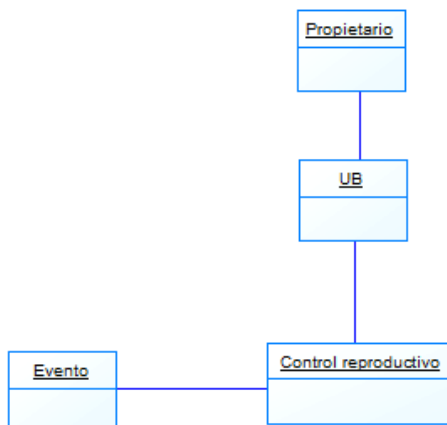


Fig. 39: Diagrama de objetos de Crear un registro.

Realizado por: Los Autores de la Tesis

DIAGRAMA DE OBJETOS
CONTROL DE SUPLEMENTOS

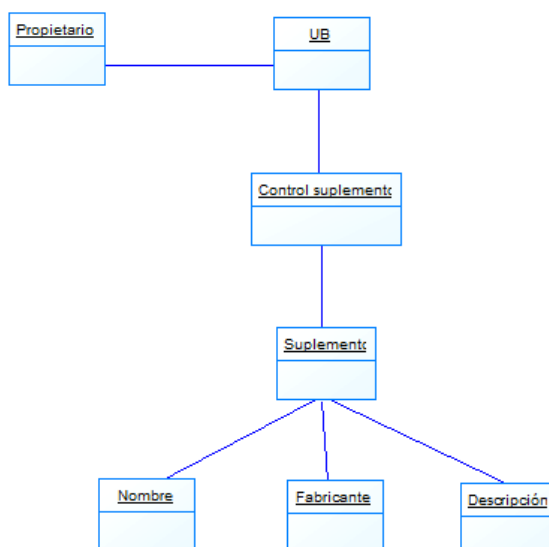


Fig. 40: Diagrama de objetos para el Control de suplementos.

Realizado por: Los Autores de la Tesis.

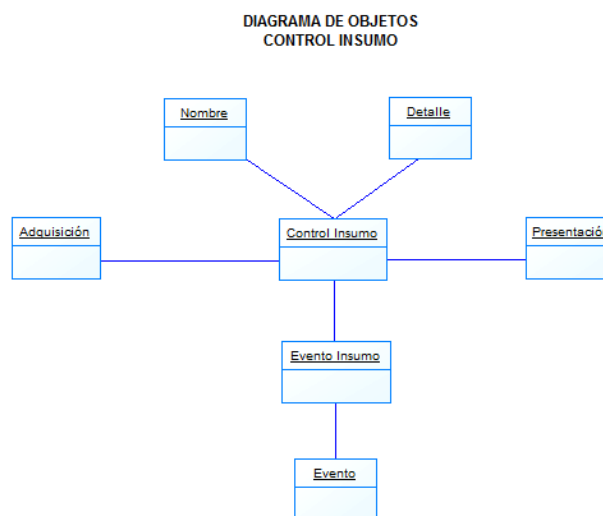


Fig. 41: Diagrama de objetos de Control Insumo.

Realizado por: Los Autores de la Tesis.

4.5.2.1.2 Modelo de interfaces

Especifica como interactúa el sistema con actores externos al ejecutar cada caso de uso. El objetivo del modelo de interfaces (abstractas) es definir las interfaces que el sistema ofrecerá para poder realizar la funcionalidad expresada en el modelo de casos de uso y en el modelo de comportamiento. No es necesario, sin embargo, entrar en detalles de la implementación de dichas interfaces.

4.5.2.1.3 Modelo del Dominio

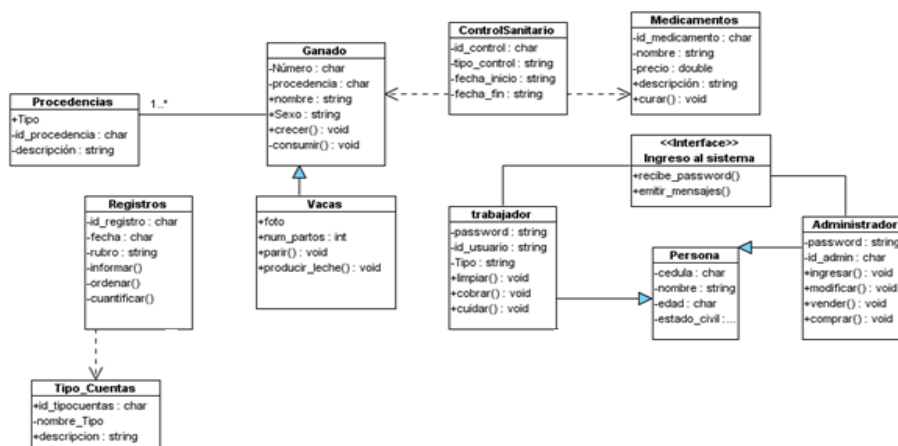


Fig. 42: Modelo de dominio.

Realizado por: Los Autores de la Tesis.

4.5.2.1.4 Diccionario de datos.

El diccionario de clases o diccionario de datos, describe textualmente las clases identificadas durante el modelo del dominio del problema. Este diccionario sirve como un glosario de términos.

A continuación se enumera todas las TABLAS de la base de datos.

NOMBRE DE LA TABLA	DESCRIPCIÓN	CLAVE PRIMARIA
ADQUISICIÓN	Esta tabla lleva un control de las adquisiciones para el manejo de las UB.	adqui_codigo
AUDITORÍA	Esta tabla contiene información referente a la auditoría que se realiza a las tablas críticas del sistema.	audi_codigo
CABECERA_ADQUISICIÓN	Esta tabla almacena datos generales de una adquisición a un proveedor.	capr_codigo
CATÁLOGO_ITEM	Esta tabla permite registrar los diferentes datos tipo catálogo.	caitem_codigo
COMPLEMENTO	Esta tabla almacena los complementos aplicados a las UB.	comp_codigo
COMPLEMENTO_SUPLEMENTO	Esta tabla almacena los diferentes suplementos que comprenden un complemento	cosu_codigo
CONTROL_COMPLEMENTO	Tabla de ruptura entre la tabla Complemento y UB, que contiene información referente al uso de los complementos en las UB	coco_codigo
CONTROL_INSUMO	Esta tabla permite llevar un control de la cantidad de todos los insumos existentes para el manejo de las UB	coin_codigo
CONTROL_REPRODUCTIVO	Esta tabla permite llevar un registro reproductivo individual de cada una de las UB en estado de preñez.	core_codigo
DETALLE_EVENTO_INSUMO	Esta tabla almacena datos de los insumos usados en un evento.	dein_codigo
ENFERMEDAD	Esta tabla contiene información sobre las enfermedades que afectan a las UB.	enfe_codigo
ENFERMEDAD_TRATAMIENTO	Esta tabla contiene información sobre los tratamientos que recibe una UB.	entra_codigo

EVENTO	Esta tabla contiene información referente a los diferentes eventos que se dan dentro de una finca ganadera	even_codigo
EVENTO_INSUMO	Tabla de ruptura entre los eventos y los insumos	evin_codigo
EVENTO_REPRODUCTIVO	Esta tabla almacena información sobre los Eventos Reproductivos de una UB.	esre_codigo
HIST_CLINICA_DETALLE	Tabla donde se almacena los detalles de una historia clínica	hcde_codigo
HISTORIA_CLINICA_UB	En esta tabla se almacenará información referente al control sanitario que va teniendo una UB.	hist_codigo
HISTORIAL_TRATAMIENTO	Almacena la dosis reales aplicadas UB por cada tratamiento	hitr_codigo
HISTORIAL_VACUNACION	Esta tabla contiene información sobre las vacunaciones que se las realiza a las UB.	hiva_codigo
INSUMO	Esta tabla contiene información sobre los insumos que se utilizan en el control reproductivo y sanitario	insu_codigo
MADUREZ	Esta tabla contiene datos referente a la edad de la UB	madu_codigo
MEDICAMENTO	Esta tabla contiene información sobre medicamentos a ser administrados a las UB.	insu_codigo
OPCION	Se almacena todas las opciones del sistema (módulos)	opci_codigo
PERFIL	Guarda los perfiles ó roles que va a tener el sistema.	perf_codigo
PERFIL_USUARIO	Esta tabla permite definir los diferentes roles o perfiles, que tienen los usuarios del sistema.	peus_codigo
PERMISO_OPCION_PERFIL	Tabla de ruptura entre las tablas perfil_usuario y opción	oppe_codigo
PRESENTACION	Esta tabla contiene información sobre las distintas presentaciones que puede tener un insumo.	pres_codigo
PROPIETARIO	Esta tabla contiene información referente a los propietarios de las UB	prope_codigo
PROVEEDOR	Almacena datos de los proveedores de insumos	prov_codigo
SUPLEMENTO	Esta tabla contiene información referente a los suplementos, generalmente, nutricionales que se administra a las UB	suple_codigo

TRATAMIENTO_MEDICAMENTO	Tabla de ruptura entre tratamiento y medicamento, contiene información sobre los medicamentos para un tratamiento de la UB.	trme_codigo
UNIDAD_BOVINA	Esta tabla contiene información de las UB	unbo_codigo_hc
USUARIO	Esta tabla contiene datos de los usuarios del sistema.	usua_codigo

Los campos que tiene cada una de las tablas, se describen a continuación.

TABLA ADQUISICIÓN		
NOMBRE DE LA COLUMNA	DESCRIPCIÓN	TIPO DE DATO
adqui_codigo	Clave principal de la tabla INSUMOS	INT4
adqui_cantidad	Cantidad comprada del insumo	INT4
adqui_unidad	Este campo almacena las unidades de un insumo adquirido	VARCHAR(6)
adqui_valorunitario	Costo por unidad del insumo	DECIMAL(10,2)

TABLA AUDITORIA		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
audi_codigo	Integer	Clave principal de la tabla Auditoria.
audi_fecha	Date & Time	Fecha en que se realizó la transacción.
audi_sentencia	Variable characters (2000)	Almacena la sentencia ejecutada en una transacción.
audi_tmodificada	Variable characters (50)	Nombre de la tabla que fue modificada.
audi_usuario	Variable characters (50)	Nombre del usuario que realiza una transacción, en una tabla de la base de datos

TABLA CABECERA_ADQUISICIÓN		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
caad_codigo	Integer	Clave principal de la tabla cabecera_adquisición
caad_descripcion	Variable characters (400)	Información adicional sobre la adquisición
caad_fecha	Date & Time	Fecha en que se realizó la adquisición de insumos.

caad_responsable	Variable characters (50)	Almacena el nombre del responsable de la adquisición de los insumos
caad_total	Decimal (10,2)	Almacena el total de la adquisición realizada
TABLA CATÁLOGO_ITEM		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
caitem_codigo	Variable characters (6)	Clave principal de la tabla Catálogo_item
caitem_codigopadre	Variable characters (6)	Describe el nombre o valor que tiene el registro padre.
caitem_descripcion	Variable characters (500)	Información adicional referente del item hijo.
caitem_item	Variable characters (50)	Es el nombre o valor que tiene los diferentes registros que tiene la tabla catálogo.
caitem_orden	Decimal (5,2)	Muestra el orden que se visualiza en el combo
caitem_tipo	Variable characters (6)	Describe el nombre del item hijo.

TABLA COMPLEMENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
comp_codigo	Integer	Clave principal de la tabla complemento
comp_descripcion	Variable characters (255)	Información adicional del complemento.
comp_estado	Variable characters (6)	Habilita ó deshabilita un complemento.
comp_nombre	Variable characters (50)	Nombre del complemento
comp_periodo	Variable characters (6)	Almacena los períodos en los que se aplica el complemento.

TABLA COMPLEMENTO_SUPLEMENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
cosu_codigo	Integer	Clave principal de la tabla complemento_suplemento
cosu_dosis	Decimal (10,2)	Dosis administrada de un complemento.
cosu_unidad	Variable characters (6)	Unidad de medida del complemento.

TABLA CONTROL_COMPLEMENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
coco_codigo	Integer	Clave principal de la tabla control_complemento
coco_fecha	Date & Time	Fecha en la que se uso o administró el complemento a la UB
coco_observacion	Variable characters (300)	Información adicional referente al registro del uso de los complementos en las UB

TABLA CONTROL_INSUMO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
coin_codigo	Integer	Clave principal de la tabla CONTROL INSUMO
coin_existencia	Decimal (10,2)	Especifica la cantidad existente de un insumo
coin_maximo	Decimal (10,2)	Especifica la cantidad máxima existente de un insumo
coin_minimo	Decimal (10,2)	Especifica la cantidad mínima que debe existir de un insumo
coin_observacion	Variable characters (400)	Información adicional sobre el uso del insumo

TABLA CONTROL_REPRODUCTIVO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
core_codigo	Integer	Clave principal de la tabla control_reproductivo
core_estado	Variable characters (4)	Estado del control reproductivo
core_estado_rep	Variable characters (30)	Aquí se pone el nombre del estado reproductivo
core_fecha	Date	En este campo se pone la fecha en la cual da inicio el evento reproductivo
core_fecha_parto	Date	Fecha en la cual se pone fin al evento reproductivo

core_fechafin	Date	En este campo se pone la fecha aproximada en la cual da fin al evento reproductivo.
---------------	------	---

TABLA DETALLE_EVENTO_INSUMO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
dein_cantidad	Decimal (10,2)	Cantidad de insumos utilizados en un evento
dein_codigo	Integer	Código de la tabla detalle evento insumo
dein_unidad	Variable characters (6)	Este campo almacena las unidades de un insumo.

TABLA ENFERMEDAD		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
enfe_codigo	Integer	Clave principal de la tabla enfermedad.
enfe_estado	Variable characters (6)	
enfe_nombre	Variable characters (50)	Nombre de la enfermedad que puede llegar a padecer la UB
enfe_sintoma	Variable characters (1000)	Describe los síntomas que presenta una enfermedad

TABLA ENFERMEDAD_TRATAMIENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
entra_codigo	Integer	Clave principal de la tabla enfermedad tratamiento.
entra_estado	Variable characters (6)	Estado de la enfermedad tratamiento
entra_nombre	Variable characters (100)	Nombre que se le dá al tratamiento
entra_observacion	Variable characters (200)	Información adicional del tratamiento
entra_tipo	Variable characters (6)	Indica el tipo de Tratamiento (Preventivo / Curativo)

TABLA EVENTO

NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
even_codigo	Integer	Clave principal de la tabla evento.
even_descripcion	Variable characters (400)	Descripción del evento.
even_estado	Variable characters (6)	Estado de un evento, puede ser activo o inactivo
even_nombre	Variable characters (50)	Nombre del Evento
even_observacion	Variable characters (400)	Información adicional referente al evento

TABLA EVENTO_INSUMO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
evin_codigo	Integer	Clave principal de la tabla EVENTO INSUMO
evin_fecha	Date	Almacena la fecha en el cual se realizó un evento y se utilizo insumos
evin_observacion	Variable characters (1000)	Observación de un evento.
evin_responsable	Variable characters (50)	Almacena el nombre del responsable que utiliza un insumo en un evento.

TABLA EVENTO_REPRODUCTIVO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
evre_codigo	Integer	Código de la tabla estado_reproductivo
evre_duracion	Integer	Indica el tiempo de duración entre el inicio y el fin del mismo.
evre_estado	Variable characters (6)	Habilita ó Deshabilita un Evento Reproductivo.
evre_nombre	Variable characters (50)	Nombre del Evento Reproductivo
evre_orden	Decimal (5,2)	
evre_unidadtiempo	Variable characters (6)	Este campo indica las unidades de tiempo para un evento reproductivo.

TABLA HISTORIA_CLÍNICA_DETALLE		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN

hcde_codigo	Integer	Código de la historia clínica detalle
hcde_fecha_control	Date & Time	Fecha de control de una UB.
hcde_fecha_tratamiento	Date & Time	Fecha de tratamiento de una UB.
hcde_observacion	Variable characters (255)	Información adicional sobre el control sanitario.
hcde_sintomas	Variable characters (1000)	Contiene los Síntomas que presenta la Unidad Bovina
hcde_tipo	Variable characters (6)	Tipo de historia

TABLA HISTORIA_CLÍNICA_UB		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
hist_codigo	Integer	Clave principal de la tabla historia_clinica_ub.
hist_fecha	Date & Time	Fecha del control sanitario.
hist_fecha_control	Date & Time	Fecha de control del tratamiento de una UB.
hist_fecha_creacion	Date	Fecha de creación de la Historia Clínica
hist_fecha_tratamiento	Date & Time	Fecha de tratamiento de una UB.
hist_foto	Image	Foto que va en la historia clínica
hist_madurez	Variable characters (4)	Madurez que va en la historia
hist_numero	Variable characters (10)	Número de la historia
hist_observacion	Variable characters (255)	Información adicional sobre el control sanitario.
hist_peso	Decimal (4,3)	Peso de la UB
hist_sintomas	Variable characters (1000)	Contiene los Síntomas que presenta la Unidad Bovina

TABLA HISTORIAL_TRATAMIENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
hitr_codigo	Integer	Clave principal de la tabla historial tratamiento.

hitr_dosis	Decimal (10,2)	Indica la dosis real aplicada en un tratamiento a una UB
hitr_unidad	Variable characters (6)	Unidad de medida real del medicamento

TABLA HISTORIAL_VACUNACIÓN		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
hiva_codigo	Integer	Clave principal de la tabla Historial Vacunación.
hiva_fecha	Date & Time	Fecha en la que se realizó la vacunación
hiva_fecha_proxima	Date & Time	Indica la próxima fecha de vacunación de las UB
hiva_observacion	Variable characters (1000)	Se registra alguna observación del proceso de vacunación a las UB

TABLA INSUMO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
insu_codigo	Integer	Clave principal de la tabla INSUMO
insu_detalle	Variable characters (100)	Detalles de un insumo
insu_estado	Variable characters (6)	
insu_nombre	Variable characters (100)	Nombre del insumo
insu_tipo	Variable characters (6)	Especifica qué tipo de insumo se está almacenando.

TABLA MADUREZ		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
madu_codigo	Integer	Código de la Madurez
madu_descripcion	Variable characters (100)	Descripción de la madurez
madu_desde	Integer	Tiempo en meses desde cuando empieza el tipo madurez
madu_estado	Variable characters (6)	Estado de la madurez, puede ser activo o inactivo

madu_hasta	Integer	Tiempo en meses hasta cuando finaliza el tipo de madurez
madu_nombre	Variable characters (50)	Nombre de la madurez
madu_observacion	Variable characters (200)	Información adicional de la madurez
madu_orden	Decimal (5,2)	Orden secuencial de la madurez
madu_sexo	Variable characters (6)	Sexo del tipo de madurez

TABLA MEDICAMENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
medi_codigo	Integer	Clave principal de la tabla medicamento.
medi_contraindicacion	Variable characters (500)	Indica cuando no usar el medicamento
medi_estado	Variable characters (6)	Habilita ó deshabilita un Medicamento
medi_indicacion	Variable characters (500)	Describe para que sirve el medicamento, es decir para que enfermedades
medi_nombre_comercial	Variable characters (50)	Nombre comercial del medicamento de acuerdo a la casa que lo produce.
medi_nombre_generico	Variable characters (50)	Nombre genérico del medicamento.
medi_observacion	Variable characters (400)	Información adicional importante referente al medicamento.
medi_posologia	Variable characters (1000)	Indica la dosis a administrar a la UB de acuerdo a ciertos parámetros tales como edad o peso.

TABLA OPCIÓN		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
opci_codigo	Variable characters (50)	Clave principal de la tabla opción
opci_descripcion	Variable characters (255)	Descripción de la opción
opci_texto	Variable characters (35)	Se almacena el texto que se muestra en el menú

TABLA PERFIL		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
perf_codigo	Integer	Clave principal de la tabla perfil.
perf_descripcion	Variable characters (255)	Describe las tareas que tiene cada perfil
perf_estado	Variable characters (6)	Habilita ó deshabilita un Perfil.
perf_nombre	Variable characters (20)	Nombre del perfil ó rol de los usuarios en el sistema.

TABLA PERFIL_USUARIO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
peus_codigo	Integer	Clave principal de la tabla perfil usuario.
peus_perfil	Variable characters (20)	Nombre del perfil de ingreso al sistema

TABLA PERMISO_OPCIÓN_PERFIL		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
oppe_codigo	Integer	Clave principal de la tabla PERMISO OPCION PERFIL

TABLA PRESENTACIÓN		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
pres_codigo	Integer	Clave principal de la tabla PRESENTACIÓN
pres_detalle	Variable characters (255)	Almacena detalles sobre las presentaciones de los insumos
pres_estado	Variable characters (6)	Estado de la presentación
pres_nombre	Variable characters (50)	Almacena los nombres de las presentaciones de los insumos

TABLA PROPIETARIO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
prope_codigo	Integer	Clave principal de la tabla propietario

prope_estado	Variable characters (6)	Habilita ó deshabilita un Propietario.
prope_imarca	Image	Imagen de la marca del propietario de la UB
prope_marca	Variable characters (6)	Es la marca que se le pone en el cuerpo de la UB, generalmente para distinguir a su propietario
prope_nombre	Variable characters (50)	Nombres completos del propietario de las UB
prope_observacion	Variable characters (100)	Información adicional del propietario

TABLA PROVEEDOR		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
prov_codigo	Integer	Clave principal de la tabla proveedor
prov_direccion	Variable characters (100)	Dirección del proveedor
prov_estado	Variable characters (6)	Estado del proveedor
prov_nombre	Variable characters (20)	Nombre comercial del proveedor
prov_telefono	Variable characters (13)	Número de teléfono de contacto del proveedor

TABLA SUPLEMENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
suple_codigo	Integer	Clave principal de la tabla suplemento
suple_descripcion	Variable characters (400)	Descripción detallada del suplemento, para que sirve.
suple_fabricante	Variable characters (50)	Es el nombre de las casa que fabrica el suplemento
suple_nombre	Variable characters (50)	Nombre comercial del suplemento para administrarse a las UB
suple_observacion	Variable characters (500)	Información adicional referente al suplemento

TABLA TRATAMIENTO_MEDICAMENTO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN

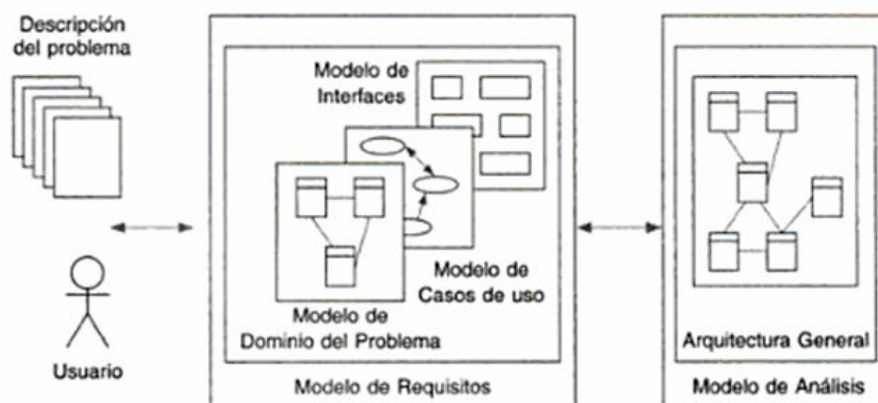
trme_codigo	Integer	Clave principal de la tabla TRATAMIENTO MEDICAMENTO
trme_dosis	Decimal (4,2)	Indica la dosis aplicada en un tratamiento a una UB
trme_unidad	Variable characters (6)	Unidad de medida del medicamento

TABLA UNIDAD_BOVINA		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
unbo_codigo_hc	Integer	Clave principal de la tabla UNIDAD_BOVINA
unbo_fbaja	Date	Fecha en el cual una UB ya no existe en la finca.
unbo_foto	Image	Es la foto de la UB
unbo_madre	Integer	En este campo se registra el nombre de la madre de la UB.
unbo_madurez	Variable characters (4)	Dato referente a la edad de la UB
unbo_nacimiento	Date	Fecha de nacimiento o llegada a la finca
unbo_nombre	Variable characters (50)	Nombre da la UB, en caso de tenerla
unbo_numero	Integer	Número que se le asigna a la UB
unbo_observacion	Variable characters (300)	Información adicional de la UB
unbo_padre	Integer	En este campo se registra el nombre del padre de la UB.
unbo_procedencia	Variable characters (6)	Procedencia de la UB, se refiere a si nacio en la finca, si fue comprada, etc.
unbo_raza	Variable characters (6)	Raza de la UB
unbosexo	Variable characters (6)	Sexo de la UB
unbo_tbaja	Variable characters (6)	Motivo por el cual una UB ya no existe en la finca.
unbo_tipoub	Variable characters (6)	Información referente al propósito que tiene la UB en la finca, por ejemplo puede ser lechera, de carne, o reproductor , etc.

TABLA USUARIO		
NOMBRE DE LA COLUMNA	TIPO DE DATO	DESCRIPCIÓN
usua_apellido	Variable characters (50)	En este campo se registra los apellidos del usuario
usua_cambioclave	Variable characters (6)	Obliga el cambio clave de un usuario
usua_celular	Variable characters (10)	Aquí se registra el número telefónico móvil del usuario.
usua_codigo	Integer	Es la clave principal de la tabla usuario.
usua_direccion	Variable characters (255)	En este campo se registra la dirección donde habita el usuario
usua_estado	Variable characters (6)	Habilita ó deshabilita un Usuario
usua_login	Variable characters (20)	Aquí se registra el nombre de usuario de ingreso al sistema.
usua_nombre	Variable characters (50)	Este campo permite registrar el o los nombres del usuario.
usua_password	Variable characters (30)	Aquí se registra la contraseña del usuario, de ingreso al sistema.
usua_telefono	Variable characters (10)	Aquí se registra el número telefónico fijo del usuario

4.5.2.2 MODELO DE ANÁLISIS

Luego de haber desarrollado y aceptado el modelo de requisitos, inicia el desarrollo del modelo de análisis siguiendo el modelo de casos de uso. El objetivo principal de este modelo es comprender y generar una arquitectura de objetos (capas) para el sistema con base en lo especificado en el modelo de requisitos. Aquí no se considera el ambiente de implementación, eso se hará durante el modelo de diseño, más bien es una representación conceptual.



El diagrama muestra conceptualmente el modelo de análisis junto con la arquitectura general de objetos, en relación con el modelo de requisitos anteriormente desarrollado.

Fig.43: Modelo de análisis.

Realizado por: Los Autores de la Tesis.

4.5.2.2.1 Arquitectura de Clases

El sistema informático manipula la información guardada en la base de datos a partir de interfaces de usuario, existe un grupo de objetos para el manejo de la funcionalidad de la aplicación y otro para interactuar con las entidades externas de la aplicación.

La jerarquía que tiene ciertas clases se hace evidente al momento de utilizar herencia como una de las principales cualidades del diseño orientado a objetos, lo que evita tener que estar implementando en cada clase ciertas características que se puedan heredar.

En la capa de interfaces es donde hay varias clases padres que tienen ciertas características que son heredadas a muchas clases hijas. Aunque en otras capas también se utiliza la herencia.

4.5.2.2 Diagramas de Secuencia

Ingresar al sistema

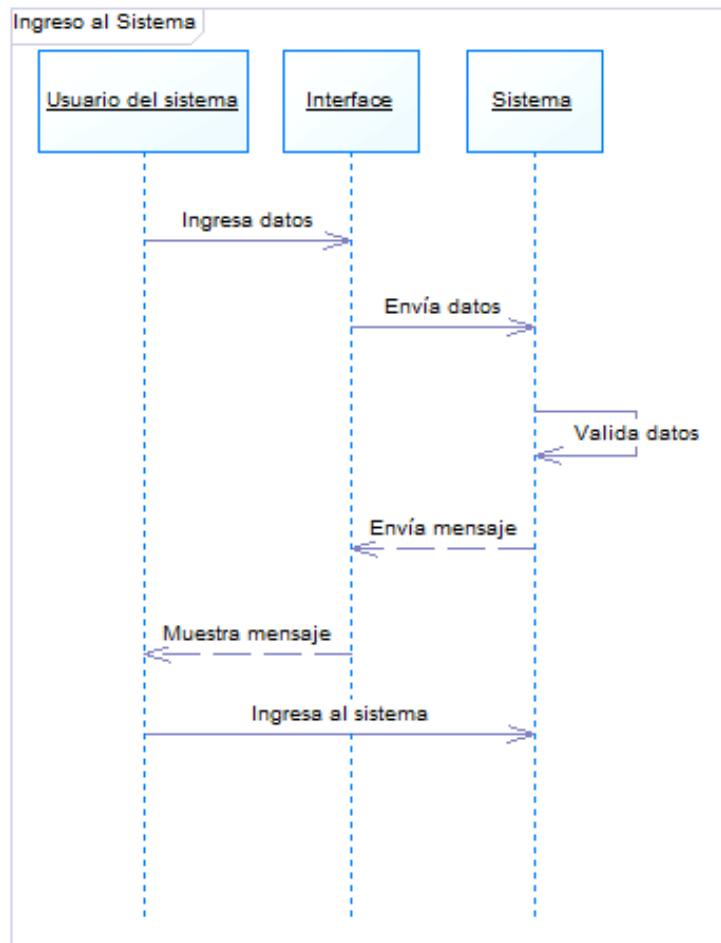


Fig. 44: Diagramas de Secuencia para Ingresar al Sistema.

Realizado por: Los Autores de la Tesis.

Crear permisos y perfiles

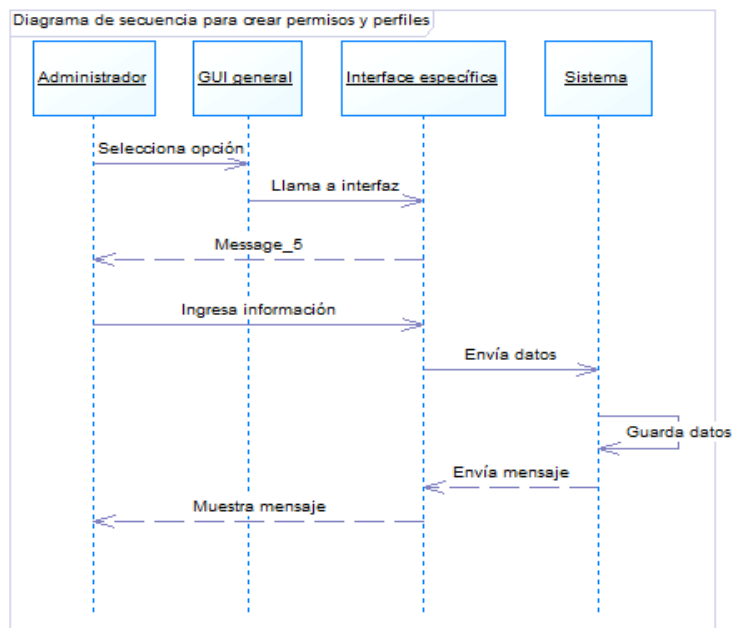


Fig. 45: Diagramas de Secuencia para Crear permisos y Perfiles.

Realizado por: Los Autores de la Tesis.

Crear usuarios

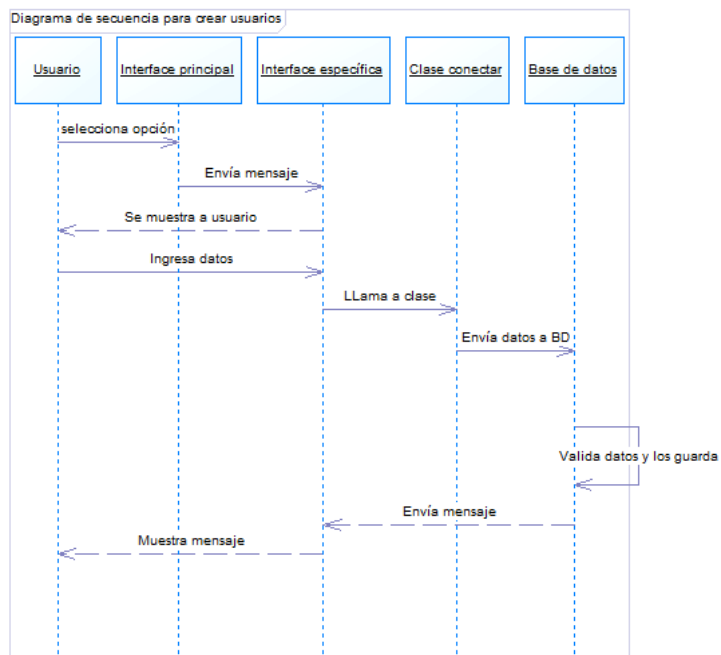


Fig. 46: Diagramas de Secuencia para Crear Usuarios.

Realizado por: Los Autores de la Tesis.

Editar y eliminar usuarios

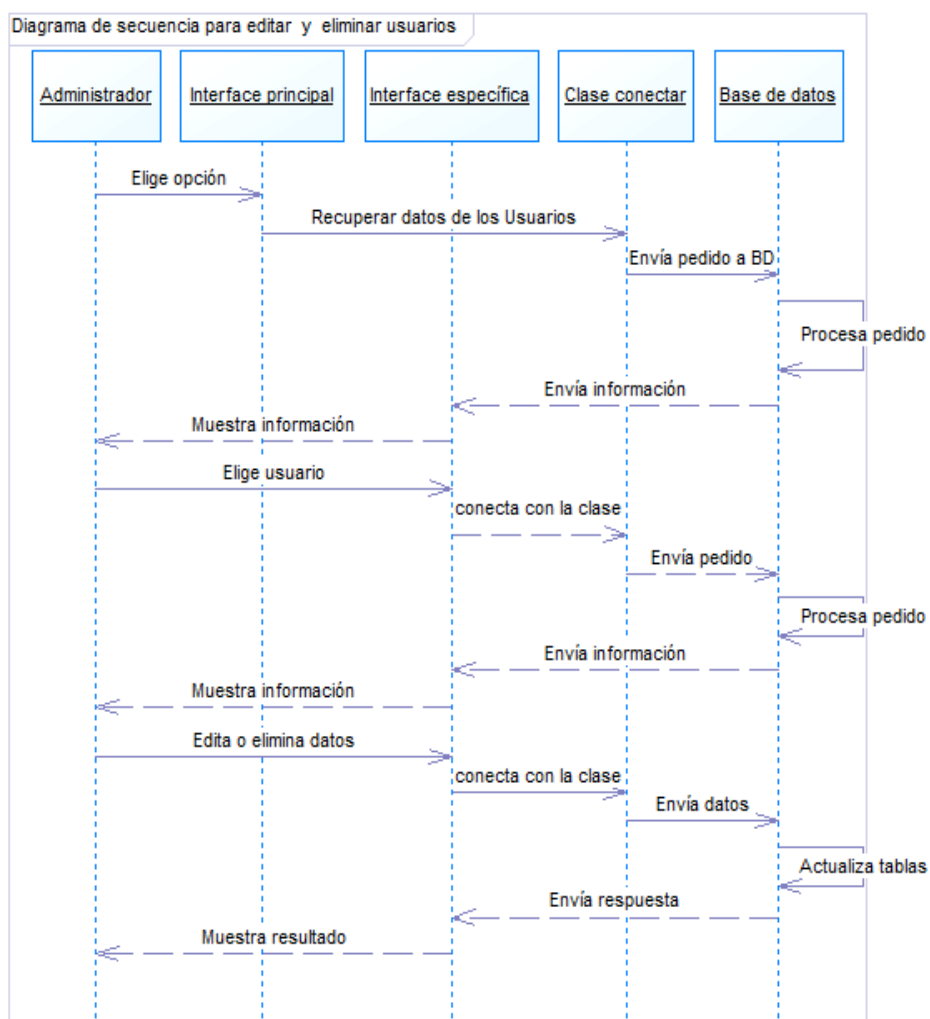


Fig. 47: Diagramas de Secuencia para Editar y Eliminar usuarios.

Realizado por: Los Autores de la Tesis.

4.5.2.2.3 Diccionario de clases según módulos

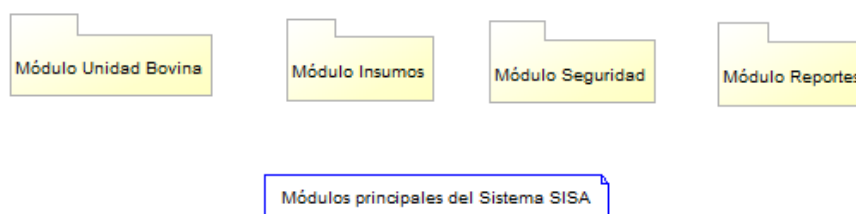


Fig. 48: Módulos principales del Sistema SISA.

Realizado por: Los Autores de la Tesis.

Módulo Unidad Bovina

Este módulo está compuesto por un conjunto de clases utilizadas para el manejo general de las Unidades Bovinas.

Módulo Insumos

Este módulo está compuesto por un conjunto de clases utilizadas para el manejo de los Insumos que se utiliza en la Finca, estos son de tres tipos:

- Uso general.
- Uso Medicinal, Medicamentos.
- Uso Alimenticio, Nutricional.

Módulo Seguridad

Este módulo está compuesto por un conjunto de clases utilizadas para el manejo de los Usuarios del Sistema, Perfiles, Permisos y Propietarios.

Módulo Reportes

Este módulo está compuesto por un conjunto de clases utilizadas para manejar los Reportes existentes en el sistema.

4.5.2.3 MODELO DE DISEÑO

4.5.2.3.1 Diseño de objetos

Aquí los objetos del sistema son refinados y se generan especificaciones muy detalladas, incluyendo las operaciones y atributos. Para este objeto utilizamos tarjetas, una por cada clase. Un ejemplo se grafica a continuación, en este caso se trata de la clase Interface.

Clase: Interface	
Descripción: toda la interacción con el usuario se hace por medio de la interface de usuario	
Módulo: InterfaceUsuario	
Estereotipo: Borde	
Propiedades	
Superclases	
Subclases	
Atributos:	
Despliega la Pantalla principal	
Envía el evento validar usuario	
Envía el evento "Salir"	

Fig. 49: Tarjeta que nos proporciona información de una clase.

Realizado por: Los Autores de la Tesis.

4.5.2.3.2Diseño y especificación de las clases

Consiste principalmente en transformar el modelo de clases lógico, en un modelo de clases de diseño. Este modelo recoge la especificación detallada de cada una de las clases, es decir, atributos, operaciones, métodos y el diseño preciso de las relaciones establecidas entre ellas, tales como agregación, asociación o jerarquía.

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

Los diagramas de Clases por definición son estáticos, esto es, representan que partes interactúan entre sí, mas no lo que ocurre, ni cuándo.

Más adelante, dentro de lo que son los diagramas UML, se muestra el diagrama de clases.

4.5.2.3.3 Diseño de algoritmos

Consiste en describir el algoritmo de los principales procesos que realiza el sistema. Para el caso se tiene: ingreso al sistema, administración de datos

El **algoritmo para ingreso al sistema**, de cualquier usuario, sin distinción, se detalla a continuación:

- ❖ Acceso a la pantalla de ingreso
- ❖ Ingreso de usuario
- ❖ Ingreso de contraseña
- ❖ Validación de la información por parte del sistema
- ❖ Si la información es correcta, entonces ingresa
- ❖ Si los datos son incorrectos, entonces
 - Mensaje informativo respecto al error
 - Permite hasta 2 veces más el ingreso de datos
 - Luego se cierra la página de acceso

Una vez detallado el primer algoritmo de ingreso al sistema, hay que distinguir los diferentes roles que tienen los usuarios, así por ejemplo, el de administrador, veterinario y usuario común.

Debido a que el administrador tiene permiso para acceder a todas las opciones que brinda el sistema, a continuación se describe algunos de los algoritmos que lleva a cabo.

Algoritmos del Administrador

Como lo dice su nombre este administra todo el sistema, sin embargo es posible que no tenga todo el conocimiento para administrar (crear, editar y eliminar) todos los módulos, pues existen algunos muy específicos para profesionales, por ejemplo, Agrónomos o Veterinarios.

Módulo Seguridades y Usuarios.

Algoritmo para crear un nuevo usuario.

- ❖ Ingreso al sistema

- ❖ Elige del menú principal la opción “Gestión Seguridad”
- ❖ Luego Usuarios
- ❖ Aparece una pantalla desde la cual se debe oprimir el botón “Nuevo”
- ❖ Se muestra otro interface, en la cual se ingresa los datos.
- ❖ Oprime el botón de “Guardar”.
- ❖ Sale de la interface
- ❖ Regresa al menú principal.

Algoritmo para actualizar los datos de un Usuario.

- ❖ Ingreso al sistema
- ❖ Elige del menú principal la opción “Gestión Seguridad”
- ❖ Luego Usuarios
- ❖ Aparece una pantalla en la cual se debe oprimir el botón “Mostrar Todos”
- ❖ Se muestra otro interface, en la cual están todos los usuarios del sistema, señalar un usuario y oprimir el botón editar.
- ❖ Se muestra otra interface con los datos del usuario a editar.
- ❖ Editar los datos.
- ❖ Oprimir el botón de “Guardar”.
- ❖ Sale de la interface
- ❖ Regresa al menú principal.

El sistema no permite eliminar usuarios, lo que si permite es inhabilitarlos, cambiar o quitar sus permisos.

Módulo UB.

Algoritmo de crear un nuevo registro de UB:

- ❖ Ingreso al sistema
- ❖ Elige del menú principal la opción “Gestión Unidad Bovina”
- ❖ Luego “Administrar UB” y “Unidad Bovina”
- ❖ Aparece una pantalla desde la cual se debe oprimir el botón “Nuevo”
- ❖ Se muestra otro interface, en la cual se debe ingresar los datos.
- ❖ Oprimir el botón de “Guardar”.

- ❖ Sale de la interface
- ❖ Regresa al menú principal.

Algoritmo de actualizar un registro de UB:

- ❖ Ingresar al sistema
- ❖ Elegir del menú principal la opción “Gestión Unidad Bovina”.
- ❖ Luego “Administrar UB” y “Unidad Bovina”.
- ❖ Aparece una pantalla en la cual se debe oprimir el botón “Mostrar Todos”
- ❖ Se muestra otra interface, en la cual están todas las Unidades Bovinas, señalar la UB y oprimir el botón editar.
- ❖ Se muestra otra interface con los datos de la UB a editar.
- ❖ Editar los datos.
- ❖ Oprimir el botón de “Guardar”.
- ❖ Sale de la interface
- ❖ Regresa al menú principal.

Algoritmo de eliminar un registro de UB:

Cabe aclarar en este momento que eliminar información, no es precisamente aquello, más bien es un cambio de estado de los datos de las UB, pues pasan de activo a inactivo. Por ejemplo si una UB ya no existe físicamente en la finca, no es que los datos de ella desaparecen de la base de datos, sino que ya no se muestran en los reportes.

Aclarado esto, a continuación se describe el algoritmo de eliminación de un registro de la UB

- ❖ Ingresar al sistema
- ❖ Elegir del menú principal la opción “Gestión Unidad Bovina”
- ❖ Luego “Administrar UB” y “Unidad Bovina”
- ❖ Aparece una pantalla en la cual se debe oprimir el botón “Mostrar Todos”
- ❖ Se muestra otra interface, en la cual están todas las Unidades Bovinas, señalar la UB y oprimir el botón editar.
- ❖ Se muestra otra interface con los datos de la UB.
- ❖ Oprimir el botón de “Eliminar”.

- ❖ Sale de la interface
- ❖ Regresa al menú principal.

Algoritmo para crear una historia clínica de la Unidad Bovina.

La historia clínica se crea cuando las unidades bovinas están enfermas, la interfaz que permite tal objetivo, permite registrar datos identificativos de la UB, de la enfermedad que padece, junto con la fecha y el tratamiento.

- ❖ Ingresar al sistema
- ❖ Elegir del menú principal la opción “Gestión Unidad Bovina”
- ❖ Luego “Control Sanitario” y “Historia Clínica”
- ❖ Aparece una pantalla desde la cual se debe oprimir el botón “Nuevo”
- ❖ Se muestra otra interface, en la cual se debe ingresar los datos.
- ❖ Oprimir el botón de “Guardar”.
- ❖ Sale de la interface
- ❖ Regresa al menú principal.

Algoritmo de actualizar una historia clínica.

Se lo hace cuando se quiere editar algún dato o también cuando se quiere completar algún campo que quedo pendiente, por ejemplo cuando se realiza un control posterior a la UB para ver cual fue el resultado del tratamiento dado.

- ❖ Ingresar al sistema
- ❖ Elegir del menú principal la opción “Gestión Unidad Bovina”
- ❖ Luego “Control Sanitario” y “Historia Clínica”
- ❖ Aparece una pantalla en la cual se debe oprimir el botón “Mostrar Todos”
- ❖ Se muestra otra interface, en la cual están todas las Unidades Bovinas, señalar el registro y oprimir el botón editar.
- ❖ Se muestra otra interface con los datos de la historia clínica a editar.
- ❖ Editar los datos.
- ❖ Oprimir el botón de “Guardar”.
- ❖ Sale de la interface
- ❖ Regresa al menú principal.

Algoritmo para generar un reporte.

- ❖ Ingresar al sistema
- ❖ Elegir del menú principal la opción “Reportes”
- ❖ Elegir la clase de reporte que quiere generar
- ❖ Generar el reporte.
- ❖ Si desea imprimir el reporte, oprimir el boton IMPRIMIR
- ❖ Sale de la interface.
- ❖ Regresa al menú principal.

Por considerar que sería repetitivo seguir describiendo más algoritmos de otros usuarios del sistema, no se describirá ninguno más.

4.5.2.3.4 Diseño de interfaces.

En el sistema se tiene básicamente las siguientes interfaces, aclarando que el resto de ellas varían solamente en los títulos de las mismas.

A continuación se muestra las interfaces del Sistema SISA.

Interfaz de ingreso al sistema



Fig. 50: Interfaz cuyo objetivo es recibir datos del usuario y su clave.

Realizado por: Los Autores de la Tesis.

Interfaz principal o menú



Fig. 51: Interfaz que es el menú principal del Sistema SISA.

Realizado por: Los Autores de la Tesis.

Formulario Máster

CODIGO	TIPO	NOMBRE	DETALLE
16	ALIMENTICIO	bolitos sabor a miel	alimenticio 3
18	ALIMENTICIO	MELASA	
15	ALIMENTICIO	platanos	alimenticio 2
14	ALIMENTICIO	sal en grano	alimenticio 1
12	MEDICINAL	aspirina	dolor de cuerpo
13	MEDICINAL	contrex	para la gripe
11	MEDICINAL	FINALIN	dolor de calabaza
9	USO GENERAL	ALAMBRE	Insumo metálico de uso múltiple
3	USO GENERAL	BALDE	Recipiente para contener sólidos o líquidos
4	USO GENERAL	BOTAS	Insumo para proteger las pies
8	USO GENERAL	CEDAZOS	Insumo para filtrar
5	USO GENERAL	CREMA PARA UBRES	Medicamento para curación de las ubres
6	USO GENERAL	DESINFECTANTE	Solución para desinfectar
1	USO GENERAL	GUANTES	Insumo para proteger las manos
2	USO GENERAL	JERINGUILLA	Insumo usado para inyectar
10	USO GENERAL	MANGUERA	Insumo para conducir líquidos o contener algún otro material.
17	USO GENERAL	Pico	Herramienta usada para varios propósitos
7	USO GENERAL	PILAS	Insumo para equipos de iluminación, sonido, etc.

Fig. 52: Interfaz principal del Módulo Insumos.

Realizado por: Los Autores de la Tesis.

Interfaz principal para administrar los diferentes procesos que se realizan en la finca y que se registran en el Sistema SISA.

Formulario Single



Fig. 53: Interfaz para Registro y actualización de registros.

Realizado por: Los Autores de la Tesis.

Interfaz de menor jerarquía, pero no de menor importancia, para recibir datos, editarlos o eliminarlos.

Formulario Filtro

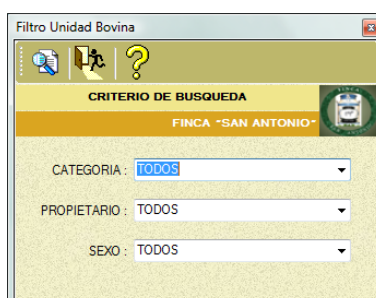


Fig. 54: Interfaz usada como filtro de información.

Realizado por: Los Autores de la Tesis.

Interfaz cuyo objetivo es filtrar la información que queremos desplegar en otra interfaz.

4.5.2.3.5 Diseño del Sistema

El objetivo del proceso de Diseño del Sistema de Información es en si la definición de la arquitectura del sistema así como el entorno tecnológico que le va a dar

soporte, junto con la especificación detallada de los componentes del sistema de información.

A partir de dicha información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la descripción técnica del plan de pruebas, la definición de los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial, éstos últimos cuando proceda.

4.5.2.3.6 Archivo o Base de Datos .

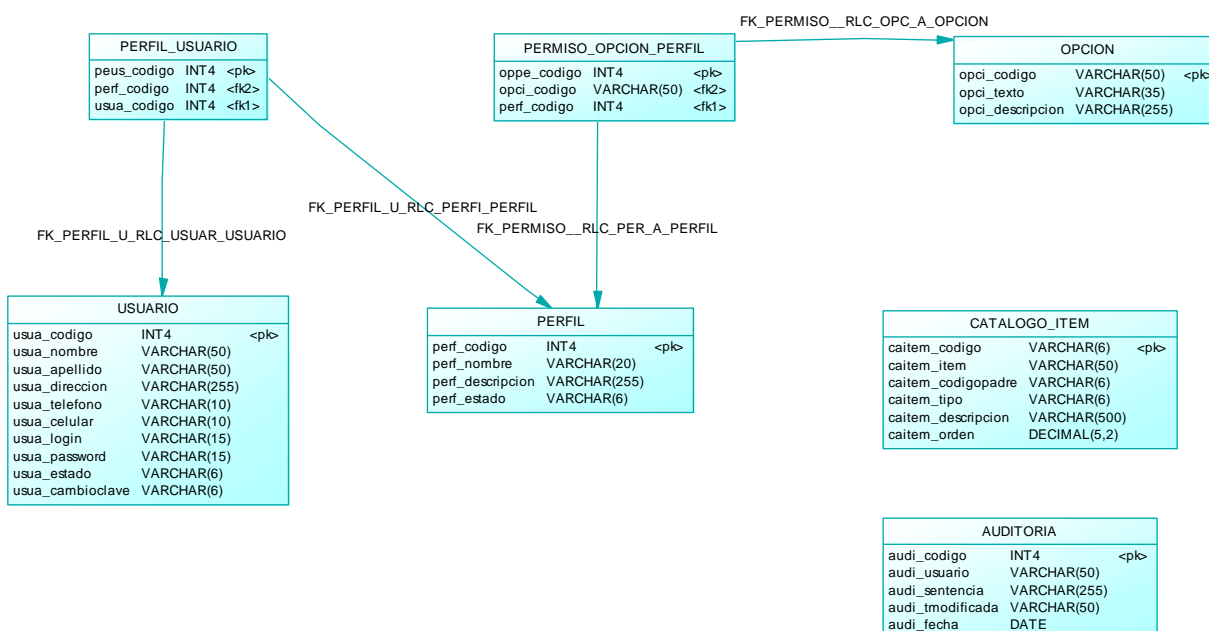


Fig. 55: Diagrama de base de datos Módulo Seguridad y Usuarios

Realizado por: Los Autores de la Tesis.

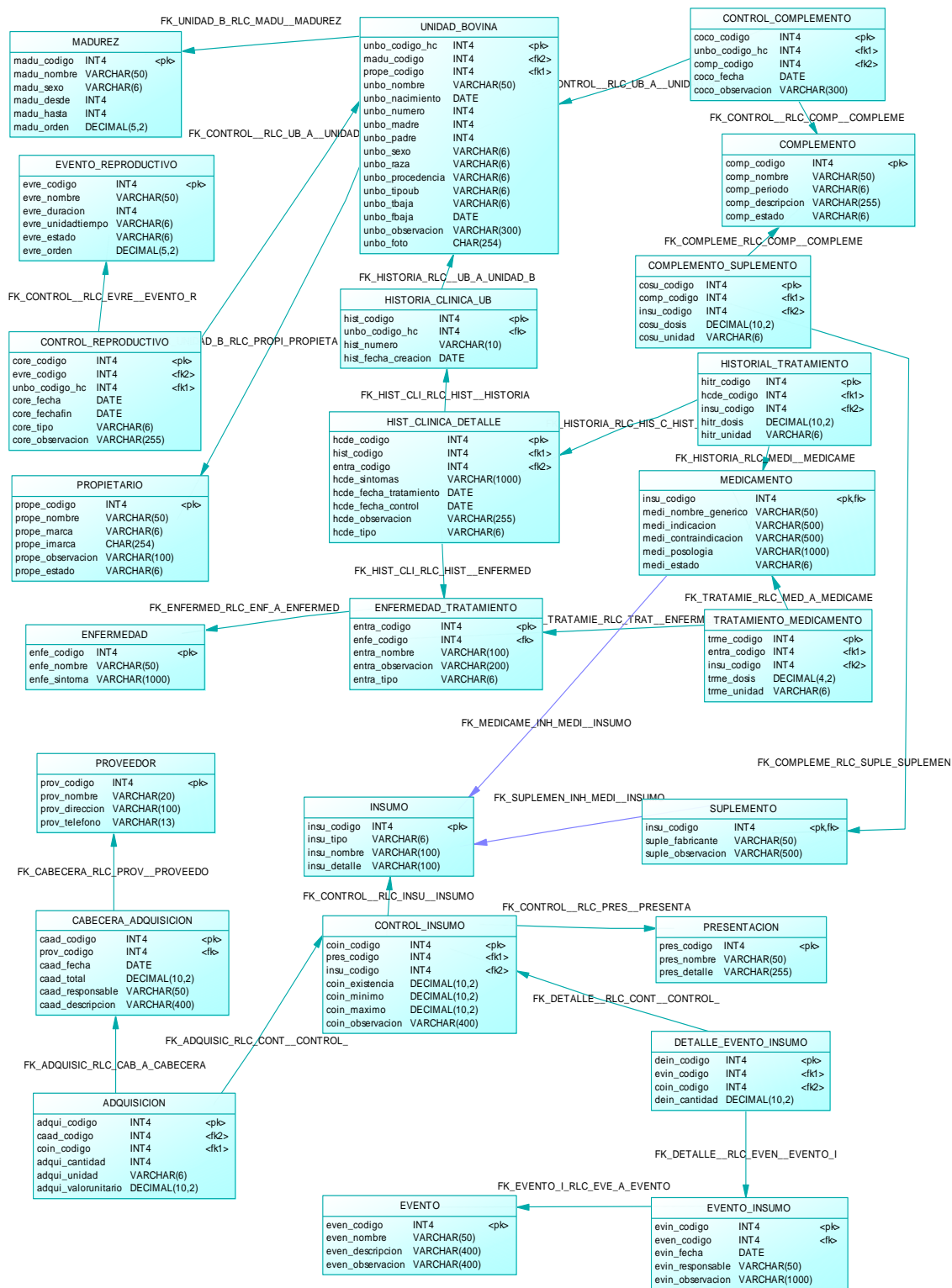


Fig. 56: Base de Datos Módulos Unidades Bovinas e Insumos
 Realizado por: Los Autores de la Tesis.

4.5.2.3.7 Diseño de menús y mapa de navegación

El menú principal del Sistema SISA, se muestra en la Fig. 57, allí se puede ver las opciones principales, que son:

- Inicio
- Ver
- Gestión Unidad Bovina
- Gestión Insumos
- Gestión Seguridad
- Reportes
- Sistema
- Ventana
- Ayuda

Cabe señalar que las opciones del menú principal se hacen visibles de acuerdo al perfil del usuario, lo que quiere decir que el menú completo solo se hace visible para el perfil "Administrador".

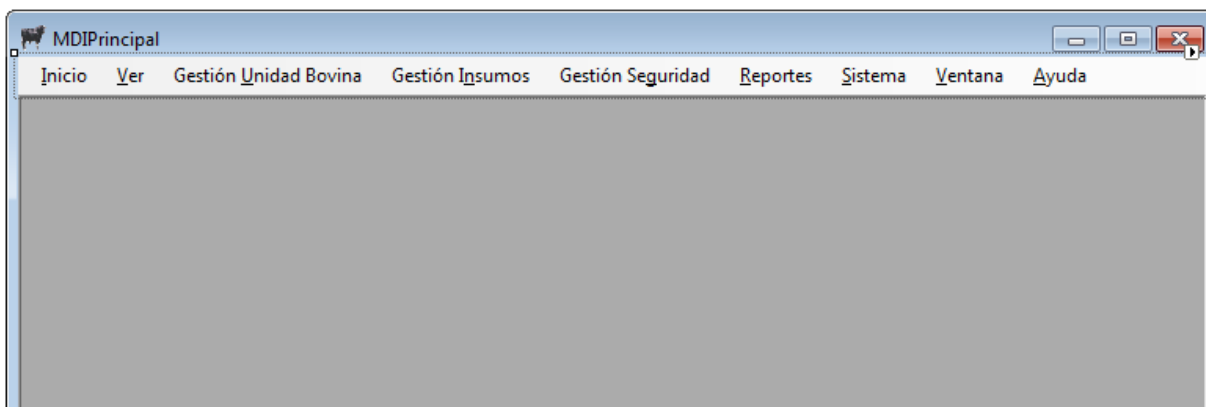


Fig. 57: Menú principal del Sistema SISA.

Realizado por: Los Autores de la Tesis.

A continuación se ve el mapa de navegación de las opciones más importantes del Sistema SISA.

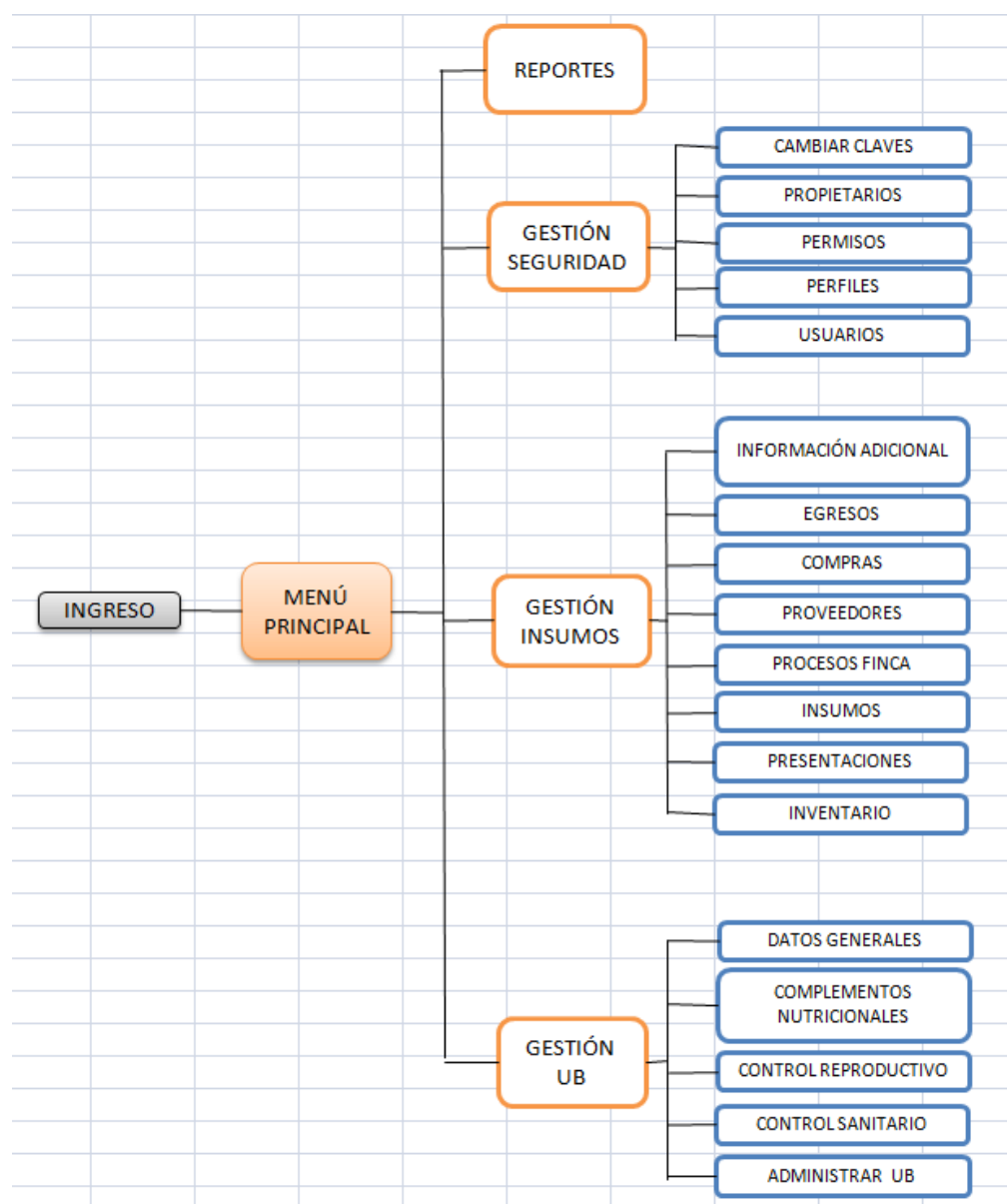


Fig. 58: Mapa navegacional.

Realizado por: Los Autores de la Tesis.

4.5.2.3.8 Diseño de Ayudas.

El sistema tiene implementado un documento de ayuda para el uso y manejo del mismo, de modo que el usuario puede guiarse en todo momento en el uso preciso del sistema, y por ende de cada formulario.

Para acceder a dicho documento simplemente se debe ir al menú principal del sistema y en la opción "Ayuda" hacer clic para que se despliegue una lista en

donde está la opción "Sistema" en la cual se debe hacer, de igual manera, clic y se muestra un documento tipo pdf en el cual se puede encontrar toda la información del sistema.

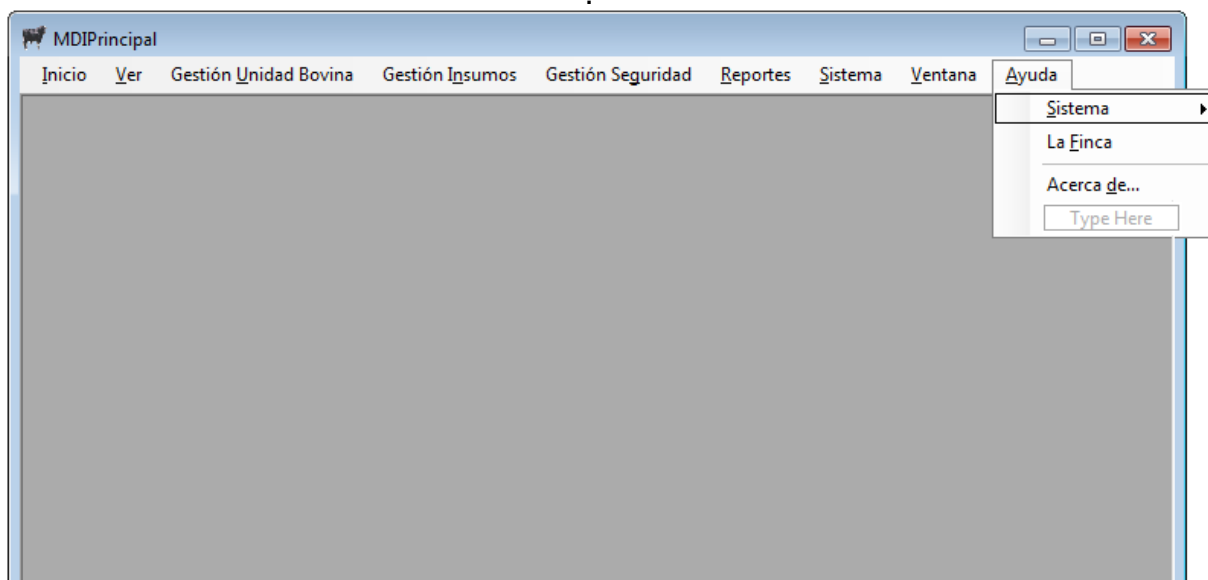


Fig. 59: Menú principal del sistema donde se ve la opción "Ayuda" y el submenú "Sistema."

Realizado por: Los Autores de la Tesis.

4.5.2.3.9 Diseño de arquitectura del sistema.

En una parte anterior de la tesis, capítulo 3.5, se hizo una descripción introductoria del tema, ahora se añade más información, pero más enfocada a este sistema.

Para el diseño de la arquitectura del sistema se ha tomado en cuenta en primer lugar la especificación del sistema, además de los requisitos de software. Lo que lleva en primer lugar a definir lo que es la especificación del sistema y los requisitos de software.

Especificación del sistema.- La especificación del sistema es un documento que sirve como fundamento para la ingeniería de hardware, software, base de datos y equipo humano. Describe la función y el rendimiento de un sistema así como sus restricciones, también describe la información que ingresa y sale del sistema.

Especificación de Requisitos de software (ERS).- Es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software. Los casos de uso también son conocidos como requisitos funcionales. Además de los casos de uso, la ERS también contiene requisitos no funcionales (o complementarios). Los requisitos no funcionales son requisitos que imponen restricciones en el diseño o la implementación (Como por ejemplo restricciones en el diseño o estándares de calidad).

En el apartado 4.5.2.1.1 se detalla los casos de uso y una serie de diagramas que permiten determinar lo que hace el sistema, que se puede resumir en lo siguiente:

Crear, editar, eliminar y ver registros de todo lo relacionado con:

- Usuarios del sistema
- Propietarios
- Unidades Bovinas
- Insumos, y Reportes.

Esto abarcaría lo que son los requisitos funcionales.

En cuanto a los requisitos no funcionales se señala que para la programación, se ha usado el lenguaje C#.

4.5.2.3.10 Métricas aplicadas al diseño.

Es importante, antes de nada, proporcionar el concepto de algunos términos que son claves para un entendimiento más cabal del tema.

Medida.- Valor asignado a un atributo de una entidad mediante una medición.

Medición.- Es el acto de determinar una medida.

Métrica.- Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición.

Indicador.- Es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software.

Error.- Fallo en un producto que se descubre antes de entregar el Software al usuario final.

Defecto.- Un fallo que se produce una vez que se ha entregado el producto final.

Las métricas son parte de un proceso que se puede resumir en el siguiente gráfico:

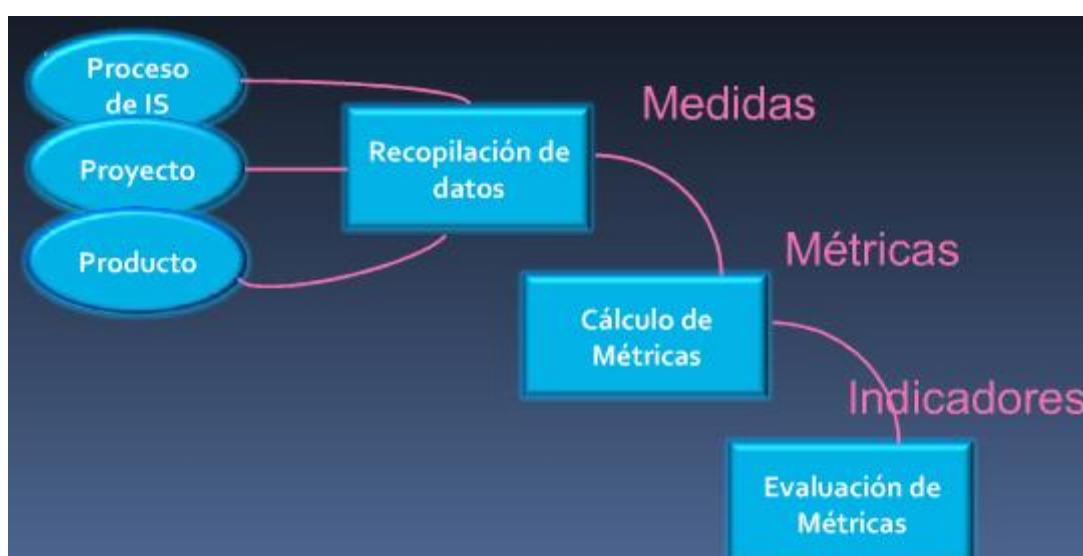


Fig. 60: Gráfico explicativo del proceso aplicado a un sistema, donde las métricas son parte del mismo.

Realizado por: Los Autores de la Tesis.

El sistema SISA se ha ido desarrollando de manera minuciosa para que la recopilación de datos sea la más correcta, acudiendo a las fuentes mismas de la información, en las diferentes etapas, de modo que el resultado sea un buen producto de software. Tomando en cuenta temas como:

- **La encapsulación y la ocultación** de la información aplicada en las diferentes clases.
- **La herencia**, aplicada principalmente en la capa de presentación.
- **La abstracción**, concentrándose en los detalles esenciales. Teniendo en cuenta que una clase es una abstracción, y que las métricas Orientadas a Objetos hace mediciones de la clase, se ha prestado mucha atención en el

número de instancias por clase así como el número de clases parametrizadas por aplicación.

4.5.2.4 CODIFICACIÓN Y PROGRAMACIÓN.

En el apartado 4.3, página 63 de la tesis, ya se desarrolló este tema.

4.5.3 DIAGRAMA GLOBAL DEL SISTEMA.

4.5.3.1 Arquitectura del sistema

La arquitectura de una aplicación viene a ser una vista conceptual de su estructura, es como diseñar los planos de una construcción, viene a ser la base sobre lo cual se va a implementar o desarrollar cualquier proyecto de ingeniería de software.

Luego del análisis correspondiente, se ha decidido que la mejor opción a implementar en el sistema, es la arquitectura de 3 capas. Pero primero es importante hacer una breve descripción de las arquitecturas multinivel.

En la etapa de diseño de los sistemas informáticos en la actualidad se usa la arquitectura multinivel o por capas, debido a que al dividir en capas la programación se logra independencia, pudiendo dividir o asignar por separado a diferentes grupos de trabajo la implementación de código.

Una ventaja muy importante de este tipo de arquitectura es al momento de hacer cambios, pues se lo hace unicamente en la capa correspondiente, lo cual permite desarrollar arquitecturas escalables.

Este sistema utiliza una arquitectura de 3 capas, por ser el diseño más utilizado en la actualidad.

1. Capa de Presentación o interfaz de usuario.
2. Capa de Negocio
3. Capa de Datos

Capa de presentación.- Esta es la capa que interactúa o ve el usuario, es la parte visible del sistema, por medio de ella un sistema recibe y muestra información, razón por la cual es importantísimo que esta sea amigable, entendible y fácil de usar.

Esta capa tiene contacto únicamente con la capa de negocio

A continuación se muestra un gráfico donde se puede ver en detalle la capa de presentación, SISA.Win, con sus carpetas que agrupan, dentro, un conjunto de interfaces.

Aquí están abiertas las carpetas de Insumo y la carpeta Principal permitiendo ver el conjunto de interfaces que contiene cada una.

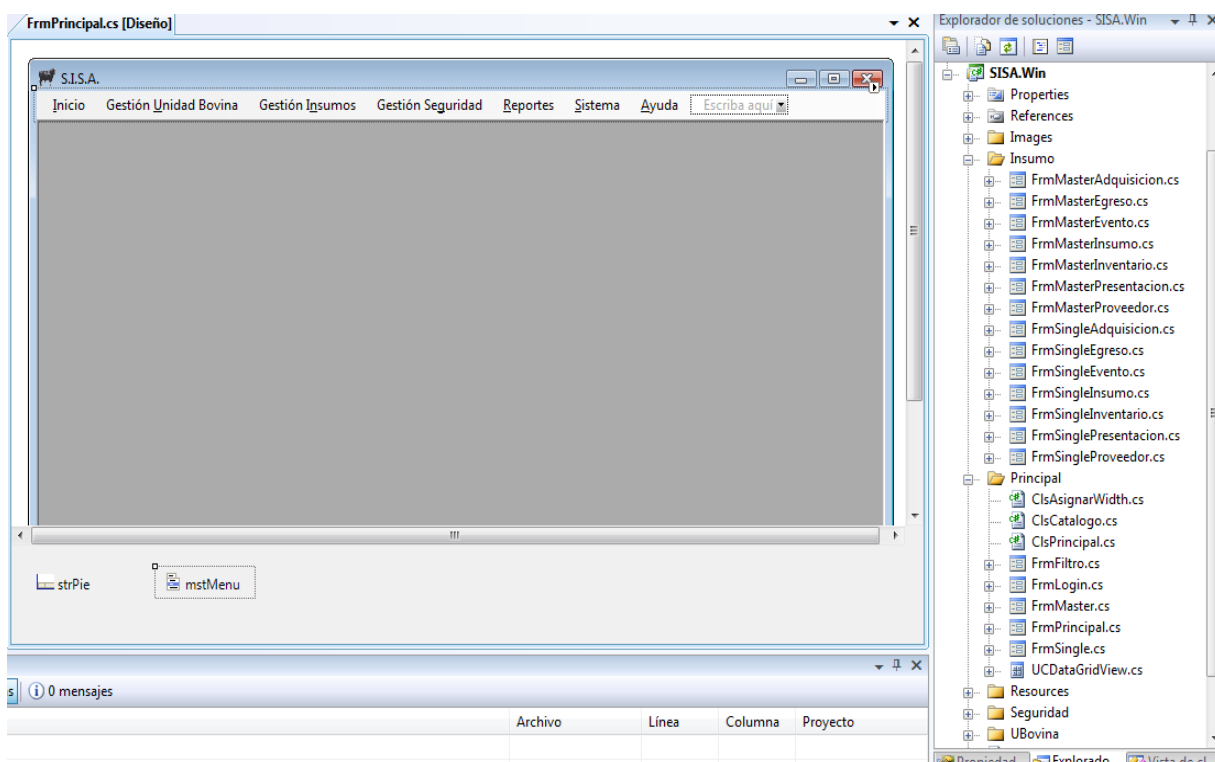


Fig. 61: Capa de Presentación del sistema SISA.

Realizado por: Los Autores de la Tesis.

Capa de negocio.- En esta capa residen las clases y métodos que proporcionan funcionalidad al sistema y se establecen todas las reglas que deben cumplirse, pues recibe las peticiones del usuario y se envían las respuestas tras el proceso.

Esta capa tiene contacto con la capa de presentación y la capa de datos.

En el siguiente gráfico se puede ver como está implementada esta capa, SISA.Negocio, además se ve el contenido de dos de sus carpetas; la carpeta NInsumo y la carpeta Nseguridad con todas las clases que contiene cada una. Una carpeta no muestra su contenido, es la carpeta Nubovina, por falta de espacio en pantalla.

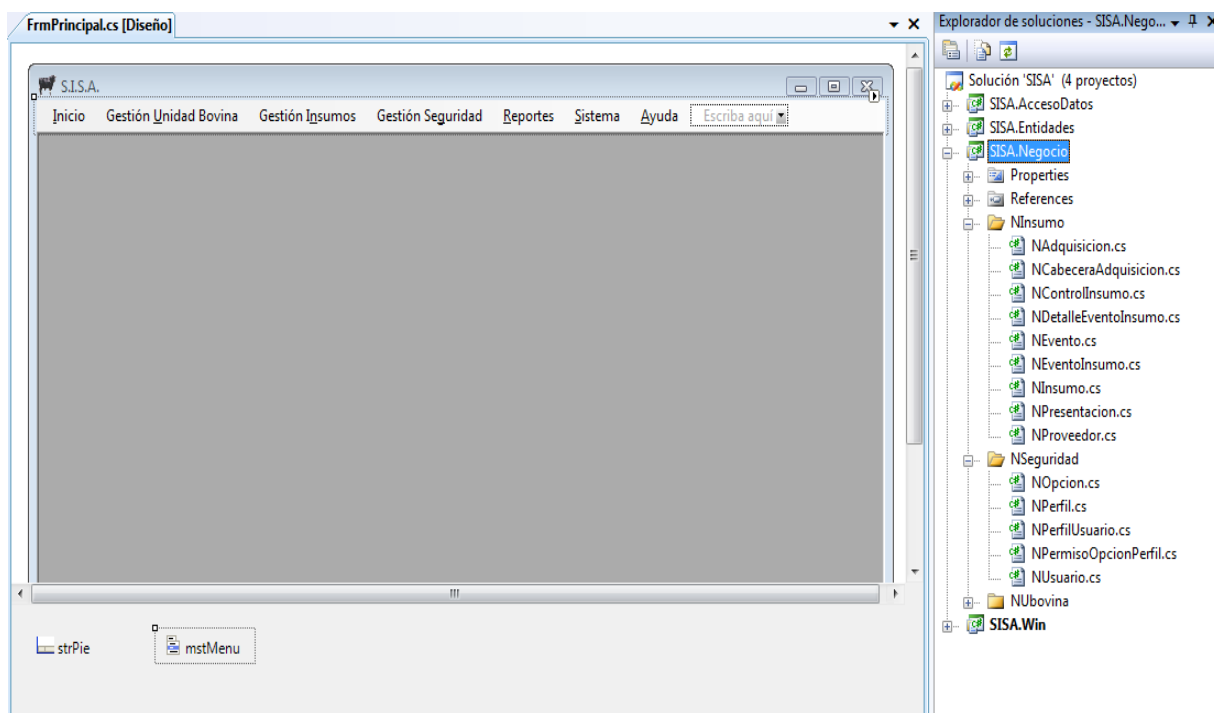


Fig. 62: Capa de Negocios del sistema SISA.

Realizado por: Los Autores de la Tesis.

Capa de datos.- Aquí residen las clases donde se implementan los métodos con sus respectivas sentencias SQL que permiten acceder a los datos que contiene la Base de Datos, que para este caso esta implementada en Postgres.

El siguiente gráfico permite ver el contenido de la capa de negocios del sistema, SISA, con sus respectivas carpetas. Para ejemplo, unicamente se ha desplegado el contenido de la carpeta ADInsumo, ADManager y ADSeguridad con todas sus clases, faltando solo ver el contenido de la carpeta ADUbovina que contiene otra serie de clases referentes a ese módulo.

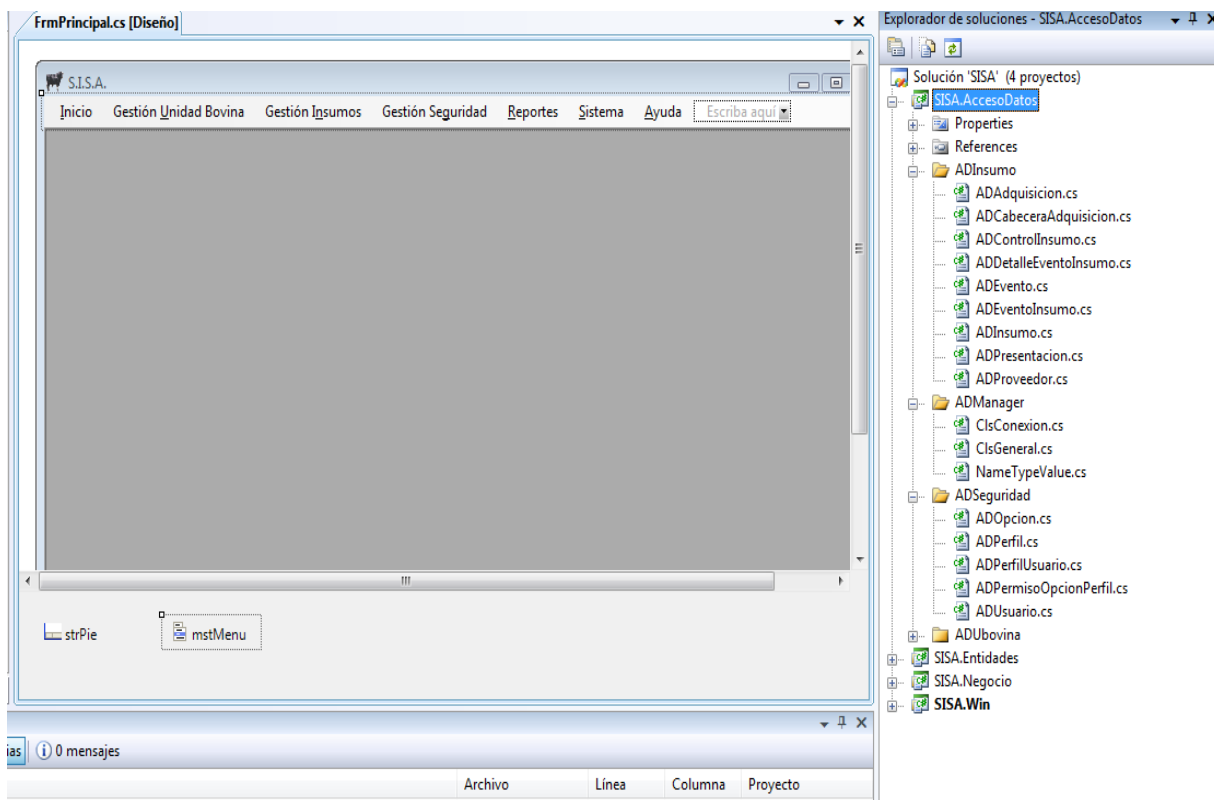


Fig. 63: Capa de Datos del sistema SISA.
Realizado por: Los Autores de la Tesis.

4.5.3.2 Diagrama de Componentes

Un componente es una pieza de software que realiza una función bien definida y posee una interfaz bien definida.

Beneficios

- La división en componentes reduce la complejidad, permite la reutilización y acelera el proceso de ensamblaje de software.
- Los creadores de componentes pueden especializarse creando objetos cada vez más complejos y de mayor calidad.
- La interoperabilidad entre componentes de distintos fabricantes aumenta la competencia, reduce los costos y facilita la construcción de estándares.
- El software se hace cada vez más rápido, de mejor calidad y a menor costo

- Los costos de mantención del software se reducen.

Este tipo de diagrama es uno de los varios diagramas UML. Representa como un sistema de software es dividido en componentes y muestra las dependencias entre estos.

Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables o paquetes.

No es necesario que un diagrama incluya todos los componentes del sistema, generalmente se lo hace por partes.

Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

Por ser un tipo de diagrama que no es de alta prioridad no se ha desarrollado para el sistema, sin embargo es conveniente poner alguna información al respecto por ser parte del Manual Técnico o Manual del Sistema.

4.5.3.3 Diagrama de Distribución.

Este tipo de diagrama se enfoca específicamente al hardware de un sistema, a diferencia de los Diagramas de Componentes. Su elemento principal es un nodo, que es un nombre genérico para todo tipo de recurso de cómputo y se representa mediante un cubo.

Al interior del nodo va información sobre el nodo, que puede ser texto o inclusive componentes de software.

A continuación se ve un Diagrama de Distribución que puede ser aplicado al sistema, de acuerdo a los recursos que se ha utilizado para desarrollarlo e implementarlo.

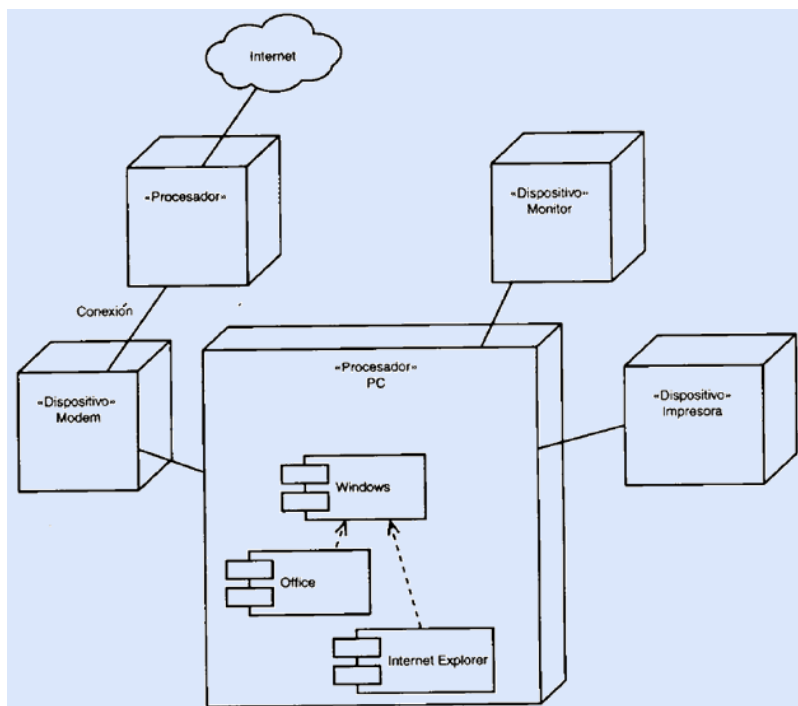


Fig. 64: Diagrama de Distribución.

Fuente: http://jaimemontoya.com/informaticprojects/uml_archivos/image017.png

4.5.4 INSTRUCCIONES DE INSTALACIÓN, EJECUCIÓN, REINICIO Y DESINSTALACIÓN.

Este tema sin tener demasiadas complicaciones, es importante abordarlo desde lo básico, es decir los requerimientos mínimos de software, para que funcione. A continuación se detalla lo principal.

INSTALACIÓN DE LOS PROGRAMAS.

Para la base de Datos.

Hay que instalar el postgresQL, lo cual está descrito de manera pormenorizada en el Manual de Instalación que está en el capítulo 4.6.

Creación del ODBC.

Para que el sistema SISA se conecte con la base de datos implementada en PostgreSQL se necesita crear el ODBC para que cada proceso del sistema se ejecute.

Esto está descrito en el Manual de Instalación en el capítulo 4.6.

Framework 3.5

El framework es un componente de software que viene incluido en los sistemas operativos de Microsoft Windows, el cual proporciona soluciones precodificadas para requerimientos comunes de los programas y gestionan la ejecución de programas escritos para este framework.

Estas soluciones cubren un gran rango de necesidades de la programación e incluyen soluciones en áreas como: interfaz de usuario, acceso a datos, conectividad a base datos, criptografía, desarrollo de aplicaciones web, algoritmos numéricos y comunicación de redes.

Este es requisito previo que debe estar instalado en las máquinas cliente, así como en el servidor. Como se explicó anteriormente este requisito ya viene instalado, lo que hay que asegurarse es que tenga una actualización reciente.

4.5.4.1 Requerimientos de Hardware y Software.

A continuación se detalla los requisitos necesarios para que funcione el sistema.

- Tener instalado Adobe Reader versión 7.0 o superior.
- Framework 3.5
- Instalador ODBC de postgresQL
- PostgreSQL en el Servidor.

4.5.5. SEGURIDAD Y CONTROL

4.5.5.1 Seguridad en el acceso a la información.

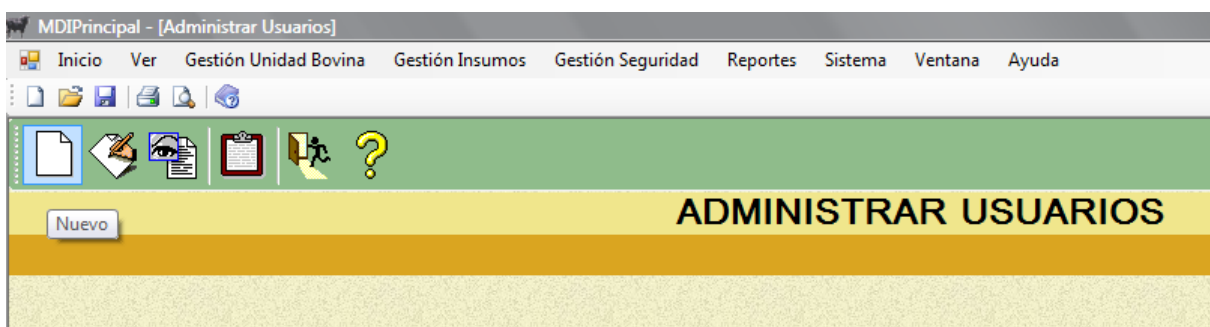
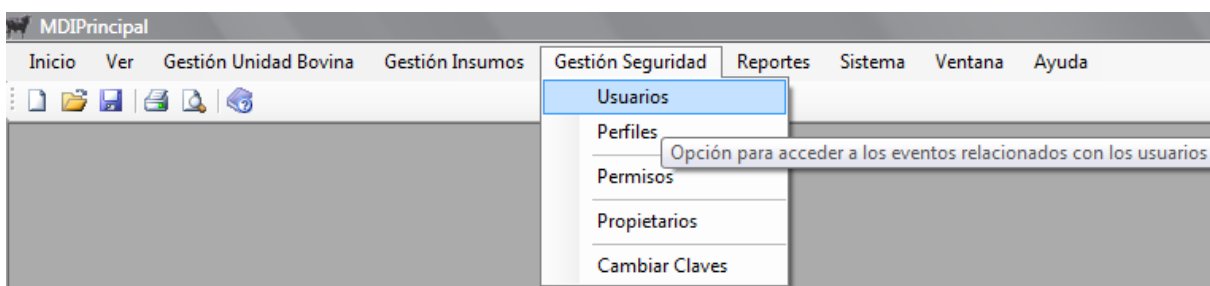
Este tema es bastante delicado, razón por la cual se ha creado una serie de perfiles que permiten un acceso restringido a las opciones que tiene el sistema lo que se refleja en seguridad en el acceso a la información. Pero esto no es suficiente y se ha creado en la base de datos una tabla, "Auditoria", donde se guarda datos las sentencias SQL efectuadas, el nombre del usuario que las hizo, y en qué fecha.

Dicho lo anterior, a continuación se enumera las principales tablas en las que se realiza la auditoria, y son las siguientes:

- Unidades Bovinas
- Historias Clínicas
- Inventarios
- Compras

4.5.5.2 Creación de usuarios

Esta tarea está restringida únicamente al perfil "Administrador", el mismo que por medio del módulo de Seguridad elige la opción "Usuarios", luego "Nuevo" y en la pantalla, que se muestra más adelante, lo crea, eligiendo su perfil con las opciones que tiene cada perfil.



The image shows a Windows application window titled "Usuarios". At the top, there is a green header bar with the text "CREAR USUARIO" and a sub-header "FINCA *SAN ANTONIO*" next to a small circular logo. Below the header, the form is organized into two columns. The left column contains text labels followed by input fields: "NOMBRES:", "APELLIDOS:", "TELEFONO:", "CELULAR:", and "DIRECCION:". The right column contains "USUARIO:", "CLAVE:", "ESTADO:" (with a dropdown menu currently set to "ACTIVO"), and "PERFIL:" (with a dropdown menu open, showing a list of roles: "ADMINISTRADOR", "PROPIETARIO", "USUARIO", and "VETERINARIO"). The "ADMINISTRADOR" role is currently selected in the dropdown. The window title bar shows "Usuarios" and standard Windows window controls (minimize, maximize, close).

Fig. 65: Secuencia de interfaces para la creación de un Usuario del sistema.

Realizado por: Los Autores de la Tesis.

4.5.5.3 Seguridad de acceso

El sistema brinda toda la seguridad de acceso necesaria, ya que se ha implementado clases que se encargan de validar los datos que los usuarios ingresan, sumado a esto se ha definido que el password tenga cuatro caracteres dado el tamaño de este proyecto y debido a que los usuarios del sistema son pocos, se considera que son suficientes.

Además el sistema tiene la clase "ClsCrypto.cs" que se encuentra en la carpeta "Principal" de la Capa "SISA.Win", que se encarga de encriptar los passwords para que ni en la base de datos se pueda verlos.

4.5.5.4 Seguridad de usuario

Este tema tiene estrecha relación con el anterior, pudiendo decir que se complementa.

El sistema tiene implementado una clase que permite, para seguridad del usuario, ofrecer un nuevo password en caso que por algún motivo la haya olvidado, este es provisional y le sirve para un solo ingreso ya que se le obliga al usuario a cambiar de clave.

4.5.5.5 Posibles usuarios

Los usuarios del sistema son:

- Los propietarios
- El administrador
- El veterinario
- El trabajador común.

En algunos casos se tiene que un mismo usuario tiene más de un perfil, como por ejemplo el usuario administrador, también es un usuario propietario o el usuario veterinario que también es usuario propietario.

4.5.5.6 Copias de Seguridad.

Es muy importante respaldar la información que maneja la finca, la cual podrían hacerse de manera periódica: semanal, mensual, trimestral, etc., esto lo determina el administrador del sistema, aunque se recomienda hacerlo trimestralmente, esto debido al volumen de datos que se maneja en la finca "San Antonio".

Para realizar los respaldos necesarios, se utiliza pgAdmin III. A continuación se describe los pasos necesarios para realizar los back ups de la base de datos.

1.- Ir al pgAdmin III y abrirlo.

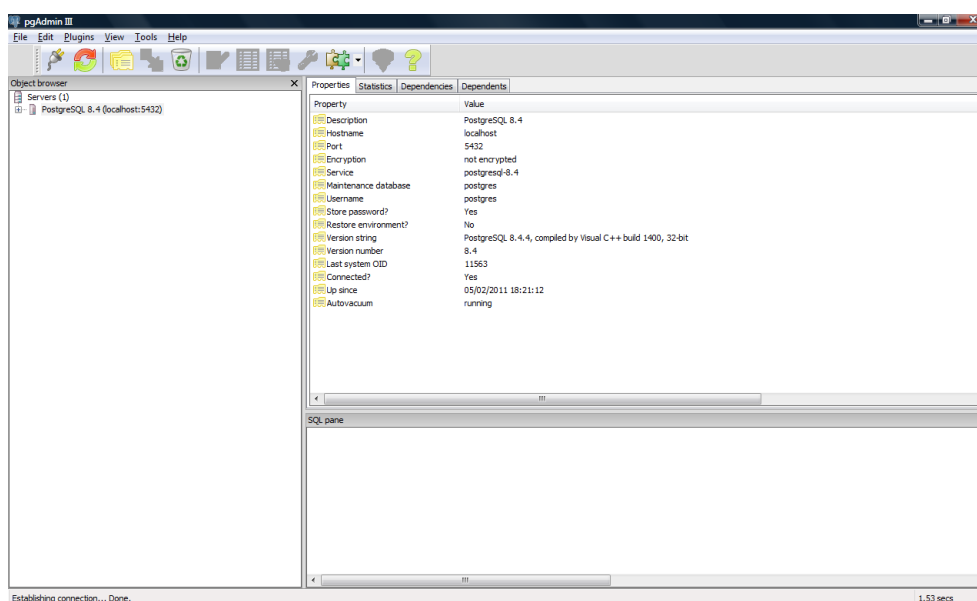


Fig.66: Pantalla principal del pgAdmin III.

Realizado por: Los Autores de la Tesis.

Hacer clic sobre la pestaña que está junto a PostgreSQL 8.4 (localhost:5432), mostrándose la siguiente pantalla.

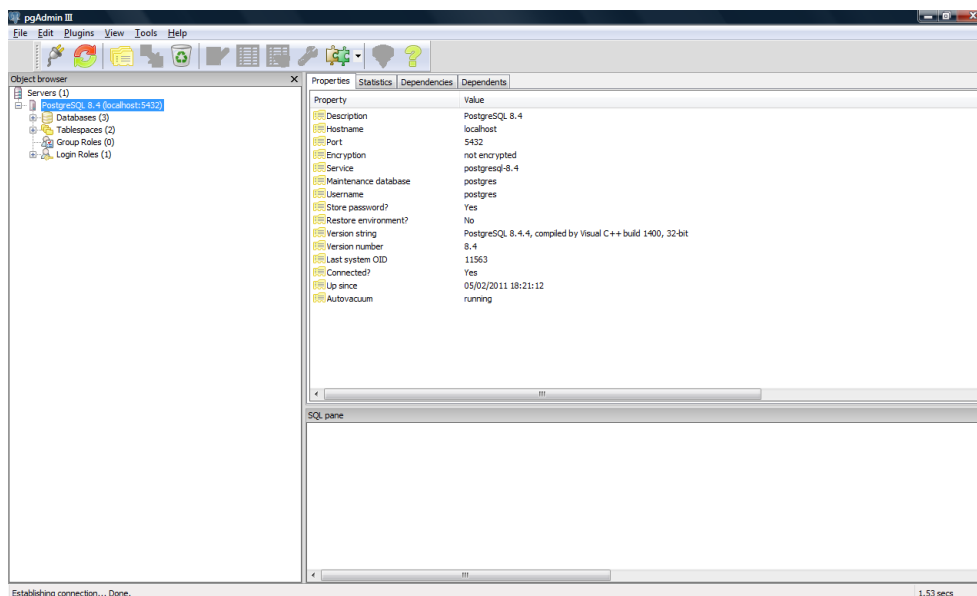


Fig.67: Pantalla principal del pgAdmin III.

Realizado por: Los Autores de la Tesis.

Aquí se puede ver las bases de datos que existen, elegir la base de datos, SISA_DB y hacer clic izquierdo con el mouse sobre este nombre y se despliega una lista, como se ve en la siguiente gráfica.

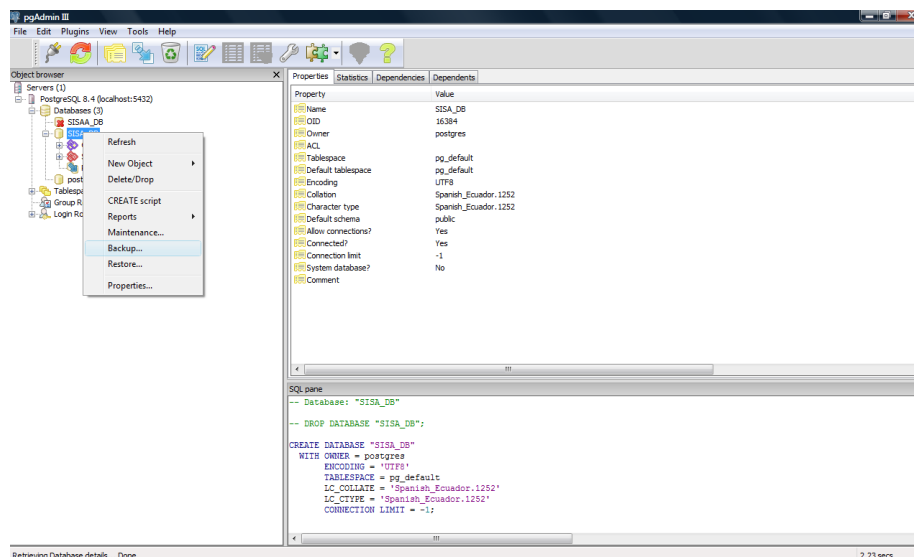


Fig.68: Pantalla principal del pgAdmin III.

Realizado por: Los Autores de la Tesis.

Como se puede ver entre las opciones aparece la palabra "Backup", sobre ella hacer clic y aparece la siguiente pantalla, donde se registra los datos que nos pide llenar.

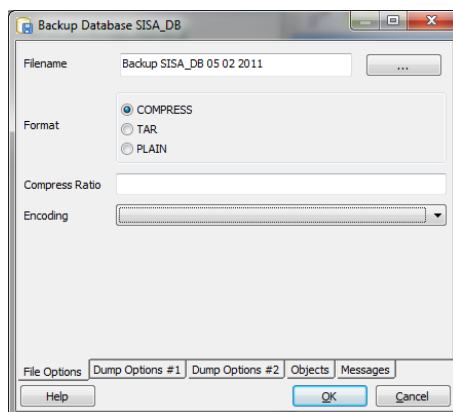


Fig. 69: Pantalla para realizar Backup de la Base de Datos.

Realizado por: Los Autores de la Tesis.

Hacer clic en "OK" y la copia de la base de datos quedará lista.

4.5.5.7 Lista de mensajes de error

A continuación se detalla la lista de mensajes de error que envía el sistema en sus diferentes interfaces.

Cuando se ingresan mal los datos en cualquier campo de una interfaz, se muestran mensajes como los siguientes:

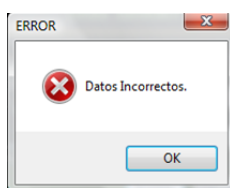


Fig.70: Mensaje de error.

Realizado por: Los Autores de la Tesis.

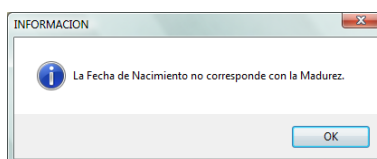


Fig. 71: Mensaje error.

Realizado por: Los Autores de la Tesis.

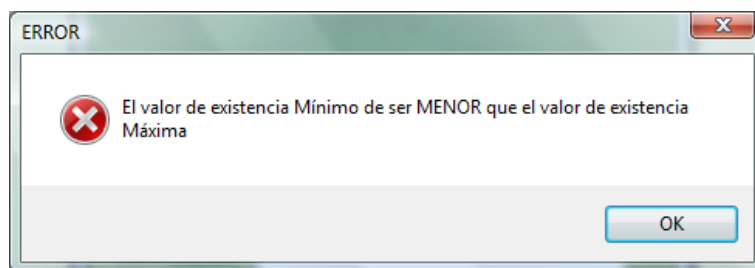


Fig. 72: Mensaje error.

Realizado por: Los Autores de la Tesis.

Cuando faltan datos en algún campo, en cambio, se muestran mensajes como los siguientes:

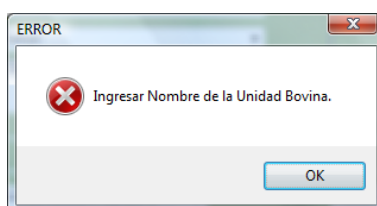


Fig. 73: Mensaje error.

Realizado por: Los Autores de la Tesis.

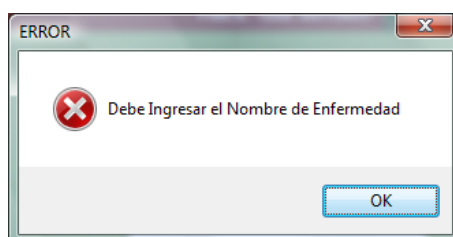


Fig. 74: Mensaje error.

Realizado por: Los Autores de la Tesis.

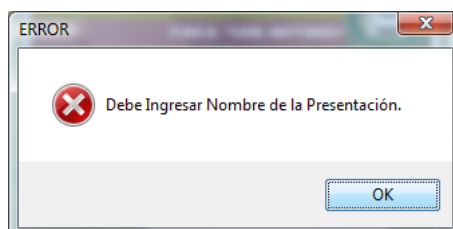


Fig. 75: Mensaje error.

Realizado por: Los Autores de la Tesis.

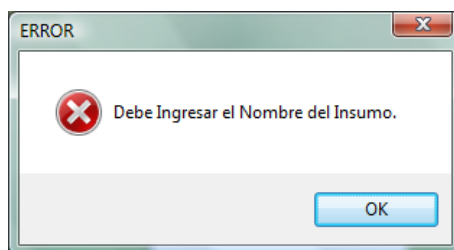


Fig. 76: Mensaje error.

Realizado por: Los Autores de la Tesis.

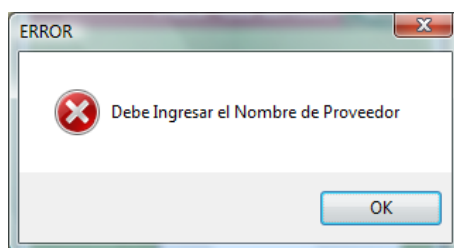


Fig. 77: Mensaje error.

Realizado por: Los Autores de la Tesis.

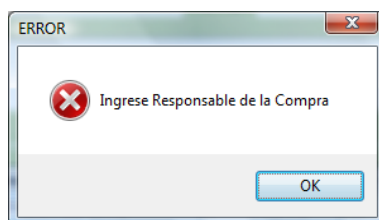


Fig. 78: Mensaje error.

Realizado por: Los Autores de la Tesis.

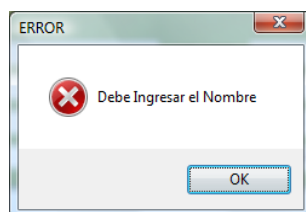


Fig. 79: Mensaje error.

Realizado por: Los Autores de la Tesis.

Cuando hay que elegir datos de un de un combo, y en lugar de hacerlo se los borran y se manda a guardar, sale el siguiente mensaje de error.

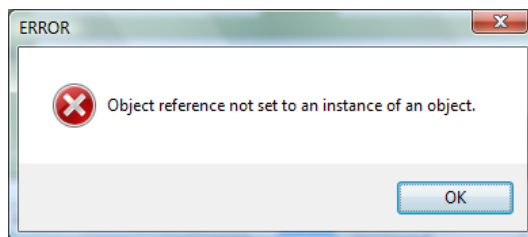


Fig. 80: Mensaje error.

Realizado por: Los Autores de la Tesis.

En el formulario que sirve para crear un complemento, cuando por ejemplo no se carga el combo con datos y a propósito se manda a guardar sale el siguiente mensaje de error.

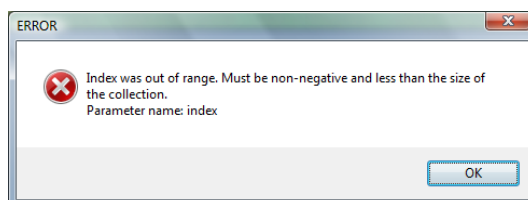


Fig. 81: Mensaje error.

Realizado por: Los Autores de la Tesis.

En el módulo de reportes cuando se manda a cargar la información en un reporte y si anteriormente no se lo diseñó bien al momento de elegir las tablas de donde se debe sacar la información, sale el siguiente mensaje de error.

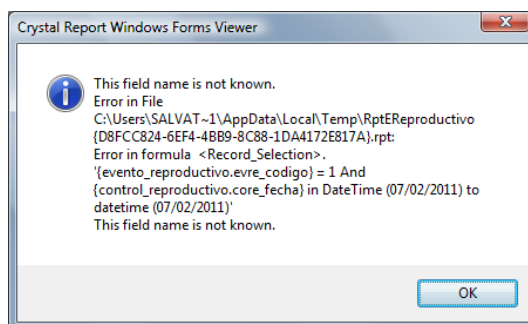


Fig. 82: Mensaje error.

Realizado por: Los Autores de la Tesis.

4.5.6. PROGRAMAS FUENTE

Esta información está contenida en un CD, que se adjunta a la presente Tesis.

4.5.7. PRUEBAS

Al sistema, de entrada, se lo somete a pruebas de caja blanca y de caja negra, pero primero conviene hacer una pequeña introducción.

Pruebas de caja blanca.- Llamadas también pruebas estructurales o pruebas de caja transparente y se denominan a las pruebas de software que se realiza sobre las funciones o métodos *internos* de una clase o módulo.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo
- Se utilizan las decisiones en su parte verdadera y en su parte falsa
- Se ejecuten todos los bucles en sus límites
- Se utilizan todas las estructuras de datos internas

Las pruebas de caja blanca comprueban la *estructura* no la funcionalidad, por ejemplo el programa puede estar bien, es decir lo que hace lo hace bien y sin embargo no hace lo que debía hacer, lo cual constituye una limitante de las pruebas de caja blanca.

Pruebas de caja negra.- Llamadas también funcionales, tienen otros nombres tales como: pruebas de caja opaca, pruebas de entrada/salida o pruebas inducidas por los datos. No están basadas en el conocimiento del código o diseño interno, determinan la funcionalidad del sistema, se aplican generalmente a las interfaces.

PRUEBAS DE CAJA BLANCA.

Se las aplicado al código del sistema para comprobar que se ejecute cada instrucción del programa, por lo menos una vez.

Mientras se desarrolló el código, se ha tenido el cuidado de ir probándolo, de manera que no haya problemas a posterior, aunque se ha encontrado errores, se los ha ido corrigiendo poco a poco.

A continuación se muestra un ejemplo de estas pruebas.

Caso de prueba

Validación de ingreso

```
private void tsbNuevo_Click(object sender, EventArgs e)
{
    bool orgVisible = dgvDatos.Visible;           (1)
    dgvDatos.Visible = false;

    try                                           (2)
    {
        frmSingle.AsignarDatosPantalla(null, enmModoPantalla.Nuevo, eSesion); (3)
        frmSingle.ShowDialog(this);
        if (frmSingle.EjecutoAccion && orgVisible) (4)
        {
            DataSet dsDatos = CargarDatosPantalla(dtoParametros); (5)

            if (dsDatos != null && dsDatos.Tables.Count > 0) (6)
                dgvDatos.DataSource = dsDatos.Tables[0]; (7)

            else (8)
                dgvDatos.DataSource = null; (9)
        } (10)

        dgvDatos.Visible = orgVisible; (11)
    } (12)

    catch (SystemException ex) (13)
    {
        DesplegarError(ex); (14)
    }
}
```

Fig. 83: Código a evaluar para Prueba de caja Blanca

Realizado por: Los Autores de la Tesis

Grafo del Caso de Prueba

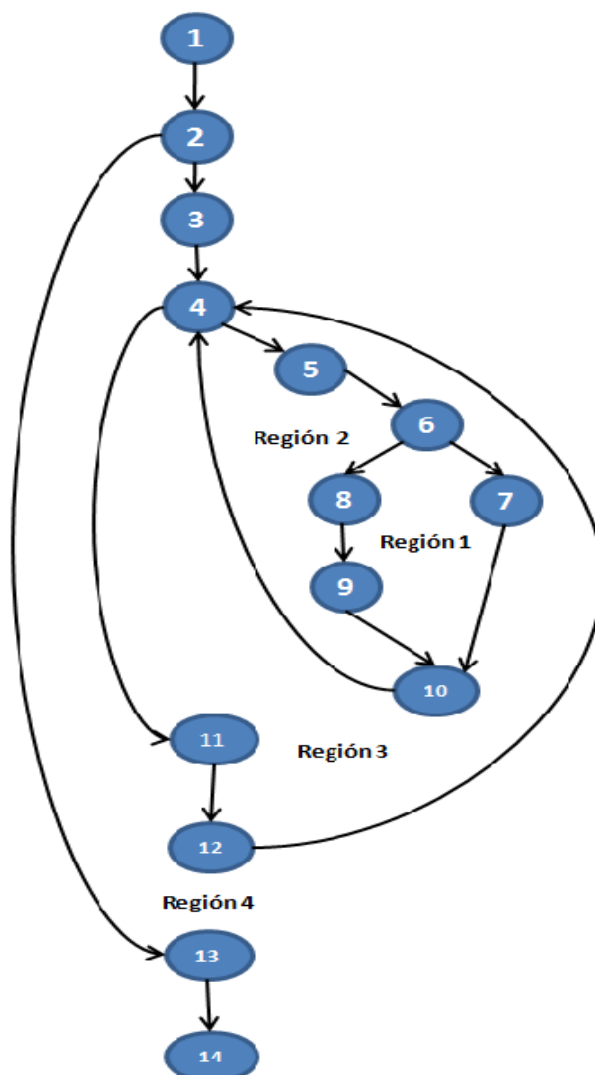


Fig. 84: Grafo de flujo

Realizado por: Los Autores de la Tesis

aristas $a=16$

nodos $n=14$

Complejidad Ciclomática $V(G) = a - n + 2 = 16 - 14 + 2 = 4$

$V(G) = 4$

Como la complejidad ciclomática es 4, entonces quiere decir que los caminos básicos tienen que ser también 4, se comprueba a continuación.

Prueba del Camino Básico

Camino 1 = 1-2-13-14

Camino 2 = 1-2-3-4-11-12-2-13-14

Camino 3 = 1-2-3-4-5-6-8-9-10-4-11-12-2-13-14

Camino 4 = 1-2-3-4-5-6-7-10-4-11-12-2-13-14

	Caso Prueba	Respuesta Esperada	Resultado Obtenido
Camino 1	No muestra el formulario	Mensaje de error	Mensaje de error
Camino 2	Mostrar formulario	Muestra formulario	Formulario en pantalla
Camino 3	Tablas sin datos	No carga datos iniciales	Formulario sin datos iniciales
Camino 4	Tablas con datos	Carga datos iniciales	Formulario con datos iniciales

Fig. 85: Grafo de caminos
Realizado por: Los Autores de la Tesis

PRUEBAS DE CAJA NEGRA.

La primera prueba que se realizó, es en el ingreso al sistema mismo, mediante la interfaz de ingreso de datos de usuario y password, es la primera que interactúa con el usuario en la entrada de datos. La cual utilizando algunas clases y métodos, ocultos para el usuario, valida los datos ingresados.

La pantalla de ingreso al sistema es la siguiente:



Fig. 86: Interfaz de ingreso al sistema.
Realizado por: Los Autores de la Tesis.

Cuando los datos ingresados, ya sea de usuario o password son incorrectos aparece una pantalla de mensaje de "ERROR", como se muestra a continuación.

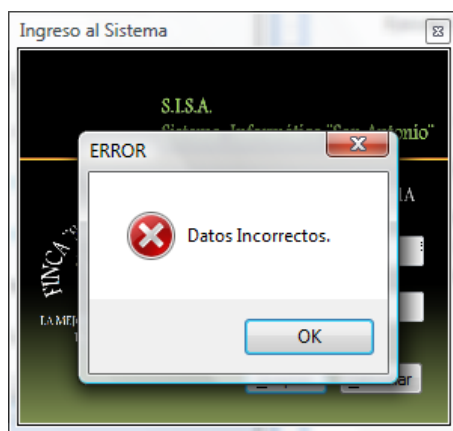


Fig. 87: Mensaje error.

Realizado por: Los Autores de la Tesis.

El sistema permite hasta dos veces que se ingrese datos incorrectos, a la tercera se cierra la interfaz de ingreso al sistema.

Otra prueba de caja negra que se va a realizar, ahora con salida de datos, consiste en hacer una consulta usando una interfaz de salida.


A continuación se ve la secuencia gráfica de esta prueba. Se usa el módulo Gestión de Unidades Bovinas, y lo que realiza es mostrar los datos de una UB seleccionada.

La interfaz principal de este módulo es la siguiente.



Fig. 88: Interfaz principal para la Administración de Unidades Bovinas.

Realizado por: Los Autores de la Tesis.

En ella hacer clic en el botón  , aparece un filtro y ahí hacer una selección luego de lo cual nos aparecerá la siguiente pantalla.



ARETE	NOMBRE	NACIMIENTO	MADUREZ	PROPIETARIO
0	Luchito	22/09/2010	TERNERO	Luis Modesto Chicaiza Tayupanta
0	Manujita	04/11/2010	TERNERA	María Augusta Chicaiza Tayupanta
0	Negrura	01/01/2010	VACONA MEDIA	Jesus Honorato Chicaiza Tayupanta
0	pruebita	25/12/2008	VACONA DE VIENTRE	Mónica María Chicaiza Tayupanta

Fig. 89: Interfaz principal para la Administración de Unidades Bovinas conteniendo un datagrid.

Realizado por: Los Autores de la Tesis.

Como se puede ver aparecen todas las unidades bovinas seleccionadas, esta en sí ya es una prueba de caja negra pues trae datos a la interfaz, de aquí se selecciona alguna, entonces hacer doble clic para que se muestre la información requerida, constituyéndose en otra prueba de caja negra.

En el siguiente gráfico se ve la interfaz con la información.



EDITAR UNIDAD BOVINA FINCA "SAN ANTONIO"

DATOS GENERALES

ID: 31
 NUMERO: 0
 NOMBRE: Luchito
 NACIMIENTO: miércoles, 22 de septiembre de 2010
 SEXO: MACHO
 MADUREZ: TERNERO
 PROPIETARIO: Luis Modesto Chicaiza Tayupanta
 SEÑAL: UKOPASHCA

DETALLES UB

MADRE: DESCONOCIDA, Número 0 RAZA: HOLSTEIN FRIESIAN
 PADRE: DESCONOCIDO, Raza HOLSTI PROCEDENCIA: COSTA
 TIPO DE UB: CARNE
 OBSERVACION:

DATOS DESCARTE

TIPO: NO APLICA FECHA: martes, 11 de enero de 2011

Fig. 90: Interfaz para editar un registro.

Realizado por: Los Autores de la Tesis.

Como se pudo ver en los gráficos, el sistema también pasa esta prueba, ahora se ejecutará otra prueba de caja negra y es la que corresponde al ingreso de datos desde una interfaz.

Ahora se va a crear un Usuario, utilizando el módulo de Seguridad.

La interfaz principal de este módulo es la siguiente.



Fig. 91: Interfaz principal para administrar Usuarios.

Realizado por: Los Autores de la Tesis.

Hacer clic en el botón "Nuevo" y aparece un formulario, que es el siguiente:

Fig. 92: Interfaz para crear Usuarios.

Realizado por: Los Autores de la Tesis.

En este formulario se ingresa los datos.

Usuarios

CREAR USUARIO

FINCA "SAN ANTONIO"

NOMBRES: JUAN JOSE USUARIO: juanjo

APELLIDOS: PEREZ SALAS CLAVE: ****

TELEFONO: 022965234 ESTADO: ACTIVO

CELULAR: 096345127 PERFIL: USUARIO

DIRECCION:

Fig. 93: Interfaz con datos para crear Usuarios.

Realizado por: Los Autores de la Tesis.

Una vez llenado los datos, hacer clic en el botón guardar, entonces aparece una interfaz con una pregunta, como se ve en el siguiente gráfico.

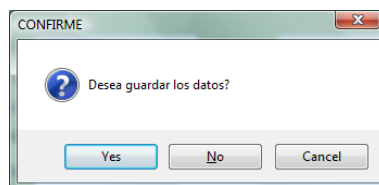


Fig. 94: Mensaje informativo de confirmación.

Realizado por: Los Autores de la Tesis.

Al hacer clic en "Yes", aparece una nueva interfaz con un mensaje, como se ve a continuación.

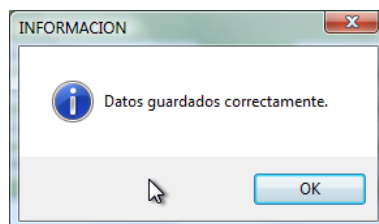


Fig. 95: Mensaje informativo.

Realizado por: Los Autores de la Tesis.

Como se ve, el sistema nuevamente pasa esta prueba. Con esto se ha terminado las pruebas de caja negra.

Otras pruebas.

Solidez (robustness testing)

Se trata de ver como reacciona el sistema ante la entrada de datos erróneos. Al respecto, asumiendo que los datos son ordenes a cumplir por el sistema, se ha probado lo siguiente:

Con datos correctos.- El sistema los administra sin ningún inconveniente, es decir, los guarda, los edita y los elimina o cambia de estado.

Órdenes con errores de sintaxis.- El sistema envía mensajes advirtiendo el error en la entrada de datos.

Órdenes correctas, pero fuera de lugar.- El sistema de igual manera envía mensajes. Esto se puede ver en el módulo de Gestión de Unidades Bovinas, al momento de crear un nuevo registro de una Unidad Bovina, cuando se pone una fecha de nacimiento de la UB que no corresponde con la madurez de la misma.

Datos con órdenes de más.- Esto se prueba principalmente en los campos de ingreso de valores numéricos. Ante esto el sistema no prosigue y envía mensajes de error en el ingreso de datos.

Aguante (stress testing).

Este tipo de prueba consiste en probar "hasta donde aguanta" un programa por razones internas, por ejemplo:

¿Es capaz de trabajar con un disco al 90%?.- Esta prueba se ha realizado y aunque no se ha tenido problemas, parece que se debe principalmente al tamaño del sistema.

¿Aguanta una carga de CPU al 90%?.- El sistema responde menos rápido, pero más allá de aquello, ningún inconveniente.

4.6 MANUAL DE INSTALACIÓN

Requisitos de software y hardware

A continuación se describe los requisitos que se deben cumplir, en hardware y software, para poder instalar el sistema.

En las máquinas clientes:

.NET Framework 3.5

ODBC de Postgres

En el servidor:

PostgreSQL 8.4 o superior, y

ODBC de Postgres

Un ordenador PC con:

- 256 Mb en Ram
- Disco Duro de 40 GB
- Procesador Pentium IV
- S.O Windows 98 o superiores

Señalado lo anterior, ahora se detalla la instalación de los programas utilizados para desarrollar el sistema.

Instalación de Visual Studio 2008

Empezar abriendo el archivo de instalación desde la ubicación donde se encuentre, entonces aparece la siguiente pantalla:



Fig. 96: Pantalla informativa de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Install Visual Studio 2008” y empieza la instalación, apareciendo la siguiente pantalla:



Fig. 97: Pantalla de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next” y sale la siguiente pantalla:

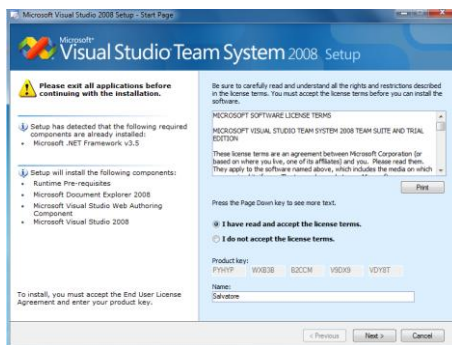


Fig. 98: Pantalla de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next” y sale la siguiente pantalla, elegir “Default”.

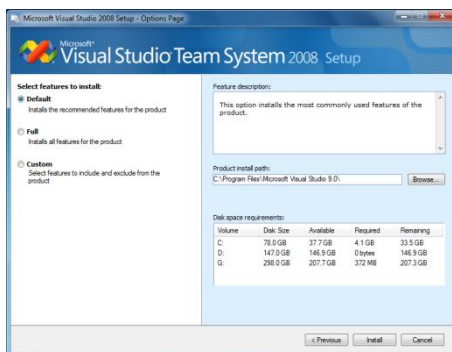


Fig. 99: Pantalla de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Install” y aparece la siguiente pantalla.

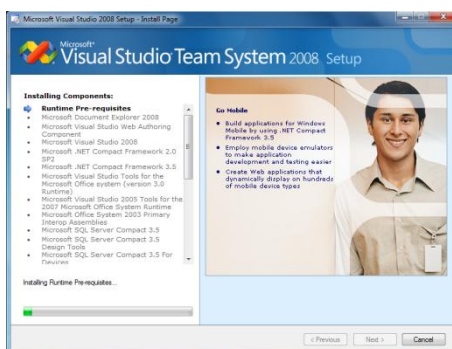


Fig. 100: Pantalla de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Entonces empieza a instalarse los prerequisites, apareciendo luego la siguiente pantalla.

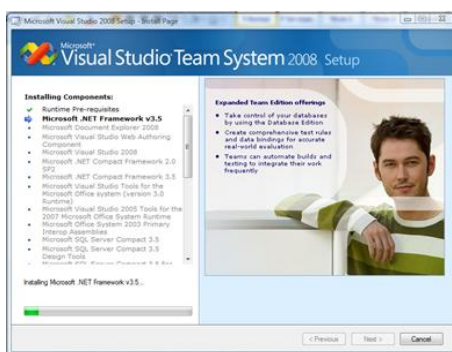


Fig. 101: Pantalla de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Y luego la siguiente.

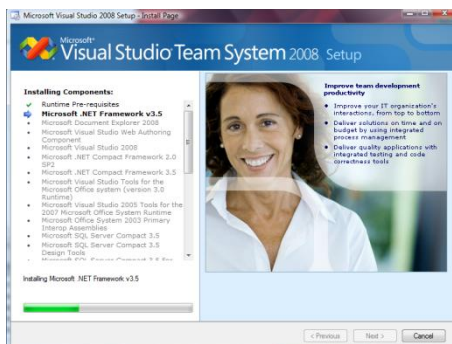


Fig. 102: Pantalla de instalación de Visual Studio 2008.

Realizado por: Los Autores de la Tesis.

Y así continua hasta finalizar.

A continuación se describe la instalación de PostgreSQL 8.4

Instalación de PostgreSQL 8.4

Empezar abriendo el archivo de instalación desde la ubicación donde se encuentre, entonces aparece la siguiente pantalla:



Fig. 103: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Siguiente”.

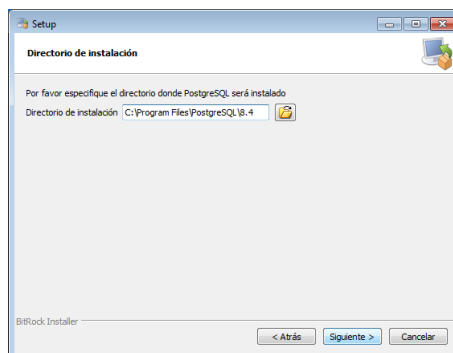


Fig. 104: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Siguiete”



Fig. 105: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Luego aparece una pantalla donde se tiene que crear e ingresar una contraseña.

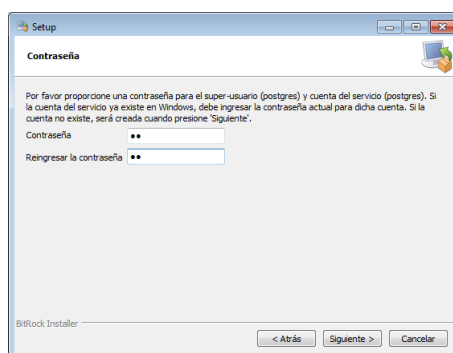


Fig. 106: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Siguiente”.

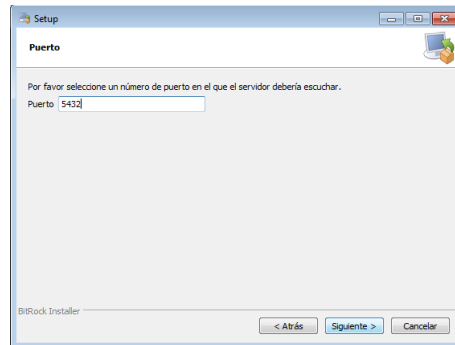


Fig. 107: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Siguiente”.

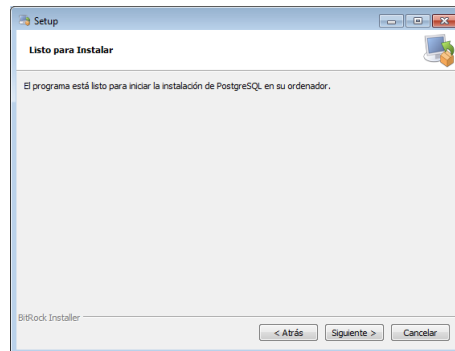


Fig. 108: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Hacer clic en siguiente y empieza la instalación, como se ve en la siguiente pantalla.

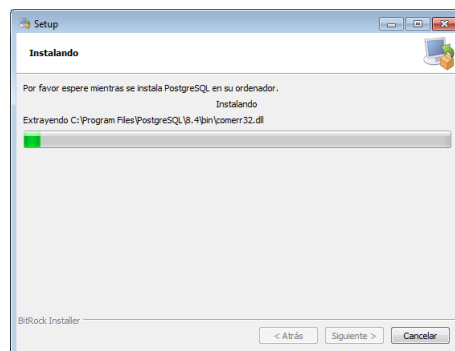


Fig. 109: Pantalla de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

Luego de instalarse aparece la siguiente pantalla en la cual se debe hacer clic en “Terminar” y termina la instalación de PostgreSQL.



Fig. 110: Pantalla final de instalación de postgresQL.

Realizado por: Los Autores de la Tesis.

En esta pantalla se tiene la opción de “¿Lanzar Stack Builder al finalizar?”, si se pone el visto en esta opción empezará a instalarse este asistente, para esto se debe estar conectados a Internet.

La primera pantalla para la instalación de este asistente es la siguiente.

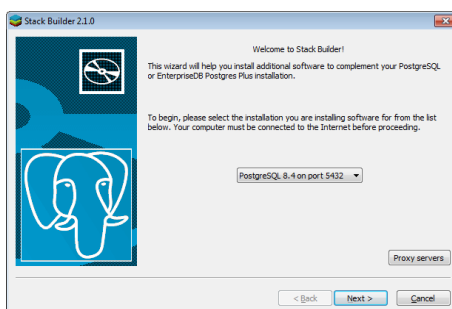


Fig. 111: Pantalla de instalación de Stack Builder.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

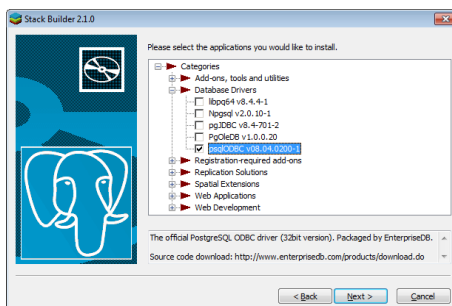


Fig. 112: Pantalla de instalación de Stack Builder.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

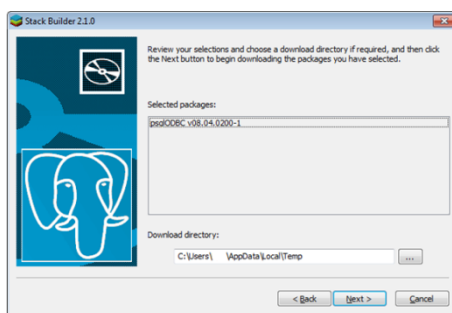


Fig. 113: Pantalla de instalación de Stack Builder.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next” y aparece la siguiente pantalla de descarga

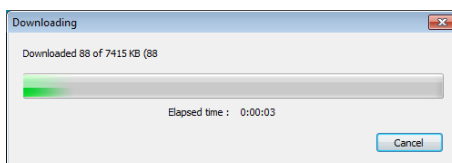


Fig. 114: Pantalla de descarga.

Realizado por: Los Autores de la Tesis.

Cuando termina de descargarse el archivo, aparece la siguiente pantalla.

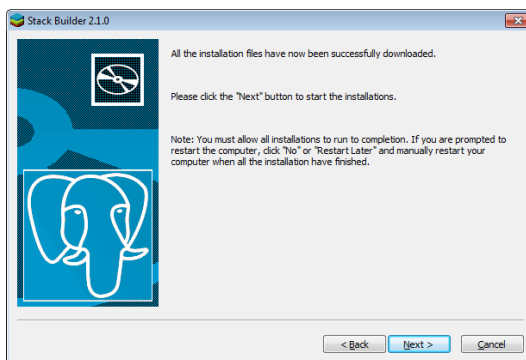


Fig. 115: Pantalla informativa de Stack Builder.

Realizado por: Los Autores de la Tesis.

Entonces está listo para que empiece la instalación del psqLODBC, para lo cual aparece la siguiente pantalla.

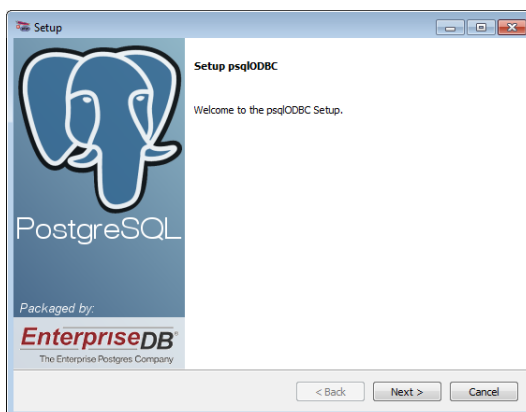


Fig. 116: Pantalla de instalación de los paquetes que contienen el psqLODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

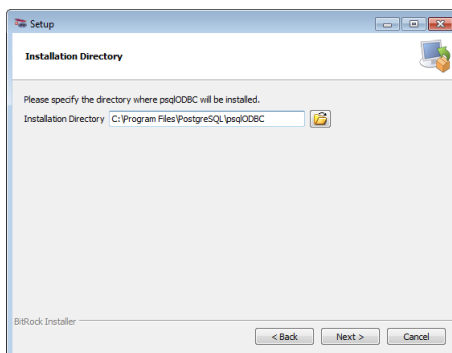


Fig. 117: Pantalla de instalación de los paquetes que contienen el psqlODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

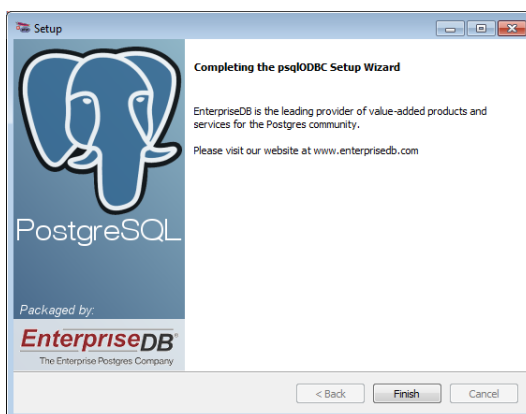


Fig. 118: Pantalla de instalación de los paquetes que contienen el psqlODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Finish” y aparece la siguiente pantalla informativa, indicando que la instalación de los paquetes seleccionados ha finalizado y que han sido guardados para una futura instalación. Lo cual quiere decir que ya se puede instalar el psqlODBC, lo cual se hará más adelante.

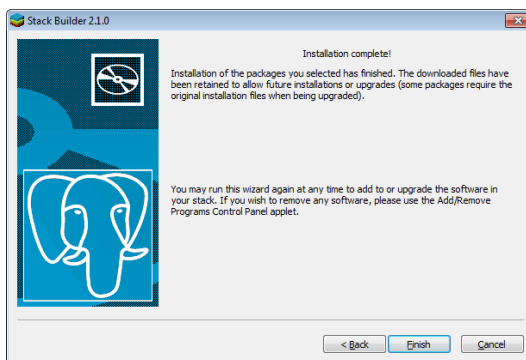


Fig. 119: Pantalla de instalación de los paquetes que contienen el psqIODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Finish”, y queda todo listo. Entonces ir a la barra de inicio para iniciar el programa pgAdmin III, se muestra la pantalla de inicio, hacer clic en "PostgreSQL" y aparece la pantalla de ingreso del password para conectar al Servidor.

La pantalla es la siguiente.

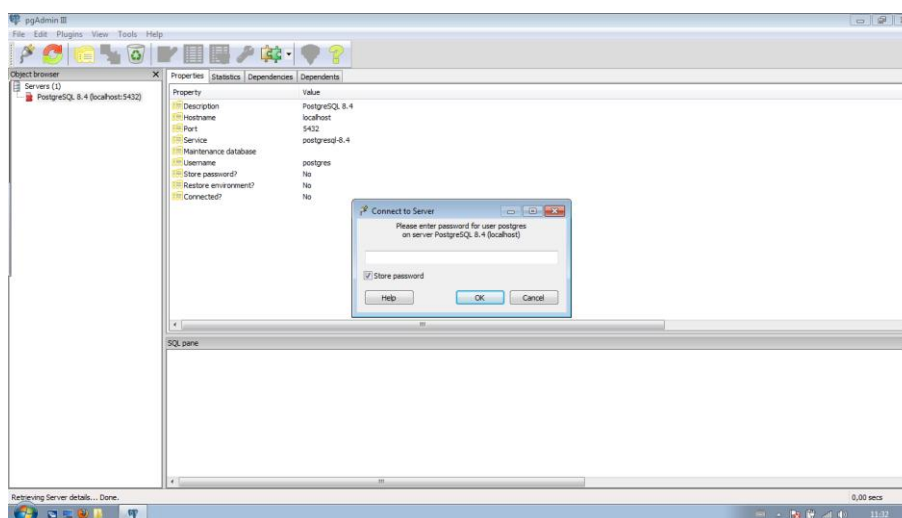


Fig. 120: Pantalla principal del pgAdmin III.

Realizado por: Los Autores de la Tesis.

Instalación del psqIODBC

Pantalla de bienvenida

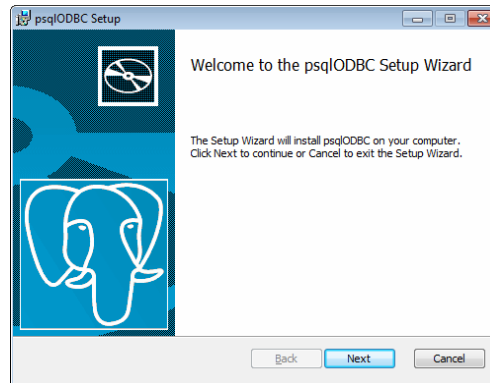


Fig. 121: Pantalla de instalación del ODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

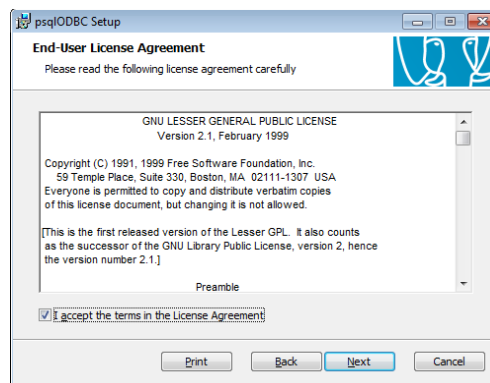


Fig. 122: Pantalla de instalación del ODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

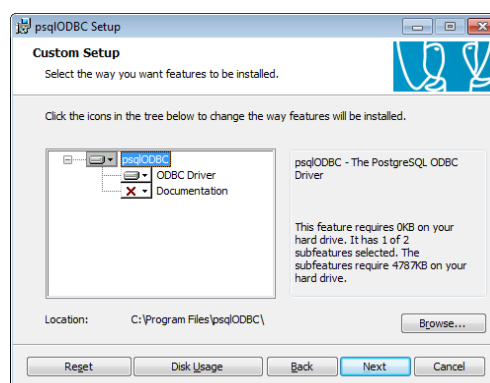


Fig. 123: Pantalla de instalación del ODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Next”

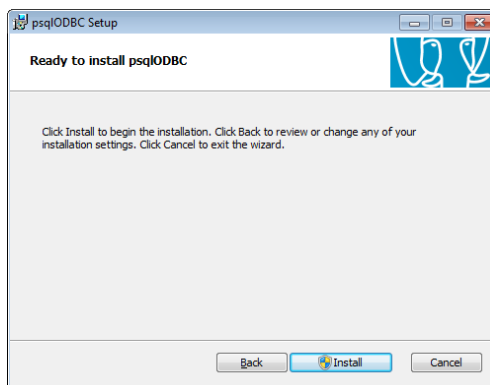


Fig. 124: Pantalla de instalación del ODBC.

Realizado por: Los Autores de la Tesis.

Hacer clic en “Install”

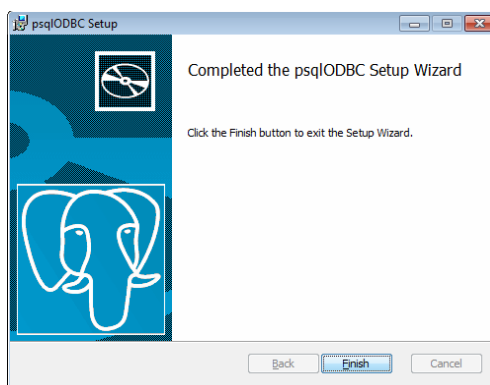


Fig. 125: Pantalla de instalación del ODBC.

Realizado por: Los Autores de la Tesis.

Para poder establecer la conexión entre el sistema informático y la base de datos, se debe primero realizar la conexión y para esto se procede de la siguiente manera.

En la computadora ir al “Panel de control”, abrirlo, entonces se ve la siguiente pantalla.



Fig. 126: Pantalla del Panel Control.
Realizado por: Los Autores de la Tesis.

Seleccionar “Herramientas administrativas”, abrirlo.

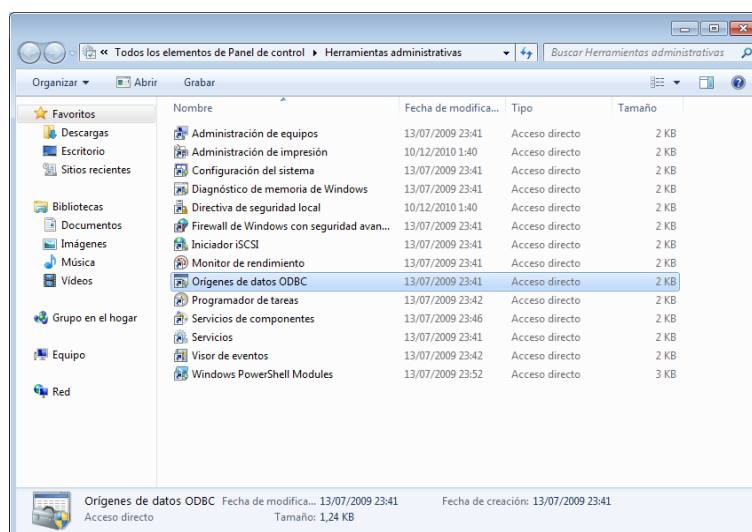


Fig. 127: Pantalla del Panel Control.
Realizado por: Los Autores de la Tesis.

Una vez en esta pantalla ir a “Orígenes de datos ODBC”, hacer doble clic, entonces aparece la siguiente pantalla.

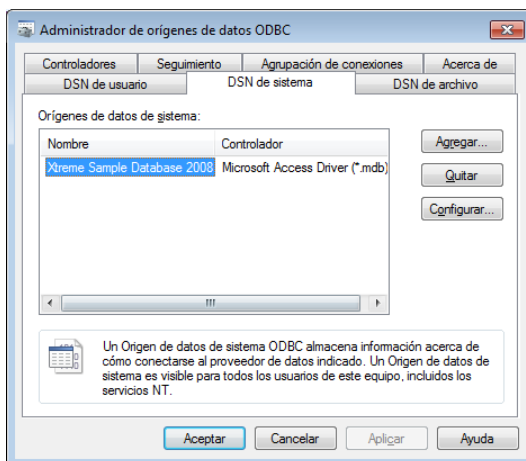


Fig. 128: Pantalla del Administrador de orígenes de datos ODBC.

Realizado por: Los Autores de la Tesis.

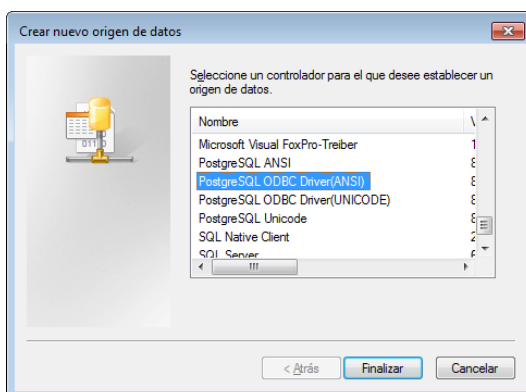


Fig. 129: Pantalla para crear un nuevo origen de datos.

Realizado por: Los Autores de la Tesis.

En la siguiente pantalla elegir los datos que nos pide, hacer clic en "Save"

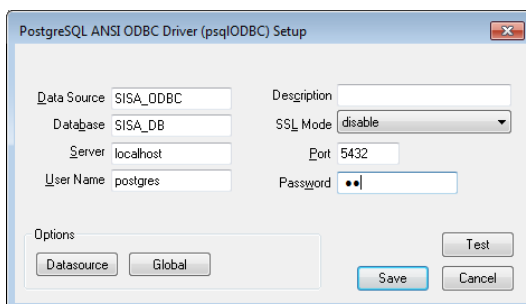


Fig. 130: Pantalla para crear un nuevo origen de datos.

Realizado por: Los Autores de la Tesis.

Y la conexión esta lista.

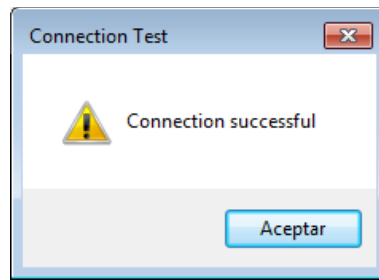


Fig. 131: Pantalla informativa.

Realizado por: Los Autores de la Tesis.

4.7 MANUAL DEL USUARIO

1. Introducción

El presente Manual de Usuario tiene como finalidad, dar a conocer de una manera clara y sencilla, el proceso de uso y funcionamiento del sistema S.I.S.A., con la finalidad de que los usuarios y administrador, se familiaricen con la aplicación y se logre administrar la información que se tiene actualmente en la finca San Antonio.

Con el uso de esta aplicación se busca crear y mantener un ambiente basado en información confiable, real y actualizada de la finca, con el propósito de tomar decisiones acertadas y coherentes para hacer uso eficiente y eficaz de los recursos, logrando con ello una buena producción lechera y de carne.

2. Acceso al sistema SISA

Para ingresar a la aplicación, se debe disponer de un nombre de usuario y una clave, asignada previamente por el administrador del sistema, luego se sigue los siguientes pasos:

Ingresar a la pantalla de ingreso, que es la siguiente.



Fig. 132: Pantalla de Ingreso al sistema SISA.

Realizado por: Los Autores de la Tesis.

En esta pantalla el usuario ingresa su usuario y la clave, luego de lo cual debe hacer clic en el botón Aceptar.

Entonces si el usuario y su clave son correctos, accede a la pantalla principal del sistema que es la siguiente:



Fig. 133: Pantalla principal del Sistema SISA.

Realizado por: Los Autores de la Tesis.

En esta pantalla se puede ver todas las opciones que tiene el sistema, las mismas que serán distintas de acuerdo al usuario.

3. Descripción de Botones de los formularios



Nuevo.- Permite acceder a una pantalla en donde se podrá ingresar los datos de un nuevo registro.



Editar.- Permite acceder a una pantalla donde se despliegan datos de un registro existente, los cuales pueden ser editados o modificados.



Ver.- Botón que permite mostrar una pantalla donde se despliegan datos de un registro existente, pero que no pueden ser modificados.



Mostrar Todos.- Botón que permite desplegar una pantalla dentro de la cual se muestra una tabla en la cual están, de manera general y sin muchos detalles, todos los registros existentes.



Guardar.- Permite guardar los datos contenidos en un formulario.



Salir.- Botón que al oprimirlo, permite salir de la aplicación o pantalla actualmente usada.



Eliminar.- Botón para eliminar datos contenidos en algún formulario.

4. Opciones principales del sistema

En esta pantalla está el menú principal, que consta de las siguientes opciones:

- Inicio
- Gestión Unidad Bovina
- Gestión Insumos
- Gestión Seguridad
- Reportes
- Sistema
- Ayuda

A continuación se detalla cada una de las opciones:

Opción “Inicio”

Esta opción del menú principal contiene a su vez otras dos opciones, que son:

- Cerrar Sesión
- Salir Sistema

Cerrar Sesión.- Esta opción permite cerrar la sesión y regresar a la pantalla de ingreso al sistema, en la cual ingresamos el usuario y la clave. Esto se puede

hacer, por ejemplo, para cambiar de usuario, en el caso de que un usuario tenga más de un perfil.

Salir Sistema.- Esta opción se usa para salir del Sistema SISA.

A continuación se puede ver la pantalla descrita en líneas anteriores.

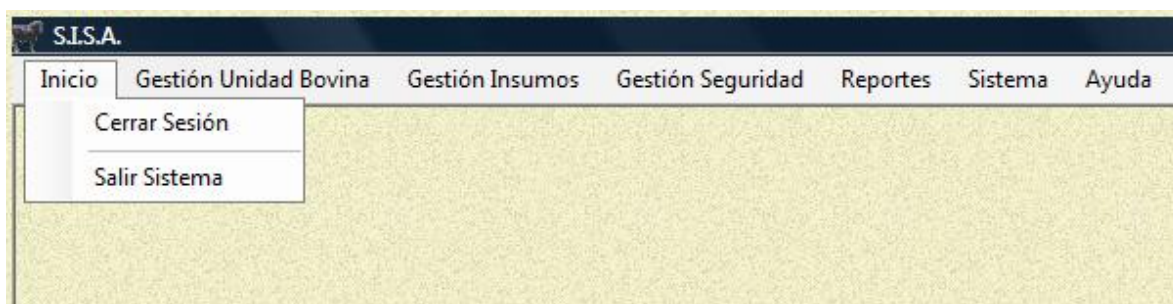


Fig. 134: Pantalla principal en la que se muestra los submenús de la opción Inicio.

Realizado por: Los Autores de la Tesis.

Antes de entrar a navegar por el resto de opciones del sistema SISA, se va a describir cada uno de los botones que se encontrará en los distintos formularios. Cabe señalar que algunos tendrán más botones que otros, según sea el tipo de formulario.

Opción “Gestión Unidad Bovina”

Esta opción del menú principal contiene a su vez a otras opciones, que son:

- Administrar UB
- Control Sanitario
 - Historia Clínica
 - Vacunación
- Control Reproductivo
- Complementos Nutricionales
 - Administrar Complementos
 - Suministrar Complementos
- Datos Generales
 - Enfermedades
 - Enfermedad Tratamiento

- Evento reproductivo
- Tipo Madurez

En las siguientes figuras se muestran los detalles de todas las opciones disponibles en “Gestión Unidad Bovina”

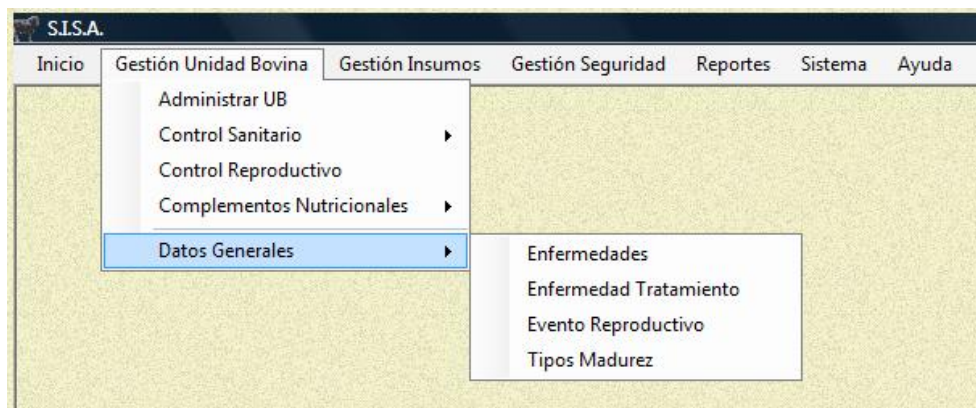
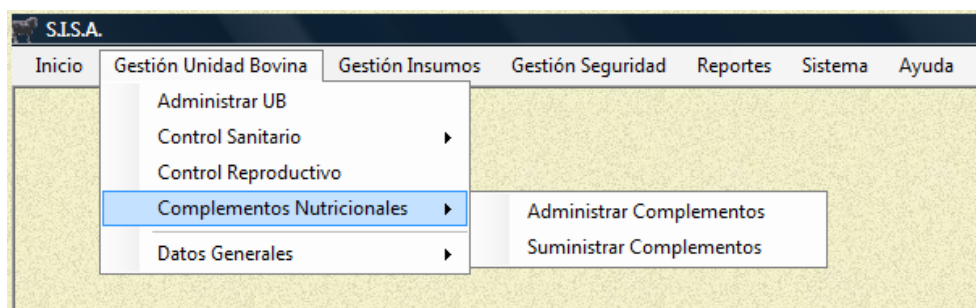
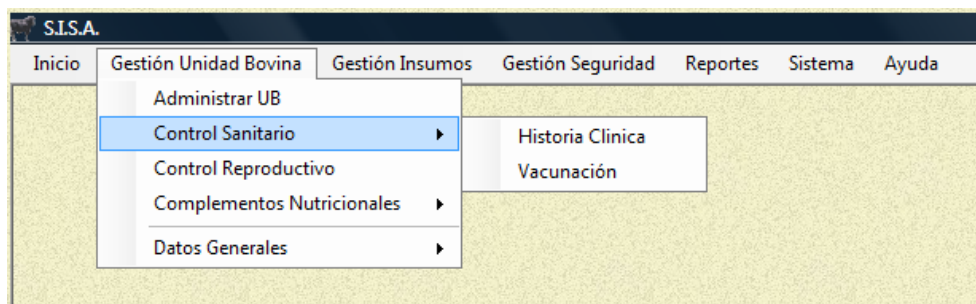


Fig. 135: Pantallas descriptivas de las opciones de “Gestión Unidad Bovina”.

Realizado por: Los Autores de la Tesis.

Opción Administrar UB

Al hacer clic en esta opción aparece la siguiente pantalla.



Fig. 136: Pantalla de la Opción Administrar UB.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo”, aparece un formulario que es el siguiente:

Fig. 137: Pantalla o formulario para crear Unidad Bovina.

Realizado por: Los Autores de la Tesis.

En este formulario hay que ingresar los diferentes datos de una Unidad Bovina, los cuales se guardan oprimiendo el botón Guardar.

Al hacer clic en el botón “Mostrar Todos”, aparece un formulario o pantalla en la cual se filtra la información, de acuerdo a la categoría, propietario o sexo de las Unidades Bovinas, que se desea muestre la tabla, ese formulario está en la Fig. 138.

Fig. 138: Pantalla para filtrar información referente a Unidades bovinas.

Realizado por: Los Autores de la Tesis.

Luego de hacer uso del criterio de búsqueda, aparece una pantalla conteniendo una tabla, Fig. 139, en la cual se muestran los registros seleccionados.

ARETE	NOMBRE	NACIMIENTO	MADUREZ	PROPIETARIO
33	manja	22/11/2010	VACA	Mónica María Chicaiza Tayupanta
9	wrefgdf	22/11/2010	VACONA DE VI...	María Feloliana Tayupanta Caizaluisa

Fig. 139: Pantalla que muestra en una tabla todas las Unidades Bovinas.

Realizado por: Los Autores de la Tesis.

Cuando se oprime el botón “Editar” aparece un formulario como el de la fig. 140, con la diferencia que este formulario muestra los datos de una Unidad Bovina, junto con su fotografía, los mismos que pueden ser modificados y luego guardados nuevamente. Aunque también hay como eliminarlos con el botón “Eliminar”.

Fig. 140: Pantalla para editar los datos de una Unidad Bovina.

Realizado por: Los Autores de la Tesis.

Cuando se oprime el botón “Ver” aparece un formulario similar al de la fig. 140, con la diferencia que en este formulario no se puede modificar ningún dato, solo se ve la información y nada más.

Control Sanitario

Esta opción tiene 2 opciones de menor jerarquía y son:

- Historia Clínica, y
- Vacunación

Historia Clínica.- Al hacer clic en esta opción aparece la pantalla desde donde se puede administrar las historias clínicas, y que al igual que todas las pantallas de administrar datos, tienen los mismos cinco botones en su parte superior; Nuevo, Editar, Ver, Mostrar Todos y Salir.

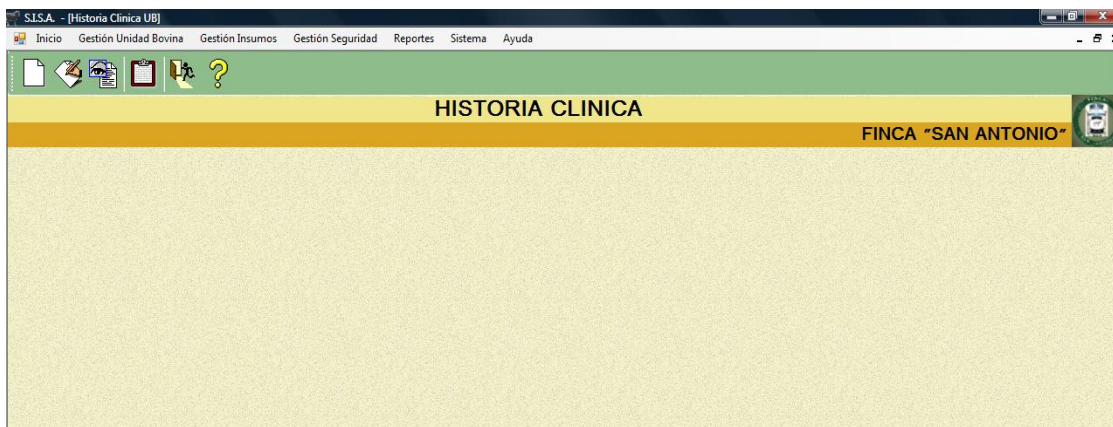


Fig. 141: Pantalla para administrar las Historias Clínicas de las Unidades Bovinas.
Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo” aparece el siguiente formulario para crear una nueva historia clínica.

CODIGO	MEDICAMENTO	DOSIS	UNIDAD
*			

Fig. 142: Pantalla para crear una nueva Historia Clínica.
Realizado por: Los Autores de la Tesis.

De igual manera cuando se hace clic en el botón “Editar” o “Ver” aparecerá el mismo formulario que el de la Fig. 142, la diferencia radica en que si se trata de “Editar”, se permitirá modificar y se trata del botón “Ver”, solo se puede ver los datos y nada más.

Vacunación.- Al hacer clic en esta opción aparece la pantalla desde donde se puede ir a administrar la Vacunación que se hace a las Unidades Bovinas pero por grupos.



Fig. 143: Pantalla para administrar la vacunación.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo” aparece el siguiente formulario para crear un nuevo registro de vacunación.

Fig. 144: Pantalla para crear un nuevo registro de vacunación.

Realizado por: Los Autores de la Tesis.

Control Reproductivo

Al hacer clic en esta opción aparece la siguiente pantalla.



Fig. 145: Pantalla de la opción Control Reproductivo.

Realizado por: Los Autores de la Tesis.

Desde esta pantalla se puede ir a crear los registros de Control Reproductivo de las Unidades Bovinas, y la secuencia para ello es la siguiente:

1. Hacer clic en el botón “Mostrar Todos” y antes que aparezca la tabla con todas las Unidades Bovinas en edad para ser preñadas así como otras que ya han tenido algún parto anterior, aparece una pantalla pequeña que es el filtro, el cual permite elegir el evento reproductivo por medio del cual quiero mostrar la información. La pantalla del filtro es la siguiente.

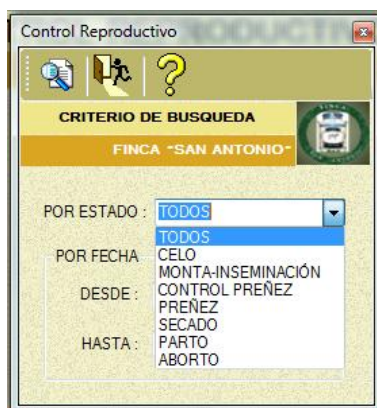


Fig. 146: Pantalla Filtro para búsqueda por evento reproductivo.

Realizado por: Los Autores de la Tesis.

La siguiente es la pantalla que se despliega después de la pantalla filtro.



Fig. 147: Pantalla conteniendo la tabla con las Unidades Bovinas a las que se puede registrar un evento reproductivo.

Realizado por: Los Autores de la Tesis.

2. Elegir una Unidad Bovina y aparece una pantalla, como la de la Fig. 148, en la cual se llena los datos que correspondan.

Fig. 148: Pantalla para crear el registro reproductivo de una UB.

Realizado por: Los Autores de la Tesis.

De igual manera cuando se hace clic en el botón “Editar” o “Ver” aparecerá el mismo formulario que el de la Fig. 148, la diferencia radica en que si se trata de “Editar”, se permitirá modificar la información y se trata del botón “Ver”, solo se puede ver los datos y nada más.

Complementos Nutricionales.

Esta opción tiene 2 opciones y son:

- Administrar complementos , y
- Suministrar Complementos.

Administrar Complementos

Al hacer clic en esta opción se despliega una pantalla, como la de la Fig. 149, desde la cual, por medio de los botones de que dispone, se puede crear nuevos complementos, editar y ver los complementos existentes.

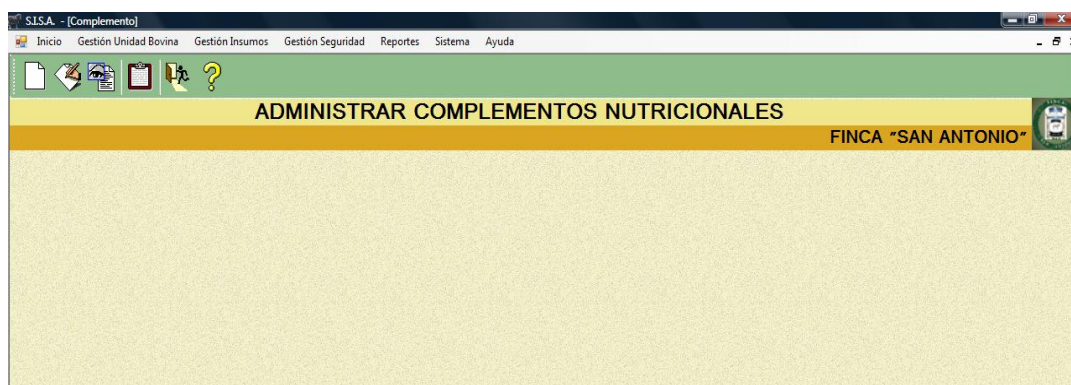


Fig. 149: Pantalla para administrar los complementos nutricionales.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo”, aparecerá una pantalla como la de la Fig. 150, en la cual se puede llenar los datos del nuevo complemento nutricional y luego guardarlo.

CODIGO	INSUMO	DOSIS	UNIDAD
*			

Fig. 150: Pantalla para crear un complemento nutricional.

Realizado por: Los Autores de la Tesis.

De igual manera cuando se hace clic en el botón “Editar” o “Ver” aparecerá el mismo formulario que el de la Fig. 150, la diferencia radica en que si se trata de “Editar”, se permitirá modificar y se trata del botón “Ver”, solo se puede ver los datos y nada más.

Suministrar Complementos

Al hacer clic en esta opción se despliega una pantalla, como la de la Fig. 151, desde la cual, por medio de los botones de que dispone, se puede crear registros del uso que se ha dado a los complementos nutricionales. Además se puede ver los registros anteriores, así como editarlos o verlos simplemente.

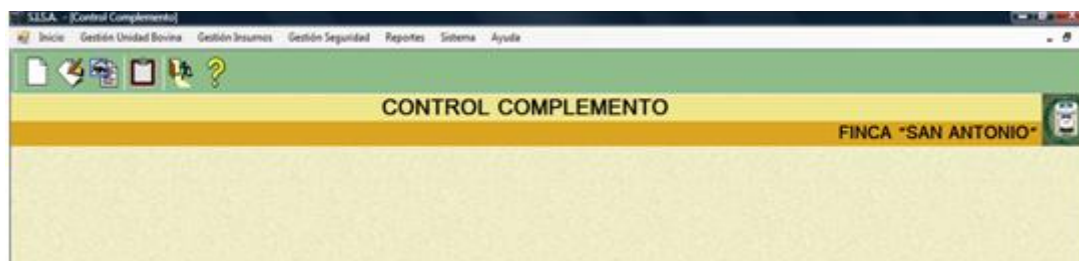


Fig. 151: Pantalla para control del uso de los complementos nutricionales.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 152, en la cual se puede crear un registro del uso del complemento.

Fig. 152: Pantalla para crear registros del uso de complementos nutricionales.

Realizado por: Los Autores de la Tesis.

De igual manera cuando se hace clic en el botón “Editar” o “Ver” aparecerá el mismo formulario que el de la Fig. 152, la diferencia radica en que si se trata de “Editar”, se permitirá modificar y se trata del botón “Ver”, solo se puede ver los datos y nada más.

Datos Generales

Esta opción tiene, a la vez, otras cuatro opciones que son:

- Enfermedades
- Enfermedad Tratamiento
- Evento Reproductivo
- Tipos Madurez

Enfermedades

Al hacer clic en esta opción se despliega una pantalla, Fig. 153, desde la cual y por medio de los botones de que dispone, se puede crear registros de enfermedades de las Unidades Bovinas, ver los registros de las enfermedades, usando el botón “Mostrar Todos”, así como editarlas o verlas simplemente.



Fig. 153: Pantalla para administrar enfermedades.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 154, en la cual se puede crear un registro de una enfermedad.



Fig. 154: Pantalla para crear el registro de una enfermedad.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece en la pantalla de la Fig. 155, una tabla con todas las enfermedades registradas, como se ve en la siguiente pantalla.

COD.	NOMBRE	SINTOMA
1	ANTRAX, CARBUNCO	Enfermedad febril, aguda infecciosa causada por el bacillus anthracis. La enfermedad puede presentarse en for...
2	BABESIOSIS, PIROPLASMOSIS BOVINA	Enfermedad protozoaria transmitida por garrapatas, causada por parásitos del genero Babesia. Las infecciones ca...
3	BOTULISMO	Intoxicación por Clostridium botulinum, debido a la ingestión de alimentos en descomposición. Parálisis motriz prog...
4	CETOSIS, ACETONEMIA, CETONEMIA	Enfermedad metabólica de las vacas lactantes que ocurre desde unos pocos días a unas pocas semanas despué...
5	CRETINISMO	Exagerado hipotiroidismo en los recién nacidos a animales jóvenes resultante de una ausencia completa de hom...
6	DERMATOBIOSIS, NUCHE, TUPE, GUSANO D...	Miasis cutánea producida por la larva de la mosca Dermatobia hominis o dermatobia cyamiventris en toda América...
7	DIARREA VIRAL BOVINA	Enfermedad infecciosa del ganado bovino, causada por un pestivirus, normalmente es subclínica o leve, con alt...
8	FIEBRE AFTOSA	Enfermedad aguda producida por un enterovirus de la familia Picornavirus, muy contagiosa en animales de pezuñ...
9	FIEBRE DE GARRAPATAS	Enfermedad de los ruminantes transmitida por ixodes spp. Fiebre, apatía, anorexia que dura de 5 a 8 días
10	MASTITIS	Inflamación de la glándula mamaria debida a infección por patógenos bacterianos o micóticos. Tumefacción, cal...

Fig. 155: Pantalla conteniendo una tabla con las enfermedades registradas.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 156, en donde están contenidos datos de la enfermedad, para editarlos o para eliminarlos usando el botón “Eliminar”.

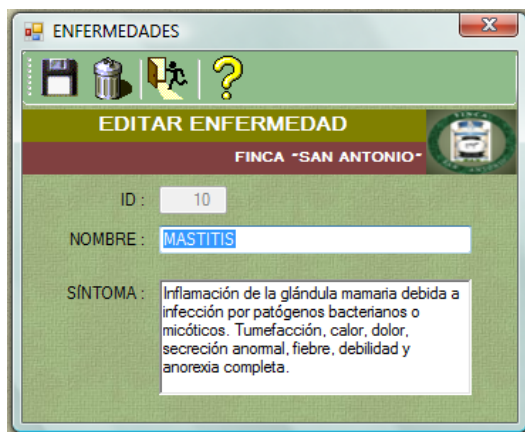


Fig. 156: Pantalla para editar datos de una enfermedad.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Ver” del menú principal, se muestra la misma pantalla anterior, pero únicamente dejará ver la información sin dejar hacer nada más.

Enfermedad Tratamiento

Al hacer clic en esta opción se despliega una pantalla, Fig. 157, desde la cual y por medio de los botones de que dispone, se puede crear registros de tratamientos guías para enfermedades de las Unidades Bovinas, ver los tratamientos existentes, usando el botón “Mostrar Todos”, así como editarlos o verlos simplemente.



Fig. 157: Pantalla para administrar tratamientos para las enfermedades de las Unidades Bovinas.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 158, en la cual se puede crear el registro de un tratamiento.

Fig. 158: Pantalla para crear un tratamiento.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece en la pantalla de la Fig. 159, una tabla con todos los tratamientos registrados, con anterioridad, como se ve en la siguiente pantalla.

COD.	ENFERMEDAD	TIPO TRATAMIENTO	NOMBRE	OBSERVACIÓN
1	ANTRAX, CARBUNCO	PREVENTIVO	ANTRAX_PRE_G	Indicado para Unidades Bovinas Grandes.

Fig. 159: Pantalla conteniendo una tabla con los tratamientos registrados.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 160, en donde están contenidos datos del tratamiento, para editarlos o para eliminarlos, usando el botón “Eliminar”.

Enfermedad Tratamiento

EDITAR TRATAMIENTO

FINCA "SAN ANTONIO"

DATOS TRATAMIENTO

ID : 1

ENFERMEDAD : ANTRAX, CARBUNCO

TIPO TRATAMIENTO : PREVENTIVO

NOMBRE TRATAMIENTO : ANTRAX_PRE_G

OBSERVACIÓN :
Indicado para Unidades Bovinas Grandes.

	COD.	MEDICAMENTO	DOSIS	UNIDAD
▶	1	Aftobov	50,00	CM. CÚBICO
*				

1 of 1

Fig. 160: Pantalla para editar datos de un tratamiento.

Realizado por: Los Autores de la Tesis

Evento Reproductivo

Al hacer clic en esta opción se despliega una pantalla, Fig. 161, desde la cual y por medio de los botones de que dispone, se puede crear registros de eventos reproductivos de las Unidades Bovinas, ver los eventos existentes, usando el botón "Mostrar Todos", así como editarlos o verlos simplemente.



Fig. 161: Pantalla para administrar eventos reproductivos de las Unidades Bovinas.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón "Nuevo" aparece una pantalla como la de la Fig. 162, en la cual se puede crear el registro de un evento reproductivo.

Evento Reproductivo

CREAR EVENTO

FINCA "SAN ANTONIO"

NOMBRE :

DURACIÓN :

TIEMPO : HORAS

ESTADO : ACTIVO

ORDEN :

Fig. 162: Pantalla para crear un evento reproductivo.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece en la pantalla de la Fig. 163, una tabla con todos los tratamientos registrados, con anterioridad, como se ve en la siguiente pantalla.

CODIGO	EVENTO	DURACIÓN	TIEMPO	ESTADO	ORDEN
1	CELO	24	HORAS	ACTIVO	1,00
2	MONTA-INSEMINACIÓN	1	HORAS	ACTIVO	3,00
3	CONTROL PREÑEZ	60	DÍAS	ACTIVO	4,00
4	PREÑEZ	9	MESES	ACTIVO	5,00
5	SECADO	2	MESES	ACTIVO	6,00
6	PARTO	1	HORAS	ACTIVO	7,00
7	ABORTO	1	HORAS	ACTIVO	8,00

Fig. 163: Pantalla conteniendo una tabla con los eventos reproductivos registrados.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 164, en donde están contenidos datos del evento, para editarlos o para eliminarlos, usando el botón “Eliminar”.



Fig. 164: Pantalla para editar datos de un evento reproductivo.

Realizado por: Los Autores de la Tesis

Tipos Madurez

Al hacer clic en esta opción se despliega una pantalla, Fig. 165, desde la cual y por medio de los botones de que dispone, se puede crear registros de tipos de madurez de las Unidades Bovinas, ver los tipos existentes, usando el botón “Mostrar Todos”, así como editarlos o verlos simplemente.

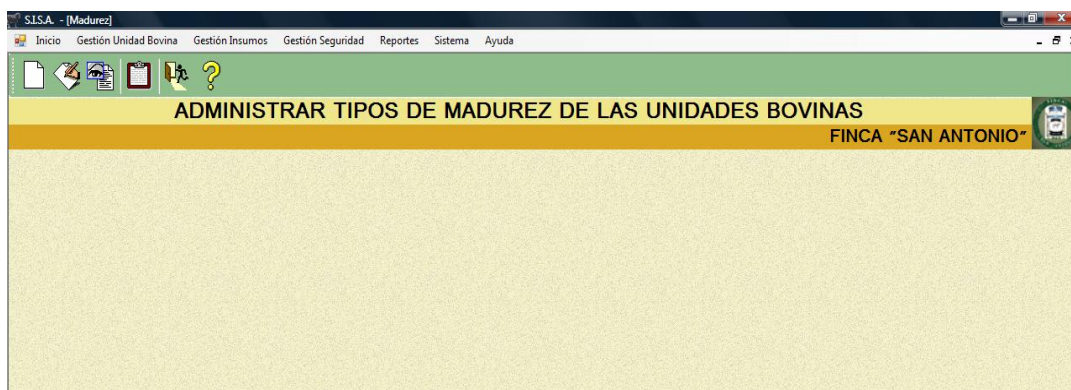


Fig. 165: Pantalla para administrar tipos de madurez de las Unidades Bovinas.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 166, en la cual se puede crear el registro de un tipo de madurez.

Fig. 166: Pantalla para crear un tipo de madurez.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece en la pantalla de la Fig. 167, una tabla con todos los tipos de madurez registrados, con anterioridad, como se ve en la siguiente pantalla.

CODIGO	NOMBRE	SEXO	DESDE	HASTA	ORDEN
1	TERNERO	MACHO	0	6	1,00
2	TORETE	MACHO	6	24	2,00
3	TORO	MACHO	24	120	3,00
4	TORO REPRODUCTOR	MACHO	24	120	4,00
5	TERNERA	HEMBRA	0	6	1,00
6	VACONA MEDIA	HEMBRA	6	12	2,00
7	VACONA FIERRO	HEMBRA	12	18	3,00
8	VACONA DE VIENTRE	HEMBRA	18	36	4,00
9	VACA	HEMBRA	36	120	5,00

Fig. 167: Pantalla conteniendo una tabla con los tipos de madurez registrados.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 168, en donde

están contenidos datos del tipo de madurez, para editarlos o para eliminarlos, usando el botón “Eliminar”.

Fig. 168: Pantalla para editar datos de un tipo de madurez.

Realizado por: Los Autores de la Tesis

Opción Gestión Insumos

Esta opción tiene otras opciones que son las siguientes:

- Inventario
- Presentaciones
- Insumos
- Procesos Finca
- Proveedores
- Compras
- Egresos
- Información Adicional

Inventario

Al hacer clic en esta opción se despliega una pantalla, Fig. 169, desde la cual y por medio de los botones de que dispone se puede ver los insumos existentes, usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 169: Pantalla para administrar el inventario de insumos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos” aparece una pantalla, Fig. 170, que es un filtro para elegir el criterio de búsqueda, en este caso se debe elegir por el tipo de insumo que queremos ver.

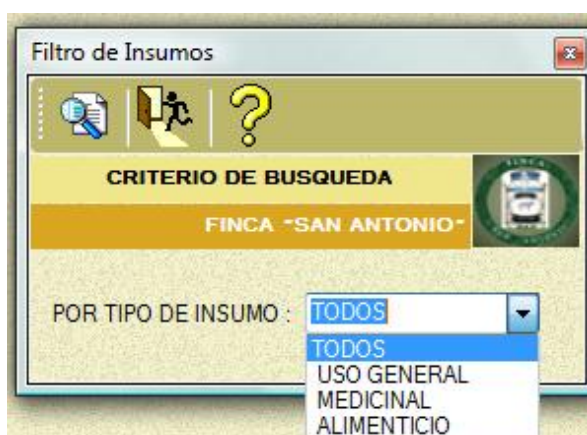


Fig. 170: Pantalla filtro para elegir el tipo de insumo a mostrar.

Realizado por: Los Autores de la Tesis

Luego de la pantalla filtro se muestra otra pantalla, Fig. 171, que contiene una tabla donde se muestran los insumos inventariados. La pantalla es la siguiente:

COD.	NOMBRE	PRESENTACIÓN	TIPO INSUMO	EXISTENCIA	OBSERVACIÓN
22	Acarex	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
21	Aclaran	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
18	Alrecho	PRESENTACIÓN GENERAL	ALIMENTICIO	0.00	Insumo de inventario
20	Albobov	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
9	Alambre	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
15	Alambre de púas	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
23	Albendex 10 %	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
3	Baldes	PLÁSTICO PEG.	USO GENERAL	0.00	Insumo de inventario de uso general
24	Bañol	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
25	Benzapen forte	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
4	Botas	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
11	Cabo fofoex	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
8	Cedazos	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
16	Clavos	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
26	Clonacum	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
30	CLOXAMBIOTIC	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
5	Crema para ubres	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
6	Desinfectante	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
13	Detergente	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
27	Grafaleón	PRESENTACIÓN GENERAL	MEDICINAL	0.00	Insumo de inventario
14	Grapas	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario
1	Guantes	DE CAUCHO	USO GENERAL	0.00	Insumo de inventario
2	Jeringuillas	PRESENTACIÓN GENERAL	USO GENERAL	0.00	Insumo de inventario

Fig. 171: Pantalla que contiene una tabla donde se despliega información de los insumos inventariados.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 172, en donde están contenidos datos del insumo, para editar la presentación del insumo así como el valor mínimo o máximo; ó para eliminarlo usando el botón “Eliminar”.

ID : 22
 INSUMO : Acarex
 PRESENTACIÓN : PRESENTACIÓN GENERAL
 EXISTENCIA : 0. MIN. : 1. MÁX. : 3.
 OBSERVACIÓN : Insumo de inventario

Fig. 172: Pantalla para editar la presentación, la mínima y la máxima cantidad de un insumo.

Realizado por: Los Autores de la Tesis

Presentaciones

Al hacer clic en esta opción se despliega una pantalla, Fig. 173, desde la cual y por medio de los botones de que dispone, se puede crear nuevas presentaciones, ver las presentaciones existentes usando el botón “Mostrar Todos”, así como editarlas, eliminarlas o verlas simplemente.



Fig. 173: Pantalla para administrar las presentaciones que pueden tener los insumos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 174, en la cual se puede crear el registro de una nueva presentación.

The image shows a screenshot of a web application window titled 'Presentacion'. The window has a toolbar with icons for home, back, forward, print, and help. The main content area has a green header with the text 'CREAR PRESENTACIÓN' and 'FINCA "SAN ANTONIO"' on the right. Below the header are two input fields: 'NOMBRE:' followed by a text box, and 'DETALLE:' followed by a larger text area.

Fig. 174: Pantalla para crear una presentación.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 175, conteniendo una tabla con todos los tipos de presentaciones.

COD.	NOMBRE	DETALLE
1	PRESENTACIÓN GENERAL	Presentación por Omisión
2	DE ACERO 1 PULG.	material con que está hecho el insumo
3	DE ACERO	material con que está hecho el insumo
4	PLÁSTICO PEQ.	material con que está hecho el insumo
5	GRUESO	diámetro del insumo
6	DELGADO	diámetro del insumo
7	PLÁSTICAS DE 5 ML.	material con que está hecho el insumo y su capacidad que ...
8	METÁLICAS	material con que está hecho el insumo
9	EN GRANO	tipo de presentación del insumo
10	DE CUERO	material con que está hecho el insumo
11	DE CAUCHO	material con que está hecho el insumo
12	ANTIMOSCOS	tipo especial del insumo
13	NORMAL	tipo común de presentación del insumo
14	EN BARRA	tipo de presentación del insumo
15	EN POLVO	tipo de presentación del insumo
16	GALVANIZADO	revestimiento del insumo
17	PLÁSTICA RÍGIDA	material y rigidez del insumo
18	PLÁSTICA FLEXIBLE	material y rigidez del insumo
19	LIQUIDA	tipo de presentación del insumo
20	EN CREMA	tipo de presentación del insumo
21	PLÁSTICA	Material del que está hecho el insumo

Fig. 175: Pantalla que contiene una tabla donde se despliega información de las presentaciones existentes.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 176, en donde están contenidos datos de la presentación, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

Presentación

EDITAR PRESENTACIÓN

FINCA "SAN ANTONIO"

ID : 1

NOMBRE : PRESENTACIÓN GENERAL

DETALLE : Presentación por Omisión

Fig. 176: Pantalla para editar datos de una presentación.

Realizado por: Los Autores de la Tesis

Insumos

Al hacer clic en esta opción se despliega una pantalla, Fig. 177, desde la cual y por medio de los botones de que dispone, se puede crear nuevos insumos, ver los existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 177: Pantalla para administrar los insumos.


Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 178, en la cual se puede crear el registro de un nuevo insumo.

 A screenshot of a dialog box titled 'Administrar Insumo'. The dialog has a green header with the text 'CREAR INSUMO' and 'FINCA "SAN ANTONIO"'. Below the header, there are three input fields: 'NOMBRE:' followed by a text box, 'TIPO INSUMO:' followed by a dropdown menu currently showing 'USO GENERAL', and 'DETALLE:' followed by a text box. The dropdown menu is open, showing options: 'USO GENERAL', 'MEDICINAL', and 'ALIMENTICIO'.

Fig. 178: Pantalla para crear insumos.

Realizado por: Los Autores de la Tesis



Administrar Insumo

CREAR INSUMO

FINCA "SAN ANTONIO"

NOMBRE : CRESTAR

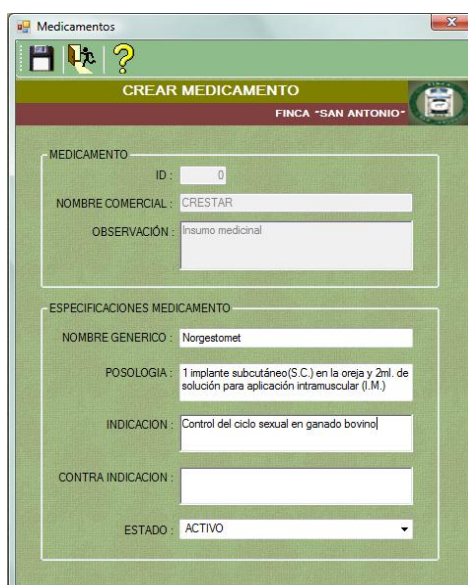
TIPO INSUMO : MEDICINAL

DETALLE : Insumo medicinal

Fig. 179: Pantalla con el registro de un insumo tipo "MEDICINAL".

Realizado por: Los Autores de la Tesis

Al crear un insumo tipo "MEDICINAL" o "ALIMENTICIO", Fig. 179, se mostrará otra pantalla para complementar datos de dicho insumo, esa pantalla la muestra la Fig. 180.



Medicamentos

CREAR MEDICAMENTO

FINCA "SAN ANTONIO"

MEDICAMENTO

ID : 0

NOMBRE COMERCIAL : CRESTAR

OBSERVACION : Insumo medicinal

ESPECIFICACIONES MEDICAMENTO

NOMBRE GENERICO : Norgestomet

POSOLOGIA : 1 implante subcutáneo(S.C.) en la oreja y 2ml. de solución para aplicación intramuscular (I.M.)

INDICACION : Control del ciclo sexual en ganado bovino

CONTRA INDICACION :

ESTADO : ACTIVO

Fig. 180: Pantalla para complementar otros datos de un insumo tipo "MEDICINAL".

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos” aparece una pantalla, Fig. 181, que es un filtro para elegir el criterio de búsqueda, en este caso se debe elegir por el tipo de insumo que queremos ver.

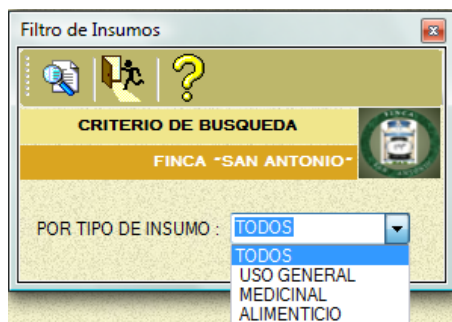


Fig. 181: Pantalla filtro para elegir el tipo de insumo a mostrar.

Realizado por: Los Autores de la Tesis

Luego de la pantalla filtro se muestra otra pantalla, Fig. 182, que contiene una tabla donde se muestran los insumos. La pantalla es la siguiente:

COD.	TIPO	INSUMO	DETALLE
20	MEDICINAL	Afrobay	Insumo medicinal
21	MEDICINAL	Azilan	Insumo medicinal
22	MEDICINAL	Acarex	Insumo medicinal
23	MEDICINAL	Albendex 10 %	Insumo medicinal
24	MEDICINAL	BaFol	Insumo medicinal
25	MEDICINAL	Benzapen forte	Insumo medicinal
26	MEDICINAL	Cloxacum	Insumo medicinal
27	MEDICINAL	Grafoleón	Insumo medicinal
28	MEDICINAL	Premaati	Insumo medicinal
29	MEDICINAL	Tadeomedin	Insumo medicinal
30	MEDICINAL	CLOXAMBIOTIC	Antibiótico para vacas en lactancia
32	MEDICINAL	CRESTAR	Insumo medicinal

Fig. 182: Pantalla que contiene una tabla donde se despliega información de los insumos tipo “MEDICINAL”.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 183, en donde están contenidos datos del insumo, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

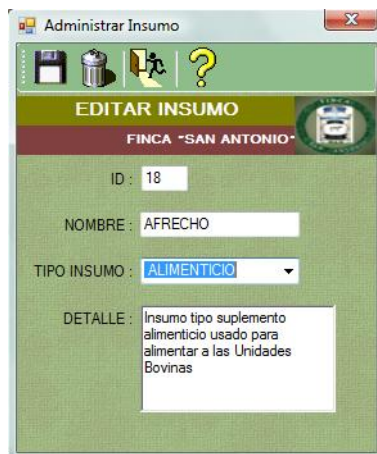


Fig. 183: Pantalla para editar un insumo.

Realizado por: Los Autores de la Tesis

Procesos Finca

Al hacer clic en esta opción se despliega una pantalla, Fig. 184, desde la cual y por medio de los botones de que dispone, se puede crear nuevos procesos, ver los existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.

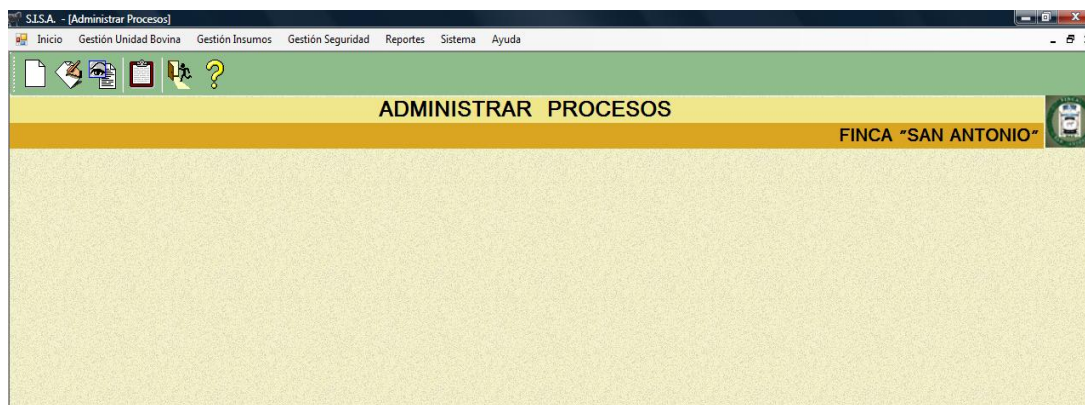


Fig. 184: Pantalla para administrar procesos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 185, en la cual se puede crear el registro de un nuevo proceso.

Procesos Finca

CREAR PROCESO

FINCA "SAN ANTONIO"

NOMBRE :

DESCRIPCION :

OBSERVACION :

Fig. 185: Pantalla para crear un proceso.
Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 186, conteniendo una tabla con todos los procesos.

S.I.S.A. - [Administrar Procesos]

Inicio Gestión Unidad Bovina Gestión Insumos Gestión Seguridad Reportes Sistema Ayuda

ADMINISTRAR PROCESOS

FINCA "SAN ANTONIO"

COD.	PROCESO	DESCRIPCIÓN	OBSERVACIÓN
1	Cercado de pastizales	Proceso que consiste en delimitar un determinado pastizal u...	Proceso periódico
3	Cercado eléctrico de pastizales	Proceso que permite tener cercado un pastizal por medios e...	
8	Control de preñez	Proceso mediante el cual se determina si una Vaca está pr...	
6	Control Mastitis	Proceso mediante el cual se realizan ciertas actividades ten...	
7	Cursaciones	Proceso mediante el cual se realizan ciertas actividades ten...	
10	Fumigación de maleza	Proceso mediante el cual se fumiga ciertas plantas consider...	
2	Mantenimiento de cercado de pastizales	Proceso que consiste en mantener en buenas condiciones l...	Este proceso se lo lleva a cabo con personal temporal
11	Mantenimiento de vías de agua	Proceso mediante el cual mantenemos en buenas condicio...	
4	Ordeño	Proceso de extraer la leche a las vacas	
9	Parto	Proceso de nacimiento de una unidad bovina	
5	Vacunación	Proceso de suministrar a las unidades bovinas virus inactiva...	

Fig. 186: Pantalla que contiene una tabla donde se despliega información de los procesos existentes.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 187, en donde

están contenidos datos del proceso, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

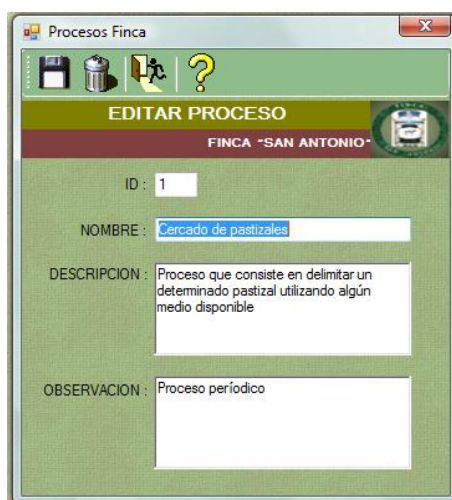


Fig. 187: Pantalla para editar un proceso.

Realizado por: Los Autores de la Tesis

Proveedores

Al hacer clic en esta opción se despliega una pantalla, Fig. 188, desde la cual y por medio de los botones de que dispone, se puede crear nuevos proveedores, ver los existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 188: Pantalla para administrar Proveedores.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 189, en la cual se puede crear el registro de un nuevo proveedor.

Fig. 189: Pantalla para crear un proveedor.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 190, conteniendo una tabla con todos los proveedores.

COD.	NOMBRE	DIRECCION	TELEFONO
1	Centro Veterinario	Tandapi	2624312
2	Ferreteria San Jorge	Tandapi	3496721
3	Bodega Universal	Tandapi	3500342

Fig. 190: Pantalla que contiene una tabla donde se despliega información de todos los proveedores.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 191, en donde

están contenidos datos del proveedor, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

The screenshot shows a window titled 'Proveedor' with a toolbar containing icons for save, delete, edit, and help. Below the toolbar is a header with the text 'EDITAR PROVEEDOR' and 'FINCA "SAN ANTONIO"'. The main area contains a form with the following fields:

- ID: 1
- NOMBRE: CENTRO VETERINARIO
- TELEFONO: 2624312
- DIRECCION: Tandapi

Fig. 191: Pantalla para editar un proveedor.

Realizado por: Los Autores de la Tesis

Compras

Al hacer clic en esta opción se despliega una pantalla, Fig. 192, desde la cual y por medio de los botones de que dispone, se puede crear nuevas compras, ver las existentes usando el botón “Mostrar Todos”, así como editarlas, eliminarlas o verlas simplemente.



Fig. 192: Pantalla para administrar las compras.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 193, en la cual se puede crear una nueva compra.

Fig. 193: Pantalla para crear una compra.

Realizado por: Los Autores de la Tesis.

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 194, conteniendo una tabla con todas las compras registradas.

COD.	PROVEEDOR	FECHA COMPRA	VALOR COMPRA	RESPONSABLE	DESCRIPCIÓN
1	Bodega Universal	07/12/2010	60.00	Isabelle Tanga	Compra realizada para arreglo de...
5	Bodega Universal	07/12/2010	40.00	Luis Chocasa	Compra hecha para arreglo de...

Fig. 194: Pantalla que contiene una tabla donde se despliega información de las compras registradas.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 195, en donde están contenidos datos de la compra, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

Cabecera Adquisición

EDITAR COMPRA FINCA "SAN ANTONIO"

DATOS COMPRA

ID: 5

PROVEEDOR: Bodega Universal FECHA: martes, 07 de diciembre de 2010

RESPONSABLE: Luis Chicaiza TOTAL: 40,00

DESCRIPCIÓN: Compra hecha para arreglo de casa

COD	INSUMO	CANTIDAD	V / U	SUBTOTAL
6	Botas DE CAUCHO	2	12,00	24,00
7	Clavos DE ACERO	3	1,00	3,00
8	Cabo fortex GRUESO	12	1,00	12,00
*				

Fig. 195: Pantalla para editar una compra.

Realizado por: Los Autores de la Tesis

Egresos

Al hacer clic en esta opción se despliega una pantalla, Fig. 196, desde la cual y por medio de los botones de que dispone, se puede crear registros de egresos, ver los existentes usando el botón "Mostrar Todos", así como editarlos, eliminarlos o verlos simplemente.



Fig. 196: Pantalla para administrar los egresos de los insumos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón "Nuevo" aparece una pantalla como la de la Fig.197, en la cual se puede crear el registro de un nuevo egreso.

The screenshot shows a web application window titled 'Egreso'. The main heading is 'CREAR EGRESO' for 'FINCA SAN ANTONIO'. The form contains the following fields:

- EVENTO:** Cercado de pastizales (dropdown menu)
- RESPONSABLE:** (text input field)
- FECHA:** martes .07 de diciembre de 2010 (dropdown menu)
- OBSERVACION:** (text area)

Below the form is a table with the following structure:

COD.	INSUMO	CANT.
**		

At the bottom of the table, there are navigation controls showing '1 of 1'.

Fig. 197: Pantalla para crear un egreso.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 198, conteniendo una tabla con todos los egresos.

The screenshot shows a web application window titled 'ADMINISTRACIÓN DE EGRESOS' for 'FINCA SAN ANTONIO'. The table displays the following data:

COD.	EVENTO	FECHA DE USO	RESPONSABLE	OBSERVACIÓN
1	Mantenimiento de vías de agua	07/12/2010	Angel Lara	

Fig. 198: Pantalla que contiene una tabla donde se despliega información de los egresos registrados.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 199, en donde están contenidos datos de los egresos, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.



Fig. 199: Pantalla para editar un egreso.

Realizado por: Los Autores de la Tesis

Información Adicional

Esta opción tiene, a la vez, otras dos opciones que son:

- Insumo Medicinal
- Insumo Alimenticio

Insumo Medicinal

Al hacer clic en esta opción se despliega una pantalla, Fig. 200, desde la cual y por medio de los botones de que dispone, se puede ver los insumos medicinales o medicamentos existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 200: Pantalla para administrar los insumos medicinales o medicamentos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 201, conteniendo una tabla con todos los medicamentos.



CODIGO	NOMBRE COMERCIAL	NOMBRE GENERICO	POSOLOGIA	INDICACION	CONTRA INDICACION
20	Aftobov	Virus de fiebre aftosa inactiv...	5 ml. Intramuscular por anim...	Vacuna para la prevención de la fiebre...	Conservar en refrigeració...
21	Acolán	Aciclovir, Óxido de zinc	Lavar y secar la zona y apli...	Tratamiento de heridas, quemaduras in...	Si se trata de vacas en p...
22	Acarex	Amitraz	Se usa en baños de asperci...	Antiparasitario externo contra garrapat...	No usar en baños de inm...
23	Albendex 10 %	Albendazol	Vía oral	Bovinos. Control y tratamiento de nem...	
24	Bañol	Amtraz(diamidina)	Dilución de 1 por 600	Control de garrapatas y piojos	No usar en equinos ni en...
25	Benzapen forte	Penicilina G. Benzatínica	Administrar por vía intramus...	Antibiótico para el control y tratamiento...	Puede causar hipersensit...
26	Cloxacum	Cloxacilina	Aplicar una jeringuilla en ca...	Tratamiento profiláctico para las mastit...	Si se ordeña el animal de...
27	Grafoleón	Benzoato de estradiol	Queda a criterio del veterina...	En la hipoplasia genital, debida a hipo...	No administrar en hembra...
28	Premasti	Yodo, polivinil pirolidona	Sumergir los pezones en la ...	Desinfectante a base yodo. Prevenció...	No ingerir, no contacto c...
29	Tadecmectin	Ivermectina	200 mcg por kg de peso vivo	Gran efectividad en el control de parás...	Uso subcutáneo
30	CLOXAMBIOTIC	Cloxacilina y Ampicilina	Ganado adulto 10 ml diarios...	Tratamiento curativo para mastitis agu...	
32	CRESTAR	Norgestomet	1 implante subcutáneo(S.C.)...	Control del ciclo sexual en ganado bov...	

Fig. 201: Pantalla que contiene una tabla donde se despliega información de los medicamentos existentes.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 202, en donde están contenidos datos del medicamento, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.



Medicamento

EDITAR MEDICAMENTO

FINCA "SAN ANTONIO"

MEDICAMENTO

ID : 20

NOMBRE COMERCIAL: Aftobov

OBSERVACION: Insumo medicinal

ESPECIFICACIONES MEDICAMENTO

NOMBRE GENERICO: Virus de fiebre aftosa inactivado

POSOLOGIA: 5 ml. Intramuscular por animal, revacunar cada 6 meses.

INDICACION: Vacuna para la prevención de la fiebre aftosa

CONTRA INDICACION: Conservar en refrigeración, pero no congeladas, proteja de la luz

ESTADO: ACTIVO

Fig. 202: Pantalla para editar un medicamento.

Realizado por: Los Autores de la Tesis

Insumo Alimenticio

Al hacer clic en esta opción se despliega una pantalla, Fig. 203, desde la cual y por medio de los botones de que dispone, se puede ver los insumos nutricionales existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 203: Pantalla para administrar suplementos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 204, conteniendo una tabla con todos los suplementos nutricionales.

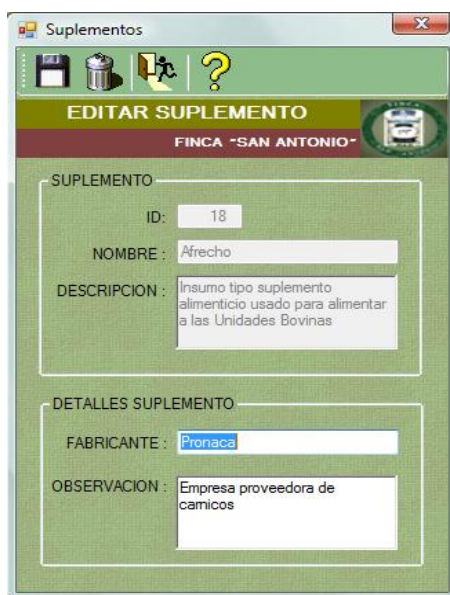
CODIGO	SUPLEMENTO	FABRICANTE	OBSERVACION
18	Alfresco	Pionaca	Empresa proveedora de camicos
19	Melaza	Agripac	Empresa fabricante de suplementos alimenticios de varios ti...

Fig. 204: Pantalla que contiene una tabla donde se despliega información de los suplementos existentes.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 205, en donde

están contenidos datos del suplemento, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.



The screenshot shows a web application window titled 'Suplementos'. The main heading is 'EDITAR SUPLEMENTO' with a sub-heading 'FINCA "SAN ANTONIO"'. The form is divided into two sections: 'SUPLEMENTO' and 'DETALLES SUPLEMENTO'. The 'SUPLEMENTO' section includes fields for 'ID' (value: 18), 'NOMBRE' (value: Afrecho), and 'DESCRIPCION' (value: Insuno tipo suplemento alimenticio usado para alimentar a las Unidades Bovinas). The 'DETALLES SUPLEMENTO' section includes fields for 'FABRICANTE' (value: Pronaca) and 'OBSERVACION' (value: Empresa proveedora de camicos).

Fig. 205: Pantalla para editar un suplemento.

Realizado por: Los Autores de la Tesis

5. Opción Gestión Seguridad

Esta opción del menú principal contiene a su vez a otras opciones, que son:

- Usuarios
- Perfiles
- Permisos
- Propietarios
- Cambiar Claves

Usuarios

Al hacer clic en esta opción se despliega una pantalla, Fig. 206, desde la cual y por medio de los botones de que dispone, se puede crear usuarios, ver los existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.

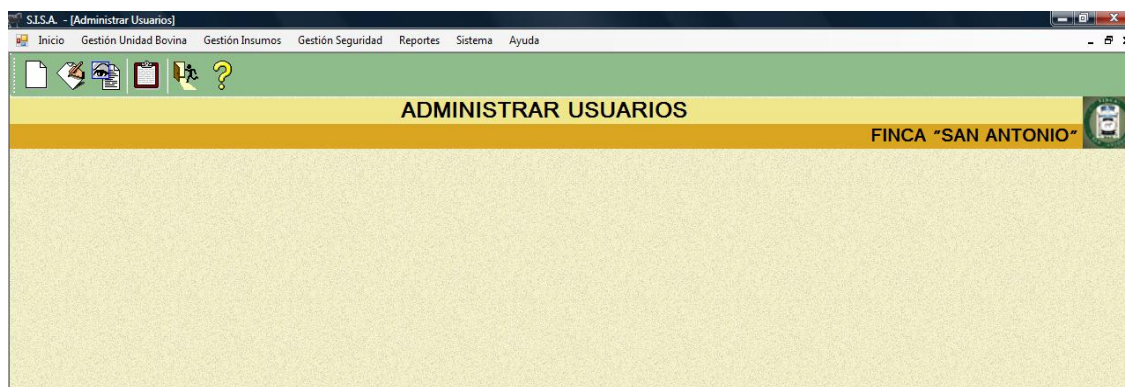


Fig. 206: Pantalla para administrar usuarios.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 207, en la cual se puede crear el registro de un nuevo usuario.

Fig. 207: Pantalla para crear un nuevo usuario.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 208, conteniendo una tabla con todos los usuarios.

COD.	NOMBRES	APELLIDOS	LOGIN	DIRECCION	TELEFONO	CELULAR	ES
3	Jesús Honorato	Chicaiza Tayupanta	jesus	Latacunga		99803052	ACTI
2	Luis Modesto	Chicaiza Tayupanta	luis	Cayambe	22110650	95008379	ACTI
7	Marco Antonio	Chicaiza Tayupanta	marco	Guayaquil	42309034	98167709	ACTI
5	María Augusta	Chicaiza Tayupanta	augusta	Quito	22647336	82123279	ACTI
6	Mónica María	Chicaiza Tayupanta	monica	Quito	2662393		ACTI
4	Segundo Rosalino	Chicaiza Tayupanta	segundo	Quito	22600101	84379261	ACTI
8	Trabajador	Finca	trabajador	Finca			ACTI
9	Veterinario	Finca	veterinario	Finca			ACTI

Fig. 208: Pantalla que contiene una tabla donde se despliega información de los usuarios existentes.

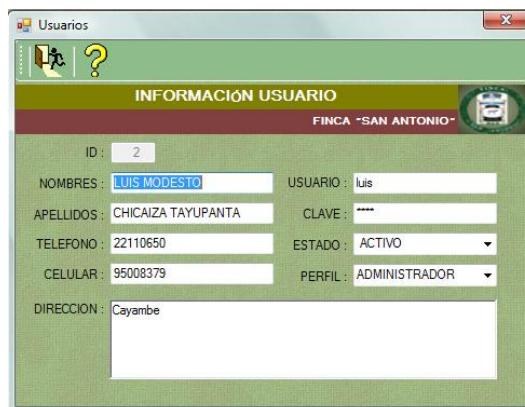
Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 209, en donde están contenidos datos del usuario, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

Fig. 209: Pantalla para editar los datos de un usuario.

Realizado por: Los Autores de la Tesis

A continuación se ve una pantalla, Fig. 210, en la cual muestra datos de un usuario, pero solo de manera informativa, como lo dice su título. No se puede modificar nada.



INFORMACIÓN USUARIO	
FINCA "SAN ANTONIO"	
ID :	2
NOMBRES :	LUIS MODESTO
APELLIDOS :	CHICAIZA TAYUPANTA
TELEFONO :	22110650
CELULAR :	95008379
DIRECCIÓN :	Cayambe
USUARIO :	luis
CLAVE :	****
ESTADO :	ACTIVO
PERFIL :	ADMINISTRADOR

Fig. 210: Pantalla informativa de los datos de un usuario.

Realizado por: Los Autores de la Tesis

Perfiles

Al hacer clic en esta opción se despliega una pantalla, Fig. 211, desde la cual y por medio de los botones de que dispone, se puede crear perfiles, ver los existentes usando el botón "Mostrar Todos", así como editarlos, eliminarlos o verlos simplemente.



Fig. 211: Pantalla para administrar perfiles.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón "Nuevo" aparece una pantalla como la de la Fig. 212, en la cual se puede crear un nuevo perfil.



Fig. 212: Pantalla para crear un perfil.
Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 213, conteniendo una tabla con todos los perfiles.

COD.	NOMBRE	DESCRIPCION	ESTADO
2	ADMINISTRADOR	Usuario con acceso a todos los módulos del sistema	ACTIVO
5	PROPIETARIO	Usuario con acceso restringido o limitado	ACTIVO
4	USUARIO	Usuario con acceso restringido o limitado	ACTIVO
3	VETERINARIO	Usuario con acceso a los módulos relacionados con el control reproductivo, sanit...	ACTIVO

Fig. 213: Pantalla que contiene una tabla donde se despliega información de los perfiles existentes.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 214, en donde están contenidos datos del perfil, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.



Fig. 214: Pantalla para editar los datos de un perfil.

Realizado por: Los Autores de la Tesis

Permisos

Al hacer clic en esta opción se despliega una pantalla, Fig. 215, desde la cual y por medio de los botones de que dispone, se puede crear permisos, ver los existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 215: Pantalla para Administrar Permisos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 216, en la cual se puede crear un nuevo permiso.



Fig. 216: Pantalla para crear permisos.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 217, conteniendo una tabla con todos los suplementos nutricionales.

CODIGO	OPCION	PERFIL USUARIO
36	Gestión & Unidad Bovina	ADMINISTRADOR
37	&Administrar UB	ADMINISTRADOR
38	Control & Sanitario	ADMINISTRADOR
39	&Historia Clinica	ADMINISTRADOR
40	Vacunación	ADMINISTRADOR
41	Control & Reproductivo	ADMINISTRADOR
42	Complementos Nutricionales	ADMINISTRADOR
43	Administrar & Complementos	ADMINISTRADOR
44	Suministrar Complementos	ADMINISTRADOR
45	Datos Generales	ADMINISTRADOR
46	Enfermedades	ADMINISTRADOR
47	Enfermedad Tratamiento	ADMINISTRADOR
48	Tipos Madurez	ADMINISTRADOR
49	Evento Reproductivo	ADMINISTRADOR
50	Gestión Insumos	ADMINISTRADOR
51	Inventario	ADMINISTRADOR
52	Insumos	ADMINISTRADOR
53	&Presentaciones	ADMINISTRADOR
54	Procesos & Finca	ADMINISTRADOR
55	Proveedores	ADMINISTRADOR
56	&Compras	ADMINISTRADOR
57	Egresos	ADMINISTRADOR
58	Información Adicional	ADMINISTRADOR
59	Insumo Medicinal	ADMINISTRADOR

Fig. 217: Pantalla que contiene una tabla donde se despliega información de los suplementos existentes.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 218, en donde están

contenidos datos del permiso, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.



Fig. 218: Pantalla para editar permisos.

Realizado por: Los Autores de la Tesis

Propietarios

Al hacer clic en esta opción se despliega una pantalla, Fig. 219, desde la cual y por medio de los botones de que dispone, se puede crear propietarios, ver los existentes usando el botón “Mostrar Todos”, así como editarlos, eliminarlos o verlos simplemente.



Fig. 219: Pantalla para administrar Propietarios.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Nuevo” aparece una pantalla como la de la Fig. 220, en la cual se puede crear un nuevo propietario.

Fig. 220: Pantalla para crear un Propietario.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 221, conteniendo una tabla con todos los Propietarios.

COD.	NOMBRES	MARCA	OBSERVACION	ESTADO
8	Común	DOBLE CHILPE	Propiedad en Común	ACTIVO
2	Jesus Honorato Chicaiza Tayupanta	MOCHO	Primer hijo del matrimonio	ACTIVO
4	Luis Modesto Chicaiza Tayupanta	UKDPASHCA	Tercer hijo del matrimonio	ACTIVO
7	Marco Antonio Chicaiza Tayupanta	JAWAPASHCA	Sexto hijo del matrimonio	ACTIVO
5	María Agusta Chicaiza Tayupanta	LLAVE	Cuarta hija del matrimonio, primera mujer	ACTIVO
1	María Feliciane Tayupanta Caizalusa	DOBLE CHILPE	Esposa del Fundador de la finca	ACTIVO
6	Mónica María Chicaiza Tayupanta	PALMA	Quinta hija del matrimonio, segunda mujer	ACTIVO
3	Segundo Rosalino Chicaiza Tayupanta	VENTANA	Segundo hijo del matrimonio	ACTIVO

Fig. 221: Pantalla que contiene una tabla donde se despliega información de los Propietarios.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 222, en donde están contenidos datos del Propietario, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

Fig. 222: Pantalla para editar datos de un Propietario.

Realizado por: Los Autores de la Tesis

A continuación se puede ver una pantalla, Fig. 223, la cual muestra datos de un usuario, pero solo de manera informativa, como lo dice su título. No se puede modificar nada.

Fig. 223: Pantalla informativa de los datos de un Propietario.

Realizado por: Los Autores de la Tesis

Cambiar Claves

Esta opción permite crear claves alternas, para los usuarios del sistema, para cuando por algún motivo las hayan olvidado. Estas claves son provisionales, pues al momento de ingresar con ellas al sistema, se les obligará a cambiarlas.

La pantalla para cambiar las claves es la siguiente.



Fig. 224: Pantalla para Crear Claves.
Realizado por: Los Autores de la Tesis

Opción Sistema

Esta opción tiene 2 opciones más, que son:

- Auditoría
- Administrar Catálogo

Auditoría

Administrar Catálogo

Al hacer clic en esta opción se despliega una pantalla, Fig. 225, desde la cual y por medio de los botones de que dispone, se puede crear propietarios, ver los existentes usando el botón "Mostrar Todos", así como editarlos, eliminarlos o verlos simplemente.

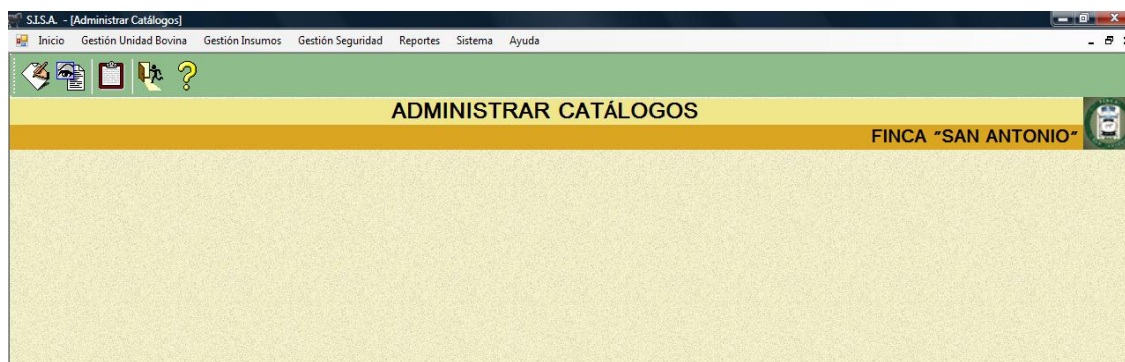


Fig. 225: Pantalla para Administrar el catálogo.

Realizado por: Los Autores de la Tesis

Al hacer clic en el botón “Mostrar Todos”, aparece una pantalla, Fig. 226, conteniendo una tabla con todos los Propietarios.

COD.	ITEM	DESCRIPCION
1	TIPO CATALOGO	Tipos de Catálogos
2	ESTADO REGISTRO	Determina el estado de un registro
3	SEXO UB	Sexo que pueden tener las Unidades Bovinas
4	RAZA	Diferentes razas, de Unidades Bovinas, que existen.
5	PROCEDECIA	Zona desde la cual procede una Unidad Bovina
6	SEÑAL	Manera de identificar al propietario de una Unidad Bovina
7	TIPO UNIDAD BOVINA	Tipo de ganado, dependiendo de su objetivo
8	TIPO DESCARTE	Razón de la inexistencia de una Unidad Bovina en la finca
9	TIPO INSUMO	Determina el tipo de insumo
10	SI/NO	Opción de realizar una acción
11	PERIODO	Periodos de tiempos
12	UNIDAD	Unidades de medida
13	UNIDAD TIEMPO	Tiempos de un evento
14	TIPO REPRODUCCIÓN	Tipo de formas de parto
15	TIPO TRATAMIENTO	Tipos de tratamientos

Fig. 226: Pantalla que muestra los registros que se tiene en el Catálogo.

Realizado por: Los Autores de la Tesis

Desde esta pantalla y usando el botón “Editar” o haciendo doble clic sobre el registro de la tabla se puede desplegar una pantalla, Fig. 227, en donde están contenidos datos del Propietario, para poderlos editar ó para eliminarlos usando el botón “Eliminar”.

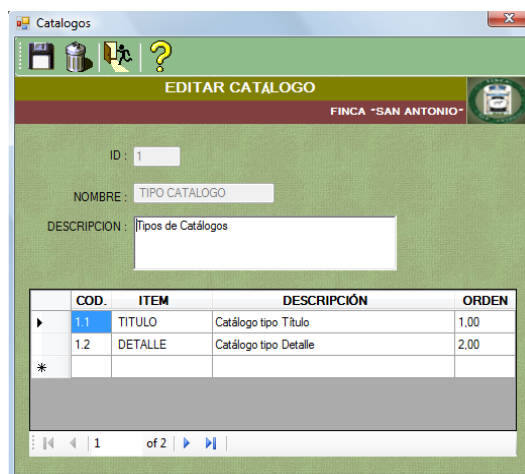


Fig. 227: Pantalla para editar el Catálogo.

Realizado por: Los Autores de la Tesis



Fig. 228: Pantalla informativa del catálogo.

Realizado por: Los Autores de la Tesis

Opción Ayuda

Esta opción tiene tres opciones más, que son las siguientes:

- Sistema
- La Finca
- Acerca De...

Sistema

Al hacer clic en esta opción se despliega el Manual del Usuario.

La Finca

Al hacer clic en esta opción se despliega un documento tipo pdf donde está la información referente a la finca "San Antonio".

Acerca De...

Al hacer clic en esta opción se despliega una pantalla con información de los diseñadores del sistema, así como otros datos relacionados al mismo.

CAPÍTULO V

PRUEBAS E IMPLEMENTACIÓN DEL SISTEMA.

5.1 Identificar Anomalías.

Anomalías en la conexión con la Base de Datos.

Una vez terminado el sistema, que fue desarrollado y codificado en lenguaje C# con base de datos en postgres, se ha procedido a instalarlo en las máquinas indicadas para el efecto, unas tenían sistemas operativos Windows Vista de 32 bits y alguna otra tenía Windows 7 de 64 bits, en los dos sistemas estaban instalados el software necesario, pero al momento de conectar a la base de datos para iniciar el sistema, en las máquinas que tenían procesadores de 64 bits, no se establecía la conexión y no funcionaba el sistema, el problema se debía a que el driver del postgres ODBC, que se había instalado, era una versión únicamente para procesadores de 32 bits.

Por esta razón se tiene que señalar de manera clara que el sistema SISA debe ser instalado, únicamente, en máquinas con procesadores de 32 bits, ya que por el momento es lo que hay disponible, esto no quiere decir que a futuro no se pueda disponer de un driver para 64 bits.

5.2 Modificación de código.

Mientras se desarrolló el sistema, de manera paralela se sujetó a las especificaciones requeridas, lo cual, supuestamente, garantizaba que el receptor del sistema estaría conforme con el mismo, pero esto no ocurrió así y más bien solicitó realizar cambios en la estructura de algunos formularios, específicamente en aquellos relacionados con las historias clínicas y el control reproductivo, por lo que hubo que modificar el código relacionado y modificar la base datos.

Más adelante al momento de implementar los reportes, se tuvo que realizar varios cambios, para tener reportes parametrizados, en algunos casos con más de un parámetro, esto ha permitido tener reportes dinámicos.

En el apartado de "Anexos" se muestra el código de la clase FrmReporteEReproductivo y el método AdicionarParametros, en el cual están implementados los parámetros.

5.3 Prueba de código modificado.

El resultado de la modificación del código se puede ver en la nueva interfaz de la historia clínica de las unidades bovinas, la cual ha sido rediseñada de manera que permita mostrar, en su diseño, todos los tratamientos que ha tenido, los cuales se pueden ver enlistados en un datagrid. Lo que equivale a decir que ahora se tiene un sistema maestro-detalle.

En el siguiente gráfico se puede ver la interfaz original, en donde originalmente se ingresaba los medicamentos con sus dosis y se mandaba a guardar.

The screenshot shows a web application window titled "Historia Clínica UB". The main heading is "CREAR HISTORIA CLINICA" with a sub-heading "FINCA 'SAN ANTONIO'". The form is divided into two main sections: "DATOS HISTORIA CLINICA" and a table for treatments.

DATOS HISTORIA CLINICA:

- UNIDAD BOVINA: Número 33, manja (dropdown)
- OBSERVACIÓN: (text area)
- SINTOMAS: (text area)
- FECHA: viernes, 03 de diciembre de 2010 (dropdown)
- ENFERMEDAD: ANTRAX, CARBUNCO (dropdown)
- FECHA CONTROL: viernes, 03 de diciembre de 2010 (dropdown)
- TIPO TRATAM.: PREVENTIVO (dropdown)
- TRATAMIENTO: (dropdown)

Treatment Table:

CODIGO	MEDICAMENTO	DOSIS	UNIDAD
*			

At the bottom of the table, there is a pagination control showing "0 of 0" records.

Fig. 229: Interfaz para crear una Historia Clínica, anterior a la modificación del código.

Realizado por: Los Autores de la Tesis

La nueva interfaz es la que se muestra a continuación, en la cual se puede ver los tratamientos que ha tenido la unidad bovina y en caso que se quiera saber detalles del tratamiento, se tiene que hacer doble clic sobre el mismo, entonces en otra interfaz se muestra detalles del tratamiento.

TRATAMIENTO	SINTOMA	F. TRATAMIENTO	F. CONTROL	TIPO REGISTRO
C ANTRAX	Fiebre	10/02/2011	10/02/2011	CURATIVO

Fig. 230: Interfaz de la Historia Clínica, después de modificar el código.

Realizado por: Los Autores de la Tesis

MEDICAMENTO	DOSIS	UNIDAD
AFTOBOV	3.00	ONZAS

Fig. 231: Interfaz donde se muestra detalles del tratamiento recibido.

Realizado por: Los Autores de la Tesis

Otra muestra de la modificación del código, se refleja en la interfaz del reporte por eventos reproductivos, el cual ahora es dinámico, pudiendo en un mismo reporte generar distinta información.

En la siguiente secuencia gráfica se puede ver un mismo reporte pero con distinto tipo de información, de acuerdo a los parámetros seleccionados.



Fig. 232: Interfaz del módulo Reportes para Eventos Reproductivos.
Realizado por: Los Autores de la Tesis



Fig. 233: Interfaz que muestra un reporte de "TODOS" los eventos reproductivos dentro de un rango de tiempo, definido por el usuario.

Realizado por: Los Autores de la Tesis



Fig. 234: Interfaz que muestra el reporte de un evento reproductivo "CELO" dentro de un rango de tiempo, definido por el usuario.

Realizado por: Los Autores de la Tesis

5.4 Entrega de Software.

Junto con la entrega del software se va a dejar el Manual del Usuario para el manejo del sistema SISA, además se capacitará a los usuarios del mismo.

El sistema en si quedará instalado en las máquinas que disponga el receptor del sistema.

5.5 Instalación del software.

Una vez listo el sistema "SISA", lo que corresponde en este apartado, es crear el instalador del sistema.

A continuación se describe paso a paso este proceso. Luego de aquello se puede decir que la instalación en las máquinas que albergarán al sistema no tiene nada de difícil, así como su desinstalación.

Creación del Instalador del Sistema SISA

En las siguientes páginas se describe paso a paso el proceso para crear el instalador del Sistema SISA.

Al estar en la pantalla principal del Microsoft Visual Studio, cambiar la opción "Debug", por la de "Release".

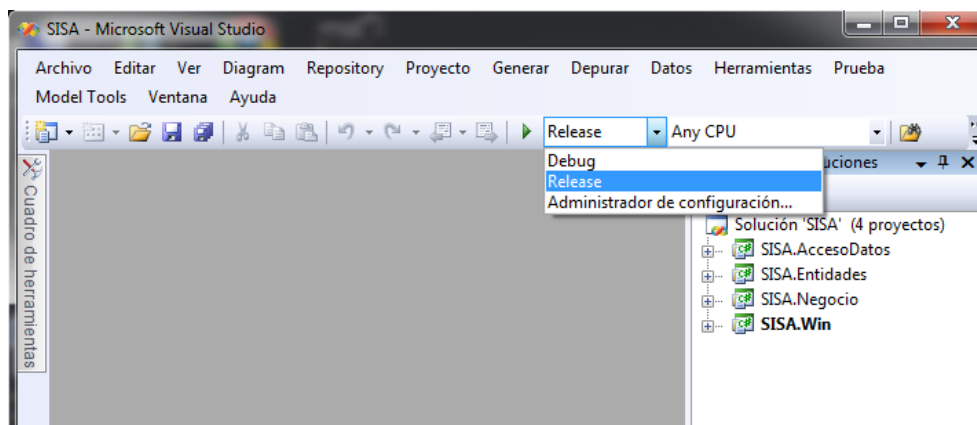


Fig. 235: Pantalla guía previa a la creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

Realizado esto, entonces hacer clic izquierdo, con el mouse, en "Ver", luego a "Explorador de soluciones". Entonces se muestra en el lado derecho el "Explorador de soluciones - SISA".

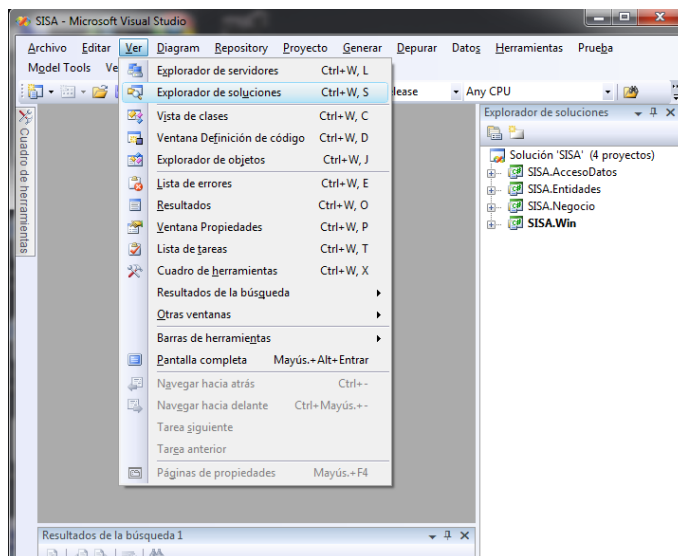


Fig. 236: Pantalla para iniciar el proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

Ubicarse con el cursor en el lado derecho de la pantalla sobre "Solución SISA", hacer clic derecho, entonces se despliega una lista, ir a "Agregar" y luego "Nuevo Proyecto".

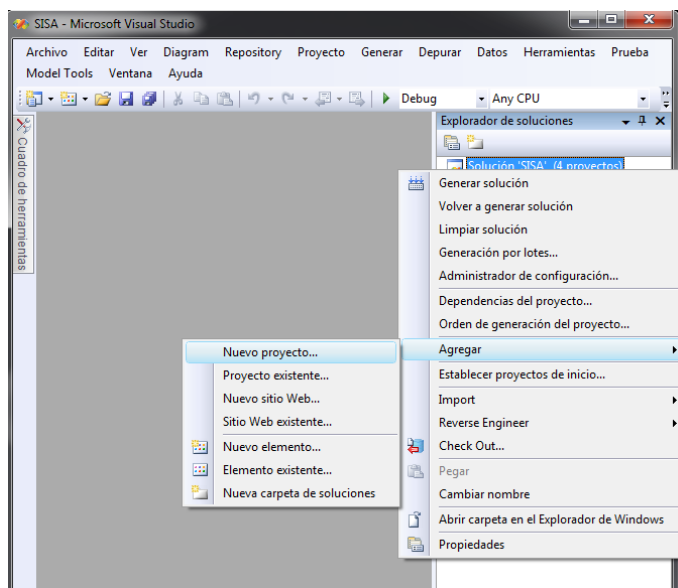


Fig. 237: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

Entonces aparece la siguiente pantalla:

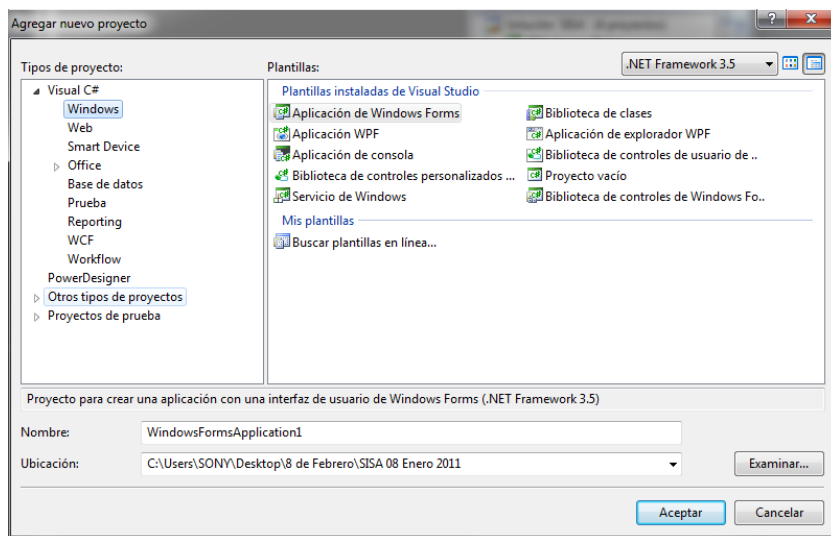


Fig. 238: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

De aquí elegir "Otros tipos de proyectos", hacer clic y se despliega otra lista, debajo de esta opción, de ahí elegir "Instalación e implementación".

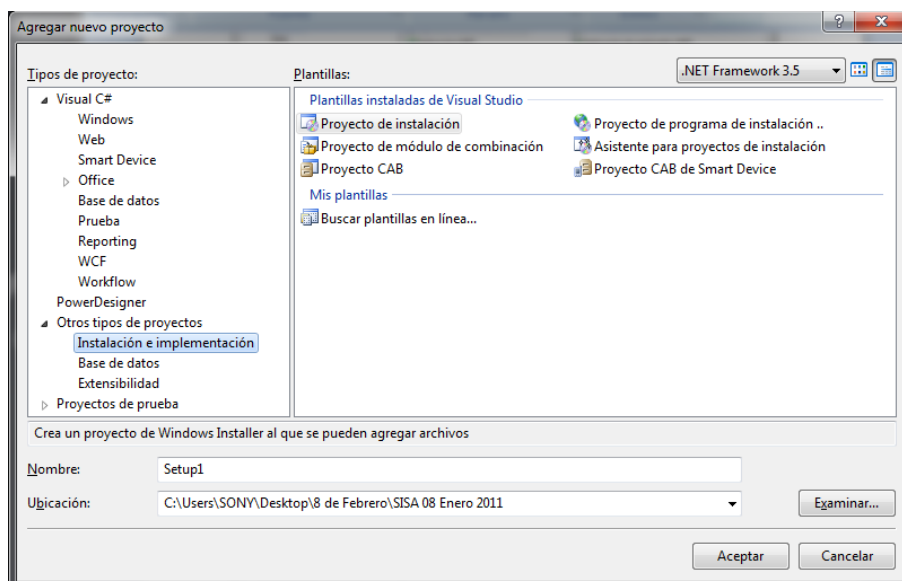


Fig. 239: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

Poner "Nombre" y asignar una "Ubicación" al proyecto.

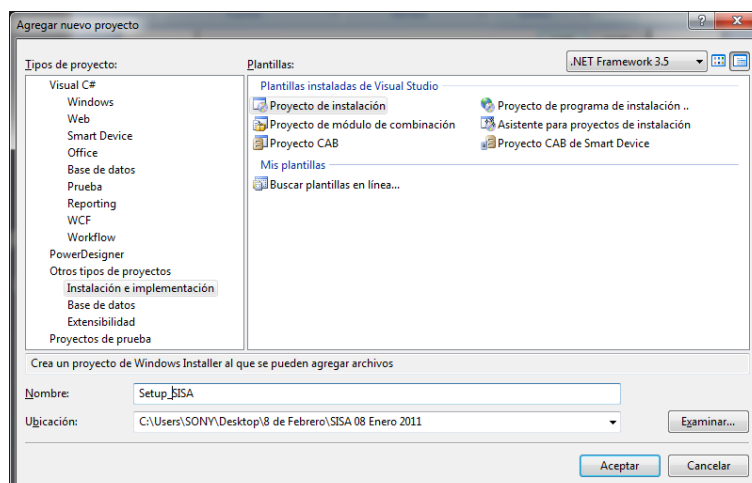


Fig. 240: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

Ir a la barra de tareas, en la opción "Generar" hacer clic para que se despliegue una lista de opciones, elegir "Generar solución", esto se puede ver en el siguiente gráfico.

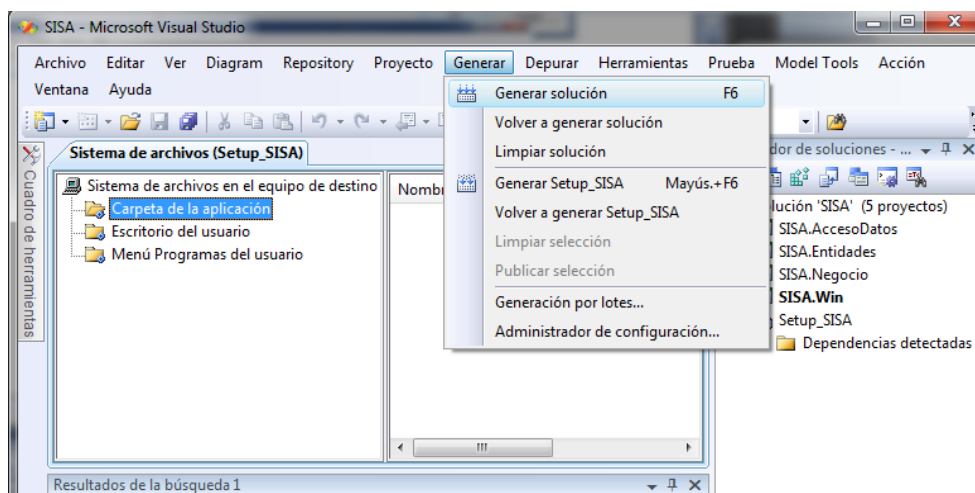


Fig. 241: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

A continuación ubicarse en el lado derecho de la pantalla, en la carpeta "Carpeta de la aplicación", "Agregar", "Resultados del proyecto" y hacer clic, esto se puede ver en el siguiente gráfico.

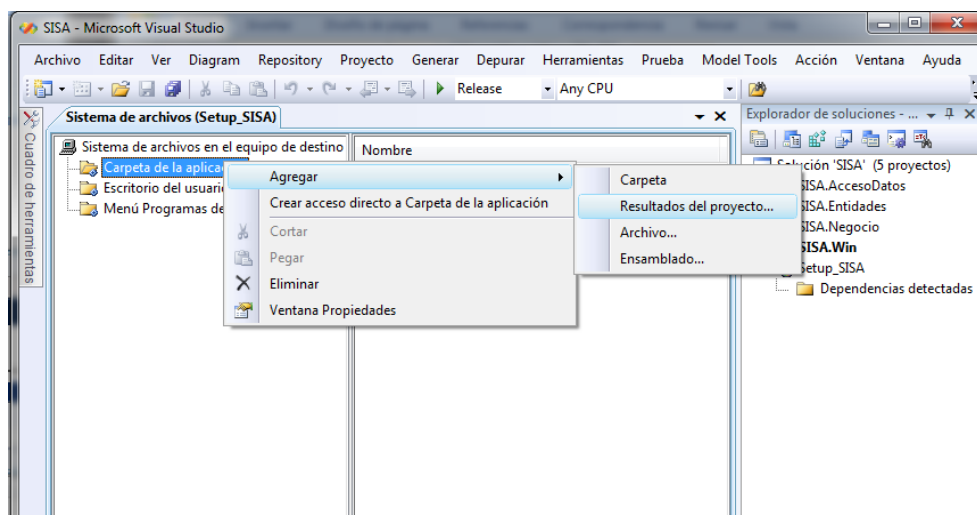


Fig. 242: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

Aparece entonces una pantalla, que se ve en el siguiente gráfico, en ella frente a la palabra Proyecto hay un combobox de donde se tiene que elegir una opción a la vez, de entre las cuatro existentes, del cuadro inferior elegir "Resultado principal", y luego aceptar.

Este mismo proceso se lo realiza para las cuatro opciones, "SISA.AccesoDatos", "SISA.Entidades", "SISA.Negocio" y "SISA.Win".

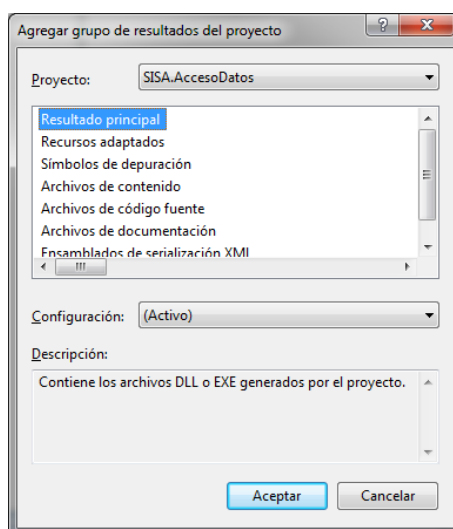


Fig. 243: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

El resultado del proceso anterior se refleja en el siguiente gráfico.

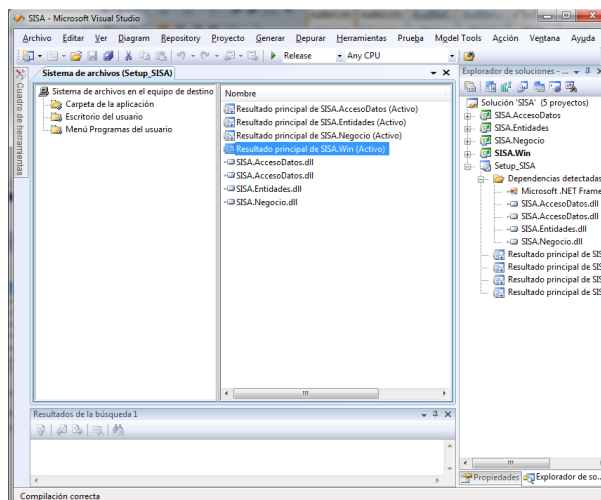


Fig. 244: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis

El sistema tiene una carpeta donde están los reportes, por lo tanto también hay que añadirlos, y esto se lo hace de la siguiente manera. Ubicarse en la carpeta "Carpeta de la aplicación", "Agregar", "Carpeta", como se ve en el siguiente gráfico.

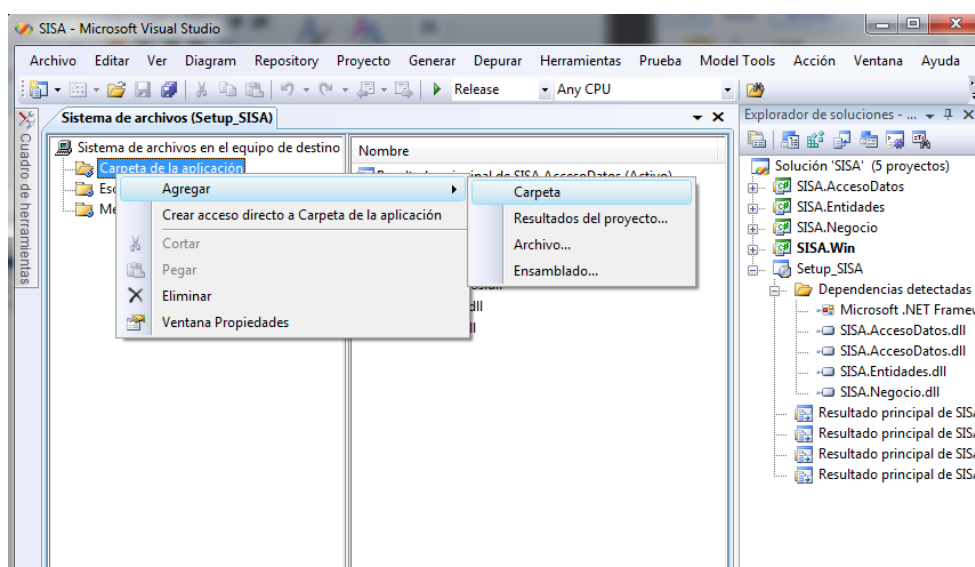


Fig. 245: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

El resultado es el siguiente.

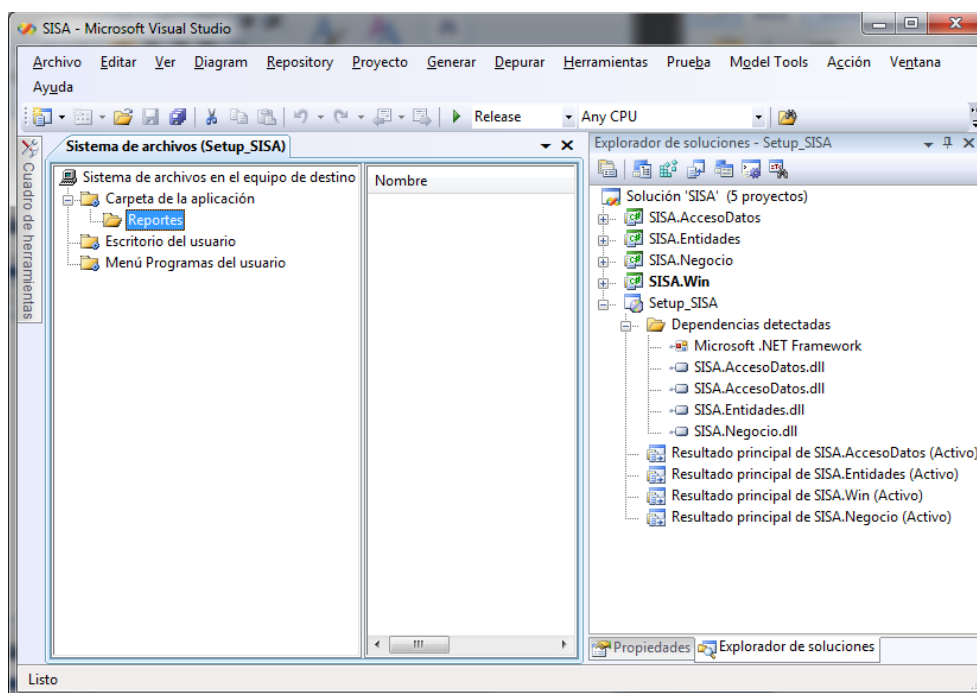


Fig. 246: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Cambiar el nombre de la carpeta, ponerle "Reportes", ubicarse en esa carpeta luego "Agregar", "Archivo", entonces aparece una pantalla para buscar donde están los archivos de los reportes del sistema, ubicarlos y añadirlos a la carpeta.

El resultado es el siguiente.

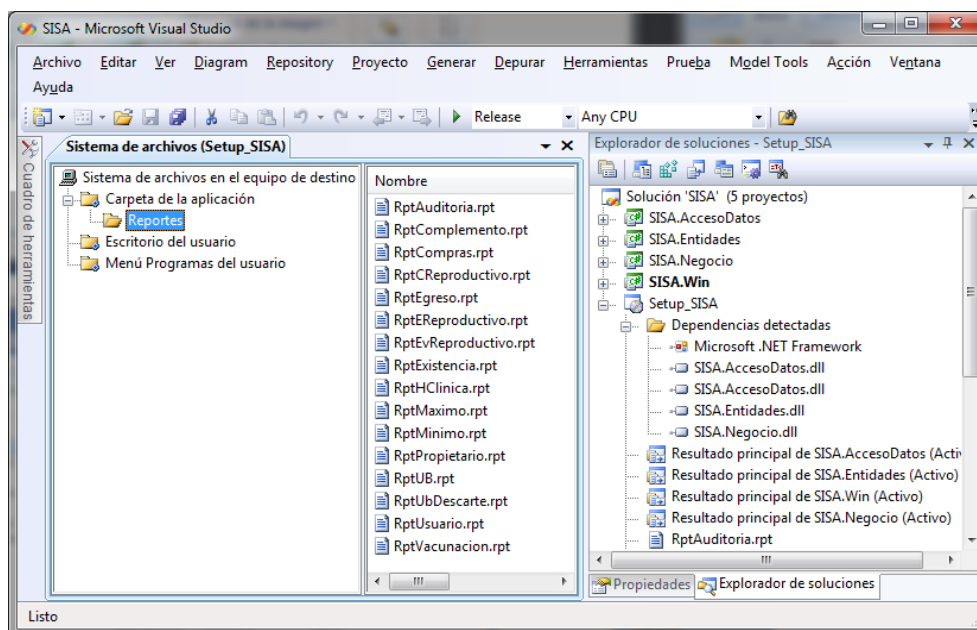


Fig. 247: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Del sistema también forman parte dos archivos adicionales, uno contiene información de la finca y otro es el manual de ayuda, los cuales deben ser añadidos. Para esto se debe seguir la siguiente secuencia.

Ubicarse en la carpeta "Carpeta de la aplicación", "Agregar", "Archivo", como se ve en el siguiente gráfico.

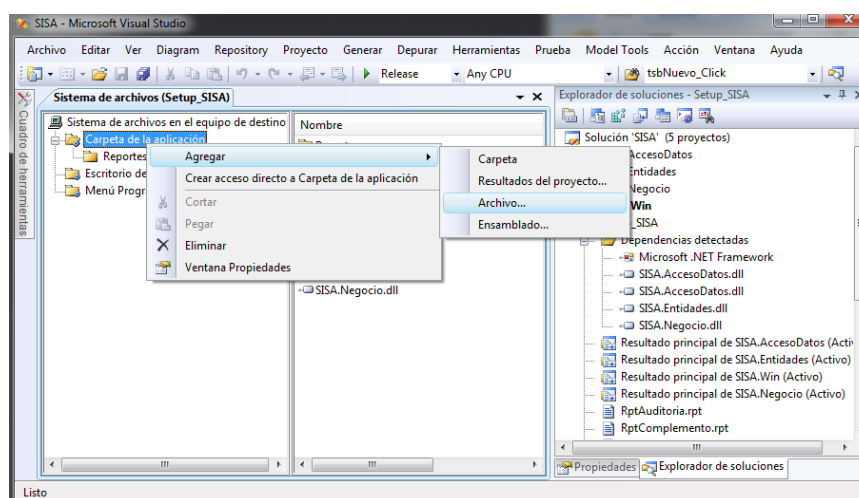


Fig. 248: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Entonces aparece una pantalla para buscar los archivos antes, ubicarlos y añadirlos a la carpeta.

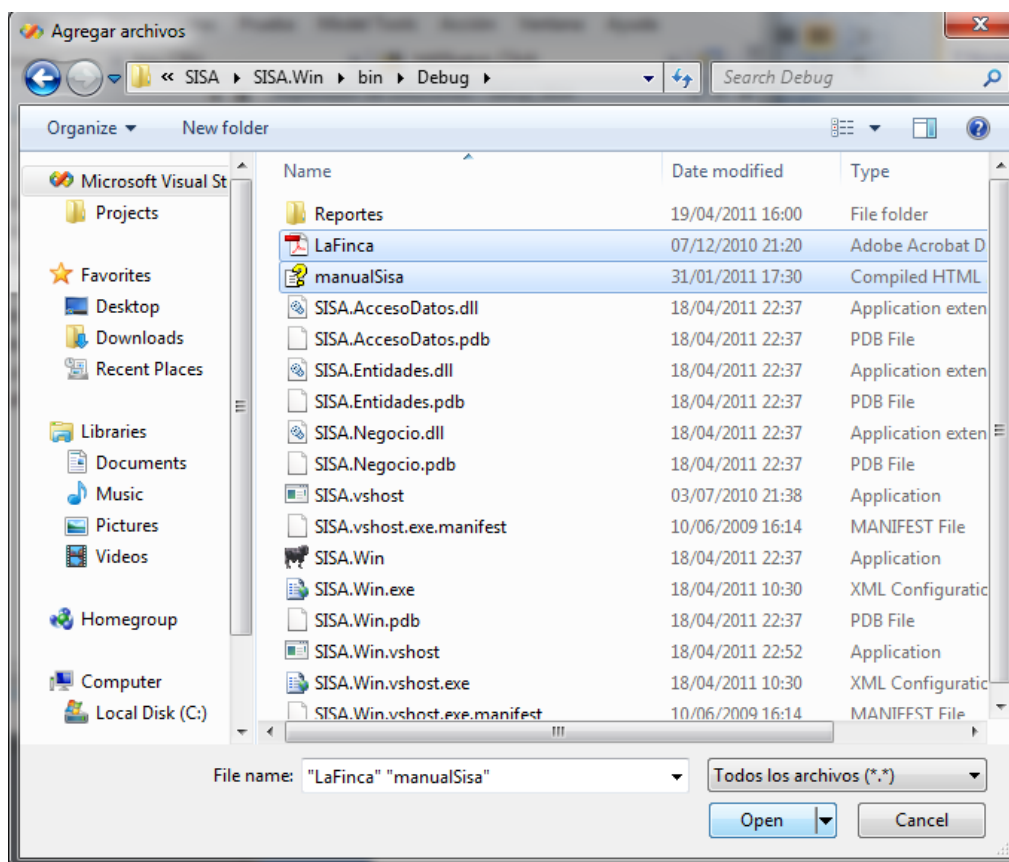


Fig. 249: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Ahora se procederá a crear el acceso directo del sistema, el cual aparecerá en el escritorio de las máquinas donde se instale el programa.

Ubicarse en el archivo "Resultado principal de SISA.Win", clic derecho, "Crear acceso directo a Resultado principal de SISA.Win(activo)". Se puede cambiar el nombre, para este caso se llamará "SISA". El resultado se ve el siguiente gráfico.

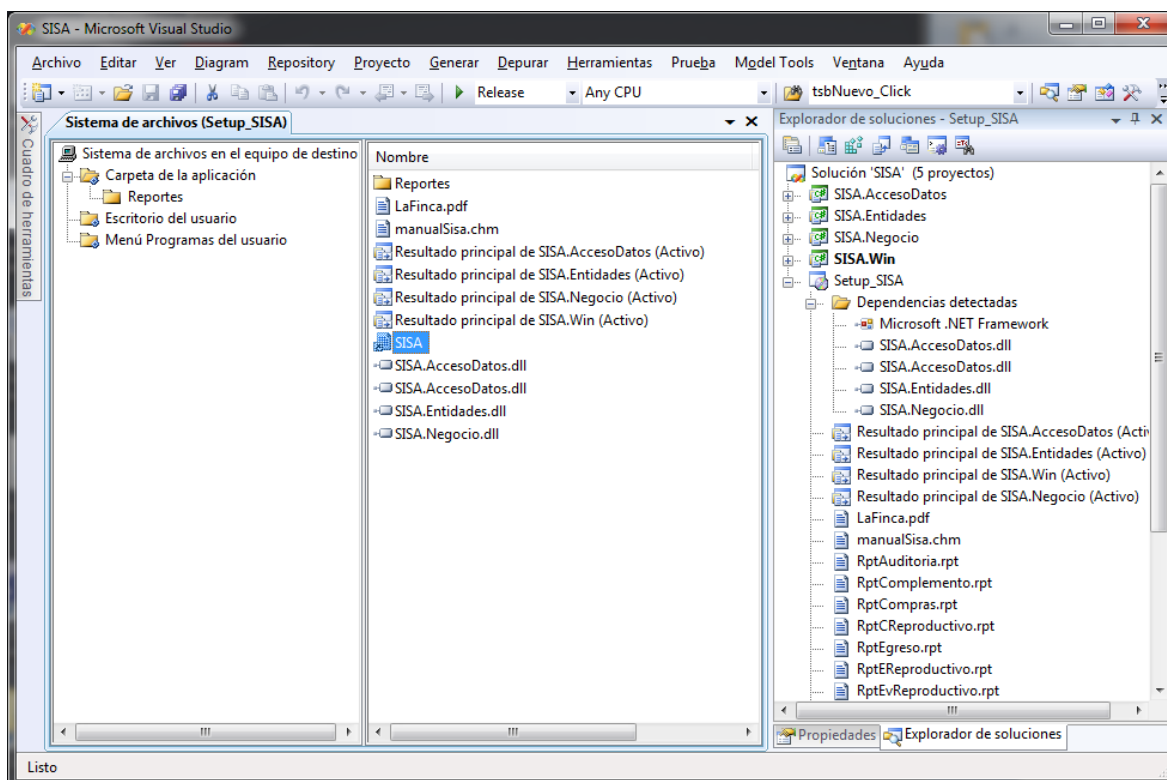


Fig. 250: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

A este archivo creado, se le va a ubicar dentro de la carpeta "Escritorio del usuario", que está ubicada en el lado derecho. Arrastrar el archivo a esa carpeta.

A este archivo se le puede cambiar la presentación, añadiéndole un ícono. Ubicado en este archivo oprimir F4, se despliega las propiedades del mismo, en lado derecho de "Icon", hacer clic en "Examinar", ubicar la imagen y añadirla. La secuencia se puede ver en el siguiente gráfico.

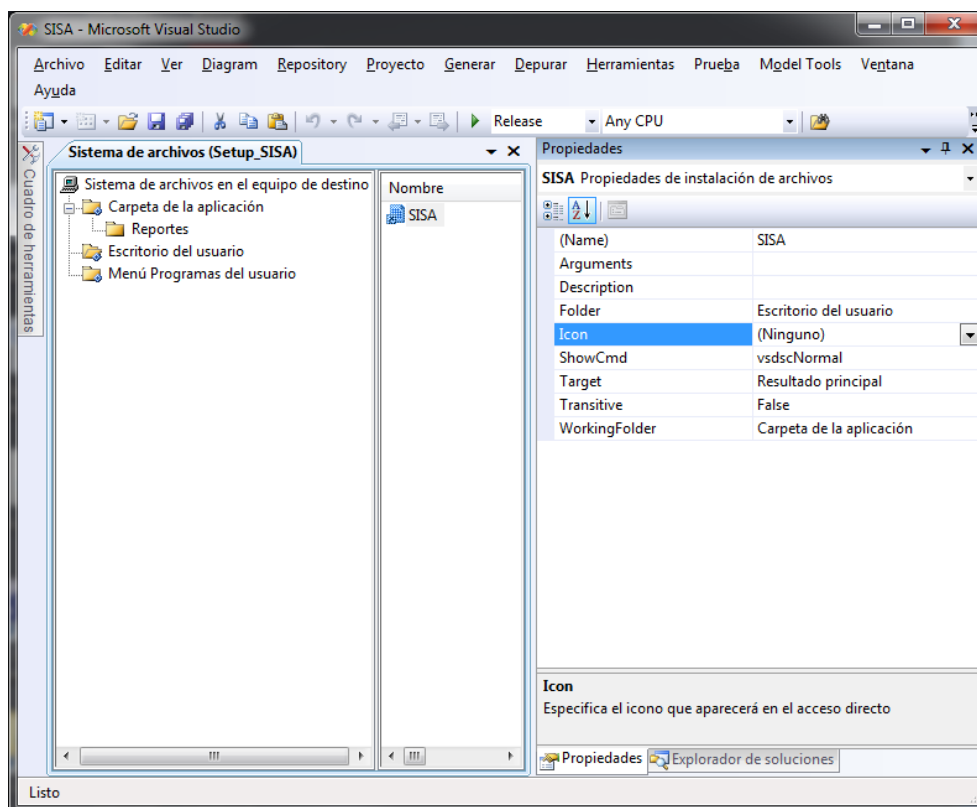


Fig. 251: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Hemos creado el acceso directo al programa, corresponde ahora describir como añadirlo al menú inicio de las máquinas donde se encuentre instalado.

Ubicarse en la carpeta "Menú Programas del usuario", que se encuentra ubicada en el lado izquierdo, "Agregar", "Carpeta", El resultado se ve en el siguiente gráfico.

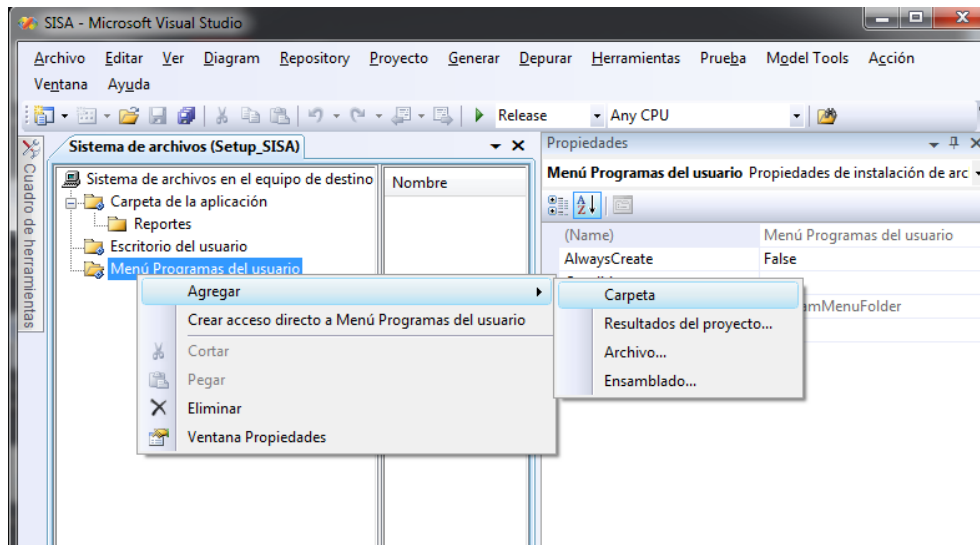


Fig. 252: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Se le puede cambiar el nombre, ahora tiene el nombre de "SISA". Luego se procede de idéntica forma, como para crear el acceso directo al escritorio.

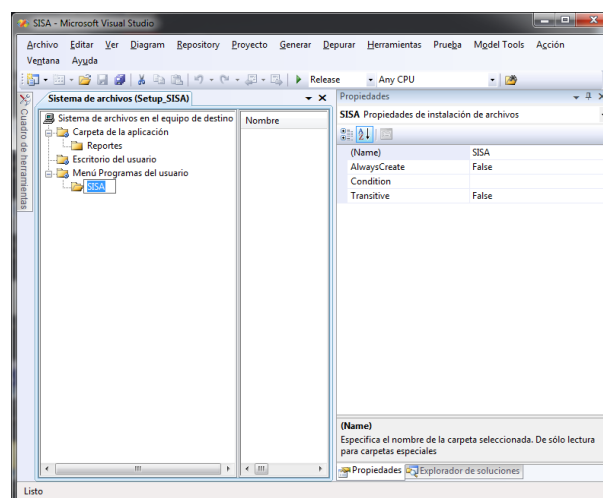


Fig. 253: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Realizado todo lo anterior, regresar al "Explorador de soluciones", ubicarse en "Setup_SISA" y hacer clic derecho, luego clic izquierdo en "Generar" y empieza a compilar, acabado este proceso y para determinar ciertas características del

instalador, ubicarse nuevamente en "Setup_SISA" hacer clic derecho, luego clic izquierdo en "Propiedades", apareciendo la siguiente pantalla.

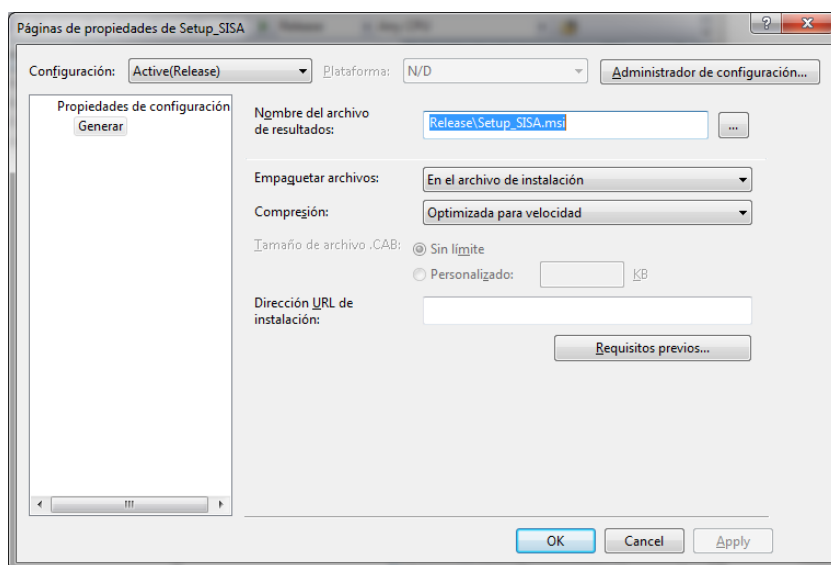


Fig. 254: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

En esta pantalla se define que archivos adicionales puede contener el instalador, por ejemplo el ".NET framework 3.5 ", en caso que la máquina donde se va a instalar el programa no tenga instalado este requisito de funcionamiento.

Haciendo clic en "Requisitos previos", aparece otra pantalla donde se puede señalar que archivos se adjuntan al instalador. La pantalla es la siguiente.

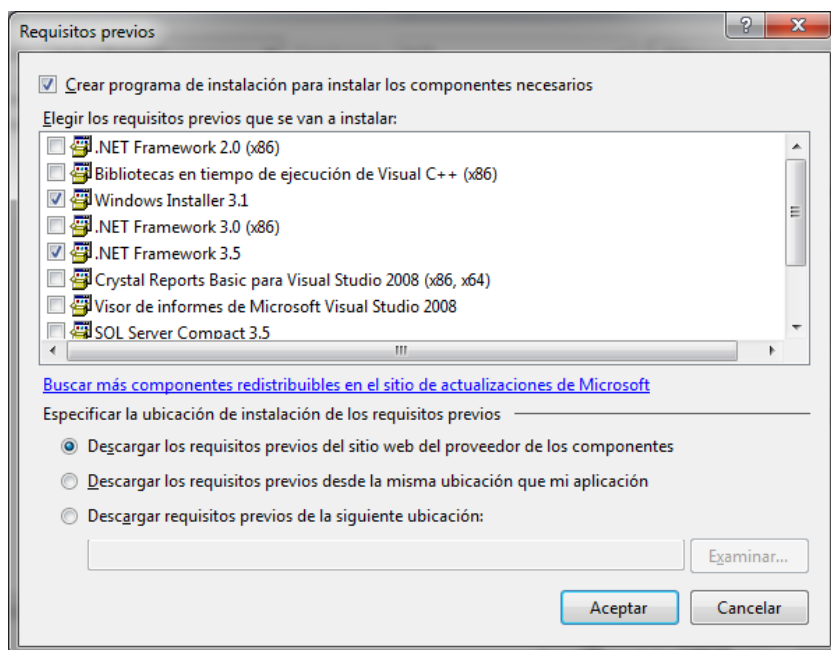


Fig. 255: Pantalla del proceso de creación del instalador del sistema "SISA".

Realizado por: Los Autores de la Tesis.

Hacer clic en aceptar, quedando completamente listo el instalador.

Manual de instalación del Sistema SISA.

Una vez creado el instalador del sistema SISA, corresponde describir la forma de instalarlo en cualquier máquina cliente.

A continuación se detalla este proceso.

Acceder a la dirección donde este el instalador, la carpeta que lo contiene, para este caso, está designada con el nombre de “Release”, al abrirla se muestra algunos archivos como se puede ver en el siguiente gráfico.

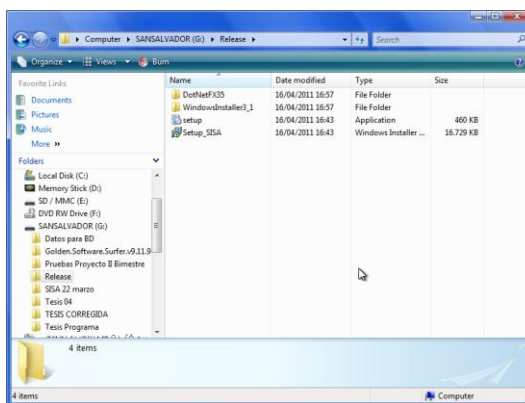
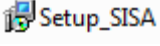
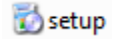


Fig. 256: Pantalla que muestra el contenido de la carpeta “Release”.

Realizado por: Los Autores de la Tesis

En esta carpeta hay dos íconos de instalación, setup y Setup_SISA.

Si se hace doble clic en el ícono Setup_SISA  , empieza la instalación sin requerir información en cuanto a los requisitos previos necesarios para que funcione el programa.

Cuando se hace doble clic sobre el ícono setup  , primero valida los requisitos que debe tener instalados la máquina para que funcione el programa SISA, como por ejemplo tener instalado el framework 3.5, si es que no está instalado, procede a instalarlo para luego empezar la instalación del programa o sistema SISA.

La primera pantalla de instalación es la siguiente:

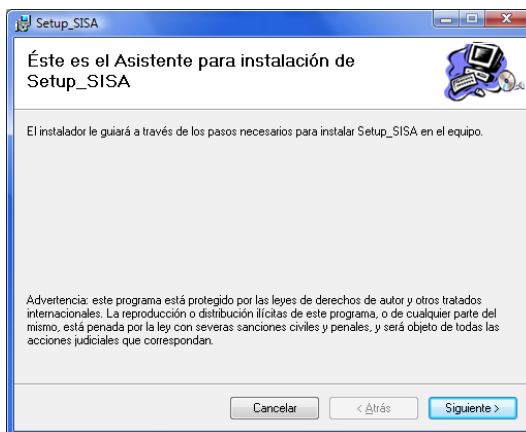


Fig. 257: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

Al hacer clic en “Siguiente” aparece la siguiente pantalla.



Fig. 258: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

Esta pantalla permite elegir y a la vez restringir el acceso al programa SISA. La opción “Para todos los usuarios” habilita a cualquier usuario de la máquina u ordenador tenerlo entre sus programas disponibles. En cambio la opción “Solo para este usuario” lo restringe al usuario que hace la instalación.

Luego de hacer la elección, hacer clic en “Siguiente”, aparece la siguiente pantalla.

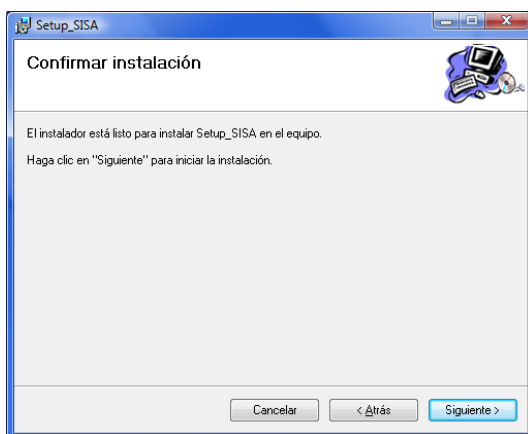


Fig. 259: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

Hacer clic en “Siguiente”.

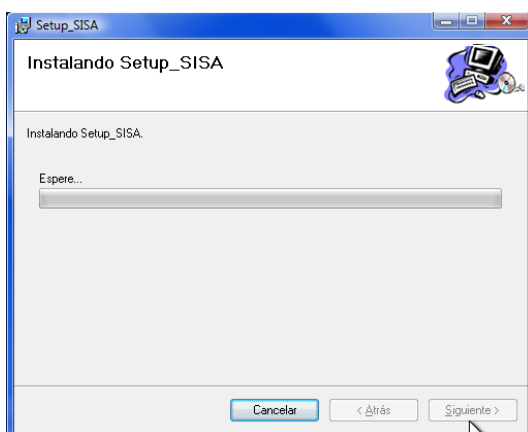


Fig. 260: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

Al terminar la instalación aparece la siguiente pantalla.

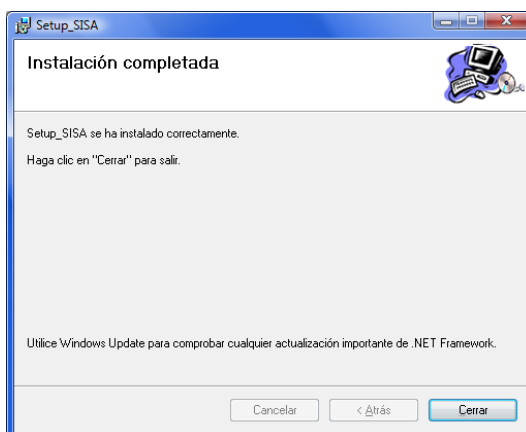


Fig. 261: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

Cerrar, y listo, quedando instalado el programa.

Antes de empezar a usarlo hay que realizar una tarea previa, que consiste en poner la dirección IP del servidor en un archivo que se creó al instalar el programa.

Este proceso se lo realiza de la siguiente manera.

Primero hay que ir hacia la carpeta "Program Files", dentro de esta carpeta se encuentra la carpeta "San Antonio", dentro de esta hay otra carpeta "Setup_SISA" y dentro de esta hay varios archivos, uno de ellos es el "SISA.Win.exe". En el siguiente gráfico se puede ver el contenido de la carpeta "Setup_SISA".

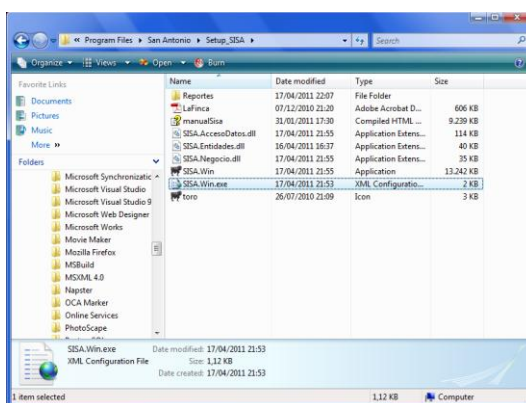
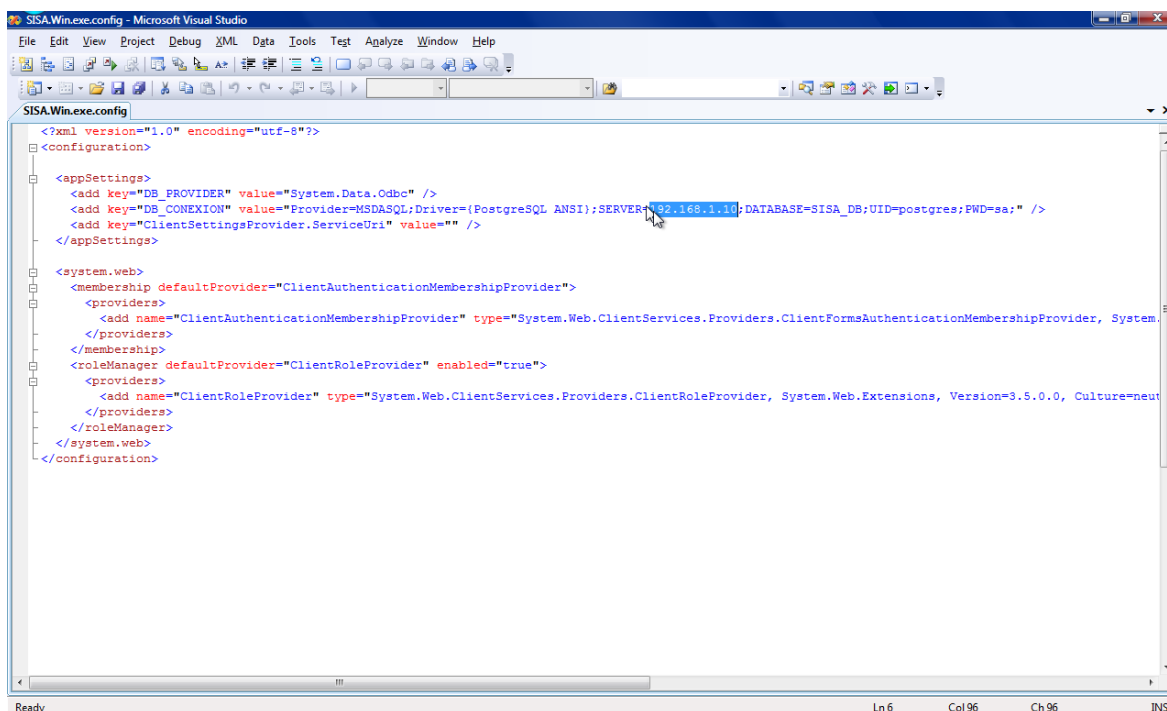


Fig. 262: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

Quitar este archivo de la carpeta, para poder modificarlo, se lo puede poner en el escritorio.

El archivo una vez abierto se puede ver en la siguiente figura.



```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="DB_PROVIDER" value="System.Data.Odbc" />
    <add key="DB_CONEXION" value="Provider=MSDASQL;Driver={PostgreSQL ANSI};SERVER=92.168.1.10;DATABASE=SISA_DB;UID=postgres;PWD=sa;" />
    <add key="ClientSettingsProvider.ServiceUri" value="" />
  </appSettings>
  <system.web>
    <membership defaultProvider="ClientAuthenticationMembershipProvider">
      <providers>
        <add name="ClientAuthenticationMembershipProvider" type="System.Web.ClientServices.Providers.ClientFormsAuthenticationMembershipProvider, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
      </providers>
    </membership>
    <roleManager defaultProvider="ClientRoleProvider" enabled="true">
      <providers>
        <add name="ClientRoleProvider" type="System.Web.ClientServices.Providers.ClientRoleProvider, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" />
      </providers>
    </roleManager>
  </system.web>
</configuration>
```

Fig. 263: Pantalla de instalación del sistema SISA.

Realizado por: Los Autores de la Tesis

En este archivo se debe poner la dirección IP de la máquina que es el servidor, en el gráfico anterior está resaltada dicha dirección.

Realizado esto, cierre el archivo y vuelva a ponerlo en la misma carpeta que lo contenía anteriormente. Ahora si ya está completamente listo el programa SISA.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES.

- Los conocimientos adquiridos en los años de estudio, no han sido suficientes para poder hacer el sistema informático, pero han servido como una buena guía. Razón por la cual se ha tenido que investigar sobre varios temas relacionados con el sistema.
- Las herramientas elegidas para desarrollar e implementar el sistema SISA, han sido fundamentales para obtener un producto correcto, fiable, seguro y amigable.
- El lenguaje de programación, C#, usado para codificar el sistema, por ser un lenguaje muy difundido en el mundo de la programación, ha permitido obtener mucha información, lo cual ha sido de gran ayuda.
- El uso de Postgres, para la Base de Datos, por ser un software libre, ha permitido usar sin restricciones la mayoría de sus características.
- La metodología en espiral, usada para el desarrollo del software ha permitido realizar cambios, en el sistema o en la base de datos, sin que el costo tiempo sea demasiado alto.
- La arquitectura de 3 capas ha permitido tener independencia, es decir que cada capa es un proceso separado, lo cual ha permitido reemplazar o modificar cada capa por separado sin que tenga que afectar a las otras capas.
- El haber implementado la tabla "Auditoria", le ha dado un agregado de seguridad al sistema, pues en ella se puede registrar los cambios que han tenido ciertas tablas que se consideran contienen datos más sensibles, además que usuario hizo los cambios y la fecha.

6.2 RECOMENDACIONES.

- Guardar toda la información entregada por los profesores, durante los años de estudio, ya que es muy útil al momento de desarrollar la tesis.
- Elegir el lenguaje de programación más adecuado para desarrollar un sistema, pues este influirá en gran medida en el éxito del mismo.
- Evaluar las ventajas y desventajas que tiene el Sistema de Gestión de Base de Datos, que se va a usar, cuando se propone implementar un sistema informático.
- Para evitar hacer cambios permanentes en los requerimientos del sistema, es necesario poner una fecha tope para realizarlos.
- El administrador del sistema debe tener un mínimo de conocimientos en la materia. Por lo menos debe saber instalar el sistema y hacer respaldos de la información.
- Instruir de manera adecuada y suficiente a todos los usuarios del sistema sobre el funcionamiento y buen uso del mismo.
- Este tipo de sistema es recomendado para el uso en pequeñas y medianas fincas ganaderas, aunque se podría aumentar e implementar otros módulos de modo que su uso sea más amplio, en esta área productiva.
- Tener siempre a mano el Manual del Usuario, para que sea una ayuda permanente para los usuarios, al menos hasta que se familiaricen con el funcionamiento del sistema SISA.
- Los usuarios del sistema deben tener cuidado en no olvidar sus contraseñas, o que alguien se las copie, ya que ellas definen el tipo de información a la cual accederán, de acuerdo a sus perfiles.
- Implementar una forma adecuada de control para el manejo de la información contenida en la Base de Datos, para saber quien cambio los datos y cuando lo hizo.

BIBLIOGRAFIA.

Libros

PRESSMAN, Roger S.: *Ingeniería del Software. Un enfoque Práctico*. Madrid: McGRAW-HILL, 2002.

TAYLOR, Allen G.: *Crystal Reports 10. For Dummies*. Indiana: Wiley Publishing, 2004.

DIETEL Harvey M. y DIETEL Paul J.: *Como programar C#*. México: Pearson, 2005.

RODRIGUEZ GÓMEZ-STERN Miguel y BESTEIRO GOROSTIZAGA Marco Antonio: *Desarrollo de Aplicaciones .Net con Visual C#*. Madrid: McGRAW-HILL, 2002.

SILBERSCHATZ Abraham, KORTH Henry y SUDARSHAN S.: *Fundamentos de Bases de Datos*. Madrid: McGRAW-HILL, 2002.

COX Brad J. y NOVOBILSKY Andrew J.: *Programación Orientada a Objetos*. Madrid: ADDISON WESLEY/DIAZ DE SANTOS, 1998.

FOWLER Martin con SCOTT Kendall: *UML GOTA A GOTA*. Madrid: ADDISON WESLEY, 2004.

Bibliografía Web

Tema: C#

- www.microsoft.com/express/Windows
- [http://msdn.microsoft.com/en-us/library/kx37x362\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/kx37x362(v=vs.80).aspx)

Tema: Postgres

- http://www.eqsoft.net/presentas/introduccion_a_postgresql.pdf
- <http://archives.postgresql.org/pgsql-es-ayuda/>
- <http://www.iessanvicente.com/colaboraciones/postgreSQL.pdf>
- <http://www3.uji.es/~mmarques/f47/apun/node74.html>

Tema: Arquitectura de base datos y de software.

- <http://www.mitecnologico.com/Main/ArquitecturaBaseDeDatos>
- <http://www.monografias.com/trabajos37/arquitectura-de-sistemas/arquitectura-de-sistemas.shtml>
- <http://www.josecuellar.net/arquitectura-de-software/el-diseno-de-la-arquitectura-de-un-sistema>
- <http://www.mitecnologico.com/Main/ArquitecturaAplicacionesWeb>

Tema: Métricas

- http://www.slideshare.net/galo_priva/mtricas-del-proceso-y-proyecto-procesos-de-ingeniera-de-software
- http://ficcte.unimoron.edu.ar/wicc/Trabajos/III%20-%20isbd/710-moa_wicc_2006.pdf
- <http://www.scribd.com/doc/408149/Metricas-basadas-en-el-Disenio-Orientado-a-Objetos-Ingenieria-de-Software>.

Tema: Diagrama de Componentes

- http://www.economicasunp.edu.ar/02Egrado/materias/trelew/236nálisis_sistemas%20I/info/uml2%20diagrama.pdf

Tema: Pruebas de software

- <http://www.ambyssoft.com/essays/flootSpanish.html>

Tema: especificaciones del sistema

- [sentencias SQL enhttp://boards4.melodysoft.com/2005AISI0202/re-c-especificacion-del-sistema-21.html](http://boards4.melodysoft.com/2005AISI0202/re-c-especificacion-del-sistema-21.html)

Tema: Metodología de desarrollo del software.

- <http://geeks.ms/blogs/rcorral/archive/2007/01/15/iquest-que-metodologiacute-a-de-desarrollo-elegir.aspx>
- <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

Tema: Generalidades.

- <http://www.unlz.edu.ar/biblioteca/tutores/tutor2c/dos.htm>
- <http://es.kioskea.net/forum/affich-95992-para-que-es-el-net-framework>
- [http://www.elguille.info/NET/dotnet/setupWin.htm//INSTALADOR SISTEMA.](http://www.elguille.info/NET/dotnet/setupWin.htm//INSTALADOR SISTEMA)

ANEXOS

Capa de Acceso a Datos
Carpeta "ADManager".
Clase "ClsGeneral.cs".

Código del método EjecutarConsulta

```
protected DataSet EjecutarConsulta(string sentencia, string nombreTabla, params
NameTypeValue[] parameters)
{
    DataSet datos = new DataSet();
    if (manejaConeccion)
        Connect();
    CreateCommand(sentencia, CommandType.Text, parameters);
    datos = ExecuteDataSet(nombreTabla);
    if (manejaConeccion)
        Disconnect();

    return datos;
}
```

Código del método EjecutarInstrucción

```
protected int EjecutarInstruccion(string sentencia, params NameTypeValue[]
parameters)
{
    int resultado;
    if (manejaConeccion)
        Connect();
    CreateCommand(sentencia, CommandType.Text, parameters);
    resultado = ExecuteCommand();
    if (manejaConeccion)
        Disconnect();

    return resultado;
}
```

Código del método EjecutarInsert

```
protected object EjecutarInsert(string sentencia, params NameTypeValue[]
parameters)
{
    int resultado;
    if (manejaConeccion)
        Connect();
```

```

        CreateCommand(sentencia, CommandType.Text, parameters);
        resultado = ExecuteScalar();
        if (manejaConeccion)
            Disconnect();

        return resultado;
    }

```

Capa de Acceso a Datos
Carpeta "ADInsumo".
Clase "ADInsumo.cs".

Código del método InsertarInsumo.

```

public long InsertarInsumo(EInsumo elInsumo)
{
    NameTypeValue[] parametros = new NameTypeValue[]
    {
        new NameTypeValue("insu_tipo", DbType.String, elInsumo.InsuTipo),
        new NameTypeValue("insu_nombre", DbType.String, elInsumo.
            InsuNombre),
        new NameTypeValue("insu_detalle", DbType.String, elInsumo.InsuDetalle)
    };
    StringBuilder str = new StringBuilder(2);
    str.Append(@"INSERT INTO INSUMO (insu_tipo, insu_nombre, insu_detalle)
VALUES (");
    str.Append(@" ?, ?, ?) RETURNING insu_codigo");

    return Convert.ToInt64(EjecutarInsert(str.ToString(), parametros));
}

```

Código del método ActualizarInsumo.

```

public int ActualizarInsumo(EInsumo elInsumo)
{
    NameTypeValue[] parametros = new NameTypeValue[]
    {
        new NameTypeValue("insu_tipo", DbType.String, elInsumo.InsuTipo),

```



```

new NameTypeValue("insu_nombre", DbType.String, elInsumo.InsuNombre),
new NameTypeValue("insu_detalle", DbType.String, elInsumo.InsuDetalle),
new NameTypeValue("insu_codigo", DbType.Int64, elInsumo.InsuCodigo)
};
    StringBuilder str = new StringBuilder(4);
    str.Append(@"UPDATE INSUMO SET insu_tipo = ?,");
    str.Append(@" insu_nombre = ?,");
    str.Append(@" insu_detalle = ?");
    str.Append(@" WHERE insu_codigo = ?");

    return EjecutarInstruccion(str.ToString(), parametros);
}

```

Código del método SeleccionarInsumo

```

public DataSet SeleccionarInsumo(long insuCodigo)
{
    NameTypeValue[] parametros = new NameTypeValue[] {
        new NameTypeValue("insu_codigo", DbType.Int64, insuCodigo)
    };
    DataSet dsDatos = new DataSet();
    StringBuilder str = new StringBuilder(2);
    str.Append(@"SELECT insu_codigo, insu_tipo, insu_nombre, insu_detalle
FROM INSUMO ");
    str.Append(@" WHERE insu_codigo = ?");

    return EjecutarConsulta(str.ToString(), "INSUMO", parametros);
}

```

Código del método SeleccionarInsumoTodos.

```

public DataSet SeleccionarInsumoTodos()
{
    DataSet dsDatos = new DataSet();
    StringBuilder str = new StringBuilder(2);
    str.Append(@"SELECT insu_codigo, insu_tipo, insu_nombre, insu_detalle
FROM INSUMO ");
    str.Append(@" WHERE insu_codigo > 0 ORDER BY insu_tipo DESC,
insu_nombre ASC ");
}

```

```

    return EjecutarConsulta(str.ToString(), "INSUMO", null);
}

```

Código del método SeleccionarInsumoFiltro.

```

public DataSet SeleccionarInsumoFiltro(DTOInsumo dtoInsumo)
{
    NameTypeValue[] parametros = new NameTypeValue[] {
        new NameTypeValue("insu_codigo", DbType.String,
dtoInsumo.InsuTipo)
    };
    DataSet dsDatos = new DataSet();
    StringBuilder str = new StringBuilder(6);
    str.Append(@"SELECT insu_codigo, insu_tipo, insu_nombre, insu_detalle
FROM INSUMO WHERE");
    if (dtoInsumo.InsuTipo == "0")
        str.Append(@" insu_tipo > ?");
    else
        str.Append(@" insu_tipo = ?");

    str.Append(@" ORDER BY insu_tipo DESC, insu_nombre ASC ");

    return EjecutarConsulta(str.ToString(), "INSUMO", parametros);
}

```

Código del método EliminarInsumo.

```

public int EliminarInsumo(long insuCodigo)
{
    NameTypeValue[] parametros = new NameTypeValue[] {
        new NameTypeValue("insu_codigo", DbType.Int64, insuCodigo)
    };
    StringBuilder str = new StringBuilder(2);
    str.Append(@"DELETE FROM INSUMO WHERE insu_codigo = ?");

    return EjecutarInstruccion(str.ToString(), parametros);
}

```

Capa de Interfaces
Carpeta "Reportes".
Clase " FrmReporteERreproductivo.cs".

Código del método AdicionarParametros.

```
protected override void AdicionarParametros()
{
    ParameterValues prFechalnicio = new ParameterValues();
    ParameterValues prFechaFin = new ParameterValues();
    ParameterValues prNombreEvento = new ParameterValues();
    ParameterValues prTipoFecha = new ParameterValues();

    ParameterDiscreteValue pfDFechalnicio = new ParameterDiscreteValue();
    pfDFechalnicio.Value = dtpInicio.Value.ToShortDateString();
    prFechalnicio.Add(pfDFechalnicio);

    ParameterDiscreteValue pfDFechaFin = new ParameterDiscreteValue();
    pfDFechaFin.Value = dtpFin.Value.ToShortDateString();
    prFechaFin.Add(pfDFechaFin);

    ParameterDiscreteValue pfDNombreEvento = new
ParameterDiscreteValue();
    pfDNombreEvento.Value = cmbEvento.Text;
    prNombreEvento.Add(pfDNombreEvento);

    ParameterDiscreteValue pfDTipoFecha = new ParameterDiscreteValue();
    pfDTipoFecha.Value = (chkPorFechalnicio.Checked?"Inicio":"Fin");
    prTipoFecha.Add(pfDTipoFecha);

    crdComponente.DataDefinition.ParameterFields["Fechalnicio"].ApplyCurrentValues(prFechalnicio);

    crdComponente.DataDefinition.ParameterFields["FechaFin"].ApplyCurrentValues(prFechaFin);

    crdComponente.DataDefinition.ParameterFields["EventoNombre"].ApplyCurrentValues(prNombreEvento);

    crdComponente.DataDefinition.ParameterFields["TipoFecha"].ApplyCurrentValues(prTipoFecha);
}
```