



**UNIDAD ACADÉMICA DE CIENCIAS DE LA INGENIERIA Y
APLICADAS**

INGENIERÍA EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES

TESIS DE GRADO

**“IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL
CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN
VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY
ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”**

**TESIS DE GRADO, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN INFORMÁTICA Y SISTEMAS COMPUTACIONALES**

AUTORES:

**MÓNICA FERNANDA BRASALES PANCHI
LORENA ELIZABETH OSORIO YUGSE**

DIRECTOR:

ING. JUAN CARLOS RODRIGUEZ

AUTORÍA

Las ideas, opiniones y comentarios que se muestran en este documento son de exclusiva responsabilidad de los autores, Sra. Mónica Fernanda Brasales Panchi con C.I. 050249432-1 y Srta. Lorena Elizabeth Osorio Yugse con C.I. 050297871-1, así como también declaramos que la programación y el diseño de ésta tesis es totalmente original y auténtica.



Mónica Fernanda Brasales Panchi



Lorena Elizabeth Osorio Yugse



AVAL DEL DIRECTOR DE TESIS

Debo certificar que cumpliendo con lo estipulado en el reglamento de la Universidad Técnica de Cotopaxi que el tema de tesis titulado “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”, propuesto y desarrollado por las Egresadas Brasales Panchi Mónica Fernanda y Osorio Yugse Lorena Elizabeth ha concluido de acuerdo a los planteamientos formulados en el plan de tesis, por lo que apruebo para que sean sometidos a la evaluación del Tribunal de Validación de Tesis que el Honorable Consejo Académico de la Unidad Académica de Ciencias de la Ingeniería y Aplicadas de la Universidad Técnica de Cotopaxi designe, para su estudio y calificación.

Latacunga, Marzo del 2010.


Juan Carlos Rodríguez
Director de Tesis

Yo, GONZALO TOAPANTA YANCHATIPÁN, portador de la cédula # 050153264-2, en calidad de representante del Área de Producción de la “ASOCIACIÓN VOLCANO FLOWERS” que es parte de la ASOCIACIÓN CAMINO A LA ESPERANZA, tengo a bien certificar que las Egresadas Brasales Panchi Mónica Fernanda, con C.I 050249432-1 y Osorio Yugse Lorena Elizabeth con C.I 050297871-1, han finalizado el proyecto de tesis con el tema “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”, el mismo que ha sido entregado, probado en su funcionamiento y está de acuerdo con las especificaciones técnicas, administrativas y de gestión solicitadas.

Motivo por el cual, faculto a las interesadas hacer uso del presente Certificado, como a bien estimaren conveniente.

Latacunga, Noviembre de 2009.


RUC 0591700782001
GONZALO TOAPANTA YANCHATIPÁN

Yo, Lic. Ana Patricia Gómez Viera, portadora de la Cédula de Ciudadanía No. 050233926-0 en calidad de Supervisora en cuanto se refiere a Inglés, tengo a bien CERTIFICAR: que las egresadas Brasales Panchi Mónica Fernanda, con C.I 050249432-1 y Osorio Yugse Lorena Elizabeth con C.I 050297871-1, han realizado la correcta estructuración del Abstract de la Tesis de Grado con el Tema: “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”, por lo que doy fe del presente trabajo.

Por tal motivo faculto a las interesadas hacer uso del presente certificado como a bien lo consideren.

Latacunga, Marzo del 2010.

Lic. Patricia Gómez Viera
LICENCIADA EN INGLÉS

Nuestro eterno agradecimiento a la Universidad Técnica de Cotopaxi, a la Unidad Académica de Ciencias de la Ingeniería y Aplicadas y a través de ella a sus Autoridades y Docentes, por habernos formado como profesionales al servicio de la sociedad, así como también a la Asociación Volcano Flowers y a su representante en el Área de de Producción el Sr. Gonzalo Toapanta, por habernos dado su apoyo desde el inicio del proyecto y habernos motivado constantemente, ya que gracias a él se pudo concretar nuestro proyecto en la plantación.

Con profunda gratitud a nuestro Director de Tesis Ing. Juan Carlos Rodríguez, por habernos brindado la orientación necesaria en el desarrollo de nuestra investigación, y a todos los que colaboraron con su aporte académico para cumplir con nuestro objetivo.

Queremos expresar finalmente nuestro eterno y sentido agradecimiento a Mauro, a las familias, amigos, amigas y a todas las personas que de una u otra manera nos brindaron apoyo y nos acogieron durante toda la preparación profesional.

A mi padre Dios, por haberme dado la suficiente entereza para mantenerme firme de alcanzar esta meta, esta alegría, que si pudiera hacerla material, con seguridad te la entregaría. A mi mami Elsa, que serás siempre mi inspiración para alcanzar mis metas, por enseñarme que todo se aprende y que todo esfuerzo es al final recompensa. A mi papi Pedro, por haberme dado los más grandes consejos, por haber hecho de mí una persona positiva y paciente, para no decaer en el camino hacia mi tan anhelada meta. A mis Hermanos Richard y Cristian porque mi felicidad siempre fue de los dos, el apoyo constante sin saber de lo que se trataba este trabajo, las palabras de aliento que recibí y el tiempo que verdaderamente debí dedicarles a ellos y que sin embargo no pude compartir, hoy se encuentra plasmado en este escrito que de corazón se los dedico. A una persona muy especial en mi vida que fue el pilar fundamental en los ánimos y desarrollo de esto, gracias por estar siempre conmigo, mi querida Tía Rita.

Lorena Elizabeth Osorio Yugse.

El presente trabajo esta dedicado a ti mi DIOS por haberme regalado el privilegio de existir, por enseñarme el camino correcto de la vida, guiándome y fortaleciéndome día a día. A mis Padres, Margarita y Alfonso por su apoyo todos estos años, por su infinito amor, comprensión y por ayudarme a que este momento llegara. A mis Hermanas, Paulina y Patricia por creer y confiar siempre en mí, apoyándome en todas las decisiones que he tomado en la vida. A mi querida Hija Melany y Esposo Jorge, que tuvieron que soportar mis ausencias, por su amor, paciencia y comprensión, sin lo que hubiese sido imposible llegar alcanzar esta meta. A Lorena amiga y compañera por el apoyo y motivación, por su entusiasmo y empeño para lograr este objetivo. A mis maestros, por sus consejos y por compartir desinteresadamente sus amplios conocimientos y experiencia.

Monica Fernanda Brasales Panfili

CONTENIDO

Portada.....	i
Autoría.....	ii
Aval Director de Tesis.....	iii
Certificado de Implementación.....	iv
Certificación Abstract.....	v
Agradecimiento.....	vi
Dedicatoria.....	vii
Índice General.....	ix
Resumen.....	xvi
Abstract.....	xvii
Introducción.....	xix

CAPÍTULO I

CREACIÓN DE UN SISTEMA INFORMÁTICO BAJO LA PLATAFORMA .NET, CON ALMACENAMIENTO DE DATOS EN SQL SERVER Y DE ARQUITECTURA CLIENTE SERVIDOR.

1. ANTECEDENTES.....	1
1.1 SISTEMA INFORMÁTICO.....	3
1.1.1 DEFINICIÓN.....	3
1.1.2 COMPONENTES Y FUNCIONAMIENTOGENERAL DE UN SISTEMA INFORMÁTICO.....	4
1.1.3 EVOLUCIÓN DE LOS SISTEMAS INFORMÁTICOS.....	6
1.1.4 IMPORTANCIA.....	7
1.2 BASE DE DATOS.....	8

1.2.2.2 CONSISTENCIA DE DATOS.....	11
1.2.2.3 COMPARTICIÓN DE DATOS.....	11
1.2.2.4 MANTENIMIENTO DE ESTÁNDARES.....	12
1.2.2.5 MEJORA EN LA INTEGRIDAD DE DATOS.....	12
1.2.2.6 MEJORA EN LA SEGURIDAD.....	12
1.2.2.7 MEJORA EN LA ACCESIBILIDAD A LOS DATOS.....	12
1.2.2.8 MEJORA EN LA PRODUCTIVIDAD.....	13
1.2.2.9 MEJORA EN EL MANTENIMIENTO.....	13
1.2.2.10 AUMENTO DE LA CONCURRENCIA.....	13
1.2.2.11 MEJORA EN LOS SERVICIOS DE COPIAS DE SEGURIDAD.....	14
1.2.3 DESVENTAJAS DE LAS BASES DE DATOS.....	14
1.2.3.1 COMPLEJIDAD.....	14
1.2.3.2 COSTE DEL EQUIPAMIENTO ADICIONAL.....	14
1.2.3.3 VULNERABLE A LOS FALLOS.....	15
1.2.4 TIPOS DE CAMPOS.....	15
1.2.5 MODELOS DE BASES DE DATOS.....	15
1.2.5.1 BASES DE DATOS JERÁRQUICAS.....	16
1.2.5.2 BASE DE DATOS DE RED.....	16
1.2.5.3 BASE DE DATOS RELACIONAL.....	17
1.2.5.4 BASES DE DATOS ORIENTADAS A OBJETOS.....	18
1.2.5.5 BASE DE DATOS DEDUCTIVAS.....	19
1.2.6 SQL.....	19
1.2.6.1 DEFINICIÓN.....	19
1.2.6.2 CARACTERÍSTICAS DE SQL SERVER.....	20
1.3 LENGUAJES DE PROGRAMACIÓN.....	22
1.3.1 VISUAL STUDIO .NET.....	23
1.4 ARQUITECTURA TRES CAPAS EN TECNOLOGÍA CLIENTE/SERVIDOR.....	24
1.4.1 CAPAS.....	25
1.4.1.1 CAPA DE PRESENTACIÓN.....	25
1.4.1.2 CAPA DE NEGOCIOS.....	26
1.4.1.3 CAPA DE DATOS.....	27

CAPITULO II

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

2. INFORMACIÓN DE LA ASOCIACIÓN VOLCANO FLOWERS.....	31
2.1 PATUTÁN.....	31
2.1.1 INFORMACIÓN GENERAL DE PATUTÁN.....	32
2.1.2 ASOCIACIÓN COMUNAL CAMINO A LA ESPERANZA.....	34
2.1.3 CHRISTIAN CHILDREN’S FUND, INC.....	35
2.1.3.1 CCF DEL ECUADOR.....	35
2.1.3.2 CCF EN EL ECUADOR.....	36
2.1.3.3 ¿DÓNDE OPERA CCF?.....	36
2.1.3.4 ¿CÓMO TRABAJA CCF?	37
2.1.3.4.1 AUTOGESTIÓN.....	37
2.1.3.4.2 CAPACITACIÓN Y ACOMPAÑAMIENTO.....	37
2.1.3.4.3 DESARROLLO DEL NIÑO	37
2.1.3.5 ¿QUÉ HACE CCF?.....	37
2.1.3.5.1 SALUD.....	38
2.1.3.5.2 EDUCACIÓN.....	38
2.1.4 ASOCIACIÓN VOLCANO FLOWERS.....	38
2.2 POBLACIÓN Y MUESTRA.....	40
2.2.1 POBLACIÓN.....	40
2.2.2 MUESTRA.....	40
2.3 DISEÑO METODOLÓGICO.....	40
2.3.1 TIPO DE INVESTIGACIÓN.....	40
2.3.2 METODOLOGÍA.....	41
2.3.2.1 MÉTODOS.....	42

2.3.2.2 TÉCNICAS.....	43
2.3.2.2.1 OBSERVACIÓN.....	43
2.3.2.2.2 LA ENTREVISTA.....	43
2.3.2.2.3 ENCUESTA.....	43
2.3.2.3 INSTRUMENTOS.....	43
2.3.2.3.1 GUÍAS DE ENTREVISTA.....	43
2.3.2.3.2 FICHAS DE OBSERVACIÓN.....	43
2.3.2.3.3 CUESTIONARIO.....	43
2.4 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS DE LA ENTREVISTA APLICADA A LAS AUTORIDADES DE LA ASOCIACIÓN VOLCANO FLOWERS.....	44
2.5 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS DE LA ENCUESTA APLICADA A LOS TRABAJADORES, PROVEEDORES Y ASOCIADOS DE VOLCANO FLOWERS.....	45
2.6 VERIFICACIÓN DE HIPÓTESIS.....	50
2.6.1 HIPÓTESIS.....	50
2.6.1.1 VERIFICACIÓN.....	50
2.7 RECURSOS NECESARIOS (humanos, materiales, técnicos, presupuesto).....	51
2.7.1 HUMANOS.....	51
2.7.2 MATERIALES: DESGLOSE DE MATERIALES.....	51
2.7.3 TÉCNICOS: DESGLOSE DEL RECURSO TÉCNICO.....	52
2.7.4 PRESUPUESTO.....	52
2.7.4.1 COSTOS DIRECTOS: DETALLE DE COSTOS.....	52
2.7.4.2 COSTOS INDIRECTOS: DETALLE DE COSTOS.....	52

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL
SISTEMA VOLCANO CONTROL, CODIFICADO EN VISUAL
BASIC .NET 2005 Y BASE DE DATOS EN SQL SERVER 2005
(APLICACIÓN DE ESCRITORIO).

3. Tema: “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”.....	53
3.1 PRESENTACIÓN.....	53
3.2 JUSTIFICACIÓN DE LA PROPUESTA.....	54
3.3 OBJETIVOS.....	56
3.3.1 OBJETIVO GENERAL.....	56
3.3.2 OBJETIVOS ESPECÍFICOS.....	56
3.4 FACTIBILIDAD DE IMPLEMENTAR EL SISTEMA INFORMÁTICO PLANTEADO.....	57
3.5 DESARROLLO DE LA PROPUESTA.....	58
3.6 REQUERIMIENTOS DE LA ASOCIACIÓN VOLCANO FLOWERS PARA EFECTUAR EL CONTROL SOBRE LA PRODUCCIÓN DE CLAVELES.....	59
3.7 ESPECIFICACIONES DEL SISTEMA VOLCANO CONTROL.....	61
3.8 ANÁLISIS Y DISEÑO DEL SISTEMA VOLCANO CONTROL.....	63
3.8.1 MODELADO Y DISEÑO DEL SISTEMA.....	65
3.8.1.1 DIAGRAMA DE FLUJO DE DATOS.....	65
3.8.1.2 DIAGRAMA DE CASOS DE USO.....	66
3.8.1.3 DIAGRAMA DE SECUENCIA.....	66
3.8.1.4 DIAGRAMA DE CLASES.....	67
3.8.1.5 MODELO ORIENTADO A OBJETOS.....	68

3.8.2 IMPLEMENTACION DEL SISTEMA VOLCANO CONTROL EN LA ASOCIACIÓN VOLCANO FLOWERS.....	69
3.8.3 PRUEBAS REALIZADAS AL SISTEMA VOLCANO CONTROL.....	70
3.8.4 MANTENIMIENTO DEL SISTEMA.....	70
3.9 ASPECTOS QUE DIFICULTARON EL DESARROLLO DEL SISTEMA.....	71
3.10 CRONOGRAMA DE ACTIVIDADES DE LA TESIS.....	71
3.11 CONCLUSIONES.....	72
3.12 RECOMENDACIONES.....	73
3.13 REFERENCIAS BIBLIOGRÁFICAS.....	74
3.14 GLOSARIO DE ABREVIACIONES.....	76
3.15 GLOSARIO DE TÉRMINOS.....	78

ÍNDICE DE ANEXOS

ANEXO DE GRÁFICOS

GRAFICO N° 1.1: ESTRUCTURA DE UN SISTEMA INFORMÁTICO.....	85
GRAFICO N° 1.2: COMPONENTES DE MICROSOFT .NET.....	85
GRAFICO N° 1.3: ARQUITECTURA TRES CAPAS EN TECNOLOGÍA CLIENTE/SERVIDOR.....	86
GRÁFICO N° 1.4: CICLO DE VIDA EN CASCADA.....	86
GRÁFICO 2.1. POBLACIÓN Y MUESTRA.....	87
GRAFICO 2.2. PREGUNTA 1.....	87
GRAFICO 2.3 PREGUNTA 2.....	88
GRAFICO 2.4 PREGUNTA 3.....	88
GRAFICO 2.5 PREGUNTA 4.....	89
GRAFICO 2.6 PREGUNTA 5.....	89
GRAFICO 2.7 PREGUNTA 6.....	90
GRAFICO 2.8 PREGUNTA 7.....	90
GRAFICO 2.9 PREGUNTA 8.....	91
GRAFICO 2.10 PREGUNTA 9.....	91

GRÁFICO 3.3 DIAGRAMA DE SECUENCIA.....	95
GRÁFICO 3.4 DIAGRAMA DE CLASES.....	96
GRÁFICO 3.5 MODELO CONCEPTUAL Y FISICO DE DATOS.....	97

ANEXO DE TABLAS

Tabla 2.1. Listado de Asociados Fundadores.....	101
Tabla 2.2. Población y Muestra.....	101
Tabla 2.3. Pregunta 1.....	101
Tabla 2.4. Pregunta 2.....	102
Tabla 2.5. Pregunta 3.....	102
Tabla 2.6. Pregunta 4.....	102
Tabla 2.7. Pregunta 5.....	102
Tabla 2.8. Pregunta 6.....	102
Tabla 2.9. Pregunta 7.....	102
Tabla 2.10. Pregunta 8.....	103
Tabla 2.11. Pregunta 9.....	103
Tabla 2.12. Pregunta 10.....	103
Tabla 2.13. MATERIALES.....	104
Tabla 2.14. DESGLOSE DEL RECURSO TÉCNICO.....	104
Tabla 2.15. COSTOS DIRECTOS: DETALLE DE COSTOS.....	105
Tabla 2.16. COSTOS INDIRECTOS: DETALLE DE COSTOS.....	105

OTROS ANEXOS

FORMATO DE ENCUESTA.....	106
BASE DE DATOS.....	108
INTERFACES DEL SISTEMA.....	196
CRONOGRAMA DE ACTIVIDADES.....	244

RESUMEN

La tesis “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”, ha sido realizada en base a los requerimientos planteados por parte de las personas que laboran en la Asociación Volcano Flowers y que son las responsables de la producción de claveles y de su debido control, cabe mencionar que dichos requerimientos fueron verificados mediante la aplicación de entrevistas y encuestas, esta investigación tiene como finalidad aportar de forma significativa al tratamiento de la información en el control de la producción de claveles para mejorar la productividad de dicha Asociación.

El software que actualmente ya se encuentra implementado, presenta una interfaz desarrollada con Microsoft Visual Studio .Net 2005 y en cuanto al almacenamiento de datos que se prevé que se generaran con el transcurso del tiempo, se ha empleado una base de datos diseñada en SQL Server 2005, pues son herramientas que proporcionan múltiples facilidades.

El Software cuenta con seguridades que garantizan el buen manejo y rendimiento del mismo, la forma de acceso es completamente segura ya que solo el personal autorizado puede hacer uso de este, es así como el Departamento de Control de

responsables serán consideradas como usuarios, los cuales tendrán privilegios que se estimen convenientes de acuerdo a las necesidades que se presenten.

El sistema implementado, permite que el usuario tenga absoluta confianza, puesto que la información almacenada en el mismo será confidencial y de uso exclusivo de la Asociación al igual que toda la información enviada y recibida. El proceso de producción será controlado por el sistema en todas sus etapas, es decir desde el momento de siembra hasta la exportación de los claveles.

VOLCANO CONTROL, es un sistema de interfaz sencilla fácil de manejar, cubre los requerimientos de la AVF y sobre todo es de alto rendimiento en su funcionamiento.

The thesis “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”. The present investigation has been developed on the requirements posed by people who work in the Volcano Flowers Association and are responsible for the production of carnations and its proper control, it is noteworthy that these requirements were verified through the application of interviews and surveys, this investigation has to contribute to provide a significant the processing of information in control the production of carnations to improve productivity of the Association.

The software is already deployed today, features an interface developed with Microsoft Visual Studio.Net 2005 and as for the storage of data that is predicted to be generated in the course of time, has used a database designed in SQL Server 2005, they are tools that provide multiple facilities.

The Software offers assurances to guarantee the good handling and performance and the access method is completely safe because only authorized personal can use this, this is how the Production Control Department of the Association will act as Volcano Flowers Administrator and who shall have full control, while each attached plantation belonging to different partners with their respective officers shall be considered as users who have privileges that may be appropriate according to the needs that arise.

The implemented system allows the user to have absolute confidence, since the information sent therein will be confidential and exclusive use of the Association as well as all information sent and received. The production process is controlled by the system at all stages, from the time of planting to the export of carnations.

VOLCANO MONITORING, is a simple system easy to use interface, it covers the requirements of the AVF and is especially high performance in its operation

A través de los años, el hombre ha utilizado diferentes formas para organizar la información que se genera dentro de las Empresas para las cuales prestan sus servicios es así que, se han ido tomando en consideración varias de las alternativas que el mismo hombre ha creado y ofrece para aminorar el arduo trabajo de escribir en forma manual y efectuar cálculos en forma exacta con resultados a tiempo.

La necesidad de mantener información actualizada de los procesos generados en una empresa y de manejar grandes cantidades de información de manera organizada, han hecho que el hombre en base a sus necesidades cree nuevas herramientas para poder controlar el flujo de información y desempeñarse de manera más eficiente, inclinándose totalmente al desarrollo de Tecnologías de la Información que en la actualidad se ha incrementado exponencialmente, pero no se puede afirmar que estas se hayan manifestado de igual manera alrededor el mundo. Los recursos de Tecnologías y Sistemas de Información en el Ecuador son cada vez más utilizados, razón por la cual los directivos de Empresas han optado por tomar medidas que permitan gestionarlas eficientemente y de esta manera mejorar el control y manejo de sus negocios, si bien es cierto, se ha logrado adquirir e Implementar Software en la mayoría de Empresas, pero no existe todavía usuarios suficientemente preparados para obtener su máximo potencial.

La importancia de esta Tesis, radica en que se muestra la forma en la que se llevará el control de la Producción de Claveles con la Implementación de un Software creado en base a las necesidades que tiene la Asociación Volcano Flowers, para mejorar el entorno laboral, tecnológico y económico de la Empresa.

Además de lo expuesto, mediante la recopilación teórica plasmada en el Capítulo I se pretende cubrir toda la terminología empleada en el desarrollo del Sistema Volcano Control, así como, también, poner a consideración del lector mediante el

sistema desarrollado e implementado en la AVF, así como también la metodología empleada, finalmente un Capítulo III en el que se detalla la forma en que fue desarrollado el Sistema y además las conclusiones y recomendaciones.

Cabe mencionar que con el Software Implementado se prevé ahorros significativos de tiempo, cálculos exactos operaciones en tiempo real, y un almacenamiento seguro y confiable de datos.

El Sistema Volcano Control automatizará los procesos dentro de la Florícola, además tendrá un aporte significativo en la toma de decisiones para crear soluciones competitivas y mejorar la calidad de producción.

En definitiva el Software desarrollado, pretende cubrir tres aspectos importantes ser un Sistema transaccional, de apoyo en la toma de decisiones y a la vez estratégico

CAPÍTULO I

CREACIÓN DE UN SISTEMA INFORMÁTICO BAJO LA PLATAFORMA .NET, CON ALMACENAMIENTO DE DATOS EN SQL SERVER Y DE ARQUITECTURA CLIENTE SERVIDOR.

1. ANTECEDENTES

En todo proceso de investigación, uno de los elementos que direcciona el camino a seguir es la sustentación teórica, ya que en base a éste se inicia un trabajo científico adentrándose de manera profunda y real en la actividad científica con la finalidad de encontrar el sentido de la investigación que se desea hacer.

Uno de los propósitos básicos de ésta investigación es crear un estudio único dentro de la Asociación Volcano Flowers ubicada en la Parroquia Eloy Alfaro del Cantón Latacunga, es necesario enfatizar que dentro de la Asociación en mención anteriormente no se ha dado ningún tipo de investigación, por lo que el proyecto planteado e implementado se considera como el primer aporte informático con el cual se inicia la automatización de los procesos que generan información en la florícola.

Debido a que la información de los procesos de producción, van acrecentando el volumen de documentación y que las exigencias de cada empresa en la actualidad es la migración hacia los Sistemas Informáticos, la Asociación Volcano Flowers también adopta la tendencia de automatizar sus procesos para mantener una organización adecuada en los datos que se manejan, es así que el desarrollar un Software de calidad y que cubra todos los requerimientos de la Empresa implica

En la actualidad existen varios programas bajo los cuales se puede codificar un Software, unos de codificación abierta (Open Source) y otros que se manejan bajo licencia, tal es el caso del lenguaje empleado en el desarrollo del Sistema creado para la Asociación denominado Volcano Control, el cual ha sido programado bajo la plataforma .NET versión 2005 específicamente en Visual Basic.NET

Según investigaciones anteriores Visual Basic.NET es una gran alternativa para construir aplicaciones robustas y escalables, además es un lenguaje de propósito general, ya que se pueden crear aplicaciones de escritorio, utilitarios, juegos, aplicaciones multimedia, sistemas con manejo de bases de datos, componentes reutilizable, aplicaciones distribuidas y mucho más.

Hoy en día la conceptualización emitida en investigaciones anteriores no ha cambiado, pues se sigue utilizando el lenguaje con el mismo propósito inicial, se puede hablar de mejoras en la plataforma pues cada nueva versión trae consigo características avanzadas que lo convierten en uno de los lenguajes mas poderosos en la actualidad.

En cuanto al almacenamiento de datos, existen varios Motores de bases de datos que permiten que el usuario mantenga la información organizada y con seguridad, es así que, SQL Server desde versiones anteriores ha brindado a los usuarios grandes ventajas en su utilización, hoy por hoy la versión 2005 de este Motor de Bases de Datos proporciona múltiples ventajas, con mejoras en su funcionalidad haciendo de este Motor una herramienta dinámica de fácil utilización.

Con la argumentación que antecede a está investigación, la creación del Software Volcano Control es la premisa básica que dará apertura al cambio del Sistema Manual que se emplea en la Asociación Volcano Flowers para controlar la información de la producción de claves a un Sistema Informático que sirva para el tratamiento automático de la información.

1.1 SISTEMA INFORMÁTICO

1.1.1 DEFINICIÓN

Podrían darse muchas definiciones de sistema informático. Dos de ellas son las siguientes:

Conjunto de elementos que hacen posible el tratamiento automático de la de información.

Sistema compuesto de equipos y de personal que realiza funciones de entrada, proceso, almacenamiento, salida y control con el fin de llevar a cabo una secuencia de operaciones con datos.

En términos genéricos se puede afirmar que el sistema informático es un conjunto de elementos interrelacionados entre sí y relacionados a su vez con el sistema global en que se encuentra que pretende conseguir unos fines determinados. Los elementos constitutivos de un sistema informático serán físicos lógicos y humanos.

Estructuralmente un sistema se puede dividir en partes pero, funcionalmente es indivisible, ya que si se dividiera perdería alguna de sus propiedades esenciales. Así, un sistema informático sin alguno de sus componentes, no funcionaría. Como características globales de un sistema informático podríamos señalar las siguientes:

Las propiedades o comportamiento de cada uno de los elementos del sistema influyen en las propiedades y funcionamiento del sistema completo.

El tipo de influencia que ejerce cada elemento del sistema depende, al menos, del comportamiento de otro elemento.

Cada sistema informático se compone, a su vez, de subsistemas que son sistemas informáticos por sí mismo. Al final de la descomposición se llegará al sistema

Habrá que determinar en que sentido y nivel de descomposición estamos hablando cuando nos referimos a un sistema informático.

Según LARRY LONG, Pág. 300-317, manifiesta que: “Los sistemas informáticos marcan el inicio de una época importante para el hombre. Son una herramienta poderosa e inigualable, particularmente para ayudar a los hombres de negocios y profesionales a mejorar el manejo y rendimiento de sus trabajos. Permiten además expandir a un rango de aplicaciones aún más grandes; y que los trabajos se realicen de manera automática, rápida y segura, liberando al usuario de los procesos complejos que se realizan internamente”.

Normalmente, el rendimiento de un sistema informático depende más de la relación y coordinación entre sus componentes que del funcionamiento de cada uno de ellos individualmente. Por eso, a veces, el funcionamiento de un sistema informático no se mejora usando los mejores componentes físicos, lógicos y humanos sino armonizando y coordinando efectivamente sus relaciones, a continuación se pone ha consideración cada uno de los componentes de un Sistema Informático y su respectivo funcionamiento.

1.1.2 COMPONENTES Y FUNCIONAMIENTO GENERAL DE UN SISTEMA INFORMÁTICO.

Un sistema informático está compuesto por:

- a) Componente físico: que constituye el hardware del sistema informático y lo conforman, básicamente, los ordenadores, los periféricos y el sistema de comunicaciones. Los componentes físicos proporcionan la capacidad y la potencia de cálculo del sistema informático.
- b) Componente lógico: que constituye el software del sistema informático y lo conforman, básicamente, los programas, las estructuras de datos y la documentación

El software se encuentra distribuido en el hardware y lleva a cabo el proceso lógico que requieren los datos.

c) Componente humano: constituido por todas las personas participantes en todas las fases de la vida de un sistema informático (diseño, desarrollo, implantación, explotación). Este componente humano es sumamente importante ya que los sistemas informáticos están desarrollados por humanos y para uso de humanos.

Ver Anexo de Gráficos, GRAFICO N° 1.1: ESTRUCTURA DE UN SISTEMA INFORMÁTICO, Pág. 85

Este camino hacia la implantación progresiva de sistemas distribuidos ha pasado por diferentes fases y se puede considerar que aún no ha finalizado. Veamos estas fases más detenidamente:

- En una primera fase, al inicio de la informatización de las organizaciones, los recursos están totalmente centralizados.
- En una segunda fase, se distribuyen los componentes físicos (y, en ocasiones, los humanos) del sistema. La capacidad de proceso y almacenamiento sigue estando centralizada pero las entradas y salidas de datos se han distribuido físicamente (terminales "tontos" conectados a un equipo central).
- En una tercera fase se distribuyen, además, los componentes lógicos del sistema. Las capacidades de proceso también se empiezan a distribuir pero no totalmente (terminales con cierta capacidad de proceso conectados a un equipo central).
- Por último, se llega al modelo más avanzado de informática distribuida en que tanto la capacidad de proceso como la capacidad de almacenamiento se encuentran distribuidas en diferentes lugares.

Una vez analizadas las diferentes fases de los sistemas distribuidos, se puede establecer que el sistema informático ha evolucionado desde una primera situación en que todos los componentes del sistema se encontraban centralizados en una sala de ordenadores a la situación actual en que los componentes del sistema se encuentran, ampliamente distribuidos en diferentes lugares físicos.

Seguidamente, se expone de manera amplia la forma en que han evolucionado los Sistema Informáticos.

1.1.3 EVOLUCIÓN DE LOS SISTEMAS INFORMÁTICOS

Las necesidades productivas y organizacionales de las empresas inducen la evolución de los sistemas informáticos hacia los Sistemas de la Información, abarcando por si solos todo lo que antes se realizaba con muchos programas informáticos e incorporando una premisa básica: la adaptación al saber hacer de la empresa, razón fundamental de su posición en el mercado.

En la digitalización de un proceso empresarial cada una de sus fases debe estar reflejada en el sistema informático: así por ejemplo, las informaciones derivadas de las etapas intermedias por las que pasa un producto, hasta llegar a convertirse en el demandado por el cliente, son volcadas al sistema donde quedan almacenados todos los hechos relevantes, ya sean problemas técnicos, tiempos empleados, desabastecimiento de almacén o número de pedidos realizados. Conocer estos datos permitirá medir, planificar y analizar de cara a la revisión y optimización del proceso, proporcionando información en tiempo real de todas las acciones de la cadena productiva, repercutiendo directamente en la mejor organización del trabajo.

En una fase posterior, los sistemas entre cliente y proveedor estarán interconectados permitiendo el flujo de operaciones entre ambos de una manera sencilla y eficaz.

Conceptos como el Just in Time o la Logística Integral, donde existe una coordinación precisa entre clientes y proveedores, han logrado madurez e implantación plenas gracias a los sistemas informáticos, proporcionando beneficios cuantificables para aquellas empresas favorables a su implantación.

Según Enciclopedia Curso de Informática Personal (2002, p 15 - 17), manifiesta que: “La personalización en función del perfil profesional será clave para obtener del sistema la información útil y precisa para el desarrollo del puesto, dejando atrás accesos indiscriminados e inseguros a los datos de la empresa, así como excesos de información que desorientan y restan perspectiva para la realización eficiente del trabajo”.

La accesibilidad a estos sistemas es obligada para convertir cualquier emplazamiento en una delegación de nuestra empresa; la conexión a través de un ordenador portátil, agenda electrónica o teléfono móvil al sistema permitirá que las actividades diarias se realicen en cualquier lugar con las mismas garantías que desde la sede central.

Es tanto el valor que se ha dado a la Evolución de los Sistemas Informáticos, que es necesario enfocar la Importancia de un Sistema Informático, que será precisado seguidamente.

1.1.4 IMPORTANCIA

Como sistema, una organización transforma inputs de recursos, bienes, información y servicios para obtener un producto. En su estructura se reconocen tres grupos diferenciados:

1. Los sistemas que atienden a la captación y evolución de los recursos fundamentales, en conexión con el entorno;
2. Los sistemas que permiten la administración o gobierno del sistema mayor u

3. Los sistemas que atienden al desarrollo de las tareas que son requeridas por la actividad de la organización.

El input es captado por los SISTEMAS DE RECURSOS, que son: Recursos Humanos, Recursos Financieros, Tecnológicos y Logísticos.

El procesamiento de los recursos, intersectando personal con operaciones (definidas por los procedimientos), y aplicándose sobre los otros recursos por medio de la tecnología, constituye la función de la empresa agrupada en los SISTEMAS OPERATIVOS, cuya naturaleza corresponde a las necesidades particulares de cada organización.

Todo el conjunto de recursos y operaciones, tienden a conformar dinámicamente estructuras que son ajustadas también dinámicamente por decisiones de la dirección. Esa dirección no se ejerce espontáneamente, sino que está a su vez estructurada: El sistema de Administración, o SISTEMA DE GOBIERNO, que se compone de varios sistemas corporativos, así llamados por tratarse de sistemas que afectan a la totalidad de la organización.

Estos son el Sistema Decisional, que establece la mecánica por medio de la cuál se toman las decisiones, el Sistema de Información, que explicita la estructura a través de la cual se captan y elaboran los datos, el Sistema de Planificación y Control, que anticipan lo que ocurrirá con cierta probabilidad, y evalúan lo que ocurrió realmente, y que constituyen los aspectos complementarios de la toma de decisiones.

Las decisiones se toman juntando datos e informándose a través del sistema de información, y otra vez a través del mismo se traducen en órdenes o normas para la producción, que luego se transforman en acciones.

El sistema de gobierno esta ahí, para que las PERSONAS que dirigen una organización puedan hacerlo sistemáticamente, que dispongan de información organizada para evaluar factores dentro y fuera de la misma, y para que el control de lo actuado sea tan eficaz, como sea posible.

La planificación es una toma de decisiones anticipada, pero, de hecho, implica un sistema para realizarla. Como en ese sistema no se puede prever todo, el control es necesario para la acción correctiva. El sistema de información, obra como nexo entre ellos, al transformar decisiones en acciones y los resultados de éstas en información de control.

Las tareas de este ciclo son las siguientes: un ACONTECIMIENTO externo o interno (evento), produce una excitación que da lugar a un REGISTRO de algunas de las variables afectadas: el DATO. Este se integra en un sistema que conjuntamente a otros datos conforman lo que llamamos BASE DE DATOS. Por medios manuales, mecánicos o electrónicos, el repositorio (la base) es utilizado para el PROCESAMIENTO, dando así lugar a la INFORMACIÓN.

1.2 BASE DE DATOS

Según la ENCICLOPEDIA Curso de Informática Personal (2002, p 29 - 30) da la siguiente definición de base de datos: “Las bases de datos son aplicaciones especializadas en almacenar grandes cantidades de datos de una forma organizada para que luego puedan consultarse de la forma más cómoda posible”.

Según la dirección electrónica:

<http://www.portalplanetasedna.com.ar/computacion2.htm> da la siguiente definición a DBMS: “La DBMS (Data Base Management System) es la herramienta que las computadoras utilizan para realizar el procesamiento y almacenamiento ordenado de los datos”.

Una base de datos es un recipiente para colecciones relacionadas de datos. Cualquier conjunto de datos organizados para su almacenamiento en la memoria de un ordenador o computadora, diseñado para facilitar su mantenimiento y acceso de una forma estándar. Los datos suelen aparecer en forma de texto, números o gráficos.

Desde su aparición en la década de 1950, se han hecho imprescindibles para las sociedades industriales. Hay cuatro modelos principales de bases de datos: el modelo jerárquico, el modelo en red, el modelo relacional (el más extendido hoy en día; los datos se almacenan en tablas a los que se accede mediante consultas escritas en SQL) y el modelo de bases de datos deductivas. Otra línea de investigación en este campo son las bases de datos orientadas a objeto, o de objetos persistentes.

Sea cual fuere el modelo de Bases de Datos, presentan varias características todas de relevante importancia, las mismas que se citan en la parte posterior.

1.2.1 CARACTERÍSTICAS

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoria.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

Cada una de estas características da lugar a que se generen diferentes ventajas con un propósito determinado.

1.2.2 VENTAJAS DE LAS BASES DE DATOS

Las principales ventajas en las que se hace énfasis son las siguientes:

1.2.2.1 CONTROL SOBRE LA REDUNDANCIA DE DATOS

Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos.

En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

1.2.2.2 CONSISTENCIA DE DATOS

Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias.

Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes.

1.2.2.3 COMPARTICIÓN DE DATOS

En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

1.2.2.4 MANTENIMIENTO DE ESTÁNDARES

Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales.

Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

1.2.2.5 MEJORA EN LA INTEGRIDAD DE DATOS

La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

1.2.2.6 MEJORA EN LA SEGURIDAD

La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados.

Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros.

1.2.2.7 MEJORA EN LA ACCESIBILIDAD A LOS DATOS

Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

1.2.2.8 MEJORA EN LA PRODUCTIVIDAD

El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación.

El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

1.2.2.9 MEJORA EN EL MANTENIMIENTO

En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan.

Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados.

Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

1.2.2.10 AUMENTO DE LA CONCURRENCIA

En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

1.2.2.11 MEJORA EN LOS SERVICIOS DE COPIAS DE SEGURIDAD

Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos.

En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

1.2.3 DESVENTAJAS DE LAS BASES DE DATOS

Las desventajas más comunes que se pueden citar son las siguientes:

1.2.3.1 COMPLEJIDAD

Los SGBD son conjuntos de programas que pueden llegar a ser complejos con una gran funcionalidad. Es preciso comprender muy bien esta funcionalidad para poder realizar un buen uso de ellos.

1.2.3.2 COSTE DEL EQUIPAMIENTO ADICIONAL

Tanto el SGBD, como la propia base de datos, pueden hacer que sea necesario adquirir más espacio de almacenamiento.

Además, para alcanzar las prestaciones deseadas, es posible que sea necesario adquirir una máquina más grande o una máquina que se dedique solamente al SGBD. Todo esto hará que la implantación de un sistema de bases de datos sea más cara.

1.2.3.3 VULNERABLE A LOS FALLOS

El hecho de que todo esté centralizado en el SGBD hace que el sistema sea más vulnerable ante los fallos que puedan producirse. Es por ello que deben tenerse copias de seguridad (Backup).

1.2.4 TIPOS DE CAMPOS

Cada Sistema de Base de Datos posee tipos de campos que pueden ser similares o diferentes. Entre los más comunes podemos nombrar:

- **Numérico:** entre los diferentes tipos de campos numéricos podemos encontrar enteros “sin decimales” y reales “decimales”.
- **Booleanos:** poseen dos estados: Verdadero “Si” y Falso “No”.
- **Memos:** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados.
- **Fechas:** almacenan fechas facilitando posteriormente su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- **Alfanuméricos:** contienen cifras y letras. Presentan una longitud limitada (255 caracteres).
- **Autoincrementables:** son campos numéricos enteros que incrementan en una unidad su valor para cada registro incorporado. Su utilidad resulta: Servir de identificador ya que resultan exclusivos de un registro.

1.2.5 MODELOS DE BASES DE DATOS

Según la dirección electrónica: http://es.wikipedia.org/wiki/Base_de_datos da la siguiente definición a Modelo de Bases de Datos: “Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y

Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

1.2.5.1 BASES DE DATOS JERÁRQUICAS

Según la dirección electrónica: http://es.wikipedia.org/wiki/Base_de_datos da la siguiente definición a Bases de Datos Jerárquicas: “Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*”. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

1.2.5.2 BASE DE DATOS DE RED

Según la dirección electrónica: http://es.wikipedia.org/wiki/Base_de_datos manifiesta lo siguiente: “Éste es un modelo ligeramente distinto del jerárquico; su diferencia es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico)”.

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores.

1.2.5.3 BASE DE DATOS RELACIONAL

Según la dirección electrónica: http://es.wikipedia.org/wiki/Base_de_datos manifiesta lo siguiente: “Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente”. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por *registros* (las filas de una tabla), que representarían las tuplas, y *campos* (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia. Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

1.2.5.4 BASES DE DATOS ORIENTADAS A OBJETOS

Este modelo propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- **Encapsulación** - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- **Herencia** - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.
- **Polimorfismo** - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

Según la dirección electrónica: http://es.wikipedia.org/wiki/Base_de_datos manifiesta lo siguiente: “En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz.

Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado”.

1.2.5.5 BASE DE DATOS DEDUCTIVAS

Según la dirección electrónica: http://es.wikipedia.org/wiki/Base_de_datos manifiesta lo siguiente: “Un sistema de base de datos deductivas, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática”.

Después de haber descrito los principales modelos de bases de datos, se debe deducir el Motor de Bases de Datos que se va a utilizar, sin antes mencionar que existen diferentes Motores para Bases de Datos, pero en este caso se hará un enfoque a SQL Server.

1.2.6 SQL

Microsoft SQL Server constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios. Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

1.2.6.1 DEFINICIÓN

Según el Autor CASTRO, F. Lorenzo; UCLM-ESI; Modelos de Datos; México, 1999/2000, manifiesta lo siguiente: “SQL (Structured Query Language) Server es un lenguaje de consulta estructurada. Un lenguaje para definir la estructura y procesamiento de una base de datos relacional. Se emplea como un lenguaje de consulta único o puede incorporarse en programas de aplicación. El American National Standards Institute acepta un SQL como una norma en Estados Unidos y

1.2.6.2 CARACTERÍSTICAS DE SQL SERVER.

SQL Server es la oferta completa de base de datos y análisis. Tanto por la capacidad para consultar la base de datos mediante un explorador como por la compatibilidad con el Lenguaje de marcado extensible (XML, *Extensible Markup Language*), SQL Server es la base de datos totalmente habilitada para Web. Además, ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de una base de datos empresarial.

Si lo que se mide es la velocidad en el desarrollo de aplicaciones y la velocidad del procesamiento de transacciones, SQL Server es la base de datos más rápida, convirtiéndole en la opción principal para la empresa que busca agilidad en sus operaciones.

Según la dirección electrónica

<http://www.microsoft.com/latam/sql/evaluation/features/default.asp> pone ha consideración las siguientes características por operación:

1. **Habilitado por completo al Web.**

- **Compatibilidad con XML.-** Simplifica la integración de sistemas de servicios de fondo y la transferencia de datos.
- **Análisis habilitado para Web.-** Analiza datos desde sitios remotos accesibles a través de Web.
- **Acceso Web a los datos.-** Conecta las bases de datos de SQL Server y cubos OLAP de manera flexible, mediante el Web sin necesidad de ninguna programación adicional.
- **Alojamiento de aplicaciones.-** SQL Server permite aprovechar totalmente las inversiones en hardware de forma que múltiples aplicaciones se pueden ejecutar en un solo servidor o externamente.

- **Seguridad.-** SQL garantiza que las aplicaciones sean seguras en cualquier entorno de red, basada en funciones y el cifrado de archivos y de la red.

2. Escalable y confiable.

- **Escalabilidad.-** Puede escalar aplicaciones hasta 32 CPU y 64 gigabytes (GB) de RAM. SQL Server ha demostrado un rendimiento insuperable.
- **Vistas con particiones distribuidas.-** Permite repartir carga entre varios servidores para obtener más escalabilidad.
- **Alta disponibilidad.-** Aumenta la disponibilidad de aplicaciones empresariales con el trasvase de registros, las copias de seguridad en línea y los clústeres de conmutación por error.
- **Duplicación.-** Implementa duplicaciones de mezcla, transaccionales y de instantáneas con sistemas heterogéneos

3. Menor plazo de comercialización.

- **Administración de bases de datos simplificada.-** Las características de ajuste y mantenimiento automáticos permiten a los administradores centrar su atención en otras tareas críticas.
- **Productividad del programador mejorada.-** Las funciones definidas por el usuario, la integridad referencial en cascada y el depurador integrado de Transact SQL permiten la reutilización del código para simplificar el proceso de desarrollo.
- **Servicios de transformación de datos.-** Automatiza las rutinas de extracción, transformación y carga de datos de orígenes heterogéneos.
- **Alojamiento Web y de aplicaciones.-** Puede alojar soluciones de comercio electrónico en otros proveedores, gracias a la compatibilidad de SQL Server con varias instancias.
- **English Query (Query Analyser).-** Permite que usuarios planteen preguntas en inglés en lugar de utilizar expresiones multidimensionales.

1.3 LENGUAJES DE PROGRAMACIÓN

Según la dirección electrónica:<http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>, da la siguiente definición:

“Los lenguajes de programación son herramientas que nos permiten crear programas y software”. En la actualidad existen varios lenguajes de programación, entre ellos tenemos Visual Basic, Java, etc...

Una computadora funciona bajo control de un programa el cual debe estar almacenado en la unidad de memoria; tales como el disco duro. Los *lenguajes de programación* de una computadora en particular se conocen como código de máquinas o lenguaje de máquinas.

Por otra parte, los lenguajes de programación en código de máquina son verdaderamente difíciles de entender para una persona, ya que están compuestos de códigos numéricos sin sentido nemotécnico.

Los *lenguajes de programación* facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar.

Existen estrategias que permiten ejecutar en una computadora un programa realizado en un lenguaje de programación simbólico. Los procesadores del lenguaje son los programas que permiten el tratamiento de la información en forma de texto, representada en los lenguajes de programación simbólicos.

Hay lenguajes de programación que utilizan *compilador*.

La ejecución de un programa con compilador requiere de dos etapas:

- 1) Traducir el programa simbólico a código máquina

Otros lenguajes de programación utilizan un programa intérprete o traductor, el cual analiza directamente la descripción simbólica del programa fuente y realiza las instrucciones dadas.

El *intérprete* en los lenguajes de programación simula una máquina virtual, donde el lenguaje de máquina es similar al lenguaje fuente.

La ventaja del proceso interprete es que no necesita de dos fases para ejecutar el programa, sin embargo su inconveniente es que la velocidad de ejecución es más lenta ya que debe analizar e interpretar las instrucciones contenidas en el programa fuente.

1.3.1 VISUAL STUDIO .NET

Según la dirección electrónica: [http://msdn2.microsoft.com/es-es/library/fx6bk1f4\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/fx6bk1f4(VS.80).aspx) pone a consideración lo siguiente: “Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes.

Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML”.

.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y

1.4 ARQUITECTURA TRES CAPAS EN TECNOLOGÍA CLIENTE/SERVIDOR

La metodología RPM presentada por C. Larman presupone una estructura de tres capas que es típica de los Sistemas de Información. Estas tres capas son:

1. La capa de la Presentación
2. La capa del Dominio de la Aplicación
3. La capa del Repositorio

RPM toca muy superficialmente los problemas asociados al desarrollo de una capa de Presentación. Menciona que es conveniente usar un enfoque de prototipaje y que pueden ser útiles los casos reales de uso; sin embargo no proporciona métodos, principios o lineamientos metodológicos que apoyen el desarrollo de una *metáfora de diseño para la interfaz*, el *diseño lógico* de la presentación o su *diseño físico*.

RPM tampoco proporciona muchas luces sobre el desarrollo de la capa del Repositorio y tiende a subestimar su complejidad.

Una vez elaborado el Modelo de Uso, RPM se recomienda seguir con actividades de *análisis de requerimientos*. Para ello se elabora un *Modelo Conceptual*, en el cual se busca identificar los conceptos importantes del dominio de la aplicación, abstrayendo todo lo que puedan ser mecanismos o conceptos propios de un diseño o de una implementación.

La mayoría de estos elementos de un Modelo Conceptual terminan agrupándose naturalmente en la capa del Dominio de la Aplicación.

Este enfoque de RPM termina por sesgar la metodología hacia el desarrollo de la capa del Dominio de la Aplicación, y en la práctica, tiende a hacer presuponer que el desarrollo de las otras capas se puede ajustar al diseño de la capa del Dominio de Aplicación, lo cual no es siempre el caso, incluso dentro del ámbito de los Sistemas de Información.

Ver Anexo de gráficos, GRAFICO N° 1.3: ARQUITECTURA TRES CAPAS EN TECNOLOGÍA CLIENTE/SERVIDOR, Pág. 86

1.4.1 CAPAS

Las capas que constituyen la arquitectura Cliente/Servidor son:

1.4.1.1 Capa de presentación

Según la dirección electrónica

<http://www.geocities.com/trescapas/capadepresentacion.htm> da a conocer que:

“Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios”.

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. Aún así, existen oportunidades para identificar servicios que residen en componentes separados.

El cliente proporciona el contexto de presentación, generalmente un *browser* como Microsoft Internet Explorer o Netscape Navigator, que permite ver los datos remotos a través de una capa de presentación HTML, O también una aplicación WIN32 como ser los formularios de Visual Basic.

Mediante el uso de componentes, se separa la programación que da acceso a los datos en las bases de datos y aplicaciones desde el diseño y otros contenidos de la página Web. Esto ayuda a asegurar que los desarrolladores estén libres para enfocarse en escribir su lógica de negocios en componentes sin preocuparse acerca de cómo se muestra la salida. Recíprocamente, esto da libertad a los diseñadores de usar herramientas familiares para modificar la interfaz.

La capa de servicios de presentación es responsable de:

1. Obtener información del usuario.
2. Enviar la información del usuario a los servicios de negocios para su procesamiento.
3. Recibir los resultados del procesamiento de los servicios de negocios.
4. Presentar estos resultados al usuario.

1.4.1.2 CAPA DE NEGOCIOS

Según la dirección electrónica

<http://www.geocities.com/trescapas/capadenegocios.htm> da a conocer que: “Los servicios de negocios son el “puente” entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Cuando los datos necesarios residen en un servidor de bases de datos, garantizan los servicios de datos indispensables para cumplir con la tarea de negocios o aplicar su regla. Esto aísla al usuario de la interacción directa con la base de datos”.

Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como introducir una orden de compra o imprimir una lista de clientes. Las reglas de negocio (*business rules*) son políticas que controlan el flujo de las

Como las reglas de negocio tienden a cambiar más frecuentemente que las tareas específicas de negocios a las que dan soporte, son candidatos ideales para encapsularlas en componentes que están lógicamente separados de la lógica de la aplicación en sí.

El nivel de servicios de negocios es responsable de:

1. Recibir la entrada del nivel de presentación.
2. Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar (por ejemplo, la preparación de impuestos por ingresos, el procesamiento de ordenes y así sucesivamente).
3. Enviar el resultado procesado al nivel de presentación.

1.4.1.3 CAPA DE DATOS

El nivel de servicios de datos es responsable de:

1. Almacenar los datos.
2. Recuperar los datos.
3. Mantener los datos.
4. La integridad de los datos.

Los servicios de datos tienen una variedad de formas y tamaños, incluyendo los sistemas de administración de bases de datos relacionales (SABDs) como Microsoft SQL Server, servidores de correo electrónico como Microsoft Exchange Server y sistemas de archivos tales como el Sistema de Archivos NTFS.

1.5 SISTEMA DE CONTROL

Según la dirección electrónica: http://es.wikipedia.org/wiki/Sistema_de_control da la siguiente definición de sistemas de control: “Un sistema de control está definido como un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado”.

El diccionario de computación, FREEDMAN (2000, p227) manifiesta que: “Un Sistema de control se encarga de realizar computaciones sobre un conjunto de datos medios y presentados por una fuente física, es decir natural, de tal manera que desarrolla un modelo paramétrico óptimo de dicha fuente.”

1.6 SEGURIDAD INFORMÁTICA

Generalmente, los sistemas de información incluyen todos los datos de una compañía y también en el material y los recursos de software que permiten a una compañía almacenar y hacer circular estos datos. Los sistemas de información son fundamentales para las compañías y deben ser protegidos.

La seguridad informática consiste en garantizar que el material y los recursos de software de una organización se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto.

La seguridad informática se resume, por lo general, en cinco objetivos principales:

- **Integridad:** garantizar que los datos sean los que se supone que son
- **Confidencialidad:** asegurar que sólo los individuos autorizados tengan acceso a los recursos que se intercambian
- **Disponibilidad:** garantizar el correcto funcionamiento de los sistemas
- **Evitar el rechazo:** garantizar de que no pueda negar una operación realizada.
- **Autenticación:** asegurar que sólo los individuos autorizados tengan acceso a los recursos.

1.7 CICLO DE VIDA DE UN SISTEMA INFORMÁTICO.

Según Whitten J., Bentley L., Barlow V. 1996. *Análisis y Diseño de Sistemas de Información*. 3era. Edición. Pág: 95 manifiesta que: "Un ciclo de vida es un proceso por el cual los analistas de sistemas, los ingenieros de software, los programadores y los usuarios finales elaboran sistemas de información y aplicaciones informáticas".

COVIC (2001, p 42-51) nos dice: "La formalización del proceso de desarrollo se define como un marco de referencia denominado ciclo de desarrollo del software o ciclo de vida del software. Se puede describir como, "el periodo de tiempo que comienza con la decisión de desarrollar un producto software y finaliza cuando se ha entregado éste". Este ciclo, por lo general incluye las fases:

1. Definición del problema
2. Recopilación de información
3. Análisis y Diseño
4. Programación
5. Prueba
6. Documentación
7. Implementación

El ciclo de desarrollo de software se utiliza para estructurar las actividades que se llevan a cabo en el desarrollo de un producto. A pesar de que no hay acuerdo acerca del uso y la forma del modelo, este sigue siendo útil para la comprensión y el control del proceso".

1.7.1 CICLO DE VIDA EN CASCADA

La visión del modelo cascada del desarrollo de software es muy simple; se dice que el desarrollo de software puede ser a través de una secuencia simple de fases.

Cada fase tiene un conjunto de metas bien definidas, y las actividades dentro de una fase contribuyen a la satisfacción de metas de esa fase o quizás a una subsecuencia de metas de la fase. Las flechas muestran el flujo de información entre las fases. La flecha de avance muestra el flujo normal. Las flechas hacia atrás representan la retroalimentación.

El modelo de ciclo de vida cascada, captura algunos principios básicos:

1. Planear un proyecto antes de embarcarse en el mismo
2. Definir el comportamiento externo deseado del sistema antes de diseñar su arquitectura interna
3. Documentar los resultados de cada actividad
4. Diseñar un sistema antes de codificarlo
5. Testear un sistema después de construirlo

Las fases incluyen dentro de sí determinadas tareas que clasifican de una forma clara el trabajo a realizar.

El desarrollo de las fases, se produce de manera secuencial. Una vez se produce el análisis tanto del Sistema como de los requisitos del software demandado por el cliente, se procede a la fase de diseño de la arquitectura global del software. Un diseño elaborado de forma cuidadosa llevará a una rápida codificación. Tras haber traducido el programa a un lenguaje comprensible para el ordenador, se comprueban los elementos de forma individual y más tarde de manera homogénea. Una vez entregado el software al cliente, la fase de mantenimiento comprenderá las actualizaciones y las correcciones de errores que sean necesarias en el programa.

Ver anexo de gráficos, GRÁFICO N° 1.4: CICLO DE VIDA EN CASCADA, Pág. 86.

CAPITULO II

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

2. INFORMACIÓN DE LA ASOCIACIÓN VOLCANO FLOWERS

2.1 PATUTÁN

Antes , ésta comunidad se llamaba CHIPUXI pero cuando llegaron los padres Jesuitas le pusieron el Nombre de Patután, debido a que en el ejido existían patos y la gente decía “Patos están” lo que se deriva del nombre actual de la comunidad. Una de las señales que pusieron en este sector fue una cruz, ubicada en Guayrapungo, en el año 1958 se la declara comunidad jurídica

En la actualidad muchas personas que migraron a otras parroquias están regresando a vivir en la comunidad ya que gracias al CCF y a las capacitaciones que brinda a las familias del sector, estos están formando sus propias microempresas, entre las cuales mencionamos las siguientes:

Elaboración de lácteos

Cultivo de claveles

Elaboración de balanceados

Crianza de cuyes

Restaurant Asadero

Banco Comunitario

Asociación de Bloqueros

También se cuenta con una granja integral donde las familias aprenden hacer sus propias granjas familiares, ayuda al mejoramiento de viviendas, proporciona

2.1.1 INFORMACIÓN GENERAL DE PATUTÁN

UBICACIÓN

Provincia:	Cotopaxi
Cantón:	Latacunga
Parroquia:	Eloy Alfaro
Localización:	Está ubicada a 4,5 Km desde el centro de Latacunga
Vías de Acceso:	4 Km de asfalto vía a Saquisilí a ½ de tierra
Altitud:	2600 m.s.n.m
Comunidad:	Jurídica

LIMITES

Norte:	Comunidades de Chantillín – Canalo
Sur:	Barrio La Calera – Santa Isabel
Este:	Carretera Latacunga – Saquisilí - Chantillín
Oeste:	Barrio Escalera Loma – Hacienda la Rioja

INSTITUCIONES EDUCATIVAS

Guarderías:	Infancia Feliz cuenta con 26 niños
Escuelas:	Ramón Páez cuenta con 342 niños María Maldonado Enríquez cuenta con 95 niños

TIPO DE SUELO:	Arenoso
CLIMA:	Templado Frio
TEMPERATURA:	14 grados centígrados promedio
EPOCA DE LLUVIA:	Octubre – Abril, esporádicamente y escasas
EPOCA DE SIEMBRA:	Octubre - Abril con las lluvias
TIPO DE PRODUCCIÓN	
Vegetal:	Maíz, fréjol, quinua, morocho
Hortalizas:	Tomate riñón en invernadero
Animales:	Ganado, chanchos, ovejas, cuyes, conejos, aves
RAZA:	Indígena
IDIOMA PREDOMINANTE:	70% español, 30% quichua
DISTRIBUCIÓN DE LA ACTIVIDAD ECONÓMICA	
Agricultores:	8%
Jornaleros Agrícolas (florícolas):	74%
Pequeños Comerciantes:	10%
Empleadas Domésticas:	2%
Otros:	6%

TIPO DE TRANSPORTE: Bus urbano “Sultana del Cotopaxi”

Camionetas fletadas

SERVICIOS BÁSICOS: Luz, agua, alcantarillado

ORGANIZACIONES QUE EXISTEN EN LA COMUNIDAD

Directiva Comunal

Junta administrativa del agua Potable

Comité Coordinador de la Asociación “Camino a la Esperanza”

Comité Central de Padres de Familia de las Escuelas

Liga Deportiva Comunal

Directiva de Agua de Regadío

Comité de Padres de Familia del Colegio a Distancia

2.1.2 ASOCIACIÓN COMUNAL CAMINO A LA ESPERANZA

La Asociación Comunal Camino a la Esperanza, viene trabajando en convenio con CCF desde el mes de julio de 1994, fecha en la cual se afiliaron las Comunidades de Patutan y Cruz Loma en el año 1999 se afilia a la Comunidad de San Vicente y en el año 2005 se afilian las comunidades de Tilipulo, Zumbalica y Escalera Loma.

Durante estos años la Asociación ha entregado servicios directos e indirectos a las familias de las comunidades mencionadas.

Nuestra Asociación ha venido ejecutando programas en beneficio de los niños y las familias entre los que se pueden destacar las siguientes:

Saneamiento ambiental familiar y comunal orientado al acceso a agua segura, disposición de excretas, recolección de basura.

Salud Infantil y de Madres con actividades educativas de seguimiento y tratamiento a niños menores de 15 años y madres en estado de gestación y lactancia.

Capacitación nutricional en donde se ha realizado el control y seguimiento nutricional, formación de huertos familiares y control de alimentos.

Desarrollo del Niño orientado a fortalecer el cuidado de los niños y adecuación de rincones de estimulación temprana.

Educación apoyo con material didáctico y profesores complementarios en las escuelas.

2.1.3 CHRISTIAN CHILDREN'S FUND, INC

2.1.3.1 CCF DEL ECUADOR

El CCf es una de las organizaciones de auspicio más grandes en el mundo que promueve el Desarrollo y Protección a más de 255 millones de niños

Christian Children's Fund es una organización que impulsa el desarrollo humano sin ánimos de lucro y no sectaria, fundada en 1938. Su casa matriz se encuentra ubicada en Richmond, Virginia Estados Unidos y está constituida por una alianza de organizaciones ubicadas en Inglaterra, Alemania, Dinamarca, Suecia, Canadá, Francia, Nueva Zelandia, Australia, Japón, Corea y Taiwan.

Su propósito es de promover el mejoramiento de las condiciones de vida y de los ambientes en los que se desenvuelven los niños más necesitados y sus familias, dentro de su contexto cultural y en países en desarrollo; a través de programas financiados por las donaciones de personas, que se relacionan con los niños a través de la correspondencia y gestionados por las mismas familias quienes los co-financian con sus propios recursos o aquellos disponibles en el lugar.

2.1.3.2 CCF EN EL ECUADOR

Inicia su operación de afiliación de niños y organización de proyectos en septiembre de 1984, estableciendo como prioritaria la atención al área rural – andina, por su alto grado de pobreza y por presentarse en ésta escasa o ningún apoyo gubernamental, ni privado. Previendo a prevenir la emigración rural y la consecuente marginalidad urbana, impulsando la autogestión de los padres y la renovación del niño como sujeto activo de su propio desarrollo.

2.1.3.3 ¿DÓNDE OPERA CCF?

Opera en el área centro – norte del país, que corresponde a la zona andina, que no es una zona geográfica homogénea, pues incluye diferentes grupos étnicos: indios, afro-ecuatorianos y mestizos especialmente, quienes muestran una gran diversidad cultural y viven en pisos ecológicos distintos, desde el paramo andino hasta el subtropico

Las operaciones se encuentran divididas en tres regiones:

Región Norte:	Carchi e Imbabura
Región Centro:	Zona Urbana y Rural de Pichincha
Región Sur:	Cotopaxi y Noroccidente de Pichincha

2.1.3.4 ¿CÓMO TRABAJA CCF?

2.1.3.4.1 AUTOGESTIÓN o participación activa de los padres, de los jóvenes y los niños en el análisis de su situación, identificación y puesta en marcha de alternativas de acción que alivien, mejoren y sienten las bases para solucionar las situaciones negativas que los afectan y mantengan el impacto positivo de los programas desarrollados

2.1.3.4.2 CAPACITACIÓN Y ACOMPAÑAMIENTO del personal de CCF, que se convierte en facilitador de procesos y conductor de análisis para la planeación, seguimiento y evaluación de los programas. La capacitación es fundamental para la formación de recursos internos en las comunidades para facilitar la administración y consecución de recursos y para impulsar una organización comunitaria que proyecte, gestione y ejecute los programas

2.1.3.4.3 DESARROLLO DEL NIÑO como enfoque de los programas, puesto que orienta la definición de metas y objetivos de los planes comunitarios y considera al niño como un ser integral que se desarrolla en los aspectos físico, emocional, intelectual y social; y, se relaciona con su familia, su ambiente y la sociedad.

2.1.3.5 ¿QUÉ HACE CCF?

Trabaja organizado en grupos familiares, que son los responsables de definir sus programas de acuerdo con las necesidades de desarrollo de los niños y del ambiente familiar y comunitario, pero de manera integral. Los programas básicos son:

2.1.3.5.1 SALUD

Los niños en condición de pobreza son afectados significativamente en sus posibilidades de supervivencia y desarrollo, por lo que programas de acceso a agua segura, saneamiento ambiental, prevención y atención de la salud de niños y madres; acceso a alimentación nutritiva, ambiente familiar y comunitario saludable y seguro, son comunes en las comunidades apoyadas.

2.1.3.5.2 EDUCACIÓN

Familias numerosa y con escasos recursos causan que los niños no ingresen o retrasen su ingreso a la escuela y más tarde la abandonen para incorporarse a la producción familiar de subsistencia, por lo tanto su acceso y permanencia en los niveles de educación básica, el mejoramiento de la calidad de ésta y las oportunidades educativas alternativas y/o vocacionales son prioridades ineludibles

Se realizan también programas culturales educativos complementarios el de comunicación y cultura y el de desarrollo sico-social de los niños con estrategias que promueven la recuperación de la identidad cultural, del rol familiar como cuidador y educador primario del niño y del juego como fuente de aprendizaje y desarrollo.

2.1.4 ASOCIACIÓN VOLCANO FLOWERS

La Asociación Volcano Flowers es un pequeño grupo de floricultores dedicado exclusivamente al cultivo del clavel y la rosa, que como parte de la Asociación principal Camino a la Esperanza encamina parte de sus utilidades son destinadas al progreso del Barrio Patután.

Inicialmente la idea de crear una pequeña Asociación de floricultores nace en el año de 1998, a partir de un proyecto denominado “Claveles para la exportación”, el mismo que financiado por el CCF y capacitado por la misma Institución comenzó a tener resultados en el año de 1999.

A mediados del año 1999 se empiezan a crear las primeras plantaciones de prueba, invirtiendo un gran capital, y se empiezan a sumar personas al proyecto con lo que se empieza a gestionar dentro de la misma Asociación Camino a la Esperanza, para que se de un espacio para crear las instalaciones de lo que hoy es la Asociación Volcano Flowers.

En el año 2000 la ASOCIACIÓN VOLCANO FLOWERS, empieza con sus actividades, contando en ese entonces con 4 asociados capitalistas y productores, tras su primera cosecha cada asociado empieza a crear su propia plantación y toda la producción que individualmente se obtenía era enviada a la plantación principal VOLCANO FLOWERS (VF), para que a través de grandes plantaciones sean comercializados sus claveles. Ya para el año 2002 se incrementa el numero de asociados y se crean mas plantaciones que llevaban el mismo ritmo de trabajo pero con mejoras en su producción. A partir del año 2003 con 53 personas, entre Directivos, Asociados, y trabajadores deciden producir claveles no solo para el mercado nacional sino también para el mercado internacional y establecen contacto con la florícola Los Ilinizas con la cual empiezan a exportar sus claveles a España, es desde ese entonces que la AVF se consolida como empresa productora de claveles, creando nuevas oportunidades de trabajo.

Ver Otros Anexos. Listado de Asociados fundadores, Pág 92.

Es necesario mencionar que en el listado se encuentran Directivos, Asociados y Proveedores, y algunos trabajadores aunque no en su totalidad.

2.2 POBLACIÓN Y MUESTRA

2.2.1 POBLACIÓN

De acuerdo a la importancia de la investigación en curso se toma en consideración la siguiente población:

Ver anexo de tablas, Tabla 2.2: Población y Muestra, Pág. 101

Ver anexo de gráficos, Gráfico 2.1: Población y Muestra, Pág. 87

2.2.2 MUESTRA

Debido a que la población que intervendrá en la investigación es mínima, se ha visto conveniente tomar en cuenta el total de la población, pues 53 personas proporcionaran la ayuda suficiente para la implementación del sistema que permita llevar el control de la producción de claveles en la Asociación Volcano Flowers ubicada en la Parroquia Eloy Alfaro, Barrio Patután del cantón Latacunga.

2.3 DISEÑO METODOLÓGICO

2.3.1 TIPO DE INVESTIGACIÓN

La presente investigación, es de tipo Descriptiva, puesto que ayuda a comprender las características externas del objeto de estudio; a fin de profundizar el conocimiento objetivo del problema, y a su vez la investigación científica pues se realiza en un tiempo y lugar determinado, logrando de esta forma identificar el problema que afecta de cierta forma a la Asociación Volcano Flowers.

A través de la investigación en curso se pretende implementar un Sistema útil como estrategia de solución para mejorar la calidad de producción de claveles en la AVF, identificando las relaciones existentes entre las variables.

El grupo investigador sobre la base de una hipótesis estima posibles resultados, que aporten al conocimiento y a la implementación de un Sistema oportuno y eficiente en el manejo de la información sobre el control de la producción de claveles.

2.3.2 METODOLOGÍA

En la presente investigación se emplea el método hipotético deductivo, pues se plantea que la hipótesis sea demostrada mediante el avance y secuencia de la misma y de igual manera establecer conclusiones y recomendaciones.

Para el Software que se va a desarrollar para la AVF utilizaremos el **Modelo Cascada** siendo este el método más apropiado para la creación de este tipo de proyectos. Este modelo consta de las siguientes fases:

Análisis de requerimientos.- etapa en la que se comprende el problema de estudio, tomando en cuenta ciertos parámetros y restricciones.

Especificaciones.- dan lugar a todas y cada una de las actividades que debe realizar el sistema.

Diseño.- consiste en la creación del aspecto gráfico del software, así como también de su codificación.

Implementación.- consiste en la instalación del SW y de cada componente a fin de comprobar la funcionalidad del sistema.

Prueba.- es la fase en la que se determina si dicho SW cumple con las especificaciones registradas al inicio de su desarrollo, dentro de esta fase se realiza la respectiva detección de errores y posterior corrección de errores.

Mantenimiento.- modificaciones constantes del sistema para mejorarlo.

2.3.2.1 MÉTODOS

2.3.2.1.1 MÉTODO CIENTÍFICO

Al ser ésta investigación un aporte significativo para la Asociación Volcano Flowers se ha podido establecer:

Plantear un problema

Formular objetivos e hipótesis

Recoger datos

Realizar el análisis de datos

Establecer conclusiones

2.3.2.1.2 MÉTODO ANALÍTICO SINTÉTICO

Debido a que este método permite descomponer el todo en sus partes, para poder establecer un análisis respectivo y luego relacionar las partes motivo de estudio y volver a integrarlas, encaja perfectamente al sistema a desarrollarse ya que su diseño y desarrollo se lo realizó a través de módulos que finalmente se unificaron para su funcionamiento.

2.3.2.1.3 MÉTODO DESCRIPTIVO

Este método permite detallar las causas y efectos que son provocados por los problemas que se suscitan dentro de la Asociación Volcano Flowers, con respecto al control de la producción de claveles.

2.3.2.2 TÉCNICAS

2.3.2.2.1 OBSERVACIÓN

A través de esta técnica se podrá tomar en consideración las inexactitudes originadas en la Asociación en donde se proyectó implantar el sistema de control de producción.

2.3.2.2.2 LA ENTREVISTA

Esta técnica se basa en el dialogo directo entre el entrevistador y el entrevistado, lo cual facilita la recopilación de datos provenientes del presidente, secretario y tesorero de la Asociación Volcano Flowers.

2.3.2.2.3 ENCUESTA

Es necesario recopilar información de un número considerable de personas, por lo que esta técnica es la adecuada para aplicarla a los trabajadores, proveedores y asociados.

2.3.2.3 INSTRUMENTOS

2.3.2.3.1 GUÍAS DE ENTREVISTA.- Permite al investigador tener a su alcance un documento que sirve como soporte a la entrevista

2.3.2.3.2 FICHAS DE OBSERVACIÓN.- permitirá conocer y detallar los aspectos del entorno de trabajo, además describir las características de la institución.

2.3.2.3.3 CUESTIONARIO.- instrumento que será redactado con preguntas cerradas y de opción múltiple.

2.4 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS DE LA ENTREVISTA APLICADA A LAS AUTORIDADES DE LA ASOCIACIÓN VOLCANO FLOWERS.

De acuerdo a los resultados obtenidos mediante la entrevista aplicada a las autoridades (Presidente, Secretario, Tesorero) de la Asociación Volcano Flowers en cuanto se refiere a la forma en la que se lleva el control sobre la producción de claveles, se deduce que existe una magna preocupación, pues toda la producción dentro de la plantación es controlada en forma manual y pese ha ser una Asociación que recién está en sus inicios acarrea serios problemas, pues todavía no existe un Sistema que brinde ayuda ante tanta conflictividad.

Es muy importante conocer si es indispensable para la Asociación sistematizar la forma en la que se lleva el control de la producción, por lo que se preguntó si se cree conveniente la implementación de un sistema que permita llevar el control de la producción de claveles a fin de proporcionar ayuda a las personas que laboran en dicha plantación, ante lo cual manifestaron que sería la mejor herramienta de trabajo como una alternativa de solución ante las múltiples contrariedades.

Ratifican que la forma actual de llevar el control sobre la producción de claveles causa serias equivocaciones que a la larga se tornan en grandes pérdidas no sólo de capital, elementos químicos, sino también de tiempo, a su vez todo esto provoca que la producción frene su secuencia y se pierdan lugares en el mercado.

Se manifiesta que al ser una Asociación creada por una Asociación superior conformada por varios miembros, obligatoriamente se deben presentar reportes tras la comercialización de cada producción detallando la cantidad de materia prima o elementos químicos que se emplearon así como también los lugares de destino de la producción cantidades y costos, con la finalidad de conocer las pérdidas y ganancias que se obtuvieron en cada producción.

2.5 ANÁLISIS E INTERPRETACIÓN DE RESULTADOS DE LA ENCUESTA APLICADA A LOS TRABAJADORES, PROVEEDORES Y ASOCIADOS DE VOLCANO FLOWERS.

Dada la importancia de ésta investigación, es necesario identificar cuan provechoso sería el implementar un Sistema Informático dentro de la Asociación Volcano Flowers, así como también identificar varios de los requerimientos para cumplir con la primera fase del modelo en cascada seleccionado para la construcción del Software, para cuyo fin se ha aplicado una encuesta tanto a los trabajadores, proveedores y asociados de la AVF. A continuación se muestra el cuestionario aplicado y cada una de las interrogantes con su respectiva tabulación e interpretación.

Ver otros anexos, Formato de Encuesta, Pág. 106

1. ¿Considera Ud. que la Asociación Volcano Flowers es una empresa florícola creada para ser una fuente de trabajo única beneficiara del barrio Patután?

Ver Anexo de Tablas, Tabla 2.3. Pregunta N° 1, Pág. 101

Ver Anexo de gráficos, Gráfico 2.2. Pregunta N° 1, Pág. 87

INTERPRETACIÓN

Una vez puesta a consideración la interrogante a los trabajadores se pudo verificar que en un 86% la respuesta afirmativa de los encuestados, permite que salte a la vista de todos que la florícola destina su capital para múltiples beneficios en el barrio, es así como se ha constituido como una gran fuente de trabajo, pero también por otro lado existen quienes en un 14% dan su opinión negativa lo que conlleva a pensar que existe un cierto grado de disconformidad.

2. ¿Cree Ud. que de alguna forma las actividades que se realizan dentro de la AVF generan múltiples problemas debido a la forma en la que se lleva el control de la producción?

Ver Anexo de Tablas, Tabla 2.4. Pregunta N° 2, Pág. 101

Ver Anexo de gráficos, Gráfico 2.3. Pregunta N° 2, Pág. 88

INTERPRETACIÓN

Debido a que no todas las actividades se realizan dentro de un mismo lugar y varias de las etapas de producción se efectúan en la Asociación principal (ACE), en un 58% si considera que la forma actual de manejar la producción es causa primordial de los problemas que se dan dentro de la AVF, de la misma forma existen opiniones negativas en un 24% y ocasionales en un 18%, pero de acuerdo a los resultados obtenidos se puede llegar a la conclusión que varios de los problemas en cuanto a la producción se refiere tienen su origen en la forma empleada para llevar el control sobre la producción de claveles.

3. ¿Piensa Ud. que el control que actualmente se lleva sobre la producción de claveles es el más adecuado?

Ver Anexo de Tablas, Tabla 2.5. Pregunta N° 3, Pág. 102

Ver Anexo de gráficos, Gráfico 2.4, Pregunta N° 3, Pág. 88

INTERPRETACIÓN

Tomando en cuenta que la AVF es una pequeña empresa que tiene pocos años de funcionamiento, día a día se torna defectuoso el sistema manual empleado para llevar la información, lo que se comprueba con el resultado arrojado en las encuestas aplicadas, siendo el 76% de las personas encuestadas que están en desacuerdo con el sistema de control actual y tan solo el 24% que se halla conforme con dicho sistema.

4. ¿Con que frecuencia se presentan reportes o informes acerca de la información que se toma en cuenta en la producción de claveles?

Ver Anexo de Tablas, Tabla 2.6. Pregunta N° 4, Pág. 102

Ver Anexo de gráficos, Gráfico 2.5, Pregunta N° 4, Pág. 89

INTERPRETACIÓN

Por los resultados obtenidos al aplicar la encuesta y haciendo referencia a la interrogante planteada el 50% del personal que labora para la AVF manifiesta que los informes se realizan una vez que se comercializa la producción que se obtiene en un cierto período de tiempo, con la finalidad de establecer conclusiones en cuanto a utilidades se refiere y a la cantidad de producción que se comercializó obviamente existen trabajadores que realizan reportes tras cada etapa de producción pues se cree que es mejor el elaborar o presentar un informe por etapa a fin de cubrir el pedido del cliente.

5. ¿Por lo general en qué lapso de tiempo se comercializa el clavel?

Ver Anexo de Tablas, Tabla 2.7. Pregunta N° 5, Pág. 102

Ver Anexo de gráficos, Gráfico 2.6, Pregunta N° 5, Pág. 89

INTERPRETACIÓN

Es muy importante conocer el tiempo que tarda el clavel en estar en optimas condiciones para ser comercializado, se puede afirmar con los resultados obtenidos de las encuestas aplicadas en un 100% todos coinciden que la comercialización de esta flor es sumamente rentable pues se produce trimestralmente.

6. ¿En el caso de que toda la producción fuera de su responsabilidad que medio emplearía para llevar el control de la misma?

Ver Anexo de Tablas, Tabla 2.8. Pregunta N° 6, Pág. 102

Ver Anexo de gráficos, Gráfico 2.7, Pregunta N° 6, Pág. 90

INTERPRETACIÓN

Sin lugar a duda es evidente que la tecnología avanza con el pasar del tiempo y que cada segundo es vital para crear algo nuevo, la respuesta en un 78% reflejaba su interés por salir de medios tradicionales que solo dificultan su trabajo, con total seguridad se inclinarían por medios informáticos que den pleno respaldo y confiabilidad a las tareas realizadas minimizando tiempo y sin daños ni pérdidas de información que son el cimiento fundamental de la producción en la AVF, sin embargo todavía existe un mínimo porcentaje (22%) que por temor o por falta de preparación todavía no se siente capaz de manejar un medio informático.

7. ¿Está Ud. de acuerdo en que se lleve el control de la producción de claveles en forma automática empleando un computador?

Ver Anexo de Tablas, Tabla 2.9. Pregunta N° 7, Pág. 102

Ver Anexo de gráficos, Gráfico 2.8, Pregunta N° 7, Pág. 90

INTERPRETACIÓN

Indiscutiblemente el emplear un computador facilitaría el trabajo, en la actualidad existe información que todavía se desconoce debido a que pasa el tiempo y los escritos se deterioran y en muchas de las ocasiones se deja pasar por alto cierta información afectando directamente a la producción y en cierto modo a todas las personas que tienen la responsabilidad de velar por los intereses de toda una comunidad, es así que en un 94% están totalmente de acuerdo y tan solo en un 6% se hallan en desacuerdo.

8. ¿Cree Ud. que el registro de toda la información con proyecciones a futuro en forma automática permitirá establecer un régimen de pérdidas y ganancias desde la elaboración de un proyecto de cultivo de claveles?

Ver Anexo de Tablas, Tabla 2.10. Pregunta N° 8, Pág. 103

Ver Anexo de gráficos, Gráfico 2.9, Pregunta N° 8, Pág. 91

INTERPRETACIÓN

Se puede observar claramente con los resultados obtenidos que en un 98% toda proyección a futuro para la AVF es de total interés, pues desde la elaboración de un proyecto de cultivo de claveles ellos ya deben establecer un régimen de pérdidas y ganancias para saber si al hacer realidad el proyecto se beneficiaran con las utilidades y a partir de esto saber cuanto tienen que emplear para cada etapa y cuan rentable es cultivar y comercializar los claveles, cabe recalcar que el proceso de cultivo empieza desde la construcción del invernadero hasta la comercialización.

9. Cree Ud. que con la Implementación de un Sistema Informático se logrará mejorar la producción dentro de la AVF

Ver Anexo de Tablas, Tabla 2.11. Pregunta N° 9, Pág. 103

Ver Anexo de gráficos, Gráfico 2.10, Pregunta N° 9, Pág. 91

INTERPRETACIÓN

Con absoluta certeza el 100% opina que la producción se puede mejorar con la Implementación de un Sistema Informático, pues sólo de esta forma se puede organizar todos los procesos en un tiempo prudente y aprovechar todos los factores de producción que se emplean.

10. Que calificación proporcionaría Ud. a la posibilidad de que la AVF implemente un programa de automatización de toda la plantación con la única finalidad de proporcionar una alternativa de solución de problemas originados dentro de la misma.

Ver Anexo de Tablas, Tabla 2.12. Pregunta N° 10, Pág. 103

Ver Anexo de gráficos, Gráfico 2.11, Pregunta N° 10, Pág. 92

INTERPRETACIÓN

Con respuestas que causan satisfacción y que motivan para el desarrollo e Implementación de un Sistema Informático, dieron a conocer su punto de vista en la AVF llegando a un 86% del total, a calificaciones excelentes de entre los niveles planteados.

2.6 VERIFICACIÓN DE HIPÓTESIS

2.6.1 HIPÓTESIS

“La implementación de un sistema permitirá llevar el control de la producción de claveles en la Asociación Volcano Flowers ubicada en la Parroquia Eloy Alfaro, Barrio Patután del cantón Latacunga”.

2.6.1.1 VERIFICACIÓN

El estudio, análisis y evaluación de los diferentes requerimientos obtenidos mediante la aplicación de técnicas e instrumentos de investigación, han permitido conocer y valorar las diferentes ventajas que proporciona el implementar sistemas seguros y eficientes que se adapten y cubran las necesidades de una determinada actividad o proceso.

Es así que, con la implementación del Software Volcano Control en la Asociación Volcano Flowers, se han obtenido resultados favorables que permiten ver evidentemente el aporte que se ha brindado con dicha implementación, pues actualmente a través del Sistema se efectúa un control eficiente sobre la producción de claveles.

Las actividades que se realizaban en forma manual hoy se encuentran automatizadas, de esta forma se ha disminuido notablemente la pérdida de información de los registros, y al igual se mantiene una información ordenada confiable y a tiempo, lo que ha permitido comprobar la veracidad de la hipótesis planteada.

2.7 RECURSOS NECESARIOS (humanos, materiales, técnicos, presupuesto)

2.7.1 HUMANOS

Director de la Investigación – Ing. Juan Carlos Rodríguez

Personal de la AVF y de la ACE

Grupo Investigador:

Mónica Fernanda Brasales Panchi

Lorena Elizabeth Osorio Yugse

2.7.2 MATERIALES: DESGLOSE DE MATERIALES

Ver Anexo de Tablas, Tabla 2.13. MATERIALES: DESGLOSE DE MATERIALES, Pág. 104

2.7.3 TECNICOS: DESGLOSE DEL RECURSO TÉCNICO

Ver Anexo de Tablas, Tabla 2.14. TECNICOS: DESGLOSE DEL RECURSO TÉCNICO, Pág. 104

2.7.4 PRESUPUESTO

2.7.4.1 COSTOS DIRECTOS: DETALLE DE COSTOS

Ver Anexo de Tablas, Tabla 2.15. COSTOS DIRECTOS: DETALLE DE COSTOS, Pág. 105

2.7.4.2 COSTOS INDIRECTOS: DETALLE DE COSTOS

Ver Anexo de Tablas, Tabla 2.16. COSTOS INDIRECTOS: DETALLE DE COSTOS, Pág. 105

CAPÍTULO III

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA VOLCANO CONTROL, CODIFICADO EN VISUAL BASIC .NET 2005 Y BASE DE DATOS EN SQL SERVER 2005 (APLICACIÓN DE ESCRITORIO).

3. Tema: “IMPLEMENTACIÓN DE UN SISTEMA QUE PERMITA LLEVAR EL CONTROL DE LA PRODUCCIÓN DE CLAVELES EN LA ASOCIACIÓN VOLCANO FLOWERS (AVF) UBICADA EN LA PARROQUIA ELOY ALFARO, BARRIO PATUTÁN DEL CANTÓN LATACUNGA”

3.1 PRESENTACIÓN

El sistema VOLCANO CONTROL, desarrollado e implementado permitirá llevar el control de la producción de claveles dentro de la Asociación Volcano Flowers (AVF) en forma acertada y sencilla, dando lugar a que los procesos que se efectúan en la misma sean más organizados.

El sistema que permitirá llevar el control de la producción de claveles se basa en la información proporcionada por los Directivos, Asociados y Empleados de la AVF, quienes palpan a diario la realidad vigente dentro de la florícola , es así que la información con la que cuenta el sistema implica todo el proceso de producción, desde lo que es construcción hasta cuando se obtiene el producto final que en este caso es el clavel, los inventarios que en este caso resultan indispensables y las utilidades que recibe la Asociación o a su vez las pérdidas que se registran en la AVF, todo esto contemplando ciertas formulas

Este sistema permitirá que los usuarios agilicen los procesos y planifiquen de mejor forma todas las actividades basándose en la información real proporcionada por el mismo, con el objetivo de mejorar la producción.

Además de todo lo mencionado, se contara con un historial de producción lo cual permitirá tanto a los Asociados como a los Directivos de la AVF, realizar diferentes consultas no solo en fechas actuales sino también de fechas pasadas lo cual resulta una eficaz forma para establecer cuadros comparativos entre periodos para así mejorar cada vez la producción e incrementar las utilidades que benefician a varias familias y de cierto modo al Barrio Patután.

De igual forma se podrá a través de este sistema gestionar proyectos para aquellos Asociados que en la actualidad no se hallan produciendo y para aquellos que quieren aumentar su infraestructura y producción.

Se llevara un control absoluto de todos los procesos, de principio a fin.

En cuanto a los **Perfiles de Usuario** para establecer las seguridades que se emplearán en el Software Implementado, se maneja dos tipos de usuarios el Administrador con una clave única que permite tener acceso a todas las operaciones del Sistema (Inserción, Actualización, Modificación, Eliminación, Consultas), y el cliente o los clientes con claves identificadas en el sistema Volcano Control con acceso a Consultas, a fin de no permitir el mal uso de la información y de mantener en cierto modo toda la documentación en forma confidencial de uso exclusivo de la Asociación Volcano Flowers.

3.2 JUSTIFICACIÓN DE LA PROPUESTA

La automatización de la información que se maneja en la producción de claveles es de vital importancia, pues vivimos en una sociedad inmersa en el desarrollo tecnológico, lo que hace que las entidades tengan mayores exigencias que cumplir

El grupo de investigación tiene la necesidad de desarrollar un Sistema Informático útil y de fácil manejo, con una interfaz sencilla pero que llene las expectativas de la Asociación logrando de esta forma eliminar el sistema manual defectuoso y reemplazarlo por el sistema Informático planteado, denominado VOLCANO CONTROL.

La implementación del sistema antes mencionado es necesaria, por que permitirá la optimización de la información y una adecuada utilización de la misma, otro aspecto que ha sido considerado es el tiempo de respuesta de la información esto posibilita el desarrollo de las actividades de reportes, la elaboración de informes sistematizados rápidos y de alta calidad.

Otra de las razones por las que se planteó desarrollar la presente investigación se origina o se deriva de los accesos no autorizados para hacer uso de la información que de cierto modo dentro de la AVF se llevaba en forma manual, pues existen personas que no deberían utilizar dicha información pero que sin embargo acceden a ella de forma ilícita, se estima conveniente que una de las formas con las que se puede proteger la información que se lleva para el control de la producción de claveles es la creación de contraseñas como llaves del sistema que se ha implementado, lógicamente para que el sistema entre en funcionamiento el usuario deberá ingresar un usser (nombre-usuario) y un password (contraseña), que le permitan autenticarse y hacer uso del sistema.

Para la aplicación de la investigación contamos con el apoyo de los miembros de la ASOCIACIÓN VOLCANO FLOWERS y de todo el personal que labora en las instalaciones, a través del cual se ha logrado recolectar la información necesaria para el desarrollo del sistema y así finalmente implementarlo y dejarlo en funcionamiento.

El Sistema Informático que permitirá llevar el control de la producción de claveles se desarrolló utilizando las herramientas y lenguajes de programación como Visual Basic .NET 2005 y la Base de Datos en SQL Server 2005, permitiendo de esta manera mejorar eficientemente el llevar la información que se maneja para el control de producción de claveles en la Asociación Volcano Flowers, además el sistema será de uso exclusivo de la misma convirtiéndose así en un sistema local.

Con la Implementación del Sistema desarrollado se logrará dar un aporte significativo para el desarrollo de la Asociación Volcano Flowers, pues se constituirá en un estudio único y auténtico desarrollado dentro y para la AVF.

3.3 OBJETIVOS

3.3.1 OBJETIVO GENERAL

Implementar un sistema adecuado que va a permitir llevar el control de la producción de claveles en la Asociación Volcano Flowers (AVF) ubicada en la Parroquia Eloy Alfaro, Barrio Patután del cantón Latacunga.

3.3.2 OBJETIVOS ESPECÍFICOS

- Identificar la importancia que tiene la sustentación teórica en ésta investigación.
- Identificar las necesidades para controlar la producción de claveles en la Asociación Volcano Flowers.
- Seleccionar y analizar las herramientas informáticas que se van a emplear en el desarrollo del Sistema, a fin de construir un software de calidad.
- Implementar parámetros de Seguridad en el Software.
- Desarrollar el Sistema Informático para llevar el control de la producción de claveles en la Asociación Volcano Flowers, con una interfaz de fácil manejo para el usuario.
- Desarrollar un manual para el usuario del Sistema.

3.4 FACTIBILIDAD DE IMPLEMENTAR EL SISTEMA INFORMÁTICO PLANTEADO

Esta Investigación es un profundo análisis y estudio de una realidad vigente y palpable por el grupo investigador. Para el análisis se ha tomado en cuenta todos y cada uno de los procesos de producción de claveles a fin de automatizar toda la información que se maneja actualmente en la Producción de claveles dentro de la Asociación Volcano Flowers (AVF).

Actualmente el proceso de producción consta de las siguientes etapas:

Construcción

Preparación y adaptación del suelo

Construcción de camas

Cultivo

Cuidado de la planta

Fumigación

Recolección de claveles

Postcosecha

Comercialización

Las mismas que arrojan información que se llevan de forma manual, por lo que han sido estudiadas minuciosamente para el desarrollo de un Software de calidad que funcione como método de solución a los distintos inconvenientes que se presentan en la plantación.

Es indispensable mencionar que se cuenta con la información y documentación necesaria para la construcción del software, así como también con el Hardware, Software y el Recurso humano para el buen desarrollo del mismo.

Otro factor importante que el grupo investigador ha tomado en cuenta es el capital que se invierte en la realización de éste Software, dicho capital en un cierto porcentaje es proporcionado por la misma Asociación y en un mínimo porcentaje aportado por el grupo investigador.

Por otra parte la voluntad e interés por aprender a manejar un Software por parte de los trabajadores y de todos aquellos que laboran en las instalaciones de la AVF, son la razón principal y esencia de este proyecto.

Con todo lo expuesto queda demostrado que es factible la realización y puesta en marcha de ésta investigación, ya que se cuenta con todo lo necesario para lograr una buena implementación y garantizar un buen funcionamiento del Software.

3.5 DESARROLLO DE LA PROPUESTA

Volcano Control es un Sistema que permitirá controlar la producción de claveles dentro de la Asociación Volcano Flowers, para su desarrollo se han empleado varias herramientas informáticas que facilitan al programador crear un software de alta calidad, de fácil manejo y de codificación sencilla.

Para el desarrollo de la interfaz con el usuario se utilizó MICROSOFT VISUAL BASIC.NET 2005, ya que es una herramienta segura para la elaboración de Software, además el código generado es reutilizable, es decir que se puede utilizar en cualquier instancia en el desarrollo del proyecto y el soporte proporcionado por Microsoft es 24/7.

Dado que .NET es una plataforma que permite a los desarrolladores crear cualquier tipo de aplicación ya sea de escritorio u orientada a la Web, el lenguaje empleado ahorró a los desarrolladores el tiempo destinado para la elaboración del proyecto así como también muchas complicaciones que surgieron con la finalidad de dar solución a las necesidades de la Florícola.

En cuanto se refiere al almacenamiento de datos se utilizó SQL Server 2005, ya que es un motor de base de datos altamente confiable, con capacidad de mostrar grandes cantidades de datos de forma simultánea, en sí constituye una herramienta que contribuye significativamente al ordenamiento de datos en cualquier proceso de una empresa.

En lo que concierne a los modelos conceptual y físico del sistema se utilizó: Rational Rose 2000 Enterprise Edition, para realizar los Diagramas: Casos de Uso, Diagrama de Secuencia y Diagrama de clases una vez realizado estos diagramas se exportaron para poder realizar el diagrama Conceptual y Físico en Power Designer 9.5, es así como se obtuvo el script de la base de datos para ejecutarla en SQL Server 2005, donde se puede mostrar las tablas, operaciones y procesos almacenados que se emplearon para dar forma a través de las interfaces al Sistema.

Adicional a esto se utilizó Power Designer 6.0 para la creación de los Dfd's por niveles, y Adobe Photoshop para la edición de imágenes.

3.6 REQUERIMIENTOS DE LA ASOCIACIÓN VOLCANO FLOWERS PARA EFECTUAR EL CONTROL SOBRE LA PRODUCCIÓN DE CLAVELES.

Los requerimientos o necesidades que tiene el cliente de Software son la esencia principal para la creación de un Sistema Informático, pues a través de estos se manifiesta el verdadero origen de la problematización a la que lógicamente se debe dar solución con su desarrollo. Seguidamente se plantean los requerimientos que debe cubrir el Software Volcano Control:

Autenticación de usuarios.- dos tipos de usuarios, un Administrador y uno o varios clientes. Ingreso del nombre del usuario y su correspondiente contraseña para manipular el Sistema.

Permisos para el usuario.- en el caso de ser identificado como Administrador, tendrá acceso a todas las operaciones que se ejecuten en el Sistema y por el contrario si es identificado como cliente solo tendrá acceso a lo que son consultas, y búsquedas de los reportes.

Etapa inicial Proyecto de construcción

Etapa de construcción infraestructura

Etapa de preparación y adaptación del suelo.

Etapa de siembra

Etapa cuidado y fertilización

Etapa postcosecha

Etapa pedidos

Clientes

Compras

Ventas

Inventario – construcción

Inventario – elementos fertilización

Inventario – productos

Proveedores

Bodega

REPORTES

Reporte de bodega

Reporte de camas

Reporte de clientes

Reporte de elementos químicos (inventario)

Reporte de materiales construcción (inventario)

Reporte de Fertilización

Reporte ventas por mes

Reporte de compras por mes

Reporte pedidos por mes

Reporte de construcción materiales empleados

Reporte postcosecha

Reporte utilidades

3.7 ESPECIFICACIONES DEL SISTEMA VOLCANO CONTROL

En el Sistema Volcano Control se ejecutan cuatro tareas fundamentales: Inserciones, Eliminaciones Consultas y Modificaciones, o conocido también como un ABCM (Altas, Bajas, Consultas y Modificaciones), que son la base primordial del funcionamiento del Sistema.

Para garantizar el correcto funcionamiento del Software en mención y para mayor facilidad del usuario a fin de evitar confuciones se han diseñado en cada formulario dos secciones a partir de un tab control:

Tab 1.- denominado Búsqueda, en ésta se realiza una exploración minuciosa de datos que se han registrado anteriormente bajo ciertos criterios de búsqueda como descripción, área, fecha o de forma general dependiendo de las necesidades del usuario en cada etapa.

Tab 2.- Mantenimiento, a este Tab se podrá acceder dando un clic en el botón nuevo o a su vez dando doble clic en un registro a modificar, permite en esta sección el ingreso de nuevos registros, actualización o a su vez modificación de ciertos parámetros y eliminación de datos.

Cabe hacer énfasis en que el usuario que realice inserciones deberá llenar todos los campos solicitados, caso contrario no se podrá registrar la información en el Sistema.

En cuanto a las modificaciones dependiendo del formulario, el usuario podrá realizar los cambios solo en los campos sujetos a modificación y posterior guardar los nuevos datos.

Las eliminaciones se realizarán cuando el usuario lo estime conveniente. Finalmente los resultados de la funcionalidad del Software se ven reflejados en los reportes generados, los mismos que se realizan ingresando criterios de búsqueda, de acuerdo a los requerimientos que fueron planteados en la fase anterior.

3.8 ANÁLISIS Y DISEÑO DEL SISTEMA VOLCANO CONTROL

Después de haber realizado un estudio de todas y cada una de las necesidades de la Asociación Volcano Flowers relacionadas con la información que se maneja acerca de la producción de claveles, y haber determinado cuál será su funcionalidad se procede a interpretar de forma gráfica cada requerimiento, los cuales se pueden visualizar en la Interfaz del Sistema.

Seguidamente se analiza el procedimiento que se dio para la creación del Software Volcano Control:

Una vez creada la Base de Datos originada de las necesidades y requerimientos del Usuario se procede al Diseño y codificación de cada pantalla que interactuará con el o los usuarios. En esta Investigación se utilizó para el Diseño y codificación del Sistema un Software de Licencia, bajo la plataforma Microsoft, debido a las múltiples ventajas que ésta ofrece, es así que para la codificación se empleó el Lenguaje Visual Basic .NET 2005 y para el almacenamiento de Datos SQL Server 2005.

Es necesario mencionar que no se han utilizado herramientas de Código Abierto (Open Source), pues uno de los planteamientos de la Asociación Volcano Flowers fue que el Software sea desarrollado bajo la Plataforma de Microsoft.

A diferencia de las herramientas Open Source, Microsoft es una Plataforma que se caracteriza por cumplir a tiempo las exigencias o peticiones del cliente en forma sencilla, por ejemplo el soporte Microsoft es 24/7, es decir que el cliente efectúa su petición de ayuda en cualquier momento y ésta es cubierta en el mismo instante, en cambio con el soporte Open Source, se debe identificar con exactitud las características del problema generado, detallar y después enviar la petición del cliente para que en lo posterior sea atendida.

Otra de las cosas importantes que se debe tomar en cuenta al utilizar Software bajo Licencia, es que por ser un Software con costo brinda mayores seguridades tanto en la codificación como en el almacenamiento, algo que no sucede con Open Source que las seguridades en cambio son diseñadas y construídas por el cliente o programador.

En Open source el programador tiene que diseñar de acuerdo al proyecto que este creando, los complementos que va a utilizar a lo largo de la codificación para crear la solución. En cambio en lo que se refiere a Software bajo Licencia los complementos ya se encuentran desarrollados y se presentan como alternativas que el programador puede emplear para su solución, de esta forma se ahorra tiempo y se disminuye el margen de error en la programación de Interfaces.

Es muy importante la determinación del tipo de Software que se utilizará en el diseño de las Interfaces y el almacenamiento de datos, no solo por la comodidad del programador, sino más bien porque de la elección que se haga dependerán, el rendimiento, y correcto funcionamiento del Sistema creado.

El Sistema diseñado, desarrollado e Implementado denominado Volcano Control, emplea una programación en capas, lo que disminuye el grado de complejidad en su codificación, presenta 4 capas que se analizan a continuación:

- Una capa BEU, en la que se inicializan todos los atributos, propiedades y constructores.
- Una capa DL, en la que se establece el acceso a la base de datos mediante una cadena de conexión y los métodos que se van a emplear en la solución.
- Una capa BLL, que es muy importante porque en esta capa se hace referencia a las dos capas anteriores BEU y DL, en esta capa se codifican las operaciones del sistema Inserciones, Modificaciones, Eliminaciones, cabe destacar que para cada operación se utiliza un Proceso almacenado.

- Una capa UI, en la que se diseña la forma en que los usuarios manipularán los datos, es decir la parte gráfica del Sistema.

De esta forma se ha diseñado y desarrollado el Sistema Volcano Control y a continuación se da paso al Modelado y Diseño del Sistema a fin de conocer más detalladamente la forma en la que se diseña el Sistema antes de llegar a la Implementación.

3.8.1 MODELADO Y DISEÑO DEL SISTEMA

3.8.1.1 DIAGRAMA DE FLUJO DE DATOS

Un Diagrama de Flujo de Datos es una representación Gráfica que permite en primera instancia realizar una visualización del procesamiento de datos, además muestra la interacción entre el sistema y cada una de las entidades en forma detallada, en pocas palabras muestra en forma gráfica la funcionalidad del sistema, como se pondrá en la práctica y a parte se genera una planeación estratégica de negocios.

Los diagramas de flujo de datos pueden ser usados para proporcionar al usuario final una idea física de cómo resultarán los datos a última instancia, y cómo tienen un efecto sobre la estructura de todo el sistema. El desarrollo de un DFD ayuda en la identificación de los datos de la transacción en el modelo de datos.

Se pone a consideración del lector el siguiente DFD en el cual se muestra la funcionalidad del Sistema Volcano Control desarrollado para la Asociación Volcano Flowers.

Ver Anexo de gráficos, Gráfico 3.1. DIAGRAMA DE FLUJO DE DATOS,

3.8.1.2 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso es una de las partes más importantes en el modelado de un Sistema, pues con esta técnica se capturan los requisitos ya sean para la creación de un nuevo Software o para la actualización de un existente.

Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico.

En el Sistema Volcano Control se enmarcan tres actores: el cliente, el empleado como intermediario y el proveedor, cada uno con sus propios requerimientos pero que en general todos interactúan y se complementan en un mismo Sistema creando diferentes escenarios que muestran las características de funcionamiento del mismo.

Ver Anexo de gráficos, Gráfico 3.2. DIAGRAMA DE CASOS DE USO, Pág.94

3.8.1.3 DIAGRAMA DE SECUENCIA

Los diagramas de secuencia ilustran la interacción entre objetos y el orden secuencial en el que ocurren dichas interacciones, es decir como se comunican los objetos entre sí, proporciona un camino a partir de los escenarios para describir las operaciones en una forma más detallada. Los objetos se comunican mediante interfaces, para poder invocar a una operación.

El diagrama de secuencia es el núcleo de nuestro modelo dinámico, y muestra todos los cursos alternos que pueden tomar todos nuestros casos de uso.

Ver Anexo de gráficos, Gráfico 3.3. DIAGRAMA DE SECUENCIA, Pág. 95

3.8.1.4 DIAGRAMA DE CLASES

Un **diagrama de clases** es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Ver Anexo de gráficos, Gráfico 3.4. DIAGRAMA DE CLASES, Pág. 96

3.8.1.5 MODELO ORIENTADO A OBJETOS

3.8.1.5.1 MODELO CONCEPTUAL Y FÍSICO DE DATOS

El Modelo Conceptual de Datos al igual que el Modelo Físico de Datos, es un esquema que muestra la estructura de los objetos y asociaciones identificadas durante la etapa de análisis.

Para la elaboración de estos modelos y obtención de la base de datos para el desarrollo del Software, fue indispensable la información que proporcionó la directiva de la Asociación Volcano Flowers, así como también las necesidades de los empleados y personal que se encarga de controlar la producción de claveles, toda la información que fue proporcionada fue expuesta a diario, pues el grupo investigador tuvo que inmiscuirse a diario para conocer la situación real de la Florícola y saber a ciencia cierta que procesos se necesitaban automatizar.

Posterior a esto se organizó y clasificó toda información correspondiente a cada proceso que se lleva a cabo dentro de la Florícola, determinando que actividades se realizan en cada proceso y como se deben automatizar, cabe mencionar que todo se realizó mediante una investigación de campo y observación directa.

Pese al tiempo que se tardó en determinar todo lo que se debía automatizar se logró establecer realmente todas las entidades, atributos y relaciones que se debían establecer a fin de desarrollar un Software funcional que solvente y cubra todos los requerimientos.

A continuación se muestra como se halla estructurada la base de datos diseñada para el Software Volcano Control.

Ver Anexo de gráficos, Gráfico 3.5. MODELO CONCEPTUAL DE DATOS y MODELO FISICO DE DATOS Pág. 97

3.8.1.5.2 SCRIPT DE LA BASE DE DATOS

El script de la Base de Datos, son líneas de código obtenidas a partir de los diagramas contruidos anteriormente, es pasar del modo gráfico a la codificación para en lo posterior ser ejecutada en un Motor de Base de Datos, que en este caso es SQL Server 2005 y así crear las entidades con sus respectivas relaciones, procesos almacenados y funciones que faciliten el almacenamiento de información.

Ver Otros Anexos, BASE DE DATOS, Pág. 108

3.8.1.6 DISEÑO DE INTERFACES

El diseño de las Interfaces del Sistema es de total importancia, pues interactúan directamente con el Usuario, de éstas depende la manipulación adecuada del Software y la eficacia en rendimiento del mismo. Mientras más amigables resulten las Interfaces diseñadas mejor será su entendimiento.

Tomando en cuenta estas consideraciones, para el Diseño de las Interfaces del sistema se utilizó Visual Basic.Net 2005, Adobe Photoshop e Image Ready, para lo que son fondos, colores, edición de imágenes que se encuentran en los formularios contemplados en el Sistema. Se han creado las pantallas necesarias con lo que se cubre y respalda la base de datos.

Ver Otros Anexos, INTERFACES DEL SISTEMA, Pág. 196

3.8.2 IMPLEMENTACIÓN DEL SISTEMA VOLCANO CONTROL EN LA ASOCIACIÓN VOLCANO FLOWERS

En esta fase del modelo empleado para esta Investigación, se procede a la Implementación del Sistema diseñado y desarrollado, para lo cual es necesario tener en cuenta ciertos requerimientos que se deben cumplir para la instalación del Software con el objetivo de garantizar su correcto funcionamiento.

Se debe disponer de Equipos que cumpla con las siguientes características básicas:

Hardware:

MICROPROCESADOR	Pemtiun(R) 4 CPU 3.00 Ghz. ó superior.
MEMORIA RAM	Mínimo 512 Mb.
TARJETA DE RED	Incorporada al mainboard

Software:

- ✓ Microsoft SQL Server 2005
- ✓ Microsoft Windows XP PROFESIONAL
- ✓ Microsoft Office 2007
- ✓ Acrobat Reader 8 ó Superior.
- ✓ Microsoft .Net Framework 2.0.

Además de las características en los equipos que se van a utilizar, es importante que la persona que va hacer uso del Sistema Implementado, tenga Conocimientos básicos de computación, para afianzar una buena manipulación del Sistema, adicional a esto se debe contar con un Manual de usuario que servirá como guía en la Administración del Sistema. Solo de esta forma se podrá determinar el éxito de la Implementación y la efectividad del Software.

3.8.3 PRUEBAS REALIZADAS AL SISTEMA VOLCANO CONTROL

Una vez finalizado el desarrollo del Sistema Volcano Flowers, se efectuó la Implementación del mismo, y se llegó a poner a prueba el Software, con lo que se verificó su funcionamiento tanto en forma local como en Red, evidenciando una vez más que es un Software que cubre todos los requerimientos planteados en la fase inicial.

Volcano Control, se halla en funcionamiento satisfactoriamente en un 100%, cumpliendo con todas las actividades requeridas, es notable el aporte que brinda el Sistema dentro de la Asociación, pues se ha logrado reducir el tiempo de procesamiento de la información y la exactitud en los registros de producción.

Es necesario mencionar que el Software que actualmente se encuentra en uso funciona bajo el Sistema Operativo XP, tanto para el Servidor como para cada uno de los Clientes.

3.8.4 MANTENIMIENTO DEL SISTEMA

Después de haber realizado la Implementación del Sistema y las pruebas correspondientes, se pudo constatar que es un Sistema netamente eficiente, sin embargo con el transcurso del tiempo desde su implementación, se han dado ligeros y pequeños cambios en cuanto a reportes se refiere, pues se han planteado nuevos requerimientos que han surgido en base a su manipulación.

3.9 ASPECTOS QUE DIFICULTARON EL DESARROLLO DEL SISTEMA

Debido a que no existía la información adecuada por la desorganización de documentos dentro de la Florícola se dieron ciertos inconvenientes en la creación de la base de datos, pues las entidades que se planteaban inicialmente no cubrían de cierta forma todos los datos que se debían registrar en el sistema de automatización.

A medida que se fue recaudando la información y tras largo tiempo de realizar un profundo análisis de todos y cada uno de los procesos que se ejecutaban se pudo concretar la base de datos, que serviría para almacenar los datos en el sistema.

En cuanto se refiere a la programación no se presentaron mayores inconvenientes, se tuvo que modificar en ciertos aspectos la programación a fin de que no se creen confusiones y la aplicación no contenga errores.

Finalmente en la implementación se tuvo problemas en las configuraciones de una máquina que funcionaría como cliente pues no contaba con los requerimientos del hardware necesario. Pese a las dificultades surgidas a lo largo del desarrollo del Proyecto, se puede manifestar que la implementación ha sido de éxito.

3.10 CRONOGRAMA DE ACTIVIDADES DE LA TESIS

Ver Otros Anexos, CRONOGRAMA DE ACTIVIDADES, Pág. 244

3.11 CONCLUSIONES

Al culminar el proyecto de investigación realizado para la Asociación Volcano Flowers, se pone a consideración del lector las siguientes conclusiones:

- El marco teórico utilizado en la presente tesis está basado en los argumentos que se emplearon para el desarrollo del Software Volcano Control, el mismo que está centrado en la problematización surgida, origen de la investigación, el cual evita desviaciones del planteamiento original.
- Con la Implementación del Software Volcano Control se ha logrado llevar el control de la información que se maneja y obtiene de todas y cada una de las actividades de producción que se llevan acabo dentro de la Asociación Volcano Flowers; al cubrir todas las necesidades en la florícola se ha constituido en una herramienta de trabajo primordial tanto para los trabajadores, empleados, proveedores y Asociados.
- Se identificó todas las necesidades de la florícola, para construir el Software mencionado y así disminuir la incorrecta manipulación de la información y mejorar la calidad de producción y la optimización de recursos para elevar la calidad del trabajo.
- Con el uso de herramientas informáticas adecuadas se ha desarrollado un Software de interfaz sencilla y amigable para el usuario, con gran capacidad de almacenamiento y precisión en los datos.
- Se obtuvo un control mejorado del manejo del aplicativo en lo que concierne a las seguridades del sistema.
- Se ha capacitado al personal que manipulará el sistema directamente así como también se han entregado los manuales correspondientes para su auto capacitación y pronto entendimiento del sistema implementado.

3.12 RECOMENDACIONES.

Se recomienda lo siguiente:

- ✓ Plantear un proyecto que sirva como alternativa de solución y no como un nuevo problema.
- ✓ Crear una correcta sustentación teórica con las definiciones pertinentes en el caso.
- ✓ El Software implementado, puede ser complementado con nuevas aplicaciones si la Asociación lo considera conveniente para cubrir una necesidad.
- ✓ Optimizar recursos de Hardware y Software a fin de no perder el tiempo.
- ✓ Hacer uso de las seguridades implementadas en un Software, para disminuir el uso inescrupuloso del Software y garantizar el buen rendimiento y confiabilidad del sistema.
- ✓ Tomar en cuenta las necesidades, sugerencias y comentarios de los empleados, trabajadores, directivos y asociados.
- ✓ Tener en cuenta los aspectos que dificultan la realización de una Investigación aplicada.

3.13 REFERENCIAS BIBLIOGRÁFICAS

BIBLIOGRAFÍA CITADA

CASTRO, F. Lorenzo; UCLM-ESI; Modelos de Datos; México, 1999/2000
COVIC, Análisis y Diseño Software, (2001, p 42-51)
ENCICLOPEDIA Curso de Informática Personal (2002, p 29 - 30)
FREEDMAN, Diccionario de Computación, 5ta. Edición (2000, p227)
LARRY LONG, Sistemas Informáticos, Pág. 300-317
Whitten J., Bentley L., Barlow V. 1996, Análisis y Diseño de Sistemas de Información, 3era. Edición. Pág: 95

BIBLIOGRAFÍA VIRTUAL CITADA

http://es.wikipedia.org/wiki/Base_de_datos
http://es.wikipedia.org/wiki/Sistema_de_control
[http://msdn2.microsoft.com/es-es/library/fx6bk1f4\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/fx6bk1f4(VS.80).aspx)
<http://www.geocities.com/trescapas/capadenegocios.htm>
<http://www.geocities.com/trescapas/capadepresentacion.htm>
<http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>
<http://www.microsoft.com/latam/sql/evaluation/features/default.asp>
<http://www.portalplanetasedna.com.ar/computacion2.htm>

BIBLIOGRAFÍA CONSULTADA

Análisis Y Diseño De Base De Datos
Análisis y Diseño de Sistemas de Información
Informática Personal, Introducción a la Informática.
Metodología De La Investigación
Microsoft Encarta 2007. 1993-2006 Microsoft Corporation.
Modelos y Sistemas, Diagrama de Flujo de Datos.
Pasos Para Elaborar Proyectos y Tesis de Investigación
Programación en ASP.NET

BIBLIOGRAFÍA VIRTUAL CONSULTADA

http://64.233.169.104/search?q=cache:lzgHcDIV3CEJ:www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf+en+que+consiste+la+metodolog%C3%ADa+xp&hl=es&ct=clnk&cd=3&gl=ec

<http://albertolacalle.com/hci/interfaz.htm>

<http://elies.rediris.es/elies12/cap12.htm>

http://es.wikipedia.org/wiki/Base_de_datos

http://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas

<http://homepages.mty.itesm.mx/al792774/programacionExtrema/ProgramacionExtrema.doc>

<http://html.rincondelvago.com/el-ciclo-de-vida-del-software.html>

<http://oness.sourceforge.net/proyecto/html/ch03s02.html>

<http://wigahluk.wordpress.com/2007/06/14/percepciones-acerca-de-la-programacion-agil-y-xp/>

<http://www.agapea.com/Visual-Studio-NET-n10670i.htm>

<http://www.alegsa.com.ar/Dic/ciclo%20de%20vida%20de%20un%20sistema%20informatico.php>

http://www.docirs.cl/arquitectura_tres_capas.htm

<http://www.esxoops.com/modules/news/article.php?storyid=311>

<http://www.geocities.com/studioxl/qlp.htm>

<http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node10.html>

<http://www.infoagro.com/flores/flores/clavel.htm>

<http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>

<http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml>

http://www.lugli.org.ar/mediawiki/index.php/Programacion_Extrema

<http://www.mastermagazine.info/termino/4896.php>

<http://www.microsoft.com/spain/sql/productinfo/overview/what-is-sql-server.msp>

<http://www.monografias.com/trabajos13/sisteinf/sisteinf.shtml>

3.14 GLOSARIO DE ABREVIACIONES

A:

ACE.- Asociación Camino a la Esperanza

ASP.- Active Server Pages, Páginas de Servidor Activas.

AVF.- Asociación Volcano Flowers

C:

CCF.- CHRISTIAN CHILDREN FUND INC

CPU.- Central Processing Unit, Unidad Central de Procesamiento

D:

DBMS.- Database Management System.

DFD.- Diagrama de Flujo de Datos.

H:

HTML.- HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

I:

IBM.- International Business Machines, Empresa que fabrica y comercializa máquinas a nivel Internacional.

IDE.- "Integrated device Electronics" --Dispositivo con electrónica integrada

ISVs.- Vendedores Independientes de Software.

N:

NTFS.- New Technology File System. Sistema de Archivos de Windows NT.

O:

OLAP.- On-Line Analytical Processing. Procesamiento Analítico en Línea

P:

PC.- Personal Computer. Computador Personal.

R:

RAM.- Random-Access Memory. Memoria de Acceso Aleatorio

RDBMS.- Relational Data Base Management System. Sistema Administrador de Bases de Datos Relacionales.

RPM.- RedHat Package Manager

S:

SGBD.- Sistema de Gestión de Bases de Datos

SQL.- Structured Query Language. Lenguaje Estructurado de Consultas

SW.- SOFTWARE

X:

XML.- Extensible Markup Language, lenguaje de marcado ampliable o extensible.

3.15 GLOSARIO DE TÉRMINOS

A:

Acceso

Consulta, normalmente por teléfono, de una base de datos situada en un gran ordenador

Aplicación

Es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar un o diversos tipos de trabajo.

Automatizar

Se le denomina así a cualquier tarea realizada por máquinas en lugar de personas. Es la sustitución de procedimientos manuales por sistemas de cómputo.

B:

Base de datos

Sistema de almacenamiento de datos muy flexible que permite organizar la información de forma muy eficiente.

C:

Capas

Son todos aquellos niveles software que se sitúan entre el usuario y los dispositivos físicos de la máquina, de forma que el nivel de complejidad para la utilización de las capas o niveles aumenta conforme nos acercamos al hardware.

Cliente

Aquel proceso o dispositivo solicitante que realiza la petición de un servicio o trabajo a otro proceso denominado 'servidor'.

Código

Aquel que puede moverse fácilmente de un tipo de máquina a otra, aunque cambien las características internas de la misma.

Compilador

Es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar.

Consistencia

Coherencia en la aplicación de reglas y procedimientos prescriptos respecto a la estructuración de un sistema de clasificación y/o de un sistema de recuperación de la información.

D:**Dato**

Información en un formato que pueda ser procesado por un ordenador. La información se condensa digitalmente, de modo que un texto, imagen o sonido se pueda representar en la pantalla.

Desarrollo

Trabajo sistemático que utiliza conocimiento obtenido a partir de la investigación (aplicada) y de la experiencia práctica y que se encuentra dirigido a la producción de nuevos materiales, productos, procesos, sistemas y servicios o a la mejora sustancial de aquellos ya producidos, ofrecidos o instalados.

Diseño

Es el proceso de definición de la arquitectura software: componentes, módulos, interfaces, procedimientos de prueba y datos de un sistema, que se crean para satisfacer unos requisitos especificados.

Documentar

Compilación continua y sistémica de un proceso de acopio de información con propósito específico, clasificación y procesos de escritura en un documento.

E:

Ejecutar

Introducir un mandato en el ordenador con el fin de que éste realice una operación determinada.

Estándar

Son una serie de lineamientos técnicos detallados, destinados a establecer uniformidad en el desarrollo de programas (software) y compra de equipos (hardware).

F:

Fichero

Un **archivo informático** o **fichero** es un conjunto de bits almacenado en un dispositivo periférico.

I:

Implantación

Etapas del ciclo de vida de una solución, en la que un escenario de implementación se convierte en un diseño de implementación.

Integridad

Se salvaguarda la exactitud y totalidad de la información y los métodos de procesamiento y transmisión.

Interfaz

De un objeto es la parte de un objeto accesible desde su exterior, que nos permite utilizarlo y consultar su estado interno. La vamos a utilizar como medio de comunicación entre el objeto y cualquier ente que quiera comunicarse con él.

Intérprete

Es un programa informático capaz de analizar y ejecutar otros programas, escritos en un lenguaje de alto nivel. Los intérpretes se diferencian de los compiladores en que mientras estos traducen un programa desde su descripción en un lenguaje de programación al código de máquina del sistema, los primeros (los intérpretes) sólo realizan la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.

Interrelación

Es la asociación o conexión entre conjuntos de entidades.

Inventario

Proceso de identificación y categorización de los recursos de información de modo sistemático.

M:

Modelado

Es una técnica cognitiva que consiste en crear una representación ideal de un objeto real mediante un conjunto de simplificaciones y abstracciones, cuya validez se pretende constatar.

P:

Paradigma

El paradigma es el método que usa el código de un lenguaje

Planificación

Es una disciplina de la inteligencia artificial que tiene por objeto la producción de planes

Proceso

Un **proceso** es un programa en ejecución.

Productividad

Rendimiento del proceso económico medido en unidades físicas o monetarias, por relación entre factores empleados y productos obtenidos.

R:

Recursos

Objetos o dispositivos que son utilizados por una computadora, para poder realizar todos los trabajos y tareas que se requieren a partir de las peticiones que van realizando los procesos del sistema. Entre otros en una computadora tenemos como recursos los siguientes: memoria principal, dispositivos de memoria secundaria, y cualquier dispositivo periférico direccionable. Además estos pueden ser o no **compartibles**.

Redundancia

Se llama redundancia al hecho de que los mismos datos estén almacenados más de una vez en la base de datos.

Respaldo

Referente al equipo y a los procedimientos a los sistemas de computación o las bases de datos donde se produce una copia y se almacena en el caso de que se necesite recuperar información.

S:

Servidor

Proceso o dispositivo encargado de resolver o solucionar la petición de servicio o trabajo realizado por un proceso denominado 'cliente'. Además debe dar una respuesta con el resultado de la consulta.

Software

Los programas que se usan para dirigir las funciones de un sistema de computación.

U:

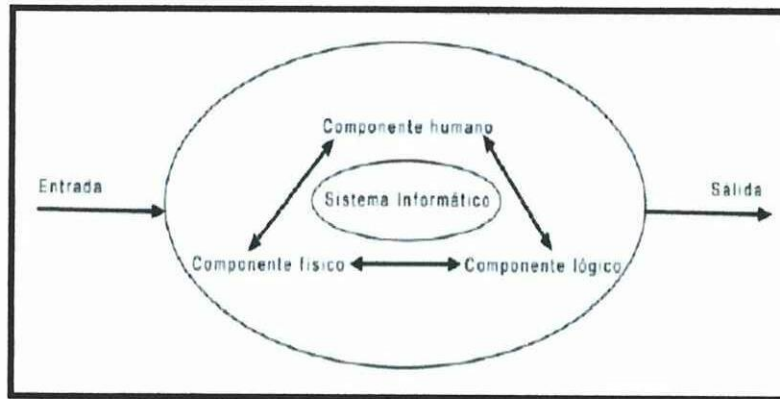
Usuario

Individuo que introduce datos a la computadora o que trabaja con ella.

ANEXOS

ANEXO DE GRÁFICOS

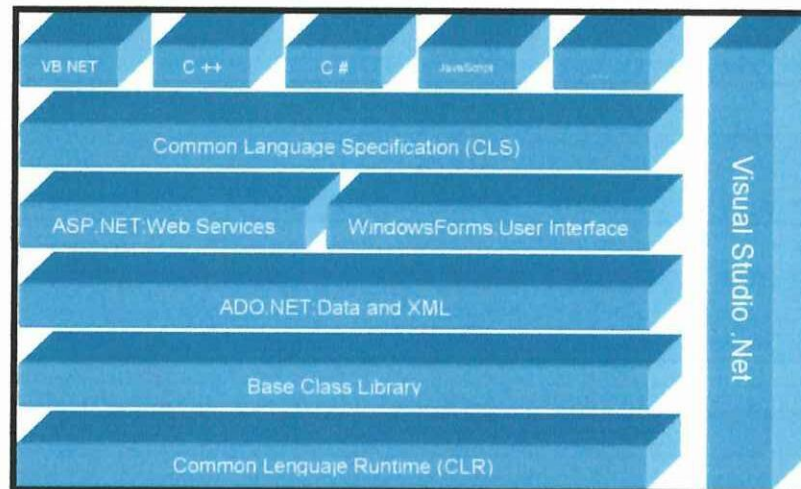
GRAFICO Nº 1.1: ESTRUCTURA DE UN SISTEMA INFORMÁTICO



FUENTE: <http://www.um.es/docencia/barzana/IAGP/Iagp1.html>

REALIZADO POR: Grupo Investigador

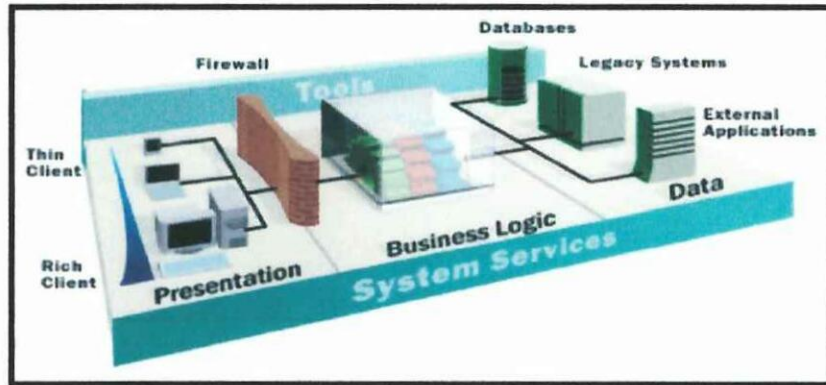
GRAFICO Nº 1.2: COMPONENTES DE MICROSOFT .NET



Fuente: Ayuda MSDN; de Microsoft Visual Studio. Net

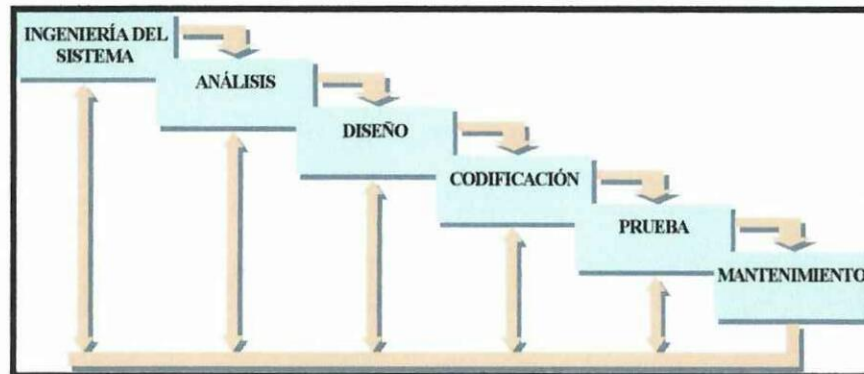
REALIZADO POR: Grupo Investigador

GRAFICO Nº 1.3: ARQUITECTURA TRES CAPAS EN TECNOLOGÍA CLIENTE/SERVIDOR



FUENTE: <http://www.um.es/docencia/barzana/IAGP/Iagp1.html>
REALIZADO POR: Grupo Investigador

GRÁFICO Nº 1.4: CICLO DE VIDA EN CASCADA



FUENTE: Modelo extraído de: (Larman 2003) Larman, C. "UML y patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado". Segunda Edición. Ed. Prentice-Hall. Madrid, 2003. Apartado 37.5.
REALIZADO POR: Grupo Investigador

GRÁFICO 2.1. POBLACIÓN Y MUESTRA



GRÁFICO 2.2. PREGUNTA 1

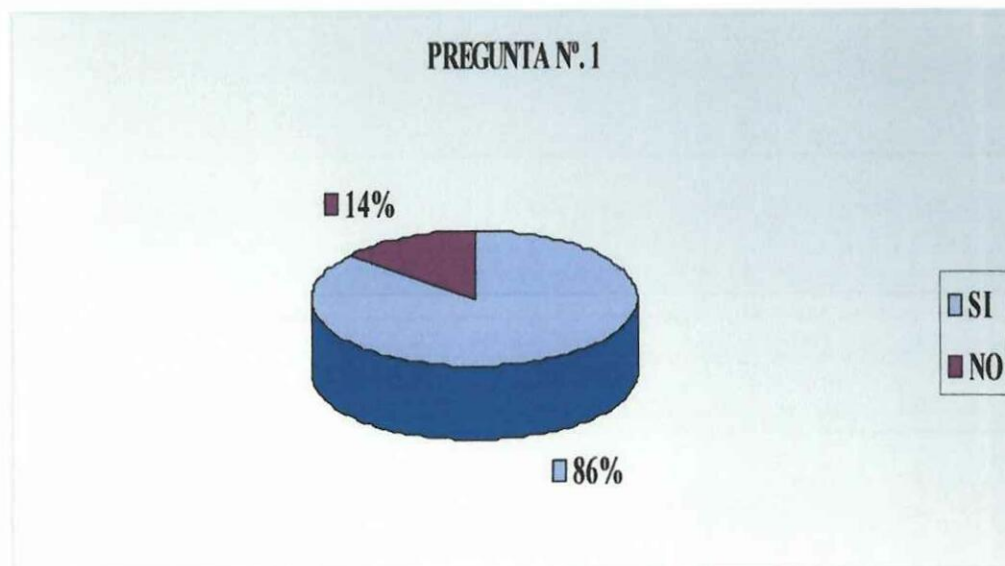


GRÁFICO 2.3 PREGUNTA 2

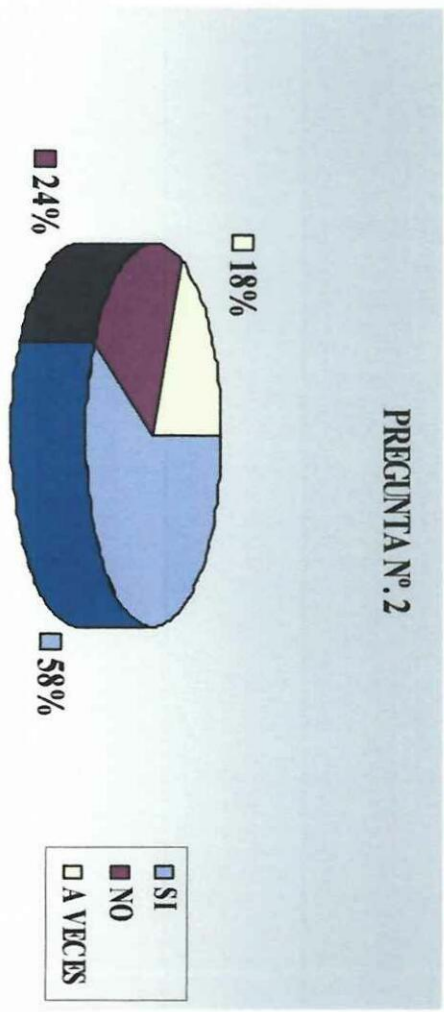


GRÁFICO 2.4 PREGUNTA 3

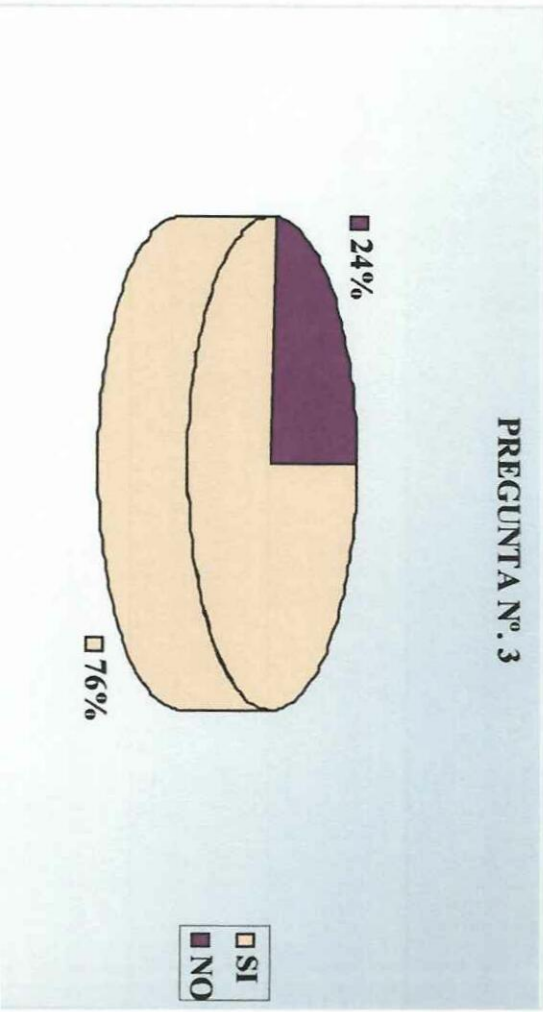


GRÁFICO 2.5 PREGUNTA 4

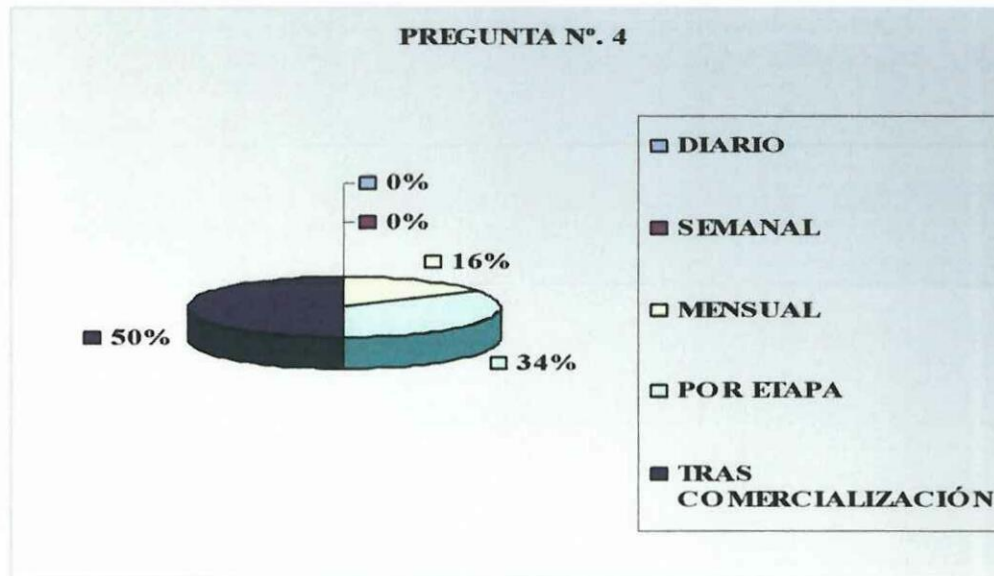


GRÁFICO 2.6 PREGUNTA 5

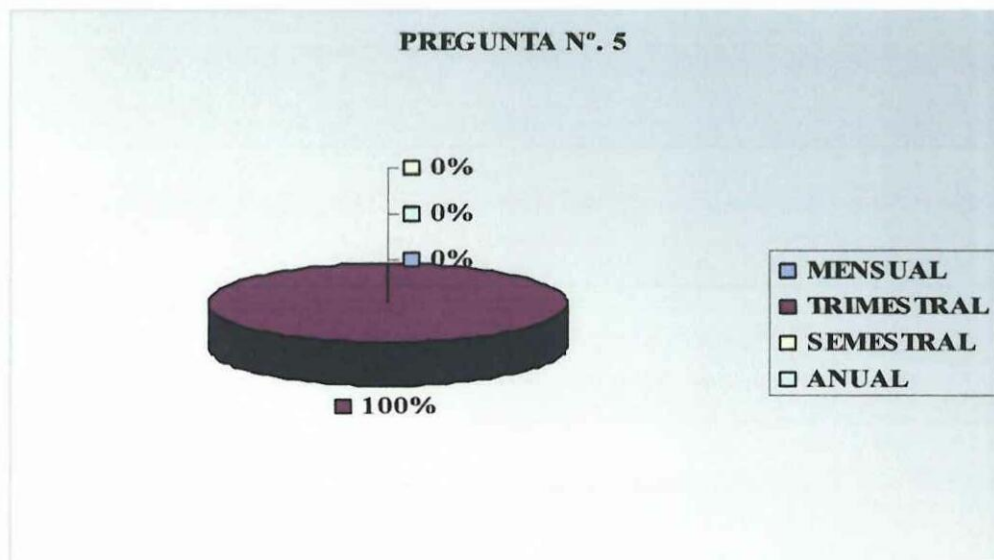


GRÁFICO 2.7 PREGUNTA 6

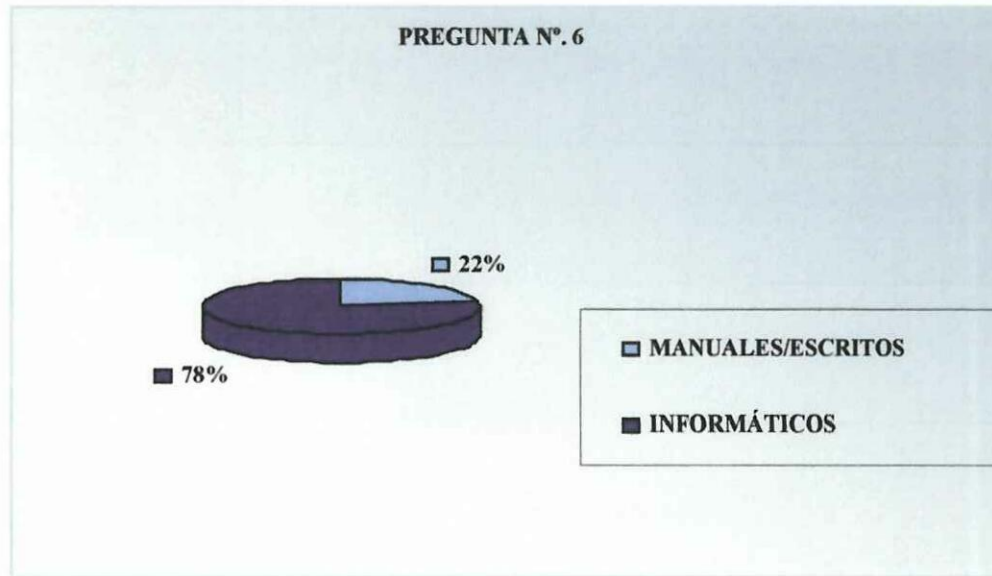


GRÁFICO 2.8 PREGUNTA 7

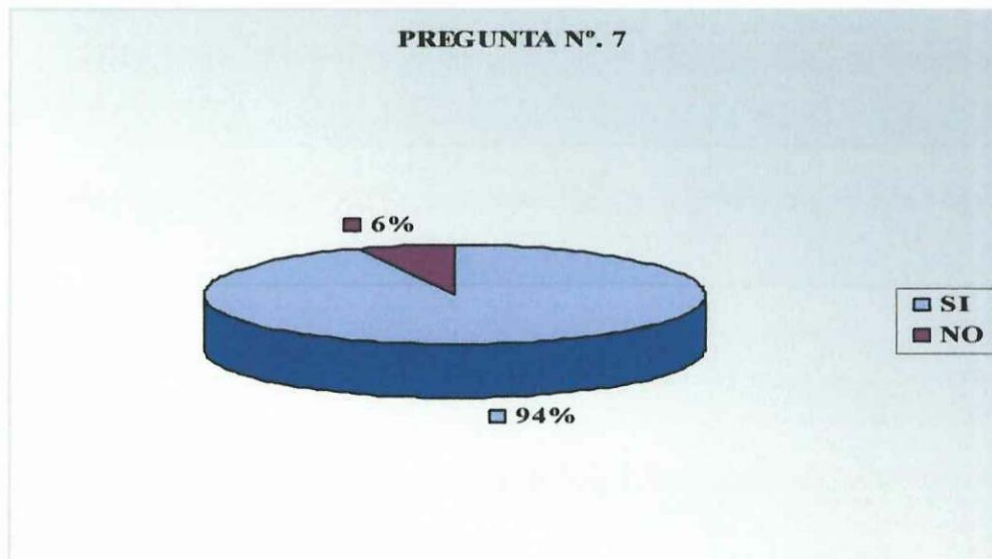


GRÁFICO 2.9 PREGUNTA 8

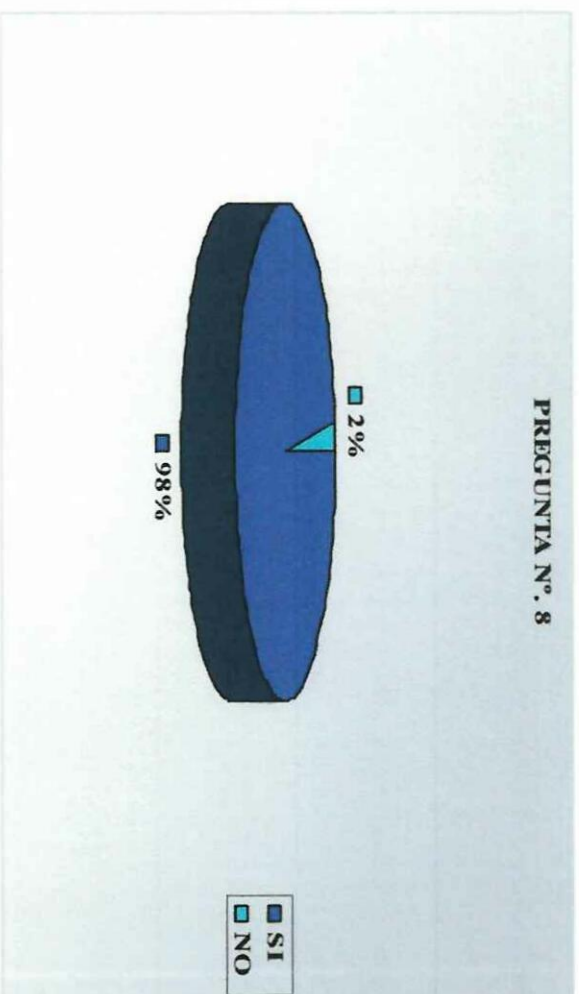


GRÁFICO 2.10 PREGUNTA 9

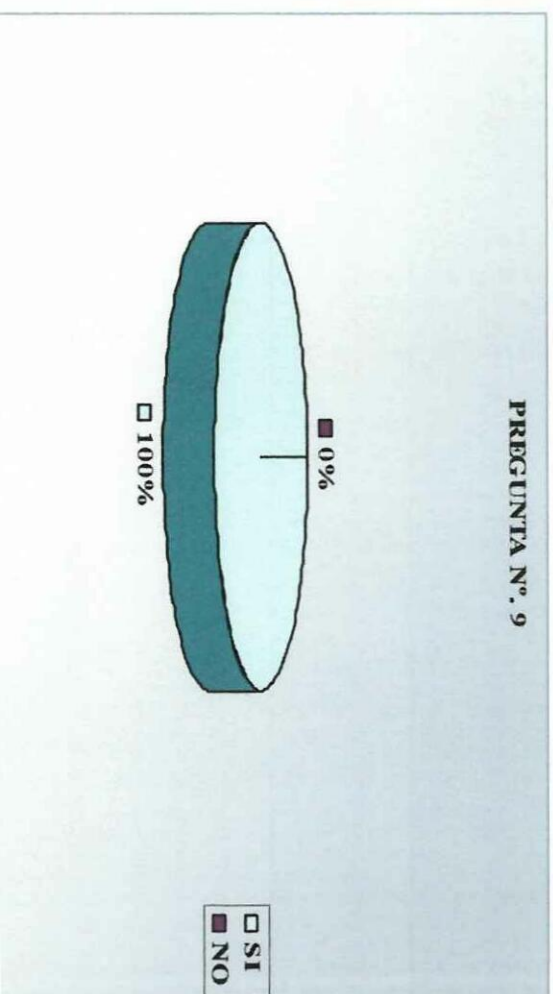


GRÁFICO 2.11 PREGUNTA 10

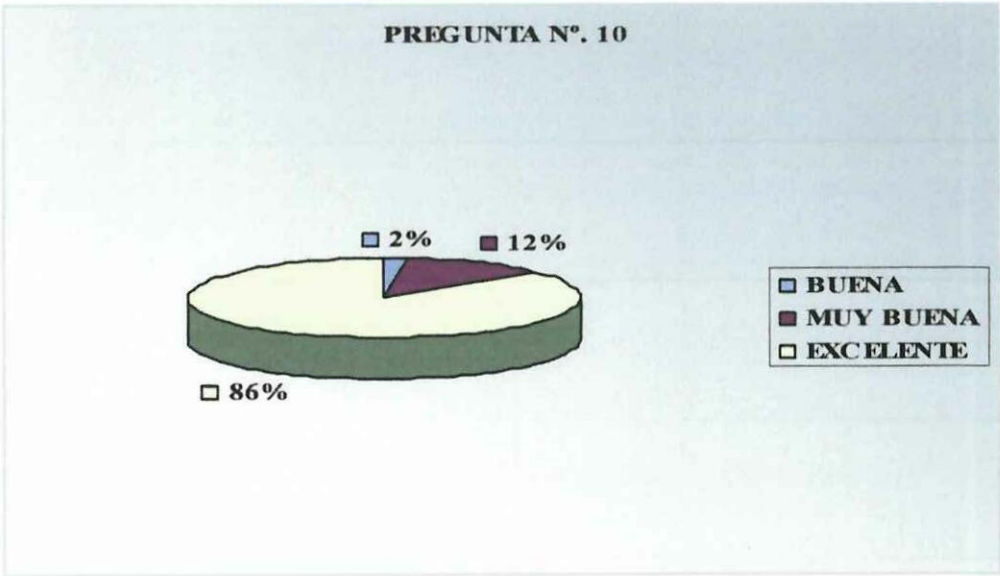


GRÁFICO 3.1 DIAGRAMA DE FLUJO DE DATOS

DIAGRAMA DE CONTEXTO

NIVEL 0

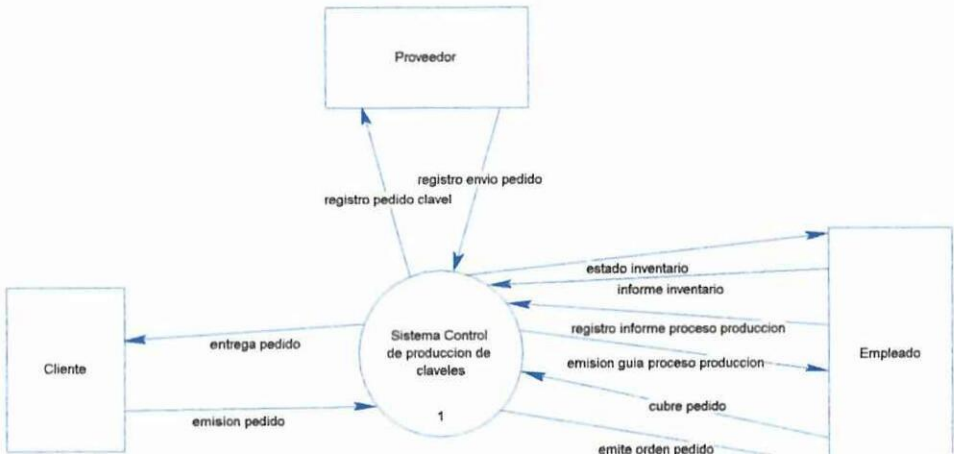


GRÁFICO 3.2 DIAGRAMA DE CASOS DE USO

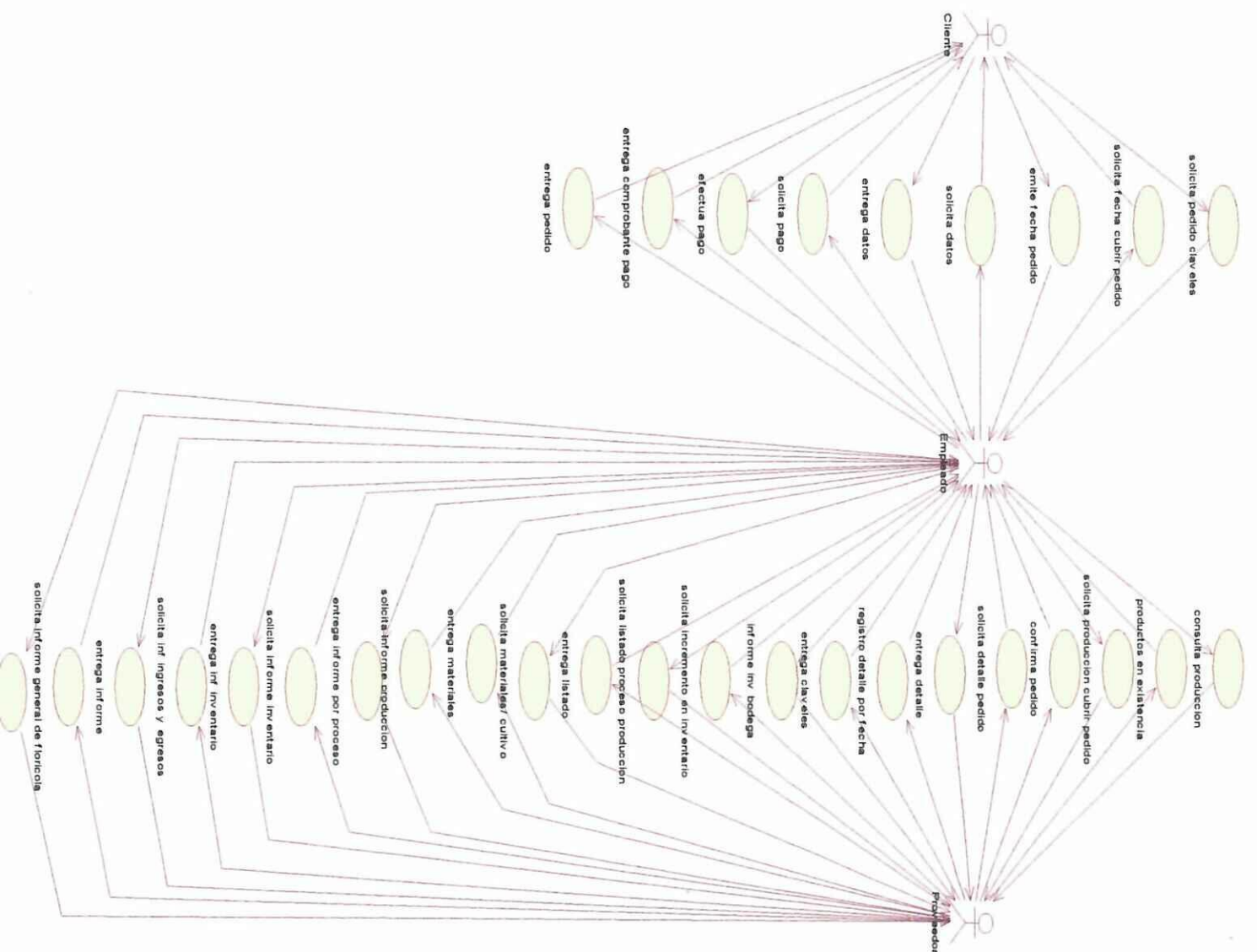
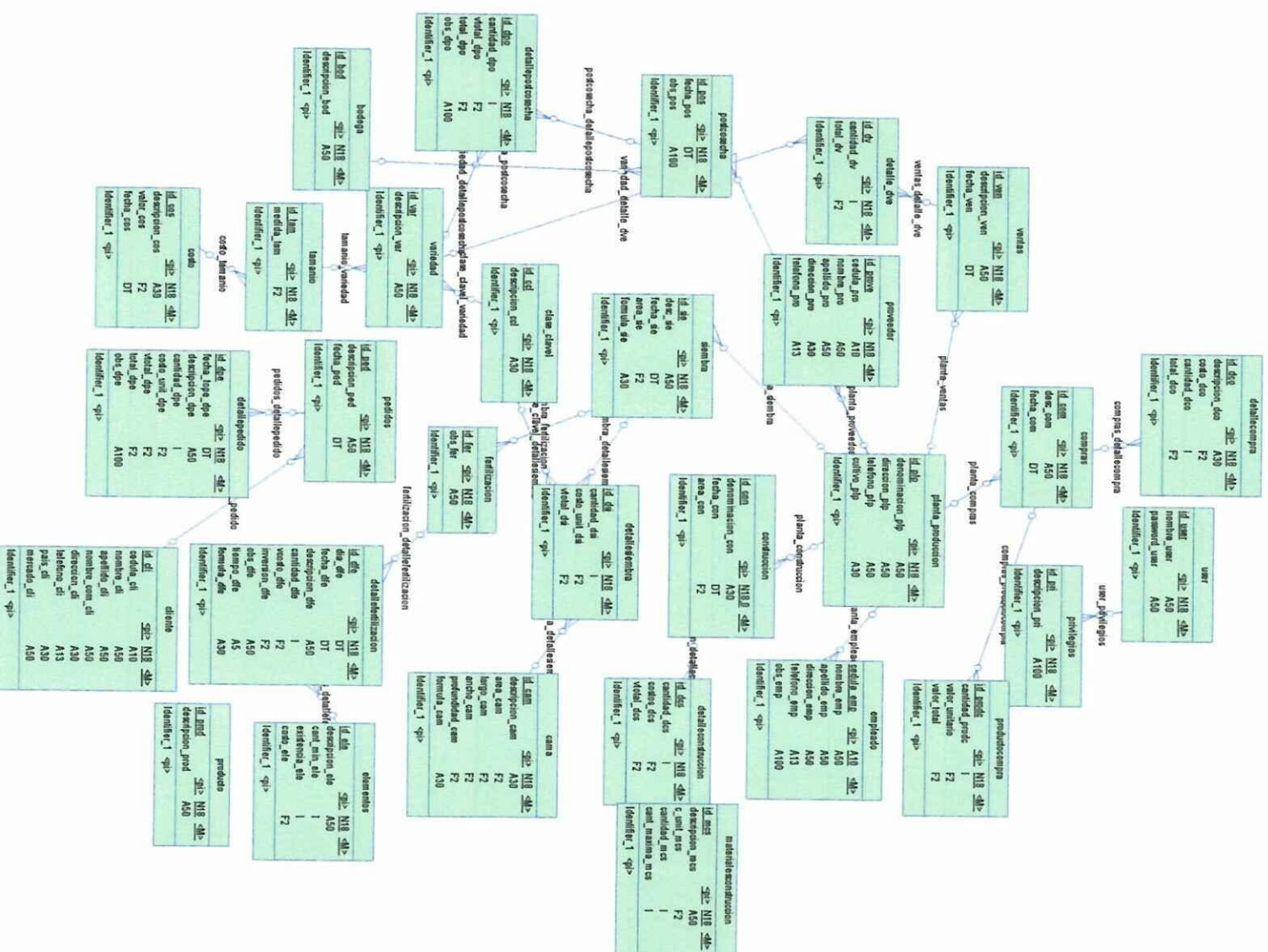
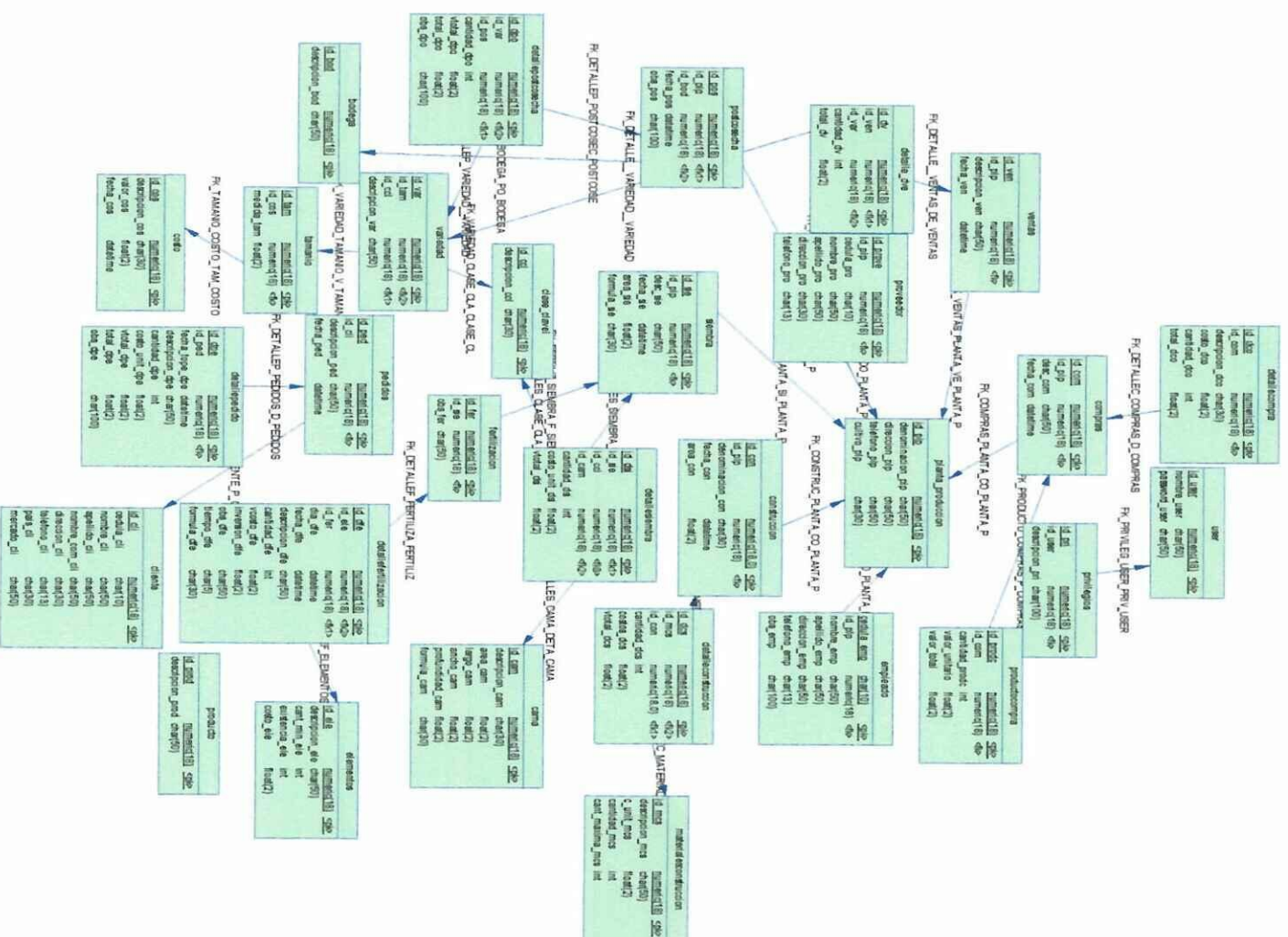


GRÁFICO 3.5 MODELO CONCEPTUAL DE DATOS



MODELO FÍSICO DE DATOS



ANEXO DE TABLAS

Tabla 2.1. Listado de Asociados Fundadores

NOMINA DE FUNDADORES			
No	APELLIDOS Y NOMBRES	C.I.	DIRECCIÓN
1	AÑARUMBA CONDOR DORIS GABRIELA	0503105066	PATUTAN S.
2	AÑARUMBA CAIZA PEDRO PABLO	0502451891	PATUTAN S. No 5
3	AÑARUMBA CONDOR CRISTIAN VINICIO	0503290140	PATUTAN S.
4	AÑARUMBA TOAPANTA LUIS ALBERTO	0501645089	PATUTAN S. No 3
5	CAISALUISA AÑARUNBA MARIA ROSA	0502332265	PATUTAN S. No 3
6	CHANGOLUISA CHINGO JOSÉ MARIA	0500813373	PATUTAN S. No 4
7	CHANGOLUISA TOAPANTA JORGE HUMBERTO	0502553175	PATUTAN S. No 3
8	CHANGOLUISA TOAPANTA LUIS ALEJANDRO	0502114705	PATUTAN S. No 3
9	CHANGOLUISA TOAPANTA LUIS FABIAN	0502333362	PATUTAN S. No 3
10	CHANGOLUISA TOAPANTA LUIS FERNANDO	0503043549	PATUTAN S.
11	CHINGO CANDO RENE PATRICIO	0502787310	PATUTAN S. No 3
12	CHINGO CHINGO MARIA NICOLASA	0500585643	PATUTAN S.
13	CHINGO LAMINGO GLADYS ERNESTINA	0502568017	PATUTAN S. No 3
14	CHINGO TIGLLA MARIA BLANCA	0502814072	PATUTAN S.
15	CONDOR LUZ MARIA IRALDA	0501401228	PATUTAN S.
16	LLANGO AÑARUNBA LUIS FRANKLIN	0502577067	PATUTAN S.
17	CONDOR LUISA MARIA	0502333362	PATUTAN S.

18	MAIGUA TIGLLA JULIO FANCISCO	0502734684	PATUTAN S. No 3
19	OBANDO CHINGO EDGAR MAURICIO	0502195571	PATUTAN S.
20	PACHECO MOLINA FREDY ORLANDO	0502293905	PATUTAN S. No 1
21	QUILUMBA GUANOLUISA MIRIAN PATRICIA	0502811524	PATUTAN S. No 2
22	TIGLLA MAIGUA MARIA	0502208838	PATUTAN S. No 3
23	TIGLLA YANCHAGUANO LUIS ANTONIO	0502326325	PATUTAN S.
24	TOAPANTA TOAPANTA MARIA ESTHER	0501466619	PATUTAN S.
25	YANCHAGUANO AREQUIPA NORMA ISABEL	0502595697	PATUTAN S.
26	YANCHATIPAN YANCHATIPAN FANNY MARISOL	0502575459	PATUTAN S.
27	YAULE CHANGOLUISA MARIA GLADYS	0502731292	PATUTAN S.
28	YAULE CHINGO BLANCA CECILIA	0502490469	PATUTAN S.
29	YAULE CHINGO LUIS NESTOR	1719552232	PATUTAN S.
30	YAULE TOAPANTA JUAN CARLOS	0503728701	PATUTAN S.
31	YAULE TOAPANTA LUIS MATIAS	0503073702	PATUTAN S. No 3
32	YAULE YANCHAGUANO JOSE FRANCISCO	0500905948	PATUTAN S.
33	YAULE YANCHAGUANO JOSE MANUEL	0500932868	PATUTAN S. No 3
34	YAULI CHANGOLUISA MAYRA ALEXANDRA	0503038226	PATUTAN S.

Tabla 2.2. Población y Muestra

SUJETO	NUMERO
Presidente de la Asociación	1
Tesorero	1
Secretario	1
Asociados y proveedores	20
Trabajadores	30
Total:	53

Tabla 2.3. Pregunta 1

SI	NO	TOTAL
43	7	50

Tabla 2.4. Pregunta 2

SI	NO	A VECES	TOTAL
29	12	9	50

Tabla 2.5. Pregunta 3

SI	NO	TOTAL
43	7	50

Tabla 2.6. Pregunta 4

DIARIO	SEMANAL	MENSUAL	POR ETAPA	TRAS COMERCIALIZACIÓN	TOTAL
0	0	8	17	25	100

Tabla 2.7. Pregunta 5

MENSUAL	TRIMESTRAL	SEMESTRAL	ANUAL	TOTAL
0	50	0	0	50

Tabla 2.8. Pregunta 6

MANUALES/ESCRITOS	INFORMÁTICOS	TOTAL
11	39	50

Tabla 2.9. Pregunta 7

SI	NO	TOTAL

Tabla 2.10. Pregunta 8

SI	NO	TOTAL
49	1	50

Tabla 2.11. Pregunta 9

SI	NO	TOTAL
50	0	50

Tabla 2.12. Pregunta 10

BUENA	MUY BUENA	EXCELENTE	TOTAL
1	6	43	50

Tabla 2.13. MATERIALES

NOMBRE	CANTIDAD	VALOR UNITARIO	TOTAL
Fotocopias	200	\$0.10	\$20.00
CD's	100	\$0.50	\$50.00
Alquiler de computador	200 h	\$1.00	\$200.00
Papel/resmas	5	\$6.00	\$30.00
impresiones	1000	\$0.15	\$150.00
Empastado	5	\$20.00	\$100.00
		TOTAL:	\$550.00

Tabla 2.14. DESGLOSE DEL RECURSO TÉCNICO

NOMBRE	CANTIDAD	VALOR UNITARIO	TOTAL
Asesoramiento técnico	2	\$100	\$200.00
Uso de Internet/horas	200	\$1	\$200.00
Capacitación externa/hrs.	2	\$100	\$200.00
		TOTAL:	\$600.00

Tabla 2.15. COSTOS DIRECTOS: DETALLE DE COSTOS

DESCRIPCION	TOTAL
MATERIALES	\$550.00
TECNICOS	\$600.00
TOTAL:	\$1.150.00

Tabla 2.16. COSTOS INDIRECTOS: DETALLE DE COSTOS

DESCRIPCION	TOTAL
TRANSPORTE	\$200.00
ALIMENTACIÓN	\$200.00
TOTAL:	\$ 400.00

Total = Costos directos + Costos indirectos + 10% de imprevistos

Total = \$1.550.00 + \$400.00 + \$195.00

Total = \$2.145.00

OTROS ANEXOS

Formato de encuesta

ENCUESTA PARA ASOCIADOS, PROVEEDORES Y TRABAJADORES DE LA ASOCIACIÓN VOLCANO FLOWERS

PROVINCIA: COTOPAXI
CANTÓN: LATACUNGA
PARROQUIA: ELOY ALFARO
BARRIO: PATUTÁN
ASOCIADO ☐ PROVEEDOR ☐ TRABAJADOR ☐

NOTA: Lea las preguntas detenidamente y marque con una X la respuesta que Ud estime conveniente.

1. ¿Considera Ud que la Asociación Volcano Flowers es una empresa florícola creada para ser una fuente de trabajo única beneficiara del barrio Patután?
SI ☐ NO ☐
2. ¿Cree Ud que de alguna forma las actividades que se realizan dentro de la AVF generan múltiples problemas debido a la forma en la que se lleva el control de la producción?
SI ☐ NO ☐ A VECES ☐
3. ¿Piensa Ud que el control que actualmente se lleva sobre la producción de claveles es el más adecuado?
SI ☐ NO ☐
4. ¿Con que frecuencia se presentan reportes o informes acerca de la información que se toma en cuenta en la producción de claveles?
DIARIO ☐ POR ETAPA ☐
SEMANAL ☐ TRAS COMERCIALIZACIÓN ☐

MENSUAL ☐

5. ¿Por lo general en qué lapso de tiempo se comercializa el clavel?

MENSUAL ☐ TRIMESTRAL ☐ SEMESTRAL ☐

ANUAL ☐

6. ¿En el caso de que toda la producción fuera de su responsabilidad que medio emplearía para llevar el control de la misma?

MANUALES / ESCRITOS ☐ INFORMÁTICOS ☐

7. ¿Está Ud de acuerdo en que se lleve el control de la producción de claveles en forma automática empleando un computador?

SI ☐ NO ☐

8. ¿Cree Ud que el registro de toda la información con proyecciones a futuro en forma automática permitirá establecer un régimen de pérdidas y ganancias desde la elaboración de un proyecto de cultivo de claveles?

SI ☐ NO ☐

9. Cree UD. que con la Implementación de un Sistema Informático se logrará mejorar la producción dentro de la AVF

SI ☐ NO ☐

10. Que calificación proporcionaría Ud a la posibilidad de que la AVF implemente un programa de automatización de toda la plantación con la única finalidad de proporcionar una alternativa de solución de problemas originados dentro de la misma.

BUENA ☐ MUY BUENA ☐ EXCELENTE ☐

BASE DE DATOS

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLECOMPRA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLECOMPRA](
    [ID_DCO] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_COM] [numeric](18, 0) NULL,
    [DESCRIPCION_DCO] [varchar](30) NULL,
    [COSTO_DCO] [real] NULL,
    [CANTIDAD_DCO] [int] NULL,
    [TOTAL_DCO] [real] NULL,
    CONSTRAINT [PK_DETALLECOMPRA] PRIMARY KEY CLUSTERED
(
    [ID_DCO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminarCompra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminarCompra]
(
    @CodigoCompra numeric
)
AS
DELETE FROM COMPRAS
WHERE ID_COM=@CodigoCompra'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[PRODUCTO]') AND type in (N'U'))
BEGIN
```

```

        [ID_PROD] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
        [DESCRIPCION_PROD] [varchar](50) NOT NULL,
CONSTRAINT [PK_PRODUCTO] PRIMARY KEY CLUSTERED
(
    [ID_PROD] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[PRODUCTO_COMPRA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PRODUCTO_COMPRA](
    [ID_COM] [numeric](18, 0) NOT NULL,
    [ID_PROD] [numeric](18, 0) NOT NULL,
    [CANTIDAD_PROD] [int] NOT NULL,
    [VALOR_UNITARIO] [float] NOT NULL,
    [VALOR_TOTAL] [float] NOT NULL,
CONSTRAINT [PK_PRODUCTO_COMPRA] PRIMARY KEY CLUSTERED
(
    [ID_COM] ASC,
    [ID_PROD] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[COSTO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[COSTO](
    [ID_COS] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_COS] [varchar](30) NOT NULL,
    [VALOR_COS] [float] NOT NULL,
    [FECHA_COS] [datetime] NOT NULL,
CONSTRAINT [PK_COSTO] PRIMARY KEY CLUSTERED

```

```

)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoTamanio]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoTamanio]
(
    @CodigoCosto int,
    @DescripcionTamanio as varchar(50),
    @Medida as float
)
AS
INSERT INTO Tamanio(ID_COS,DESCRIPCION_TAM,MEDIDA_TAM)
VALUES(@CodigoCosto,@DescripcionTamanio,@Medida)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaTamanio]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaTamanio]
(
    @CodigoTamanio as int,
    @CodigoCosto as int,
    @DescripcionTamanio as varchar(50),
    @Medida as float
)
AS
UPDATE Tamanio SET
ID_COS=@CodigoCosto,
DESCRIPCION_TAM=@DescripcionTamanio,
MEDIDA_TAM=@Medida

```



```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaTamanio]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaTamanio]
    @CodigoTamanio as numeric
AS
DELETE FROM Tamanio
WHERE ID_TAM=@CodigoTamanio
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaVenta]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaVenta]
(
    @CodigoVenta numeric
)
AS
DELETE FROM VENTAS
WHERE ID_VEN=@CodigoVenta'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaDetalleVenta]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaDetalleVenta]
(

```

```
@CodigoVentas numeric,  
@CantidadVentaDetalle integer,  
@DescripcionVentaDetalle varchar(30),  
@CostoVentaDetalle float,  
@TotalVentaDetalle float  
)
```

AS

UPDATE DETALLE_DVE

SET ID_VEN=@CodigoVentas,

CANTIDAD_DV=@CantidadVentaDetalle,

DESCRIPCIÓN_DV=@DescripcionVentaDetalle,

COSTO_DV=@CostoVentaDetalle,

TOTAL_DV=@TotalVentaDetalle

WHERE ID_DV=@CodigoDetalleVenta

,

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaSiembra]') AND type in (N'P', N'PC'))

BEGIN

EXEC dbo.sp_executesql @statement = N'

CREATE PROCEDURE [dbo].[SP_EliminaSiembra]

@CodigoSiembra int=0

AS

delete from Siembra

where ID_SIE=@CodigoSiembra

,

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresarUsuario]') AND type in (N'P', N'PC'))

BEGIN

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE

[dbo].[SP_IngresarUsuario]

```
@Contrasenia varchar(50)
)
```

AS

```
INSERT INTO USUARIO (NOMBRE_USER,PASSWORD_USER)
VALUES (@Nombre,@Contrasenia)
```

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizarUsuario]') AND type in (N'P', N'PC'))
```

BEGIN

```
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizarUsuario]
```

```
(
    @Codigo as integer,
    @Nombre varchar(50),
    @Contrasenia varchar(50)
)
```

AS

```
update USUARIO
set NOMBRE_USER=@Nombre,
PASSWORD_USER=@Contrasenia
WHERE ID_USER=@Codigo
```

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListarUsuario]') AND type in (N'P', N'PC'))
```

BEGIN

```
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListarUsuario]
```

```
(
    @Nombre varchar(50)=""
```

```

        SELECT ID_USER, NOMBRE_USER, PASSWORD_USER FROM
USUARIO
        WHERE NOMBRE_USER LIKE "%" + @Nombre + "%"
    ,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminarUsuario]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminarUsuario]
(
    @Codigo integer
)
AS
    DELETE FROM USUARIO
    WHERE ID_USER=@Codigo
    ,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[USUARIO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[USUARIO](
    [ID_USER] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [NOMBRE_USER] [varchar](50) NOT NULL,
    [PASSWORD_USER] [varchar](50) NOT NULL,
    CONSTRAINT [PK_USUARIO] PRIMARY KEY CLUSTERED
(
    [ID_USER] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminarBodega]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminarBodega]

```

```

(
    @CodigoBodega numeric
)

```

```

AS

```

```

    delete from Bodega
    where ID_BOD=@CodigoBodega
,

```

```

END

```

```

GO

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[CLIENTE]') AND type in (N'U'))

```

```

BEGIN

```

```

CREATE TABLE [dbo].[CLIENTE](
    [ID_CLI] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [CEDULA_CLI] [varchar](10) NOT NULL,
    [NOMBRE_CLI] [varchar](50) NOT NULL,
    [APELLIDO_CLI] [varchar](50) NOT NULL,
    [NOMBRE_COM_CLI] [varchar](50) NOT NULL,
    [DIRECCION_CLI] [varchar](30) NOT NULL,
    [TELEFONO_CLI] [varchar](13) NULL,
    [PAIS_CLI] [varchar](30) NULL,
    [MERCADO_CLI] [varchar](50) NULL,
    CONSTRAINT [PK_CLIENTE] PRIMARY KEY CLUSTERED

```

```

(

```

```

    [ID_CLI] ASC

```

```

)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY]

```

```

END

```

```

GO

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaDescripcionBodega]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultaDescripcionBodega]
(
    @DescripcionBodega VARCHAR(50)
)
AS
SELECT DESCRIPCION_BOD,OBS_BOD
FROM Bodega
WHERE DESCRIPCION_BOD=@DescripcionBodega'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[BODEGA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[BODEGA](
    [ID_BOD] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_BOD] [varchar](50) NOT NULL,
    CONSTRAINT [PK_BODEGA] PRIMARY KEY CLUSTERED
(
    [ID_BOD] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[MENU]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[MENU](
    [ID_MENU] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_MENU] [varchar](100) NOT NULL,
    CONSTRAINT [PK_MENU] PRIMARY KEY CLUSTERED
(

```

```

) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[ELEMENTOS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[ELEMENTOS](
    [ID_ELE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_ELE] [varchar](50) NOT NULL,
    [CANT_MIN_ELE] [int] NOT NULL,
    [EXISTENCIA_ELE] [int] NOT NULL,
    [COSTO_ELE] [float] NOT NULL,
    CONSTRAINT [PK_ELEMENTOS] PRIMARY KEY CLUSTERED
(
    [ID_ELE] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[MATERIALESCONSTRUCCION]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[MATERIALESCONSTRUCCION](
    [ID_MCS] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_MCS] [varchar](50) NOT NULL,
    [C_UNIT_MCS] [float] NOT NULL,
    [CANTIDAD_MCS] [int] NOT NULL,
    [CANT_MAXIMA_MCS] [int] NOT NULL,
    CONSTRAINT [PK_MATERIALESCONSTRUCCION] PRIMARY KEY
CLUSTERED
(
    [ID_MCS] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO

```



```

GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListarCamas]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ListarCamas]
@DescripcionCama VARCHAR(30)='',
@AreaCama as float=0
AS
set @AreaCama= ISNULL(@AreaCama,0)
SET @DescripcionCama = isNull(@DescripcionCama, '')
SELECT
ID_CAM,DESCRIPCION_CAM,AREA_CAM,LARGO_CAM,ANCHO_CAM,
PROFUNDIDAD_CAM,FORMULA_CAM
FROM CAMAS
WHERE
CASE @DescripcionCama When '' then '' else DESCRIPCION_CAM end like
Case @DescripcionCama When '' then '' else "%"+ @DescripcionCama +"%" end
and
CASE @AreaCama When 0 then 0 else AREA_CAM end = Case @AreaCama When
0 then 0 else @AreaCama end
,

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[CLASE_CLAVEL]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[CLASE_CLAVEL](
[ID_CCL] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
[DESCRIPCION_CCL] [varchar](30) NOT NULL,
CONSTRAINT [PK_CLASE_CLAVEL] PRIMARY KEY CLUSTERED
(
[ID_CCL] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON

```

```

SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[CAMAS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[CAMAS](
    [ID_CAM] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_CAM] [varchar](50) NOT NULL,
    [AREA_CAM] [float] NOT NULL,
    [LARGO_CAM] [float] NOT NULL,
    [ANCHO_CAM] [float] NOT NULL,
    [PROFUNDIDAD_CAM] [float] NOT NULL,
    [FORMULA_CAM] [float] NULL,
    CONSTRAINT [PK_CAMAS] PRIMARY KEY CLUSTERED
(
    [ID_CAM] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaCama]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaCama]
(
    @CodigoCama numeric
)
AS
    DELETE FROM CAMAS
    WHERE ID_CAM=@CodigoCama
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =

```

```

CREATE TABLE [dbo].[PLANTA_PRODUCCION](
    [ID_PLP] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DENOMINACION_PLP] [varchar](50) NOT NULL,
    [DIRECCION_PLP] [varchar](50) NOT NULL,
    [TELEFONO_PLP] [varchar](13) NULL,
    [CULTIVO_PLP] [varchar](30) NOT NULL,
    CONSTRAINT [PK_PLANTA_PRODUCCION] PRIMARY KEY CLUSTERED
(
    [ID_PLP] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoCliente]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoCliente]
(
    @CedulaCliente varchar(10),
    @NombreCliente varchar(50),
    @ApellidoCliente varchar(50),
    @NombreComercialCliente varchar(50),
    @DireccionCliente varchar(30),
    @TelefonoCliente varchar(13),
    @PaisCliente varchar(30),
    @MercadoCliente varchar(50)
)

AS
INSERT INTO CLIENTE(
    CEDULA_CLI,      NOMBRE_CLI,
    APELLIDO_CLI,NOMBRE_COM_CLI,
    DIRECCION_CLI,TELEFONO_CLI,PAIS_CLI,MERCADO_CLI)
VALUES (@CedulaCliente,@NombreCliente,@ApellidoCliente,
    @NombreComercialCliente,@DireccionCliente,@TelefonoCliente,
    @PaisCliente,@MercadoCliente)
,
END

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaCamas]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ConsultaCamas]
@CodigoCama integer=0
AS
SELECT
ID_CAM,
DESCRIPCION_CAM,
AREA_CAM,
LARGO_CAM,
ANCHO_CAM,
PROFUNDIDAD_CAM,
FORMULA_CAM
FROM CAMAS
WHERE ID_CAM =@CodigoCama
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLEFERTILIZACION]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLEFERTILIZACION](
[ID_DFE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
[DESCRIPCION_DFE] [varchar](50) NOT NULL,
[ID_FER] [numeric](18, 0) NOT NULL,
[ID_ELE] [numeric](18, 0) NOT NULL,
[FECHA_DFE] [datetime] NOT NULL,
[CANTIDAD_DFE] [int] NOT NULL,
[VCOSTO_DFE] [float] NOT NULL,
[TOTAL_DFE] [float] NOT NULL,
CONSTRAINT [PK_DETALLEFERTILIZACION] PRIMARY KEY CLUSTERED
(
[ID_DFE] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]

```

```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLECONSTRUCCION]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLECONSTRUCCION](
    [ID_DCS] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_MCS] [numeric](18, 0) NOT NULL,
    [ID_CON] [numeric](18, 0) NOT NULL,
    [CANTIDAD_DCS] [int] NOT NULL,
    [COSTOS_DCS] [float] NOT NULL,
    [VTOTAL_DCS] [float] NOT NULL,
    CONSTRAINT [PK_DETALLECONSTRUCCION] PRIMARY KEY
    CLUSTERED
    (
        [ID_DCS] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[TAMANIO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[TAMANIO](
    [ID_TAM] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_COS] [numeric](18, 0) NOT NULL,
    [DESCRIPCION_TAM] [varchar](50) NOT NULL,
    [MEDIDA_TAM] [float] NOT NULL,
    CONSTRAINT [PK_TAMANIO] PRIMARY KEY CLUSTERED
    (
        [ID_TAM] ASC
    )WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[VARIEDAD]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[VARIEDAD](
    [ID_VARI] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_VAR] [varchar](50) NOT NULL,
    [ID_CCL] [numeric](18, 0) NOT NULL,
    [ID_TAM] [numeric](18, 0) NOT NULL,
    CONSTRAINT [PK_VARIEDAD] PRIMARY KEY CLUSTERED
(
    [ID_VARI] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLEPOSTCOSECHA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLEPOSTCOSECHA](
    [ID_DPO] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_VARI] [numeric](18, 0) NOT NULL,
    [ID_POS] [numeric](18, 0) NOT NULL,
    [CANTIDAD_DPO] [int] NOT NULL,
    [VTOTAL_DPO] [float] NOT NULL,
    CONSTRAINT [PK_DETALLEPOSTCOSECHA] PRIMARY KEY CLUSTERED
(
    [ID_DPO] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLE_DVE]') AND type in (N'U'))
BEGIN

```

```

[ID_DV] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
[ID_VEN] [numeric](18, 0) NOT NULL,
[CANTIDAD_DV] [int] NOT NULL,
[ID_VARI] [numeric](18, 0) NOT NULL,
[TOTAL_DV] [float] NOT NULL,
CONSTRAINT [PK_DETALLE_DVE] PRIMARY KEY CLUSTERED
(
    [ID_DV] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLEPEDIDO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLEPEDIDO](
    [ID_DPE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_PED] [numeric](18, 0) NOT NULL,
    [CANTIDAD_DPE] [int] NOT NULL,
    [ID_VAR] [numeric](18, 0) NOT NULL,
    [VTOTAL_DPE] [float] NOT NULL,
    [COSTO_DPE] [float] NOT NULL,
    CONSTRAINT [PK_DETALLEPEDIDO] PRIMARY KEY CLUSTERED
(
    [ID_DPE] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[PRIVILEGIOS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PRIVILEGIOS](
    [ID_PRI] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_MENU] [numeric](18, 0) NOT NULL,

```



```

(
    [ID_PRI] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[PEDIDOS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PEDIDOS](
    [ID_PED] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_PED] [varchar](50) NOT NULL,
    [ID_CLI] [numeric](18, 0) NOT NULL,
    [FECHA_PED] [datetime] NOT NULL,
    CONSTRAINT [PK_PEDIDOS] PRIMARY KEY CLUSTERED
(
    [ID_PED] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[POSTCOSECHA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[POSTCOSECHA](
    [ID_POS] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_BOD] [numeric](18, 0) NOT NULL,
    [ID_PLP] [numeric](18, 0) NOT NULL,
    [FECHA_POS] [datetime] NOT NULL,
    CONSTRAINT [PK_POSTCOSECHA] PRIMARY KEY CLUSTERED
(
    [ID_POS] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[DETALLESIEMBRA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[DETALLESIEMBRA](
    [ID_DSI] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [ID_SIE] [numeric](18, 0) NOT NULL,
    [ID_CCL] [numeric](18, 0) NOT NULL,
    [CANTIDAD_DSI] [int] NOT NULL,
    [COSTO_UNIT_DSI] [float] NOT NULL,
    [VTOTAL_DSI] [float] NOT NULL,
    CONSTRAINT [PK_DETALLESIEMBRA] PRIMARY KEY CLUSTERED
(
    [ID_DSI] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SIEMBRA]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[SIEMBRA](
    [ID_SIE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESC_SIE] [varchar](50) NOT NULL,
    [ID_PLP] [numeric](18, 0) NOT NULL,
    [ID_CAM] [numeric](18, 0) NOT NULL,
    [FECHA_SIE] [datetime] NOT NULL,
    [AREA_SIE] [float] NOT NULL,
    [FORMULA_SIE] [float] NULL,
    CONSTRAINT [PK_SIEMBRA] PRIMARY KEY CLUSTERED
(
    [ID_SIE] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[FERTILIZACION]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[FERTILIZACION](
    [ID_FER] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_FER] [varchar](50) NOT NULL,
    [ID_SIE] [numeric](18, 0) NOT NULL,
    CONSTRAINT [PK_FERTILIZACION] PRIMARY KEY CLUSTERED
(
    [ID_FER] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[COMPRAS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[COMPRAS](
    [ID_COM] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESC_COM] [varchar](50) NOT NULL,
    [ID_PLP] [numeric](18, 0) NOT NULL,
    [FECHA_COM] [datetime] NOT NULL,
    CONSTRAINT [PK_COMPRAS] PRIMARY KEY CLUSTERED
(
    [ID_COM] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[CONSTRUCCION]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[CONSTRUCCION](
    [ID_CON] [numeric](18, 0) IDENTITY(1,1) NOT NULL,

```

```

[FECHA_CON] [datetime] NOT NULL,
[AREA_CON] [float] NOT NULL,
CONSTRAINT [PK_CONSTRUCCION] PRIMARY KEY CLUSTERED
(
    [ID_CON] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[EMPLEADO]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[EMPLEADO](
    [CEDULA_EMP] [varchar](10) NOT NULL,
    [ID_PLP] [numeric](18, 0) NULL,
    [NOMBRE_EMP] [varchar](50) NULL,
    [APELLIDO_EMP] [varchar](50) NULL,
    [DIRECCION_EMP] [varchar](50) NULL,
    [TELEFONO_EMP] [varchar](13) NULL,
    [OBS_EMP] [varchar](100) NULL,
    CONSTRAINT [PK_EMPLEADO] PRIMARY KEY CLUSTERED
(
    [CEDULA_EMP] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[PROVEEDOR]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[PROVEEDOR](
    [ID_PROVE] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [CEDULA_PRO] [varchar](10) NOT NULL,
    [ID_PLP] [numeric](18, 0) NOT NULL,
    [NOMBRE_PRO] [varchar](50) NOT NULL,

```

```

[TELEFONO_PRO] [varchar](13) NULL,
CONSTRAINT [PK_PROVEEDOR_1] PRIMARY KEY CLUSTERED
(
    [ID_PROVE] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[VENTAS]') AND type in (N'U'))
BEGIN
CREATE TABLE [dbo].[VENTAS](
    [ID_VEN] [numeric](18, 0) IDENTITY(1,1) NOT NULL,
    [DESCRIPCION_VEN] [varchar](50) NOT NULL,
    [ID_PLP] [numeric](18, 0) NOT NULL,
    [FECHA_VEN] [datetime] NOT NULL,
    CONSTRAINT [PK_VENTAS] PRIMARY KEY CLUSTERED
(
    [ID_VEN] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[RPTComprasMes]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[RPTComprasMes]
@Fecha datetime
AS
SELECT
PLANTA_PRODUCCION.DENOMINACION_PLP,
COMPRAS.FECHA_COM,
PRODUCTO.DESCRIPCION_PROD,
PRODUCTO_COMPRA.CANTIDAD_PROD,

```

```

FROM COMPRAS
INNER JOIN PLANTA_PRODUCCION ON COMPRAS.ID_PLP =
PLANTA_PRODUCCION.ID_PLP
INNER JOIN PRODUCTO_COMPRA ON COMPRAS.ID_COM =
PRODUCTO_COMPRA.ID_COM
INNER JOIN PRODUCTO ON PRODUCTO_COMPRA.ID_PROD =
PRODUCTO.ID_PROD
WHERE (MONTH(COMPRAS.FECHA_COM) = MONTH(@Fecha)) AND
(YEAR(dbo.COMPRAS.FECHA_COM) = YEAR(@Fecha))'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaProductoCompra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ListaProductoCompra]
@CodigoCompra as numeric=0
AS
SELECT
COMPRAS.ID_COM,
COMPRAS.DESC_COM,
COMPRAS.FECHA_COM,
PRODUCTO.ID_PROD,
PRODUCTO.DESCRIPCION_PROD,
PRODUCTO_COMPRA.CANTIDAD_PROD,
PRODUCTO_COMPRA.VALOR_UNITARIO,
PRODUCTO_COMPRA.VALOR_TOTAL,
PLANTA_PRODUCCION.DENOMINACION_PLP
FROM COMPRAS
INNER JOIN PRODUCTO_COMPRA ON COMPRAS.ID_COM =
PRODUCTO_COMPRA.ID_COM
INNER JOIN PRODUCTO ON PRODUCTO_COMPRA.ID_PROD =
PRODUCTO.ID_PROD
INNER JOIN PLANTA_PRODUCCION ON COMPRAS.ID_PLP =
PLANTA_PRODUCCION.ID_PLP
WHERE COMPRAS.ID_COM=@CodigoCompra
'
END
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoCompras]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoCompras]
(
    @CodigoPlantaProduccion numeric,
    @FechaCompra datetime,
    @DescripcionCompra varchar(50)
)
AS
INSERT INTO COMPRAS(
ID_PLP,
FECHA_COM,
DESC_COM)
VALUES(
@CodigoPlantaProduccion,
@FechaCompra,
@DescripcionCompra)

SELECT @@IDENTITY AS "Identity"
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaCompra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaCompra]
(
    @CodigoCompra numeric,
    @CodigoPlantaProduccion numeric,
    @FechaCompra datetime,
    @DescripcionCompra varchar(50)
)
AS
UPDATE COMPRAS

```

```

        DESC_COM=@DescripcionCompra
WHERE ID_COM=@CodigoCompra
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListarCompras]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListarCompras]
    @DescripcionCompra VARCHAR(50)='''
AS
SELECT
COMPRAS.ID_COM,COMPRAS.DESCRIPCION_COM,PLANTA_PRODUCCION.DENOMINACION_PLP,COMPRAS.FECHA_COM
FROM      COMPRAS INNER JOIN dbo.PLANTA_PRODUCCION ON
dbo.COMPRAS.ID_PLP = dbo.PLANTA_PRODUCCION.ID_PLP
WHERE COMPRAS.DESCRIPCION_COM LIKE '%' + @DescripcionCompra + '%'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ActualizaFertilizacion]
    (
        @CodigoFertilizacion int=0,
        @DescripcionFertilizacion VARCHAR(50),
        @CodigoSiembra int=0
    )
AS
update FERTILIZACION
set DESCRIPCION_FER=@DescripcionFertilizacion,
ID_SIE=@CodigoSiembra

```



```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaFertilizacion]
@DescripcionFertilizacion varchar(50)='''
AS
SELECT
f.ID_FER,
f.DESCRIPCION_FER,
f.ID_SIE,
s.DESC_SIE,
p.DENOMINACION_PLP,
c.DESCRIPCION_CAM,
s.FECHA_SIE,
s.AREA_SIE
FROM FERTILIZACION f
INNER JOIN SIEMBRA s ON f.ID_SIE = s.ID_SIE
INNER JOIN PLANTA_PRODUCCION p ON s.ID_PLP = p.ID_PLP
INNER JOIN CAMAS c ON s.ID_CAM = c.ID_CAM
WHERE f.DESCRIPCION_FER LIKE '%' + @DescripcionFertilizacion + '%'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoFertilizacion]
(
    @DescripcionFertilizacion VARCHAR(50),
    @CodigoSiembra int=0
)

```

```

INSERT INTO FERTILIZACION(
DESCRIPCION_FER,
ID_SIE
)
VALUES(
@DescripcionFertilizacion,
@CodigoSiembra
)

```

```

SELECT @@IDENTITY AS "Identity"
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[RPTFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[RPTFertilizacion]
@DescripcionSiembra varchar(50)='',
@FechaFertilizacion datetime
AS
SELECT
SIEMBRA.DESC_SIE,
FERTILIZACION.DESCRIPCION_FER,
DETALLEFERTILIZACION.FECHA_DFE,
ELEMENTOS.DESCRIPCION_ELE,
DETALLEFERTILIZACION.CANTIDAD_DFE,
DETALLEFERTILIZACION.VCOSTO_DFE,
DETALLEFERTILIZACION.TOTAL_DFE
FROM FERTILIZACION
INNER JOIN DETALLEFERTILIZACION ON FERTILIZACION.ID_FER =
DETALLEFERTILIZACION.ID_FER
INNER JOIN SIEMBRA ON FERTILIZACION.ID_SIE = SIEMBRA.ID_SIE
INNER JOIN ELEMENTOS ON DETALLEFERTILIZACION.ID_ELE =
ELEMENTOS.ID_ELE
WHERE (SIEMBRA.DESC_SIE=@DescripcionSiembra) AND
(DETALLEFERTILIZACION.FECHA_DFE = CONVERT(DATETIME,
@FechaFertilizacion, 102))

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SPIngresoProducto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SPIngresoProducto]
(
    @DescripcionProducto as varchar(50)
)
AS
    INSERT INTO PRODUCTO (DESCRIPCION_PROD)
VALUES(@DescripcionProducto)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SPActualizaProducto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SPActualizaProducto]
(
    @CodigoProducto as numeric,
    @DescripcionProducto as varchar(50)
)
AS
    update PRODUCTO
    set DESCRIPCION_PROD=@DescripcionProducto
    where ID_PROD=@CodigoProducto'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaProducto]') AND type in (N'P', N'PC'))

```

```

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaProducto]
(
    @CodigoProducto numeric
)
AS
DELETE FROM PRODUCTO
WHERE ID_PROD=@CodigoProducto'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaProducto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'create PROCEDURE
[dbo].[SP_ListaProducto]
    @DescripcionProducto VARCHAR(50)=""
AS
    SELECT ID_PROD,DESCRIPCION_PROD FROM producto

WHERE DESCRIPCION_PROD LIKE "%"+ @DescripcionProducto + "%'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaProducto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'create PROCEDURE
[dbo].[SP_ActualizaProducto]
(
    @CodigoProducto numeric(18, 0),
    @DescripcionProducto varchar(50)
)
AS
UPDATE Producto
SET DESCRIPCION_PROD=@DescripcionProducto
WHERE (ID_PROD=@CodigoProducto)

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresarProducto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresarProducto]
(
    @DescripcionProducto varchar(50)
)

```

```

AS
    INSERT INTO PRODUCTO (DESCRIPCION_PROD)
    VALUES (@DescripcionProducto)

```

```

RETURN
,

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaConstruccion]
    @CodigoPlantaProduccion as integer=0,
    @Area as float=0

```

```

AS
set @CodigoPlantaProduccion= ISNULL(@CodigoPlantaProduccion,0)
SET @Area = isNull(@Area,0)
SELECT
dbo.CONSTRUCCION.ID_CON,
dbo.CONSTRUCCION.ID_PLP,
dbo.PLANTA_PRODUCCION.DENOMINACION_PLP,
dbo.CONSTRUCCION.DENOMINACION_CON,
dbo.CONSTRUCCION.FECHA_CON,
dbo.CONSTRUCCION.AREA_CON,

```

```

dbo.CONSTRUCCION INNER JOIN dbo.PLANTA_PRODUCCION ON
dbo.CONSTRUCCION.ID_PLP = dbo.PLANTA_PRODUCCION.ID_PLP
WHERE
CASE @CodigoPlantaProduccion When 0 then 0 else
dbo.CONSTRUCCION.ID_PLP end like Case @CodigoPlantaProduccion When 0
then 0 else @CodigoPlantaProduccion end
and
CASE @Area When 0 then 0 else dbo.CONSTRUCCION.AREA_CON end = Case
@Area When 0 then 0 else @Area end
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoConstruccion]
(
    @CodigoPlantaProduccion numeric,
    @DenominacionConstruccion varchar(30),
    @FechaConstruccion datetime,
    @AreaConstruccion float
)
AS
INSERT INTO Construccion(
ID_PLP,
DENOMINACION_CON,
FECHA_CON,
AREA_CON
)
VALUES(
@CodigoPlantaProduccion,
@DenominacionConstruccion,
@FechaConstruccion,
@AreaConstruccion
)'
END
GO
SET ANSI_NULLS ON

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaConstruccion]
(
    @CodigoConstruccion numeric,
    @CodigoPlantaProduccion numeric,
    @DenominacionConstruccion varchar(30),
    @FechaConstruccion datetime,
    @AreaConstruccion float
)
AS
UPDATE Construccion
SET ID_PLP=@CodigoPlantaProduccion,
DENOMINACION_CON=@DenominacionConstruccion,
FECHA_CON=@FechaConstruccion,
AREA_CON=@AreaConstruccion
WHERE ID_CON=@CodigoConstruccion
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[RPTConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[RPTConstruccion]
    @Area float
AS
SELECT
PLANTA_PRODUCCION.DENOMINACION_PLP,
CONSTRUCCION.DENOMINACION_CON,
CONSTRUCCION.FECHA_CON,
CONSTRUCCION.AREA_CON,
MATERIALESCONSTRUCCION.DESCRIPCION_MCS,
DETALLECONSTRUCCION.CANTIDAD_DCS,
DETALLECONSTRUCCION.COSTOS_DCS,
DETALLECONSTRUCCION.VTOTAL_DCS

```

```

INNER JOIN DETALLECONSTRUCCION ON
MATERIALESCONSTRUCCION.ID_MCS =
DETALLECONSTRUCCION.ID_MCS
INNER JOIN CONSTRUCCION ON DETALLECONSTRUCCION.ID_CON =
CONSTRUCCION.ID_CON
INNER JOIN PLANTA_PRODUCCION ON CONSTRUCCION.ID_PLP =
PLANTA_PRODUCCION.ID_PLP
WHERE (CONSTRUCCION.AREA_CON = @Area)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultaConstruccion]
    @CodigoConstruccion as integer=0
AS
SELECT
CONSTRUCCION.ID_CON,
PLANTA_PRODUCCION.DENOMINACION_PLP,
CONSTRUCCION.DENOMINACION_CON,
CONSTRUCCION.FECHA_CON,
CONSTRUCCION.AREA_CON
FROM CONSTRUCCION
INNER JOIN PLANTA_PRODUCCION ON CONSTRUCCION.ID_PLP =
PLANTA_PRODUCCION.ID_PLP
WHERE CONSTRUCCION.ID_CON=@CodigoConstruccion
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaConstruccionTodos]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaConstruccionTodos]

```



```

AS
SELECT
CONSTRUCCION.ID_CON,
CONSTRUCCION.ID_PLP,
PLANTA_PRODUCCION.DENOMINACION_PLP,
CONSTRUCCION.DENOMINACION_CON,
CONSTRUCCION.FECHA_CON,
CONSTRUCCION.AREA_CON
FROM CONSTRUCCION
INNER JOIN PLANTA_PRODUCCION ON CONSTRUCCION.ID_PLP =
PLANTA_PRODUCCION.ID_PLP'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultarDetalleFertilizacion]') AND type in (N'P',
N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ConsultarDetalleFertilizacion]
@CodigoDetalleFertilizacion as integer=0
AS
SELECT
df.ID_DFE,
df.DESCRIPCION_DFE,
df.ID_FER,
df.ID_ELE,
e.DESCRIPCION_ELE,
df.FECHA_DFE,
df.CANTIDAD_DFE,
df.VCOSTO_DFE,
df.TOTAL_DFE
FROM DETALLEFERTILIZACION df
INNER JOIN ELEMENTOS e ON df.ID_ELE = e.ID_ELE
WHERE df.ID_DFE=@CodigoDetalleFertilizacion
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaDetalleFertilizacion]') AND type in (N'P',
N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ActualizaDetalleFertilizacion]
    @CodigoDetalleFertilizacion as integer,
    @DescripcionFertilizacion as varchar(50),
    @CodigoFertilizacion as integer,
    @CodigoElemento as integer,
    @FechaFertilizacion as datetime,
    @Cantidad as integer,
    @Costo as Float,
    @Total as float

AS
    UPDATE DETALLEFERTILIZACION
    SET DESCRIPCION_DFE=@DescripcionFertilizacion,
    FECHA_DFE=@FechaFertilizacion,
    CANTIDAD_DFE=@Cantidad,
    TOTAL_DFE=@Total
    WHERE ID_DFE=@CodigoDetalleFertilizacion
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoDetalleFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoDetalleFertilizacion]
    @DescripcionFertilizacion as varchar(50),
    @CodigoFertilizacion as integer,
    @CodigoElemento as integer,
    @FechaFertilizacion as datetime,
    @Cantidad as integer,
    @Costo as Float,
    @Total as float

```

AS

```

ID_FER,
ID_ELE,
FECHA_DFE,
CANTIDAD_DFE,
VCOSTO_DFE,
TOTAL_DFE
)
VALUES (
@DescripcionFertilizacion,
@CodigoFertilizacion,
@CodigoElemento,
@FechaFertilizacion,
@Cantidad,
@Costo,
@Total
)

```

```

,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaDetalleFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaDetalleFertilizacion]
@CodigoFertilizacion as numeric=0
AS
SELECT
df.ID_DFE,
df.DESCRIPCION_DFE,
df.ID_FER,
df.ID_ELE,
e.DESCRIPCION_ELE,
df.FECHA_DFE,
df.CANTIDAD_DFE,
df.VCOSTO_DFE,
df.TOTAL_DFE
FROM DETALLEFERTILIZACION df
INNER JOIN ELEMENTOS e ON df.ID_ELE = e.ID_ELE
WHERE df.ID_FER=@CodigoFertilizacion
'

```

```

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaDetalleFertilizacion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaDetalleFertilizacion]
(
    @CodigoFertilizacion integer
)
AS
DELETE FROM DETALLEFERTILIZACION
WHERE ID_FER=@CodigoFertilizacion'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoProductoCompra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoProductoCompra]
(
    @CodigoCompra as int=0,
    @CodigoProducto AS int=0,
    @Cantidad as INT=0,
    @ValorUnitario as float=0,
    @ValorTotal as float=0
)
AS
INSERT INTO PRODUCTO_COMPRA
(
    ID_COM,
    ID_PROD,
    CANTIDAD_PROD,
    VALOR_UNITARIO,
    VALOR_TOTAL
)
VALUES

```

```
@CodigoCompra,  
@CodigoProducto,  
@Cantidad,  
@ValorUnitario,  
@ValorTotal  
)  
,
```

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaProductoCompra]') AND type in (N'P', N'PC'))
BEGIN

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaProductoCompra]

(
 @CodigoCompra integer
)

AS

DELETE FROM PRODUCTO_COMPRA

WHERE ID_COM=@CodigoCompra'

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[RPTVentas]') AND type in (N'P', N'PC'))
BEGIN

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[RPTVentas]

 @Fecha datetime

AS

SELECT

PLANTA_PRODUCCION.DENOMINACION_PLP,

VENTAS.FECHA_VEN,

VARIEDAD.DESCRIPCION_VAR,

CLASE_CLAVEL.DESCRIPCION_CCL,

TAMANIO.DESCRIPCION_TAM,

TAMANIO_MEDIDA_TAM

```

DETALLE_DVE.CANTIDAD_DV,
DETALLE_DVE.TOTAL_DV
FROM VENTAS
INNER JOIN DETALLE_DVE ON VENTAS.ID_VEN =
DETALLE_DVE.ID_VEN
INNER JOIN VARIEDAD ON DETALLE_DVE.ID_VARI =
VARIEDAD.ID_VARI
INNER JOIN CLASE_CLAVEL ON VARIEDAD.ID_CCL =
CLASE_CLAVEL.ID_CCL
INNER JOIN TAMANIO ON dbo.VARIEDAD.ID_TAM = TAMANIO.ID_TAM
INNER JOIN COSTO ON TAMANIO.ID_COS = COSTO.ID_COS
INNER JOIN PLANTA_PRODUCCION ON VENTAS.ID_PLP =
PLANTA_PRODUCCION.ID_PLP
WHERE (MONTH(VENTAS.FECHA_VEN) = MONTH(@Fecha)) AND
(YEAR(VENTAS.FECHA_VEN) = YEAR(@Fecha))
,

```

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[RPTPostcosecha]') AND type in (N'P', N'PC'))

```

BEGIN

```

EXEC dbo.sp_executesql @statement = N'

```

```

CREATE PROCEDURE [dbo].[RPTPostcosecha]

```

```

    @FechaPostcosecha datetime

```

AS

SELECT

```

VARIEDAD.DESCRIPCION_VAR,
POSTCOSECHA.FECHA_POS,
CLASE_CLAVEL.DESCRIPCION_CCL,
TAMANIO.DESCRIPCION_TAM,
BODEGA.DESCRIPCION_BOD,
PLANTA_PRODUCCION.DENOMINACION_PLP,
TAMANIO.MEDIDA_TAM,
DETALLEPOSTCOSECHA.CANTIDAD_DPO,
COSTO.VALOR_COS,
DETALLEPOSTCOSECHA.VTOTAL_DPO
FROM DETALLEPOSTCOSECHA
INNER JOIN POSTCOSECHA ON DETALLEPOSTCOSECHA.ID_POS =
POSTCOSECHA.ID_POS

```

```

INNER JOIN BODEGA ON POSTCOSECHA.ID_BOD = BODEGA.ID_BOD
INNER JOIN VARIEDAD ON DETALLEPOSTCOSECHA.ID_VARI =
VARIEDAD.ID_VARI
INNER JOIN CLASE_CLAVEL ON VARIEDAD.ID_CCL =
CLASE_CLAVEL.ID_CCL
INNER JOIN TAMANIO ON VARIEDAD.ID_TAM = TAMANIO.ID_TAM
INNER JOIN COSTO ON TAMANIO.ID_COS = COSTO.ID_COS
WHERE POSTCOSECHA.FECHA_POS = @FechaPostcosecha
,

```

```
END
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaDetallePostcosecha]') AND type in (N'P', N'PC'))
```

```
BEGIN
```

```
EXEC dbo.sp_executesql @statement = N'
```

```
CREATE PROCEDURE [dbo].[SP_ListaDetallePostcosecha]
```

```
@CodigoPostcosecha as numeric=0
```

```
AS
```

```
SELECT
```

```
dbo.POSTCOSECHA.ID_POS,
```

```
dbo.BODEGA.ID_BOD,
```

```
dbo.BODEGA.DESCRIPCION_BOD,
```

```
dbo.PLANTA_PRODUCCION.ID_PLP,
```

```
dbo.PLANTA_PRODUCCION.DENOMINACION_PLP,
```

```
dbo.POSTCOSECHA.FECHA_POS,
```

```
dbo.DETALLEPOSTCOSECHA.ID_DPO,
```

```
dbo.DETALLEPOSTCOSECHA.CANTIDAD_DPO,
```

```
dbo.VARIEDAD.ID_VARI,
```

```
dbo.VARIEDAD.DESCRIPCION_VAR,
```

```
dbo.CLASE_CLAVEL.ID_CCL,
```

```
dbo.CLASE_CLAVEL.DESCRIPCION_CCL,
```

```
dbo.TAMANIO.ID_TAM,
```

```
dbo.TAMANIO.DESCRIPCION_TAM,
```

```
dbo.TAMANIO.MEDIDA_TAM,
```

```
dbo.COSTO.ID_COS,
```

```
dbo.COSTO.VALOR_COS,
```

```
dbo.DETALLEPOSTCOSECHA.VTOTAL_DPO
```

```
FROM dbo.BODEGA
```

```
INNER JOIN dbo.POSTCOSECHA ON dbo.BODEGA.ID_BOD =
```

```

INNER JOIN dbo.PLANTA_PRODUCCION ON dbo.POSTCOSECHA.ID_PLP =
dbo.PLANTA_PRODUCCION.ID_PLP
INNER JOIN dbo.DETALLEPOSTCOSECHA ON dbo.POSTCOSECHA.ID_POS =
dbo.DETALLEPOSTCOSECHA.ID_POS
INNER JOIN dbo.VARIEDAD ON dbo.DETALLEPOSTCOSECHA.ID_VARI =
dbo.VARIEDAD.ID_VARI
INNER JOIN dbo.CLASE_CLAVEL ON dbo.VARIEDAD.ID_CCL =
dbo.CLASE_CLAVEL.ID_CCL
INNER JOIN dbo.TAMANIO ON dbo.VARIEDAD.ID_TAM =
dbo.TAMANIO.ID_TAM
INNER JOIN dbo.COSTO ON dbo.TAMANIO.ID_COS = dbo.COSTO.ID_COS
WHERE (dbo.POSTCOSECHA.ID_POS = @CodigoPostcosecha)
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaCosto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaCosto]
(
    @CodigoCosto numeric
)
AS
DELETE FROM COSTO
WHERE ID_COS=@CodigoCosto'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaTamanio]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaTamanio]
    @DescripcionTamanio VARCHAR(50)=""
AS

```



```

TAMANIO.ID_COS,
COSTO.DESCRIPCION_COS,
COSTO.VALOR_COS,
TAMANIO.DESCRIPCION_TAM,
TAMANIO.MEDIDA_TAM
FROM TAMANIO INNER JOIN COSTO ON TAMANIO.ID_COS =
COSTO.ID_COS
WHERE DESCRIPCION_TAM LIKE "%"+ @DescripcionTamanio + "%"
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaCosto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaCosto]
    @DescripcionCosto VARCHAR(30)='''
AS
SELECT
    ID_COS,
    DESCRIPCION_COS,
    VALOR_COS,
    FECHA_COS
FROM COSTO
WHERE DESCRIPCION_COS LIKE "%"+ @DescripcionCosto + "%"
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoCosto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoCosto]
(
    @DescripcionCosto varchar(30),
    @ValorCosto float,
    @FechaCosto datetime

```

```

AS
INSERT INTO COSTO(
    DESCRIPCION_COS,
    VALOR_COS,
    FECHA_COS
)
VALUES(
    @DescripcionCosto,
    @ValorCosto,
    @FechaCosto
)
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaCosto]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaCosto]
(
    @CodigoCosto numeric,
    @DescripcionCosto varchar(30),
    @ValorCosto float,
    @FechaCosto datetime
)
AS
UPDATE COSTO
SET DESCRIPCION_COS=@DescripcionCosto,
    VALOR_COS=@ValorCosto,
    FECHA_COS=@FechaCosto
WHERE ID_COS=@CodigoCosto'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaCosto]') AND type in (N'P', N'PC'))

```

```

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultaCosto]
    @CodigoCosto INT=0
AS
SELECT
    ID_COS,
    DESCRIPCION_COS,
    VALOR_COS,
    FECHA_COS
FROM COSTO
WHERE ID_COS =@CodigoCosto'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaVentas]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ListaVentas]
    @DescripcionVenta VARCHAR(50)=""
AS
SELECT
    v.ID_VEN AS CodigoVenta ,
    dv.ID_DV AS CodigoDetalleCompra,
    va.ID_VARI,
    v.DESCRIPCION_VEN,
    pp.DENOMINACION_PLP,
    v.FECHA_VEN ,
    va.DESCRIPCION_VAR AS Variedad,
    dv.CANTIDAD_DV AS Cantidad,
    c.VALOR_COS AS VUnitario,
    dv.TOTAL_DV AS Total
FROM VENTAS v
    INNER JOIN DETALLE_DVE dv ON v.ID_VEN = dv.ID_VEN
    INNER JOIN PLANTA_PRODUCCION pp ON v.ID_PLP = pp.ID_PLP
    INNER JOIN VARIEDAD va ON dv.ID_VARI = va.ID_VARI
    INNER JOIN CLASE_CLAVEL cc ON va.ID_CCL = cc.ID_CCL
    INNER JOIN TAMANIO t ON va.ID_TAM = t.ID_TAM
    INNER JOIN COSTO c ON t.ID_COS = c.ID_COS
WHERE

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaVariedad]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultaVariedad]
@CodigoVariedad as integer
AS
SELECT
VARIEDAD.ID_VARI,
VARIEDAD.DESCRIPCION_VAR,
VARIEDAD.ID_CCL,
CLASE_CLAVEL.DESCRIPCION_CCL,
VARIEDAD.ID_TAM,
TAMANIO.MEDIDA_TAM,
TAMANIO.ID_COS,
COSTO.DESCRIPCION_COS,
COSTO.VALOR_COS,
TAMANIO.DESCRIPCION_TAM
FROM VARIEDAD
INNER JOIN TAMANIO ON VARIEDAD.ID_TAM = TAMANIO.ID_TAM
INNER JOIN CLASE_CLAVEL ON VARIEDAD.ID_CCL =
CLASE_CLAVEL.ID_CCL
INNER JOIN COSTO ON TAMANIO.ID_COS = COSTO.ID_COS
WHERE VARIEDAD.ID_VARI=@CodigoVariedad
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaVariedad]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaVariedad]
@DescripcionVariedad VARCHAR(50)=""

```

```

AS
SELECT
VARIEDAD.ID_VARI,
VARIEDAD.DESCRIPCION_VAR,
VARIEDAD.ID_CCL,
CLASE_CLAVEL.DESCRIPCION_CCL,
VARIEDAD.ID_TAM,
TAMANIO.DESCRIPCION_TAM,
TAMANIO.MEDIDA_TAM
FROM VARIEDAD INNER JOIN CLASE_CLAVEL ON VARIEDAD.ID_CCL =
CLASE_CLAVEL.ID_CCL INNER JOIN
        TAMANIO ON VARIEDAD.ID_TAM = TAMANIO.ID_TAM
WHERE DESCRIPCION_VAR LIKE "%"+ @DescripcionVariedad + "%"
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaDetalleVenta]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaDetalleVenta]
@CodigoVenta as numeric=0
AS
SELECT
DETALLE_DVE.ID_DV,
DETALLE_DVE.CANTIDAD_DV,
VARIEDAD.ID_VARI,
VARIEDAD.DESCRIPCION_VAR,
CLASE_CLAVEL.DESCRIPCION_CCL,
VARIEDAD.ID_CCL,
VARIEDAD.ID_TAM,
TAMANIO.ID_COS,
TAMANIO.DESCRIPCION_TAM,
TAMANIO.MEDIDA_TAM,
DETALLE_DVE.ID_VEN,
DETALLE_DVE.TOTAL_DV
FROM DETALLE_DVE
INNER JOIN VARIEDAD ON DETALLE_DVE.ID_VARI =
VARIEDAD.ID_VARI
INNER JOIN CLASE_CLAVEL ON VARIEDAD.ID_CCL =

```

```

WHERE DETALLE_DVE.ID_VEN=@CodigoVenta
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaVariedad]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaVariedad]
(
    @CodigoVariedad int=0,
    @DescripcionVariedad varchar(50)='',
    @CodigoClaseClavel int=0,
    @CodigoTamanio int=0
)

AS
    UPDATE Variedad SET
    DESCRIPCION_VAR=@DescripcionVariedad,
    ID_CCL=@CodigoClaseClavel,
    ID_TAM=@CodigoTamanio
    WHERE ID_VARI=@CodigoVariedad
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaVariedad]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaVariedad]
    @CodigoVariedad as numeric
AS
    DELETE FROM Variedad
    WHERE ID_VARI=@CodigoVariedad
,

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoVariedad]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoVariedad]
(
    @DescripcionVariedad varchar(50)='',
    @CodigoClaseClavel numeric,
    @CodigoTamanio numeric
)
AS
    INSERT INTO Variedad
    (
        DESCRIPCION_VAR,
        ID_CCL,
        ID_TAM
    )
    VALUES
    (
        @DescripcionVariedad,
        @CodigoClaseClavel,
        @CodigoTamanio
    )
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaDetallePedido]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaDetallePedido]
@CodigoPedido as numeric=0
AS
SELECT
dp.ID_DPE,

```

```

dp.ID_VAR,
v.DESCRIPCION_VAR,
dp.VTOTAL_DPE,
dp.COSTO_DPE
FROM DETALLEPEDIDO dp
INNER JOIN VARIEDAD v ON dp.ID_VAR = v.ID_VARI
WHERE dp.ID_PED=@CodigoPedido
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListarPostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
create PROCEDURE [dbo].[SP_ListarPostcosecha]
        @CodigoBodega int=0
AS
SELECT
dbo.POSTCOSECHA.ID_POS,
dbo.BODEGA.ID_BOD,
dbo.BODEGA.DESCRIPCION_BOD,
dbo.PLANTA_PRODUCCION.ID_PLP,
dbo.PLANTA_PRODUCCION.DENOMINACION_PLP,
dbo.POSTCOSECHA.FECHA_POS
FROM dbo.POSTCOSECHA
INNER JOIN dbo.BODEGA ON dbo.POSTCOSECHA.ID_BOD =
dbo.BODEGA.ID_BOD
INNER JOIN dbo.PLANTA_PRODUCCION ON dbo.POSTCOSECHA.ID_PLP =
dbo.PLANTA_PRODUCCION.ID_PLP
WHERE (dbo.BODEGA.ID_BOD = @CodigoBodega)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultarPostcosecha]') AND type in (N'P', N'PC'))
BEGIN

```



```

        @CodigoPostcosecha int=0
AS
SELECT
dbo.POSTCOSECHA.ID_POS,
dbo.BODEGA.ID_BOD,
dbo.BODEGA.DESCRIPCION_BOD,
dbo.PLANTA_PRODUCCION.ID_PLP,
dbo.PLANTA_PRODUCCION.DENOMINACION_PLP,
dbo.POSTCOSECHA.FECHA_POS
FROM dbo.POSTCOSECHA
INNER JOIN dbo.BODEGA ON dbo.POSTCOSECHA.ID_BOD =
dbo.BODEGA.ID_BOD
INNER JOIN dbo.PLANTA_PRODUCCION ON dbo.POSTCOSECHA.ID_PLP =
dbo.PLANTA_PRODUCCION.ID_PLP
WHERE (dbo.POSTCOSECHA.ID_POS = @CodigoPostcosecha)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaPostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_EliminaPostcosecha]
        @CodigoPostcosecha int=0
AS
delete from Postcosecha
where ID_POS=@CodigoPostcosecha
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoPostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoPostcosecha]
(
        @CodigoPostcosecha int=0

```

```

        @FechaPostcosecha dateTIME
    )

AS
INSERT INTO POSTCOSECHA(
ID_BOD,
ID_PLP,
FECHA_POS
)
VALUES(
@CodigoBodega,
@CodigoPlantaProduccion,
@FechaPostcosecha)

SELECT @@IDENTITY AS "Identity"
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaPostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
create PROCEDURE [dbo].[SP_ActualizaPostcosecha]
(
    @CodigoPostcosecha int=0,
    @CodigoBodega int=0,
    @CodigoPlantaProduccion int=0,
    @FechaPostcosecha dateTIME
)
AS
update POSTCOSECHA
set ID_BOD=@CodigoBodega,
ID_PLP=@CodigoPlantaProduccion,
FECHA_POS=@FechaPostcosecha
where ID_POS=@CodigoPostcosecha
,

END
GO
SET ANSI_NULLS ON
GO

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaPostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ListaPostcosecha]
    @CodigoBodega int=0
AS
SELECT
dbo.POSTCOSECHA.ID_POS,
dbo.BODEGA.ID_BOD,
dbo.BODEGA.DESCRIPCION_BOD,
dbo.PLANTA_PRODUCCION.ID_PLP,
dbo.PLANTA_PRODUCCION.DENOMINACION_PLP,
dbo.POSTCOSECHA.FECHA_POS
FROM dbo.POSTCOSECHA
INNER JOIN dbo.BODEGA ON dbo.POSTCOSECHA.ID_BOD =
dbo.BODEGA.ID_BOD
INNER JOIN dbo.PLANTA_PRODUCCION ON dbo.POSTCOSECHA.ID_PLP =
dbo.PLANTA_PRODUCCION.ID_PLP
WHERE (dbo.BODEGA.ID_BOD = @CodigoBodega)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoDetallePostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoDetallePostcosecha]
    @CodigoPostcosecha integer,
    @CodigoVariedad as integer,
    @Cantidad as integer,
    @Total as float
AS
INSERT INTO DETALLEPOSTCOSECHA(
    ID_POS,
    ID_VARI,
    CANTIDAD_DPO,
    VTOTAL_DPO
)

```

```

        @CodigoVariedad,
        @Cantidad,
        @Total
    )
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaDetallePostcosecha]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'create PROCEDURE
[dbo].[SP_EliminaDetallePostcosecha]
(
    @CodigoPostcosecha numeric
)
AS
DELETE FROM DETALLEPOSTCOSECHA
WHERE ID_POS=@CodigoPostcosecha'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaProveedor]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaProveedor]
    @CodigoProveedor int=0
AS
delete from PROVEEDOR
where ID_PROVE=@CodigoProveedor
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =  
OBJECT_ID(N'[dbo].[SP_IngresoProveedor]') AND type in (N'P', N'PC'))  
BEGIN  
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE  
[dbo].[SP_IngresoProveedor]
```

```
(  
    @CedulaProveedor varchar(10),  
    @NombreProveedor varchar(50),  
    @ApellidoProveedor varchar(50),  
    @DireccionProveedor varchar(30),  
    @TelefonoProveedor varchar(13),  
    @CodigoPlantaProduccion int=0  
)
```

AS

```
INSERT INTO PROVEEDOR(  
    CEDULA_PRO,  
    NOMBRE_PRO,  
    APELLIDO_PRO,  
    DIRECCION_PRO,  
    TELEFONO_PRO,  
    ID_PLP)  
VALUES (@CedulaProveedor,  
    @NombreProveedor,  
    @ApellidoProveedor,  
    @DireccionProveedor,  
    @TelefonoProveedor,  
    @CodigoPlantaProduccion)
```

END

GO

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =  
OBJECT_ID(N'[dbo].[SP_ActualizaProveedor]') AND type in (N'P', N'PC'))  
BEGIN
```

```
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE  
[dbo].[SP_ActualizaProveedor]
```

```
(  
    @CodigoProveedor int=0,  
    @CedulaProveedor varchar(10),  
    @NombreProveedor varchar(50),
```

```

        @TelefonoProveedor varchar(13),
        @CodigoPlantaProduccion int=0
    )
AS
update PROVEEDOR
set CEDULA_PRO=@CedulaProveedor,
    NOMBRE_PRO=@NombreProveedor,
    APELLIDO_PRO=@ApellidoProveedor,
    DIRECCION_PRO=@DireccionProveedor,
    TELEFONO_PRO=@TelefonoProveedor,
    ID_PLP=@CodigoPlantaProduccion
where ID_PROVE=@CodigoProveedor
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaProveedor]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaProveedor]
    @CedulaProveedor varchar(10)=""
AS
SELECT  PROVEEDOR.ID_PROVE, PROVEEDOR.CEDULA_PRO,
PROVEEDOR.NOMBRE_PRO, PROVEEDOR.APELLIDO_PRO,
PROVEEDOR.DIRECCION_PRO,
        PROVEEDOR.TELEFONO_PRO,
PLANTA_PRODUCCION.DENOMINACION_PLP,
PLANTA_PRODUCCION.ID_PLP
FROM    PROVEEDOR INNER JOIN PLANTA_PRODUCCION ON
PROVEEDOR.ID_PLP = PLANTA_PRODUCCION.ID_PLP
WHERE   (PROVEEDOR.CEDULA_PRO LIKE "%"+ @CedulaProveedor + "%")
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =

```

```

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaDetalleVenta]
(
    @CodigoVenta numeric
)
AS
DELETE FROM DETALLE_DVE
WHERE ID_VEN=@CodigoVenta'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoDetalleVenta]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoDetalleVenta]
(
    @CodigoVentas numeric,
    @CantidadVentaDetalle integer,
    @CodigoVariedad numeric,
    @TotalVentaDetalle float
)
AS
INSERT INTO DETALLE_DVE(
ID_VEN,
CANTIDAD_DV,
ID_VARI,
TOTAL_DV
)
VALUES(
@CodigoVentas,
@CantidadVentaDetalle,
@CodigoVariedad,
@TotalVentaDetalle
)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultarUsuario]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultarUsuario]
(
    @Nombre varchar(50),
    @Contrasenia varchar(50)
)
AS
    SELECT ID_USER, NOMBRE_USER,PASSWORD_USER FROM
USUARIO
    WHERE NOMBRE_USER=@Nombre AND
PASSWORD_USER=@Contrasenia
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaVenta]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaVenta]
(
    @CodigoVenta numeric,
    @CodigoPlantaProduccion numeric,
    @FechaVenta datetime,
    @DescripcionVenta varchar(50)
)
AS
UPDATE VENTAS
SET ID_PLP=@CodigoPlantaProduccion,
FECHA_VEN=@FechaVenta,
DESCRIPCION_VEN=@DescripcionVenta
WHERE ID_VEN=@CodigoVenta'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```



```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoVentas]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoVentas]
(
    @CodigoPlantaProduccion numeric,
    @FechaVenta datetime,
    @DescripcionVenta varchar(50)
)
AS
INSERT INTO VENTAS(
ID_PLP,
FECHA_VEN,
DESCRIPCION_VEN)
VALUES(
    @CodigoPlantaProduccion,
    @FechaVenta,
    @DescripcionVenta)
SELECT @@IDENTITY AS "Identity"
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[RPTPedidos]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[RPTPedidos]
@Fecha datetime
AS
SELECT
PEDIDOS.DESCRIPCION_PED,
CLIENTE.NOMBRE_COM_CLI,
CLIENTE.CEDULA_CLI,
PEDIDOS.FECHA_PED
FROM PEDIDOS
INNER JOIN CLIENTE ON dbo.PEDIDOS.ID_CLI = CLIENTE.ID_CLI
WHERE (MONTH(dbo.PEDIDOS.FECHA_PED) = MONTH(@Fecha) AND
YEAR(dbo.PEDIDOS.FECHA_PED) = YEAR(@Fecha))'

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaPedido]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaPedido]
@DescripcionPedido varchar(50)=""
AS
SELECT
p.ID_PED,
p.DESCRIPCION_PED,
p.ID_CLI,
c.NOMBRE_CLI,
c.APELLIDO_CLI,
c.NOMBRE_COM_CLI,
p.FECHA_PED
FROM PEDIDOS p
INNER JOIN CLIENTE c ON p.ID_CLI = c.ID_CLI
WHERE p.DESCRIPCION_PED LIKE "%" + @DescripcionPedido + "%"
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListarCliente]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ListarCliente]
@CedulaCliente VARCHAR(10)=""
AS
SELECT * FROM CLIENTE
WHERE CEDULA_CLI like "%" + @CedulaCliente + "%"
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaCliente]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ActualizaCliente]
(
    @CodigoCliente numeric,
    @CedulaCliente varchar(10),
    @NombreCliente varchar(50),
    @ApellidoCliente varchar(50),
    @NombreComercialCliente varchar(50),
    @DireccionCliente varchar(30),
    @TelefonoCliente varchar(13),
    @PaisCliente varchar(30),
    @MercadoCliente varchar(50)
)
AS
UPDATE CLIENTE
SET
    CEDULA_CLI=@CedulaCliente,
    NOMBRE_CLI=@NombreCliente,
    APELLIDO_CLI=@ApellidoCliente,
    NOMBRE_COM_CLI=@NombreComercialCliente,
    DIRECCION_CLI=@DireccionCliente,
    TELEFONO_CLI=@TelefonoCliente,
    PAIS_CLI=@PaisCliente,
    MERCADO_CLI=@MercadoCliente
WHERE (ID_CLI=@CodigoCliente)
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminarCliente]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_EliminarCliente]
(
    @CodigoCliente NUMERIC
)

```

```

AS
DELETE FROM CLIENTE
WHERE ID_CLI=@CodigoCliente
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListarBodega]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListarBodega]
@DescripcionBodega VARCHAR(50)='''
AS
        Select ID_BOD, DESCRIPCION_BOD from Bodega
        WHERE DESCRIPCION_BOD LIKE "%"+ @DescripcionBodega + "%"
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresarBodega]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresarBodega]
(
        @DescripcionBodega varchar(50)
)
AS
        INSERT INTO BODEGA (DESCRIPCION_BOD)
        VALUES (@DescripcionBodega)
RETURN
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaBodega]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaBodega]
(
    @CodigoBodega numeric(18, 0),
    @DescripcionBodega varchar(50)
)
AS
UPDATE Bodega
SET DESCRIPCION_BOD=@DescripcionBodega
WHERE (ID_BOD=@CodigoBodega)
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_LISTARMENU]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_LISTARMENU]
AS
    SELECT ID_MENU,DESCRIPCION_MENU FROM MENU'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaElementos]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaElementos]
    @CodigoElementos as numeric
AS
DELETE FROM ELEMENTOS
WHERE ID_ELE=@CodigoElementos

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaElementosFertilizacion]') AND type in (N'P',
N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaElementosFertilizacion]
    @CodigoElementos as numeric,
    @CantidadElementos as INT=0
AS
DECLARE @Aux as int
DECLARE @Aux2 as int
select @Aux =EXISTENCIA_ELE from ELEMENTOS WHERE
ID_ELE=@CodigoElementos
select @Aux2=@Aux-@CantidadElementos
UPDATE ELEMENTOS
SET EXISTENCIA_ELE=@Aux2
WHERE ID_ELE=@CodigoElementos
'

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaElementos]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaElementos]
    @DescripcionElementos VARCHAR(50)=""
AS
SELECT
    ID_ELE,
    DESCRIPCION_ELE,
    CANT_MIN_ELE,
    EXISTENCIA_ELE,
    COSTO_ELE

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaElementos]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaElementos]
(
    @CodigoElementos as numeric,
    @DescripcionElementos AS VARCHAR(50)='',
    @CantidadElementos as INT=0,
    @ExistenciaElementos as int=0,
    @CostoElementos as float=0
)
AS
UPDATE ELEMENTOS
SET
DESCRIPCION_ELE=@DescripcionElementos,
CANT_MIN_ELE=@CantidadElementos,
EXISTENCIA_ELE=@ExistenciaElementos,
COSTO_ELE=@CostoElementos
WHERE ID_ELE=@CodigoElementos
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresarElementos]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresarElementos]
(
    @DescripcionElementos AS VARCHAR(50)='',
    @CantidadElementos as INT=0,
    @ExistenciaElementos as int=0,
    @CostoElementos as float=0
\

```

```

AS
    INSERT INTO ELEMENTOS
    (
        DESCRIPCION_ELE,
        CANT_MIN_ELE,
        EXISTENCIA_ELE,
        COSTO_ELE
    )
    VALUES
    (
        @DescripcionElementos,
        @CantidadElementos,
        @ExistenciaElementos,
        @CostoElementos
    )
    ,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaElemento]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ConsultaElemento]
@CodigoElemento integer=0
AS
SELECT
ID_ELE,
DESCRIPCION_ELE,
CANT_MIN_ELE,
EXISTENCIA_ELE,
COSTO_ELE
FROM ELEMENTOS
WHERE ID_ELE =@CodigoElemento
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```



```

IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaConstruccion]
    @CodigoConstruccion as integer
AS
DELETE FROM DETALLECONSTRUCCION
WHERE ID_CON=@CodigoConstruccion
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoDetalleConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoDetalleConstruccion]
    @CodigoMaterialConstruccion integer,
    @CodigoConstruccion as integer,
    @Cantidad as integer,
    @Costo as Float,
    @Total as float
AS
INSERT INTO DETALLECONSTRUCCION (
    ID_MCS,
    ID_CON,
    CANTIDAD_DCS,
    COSTOS_DCS,
    VTOTAL_DCS
)
VALUES (
    @CodigoMaterialConstruccion,
    @CodigoConstruccion,
    @Cantidad,
    @Costo,
    @Total
)
,

END

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaDetalleConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaDetalleConstruccion]
        @CodigoConstruccion int=0
AS
SELECT
dt.ID_DCS as CodigoDetalleConstruccion,
dt.ID_MCS as CodigoMaterialConstruccion,
dt.ID_CON as CodigoConstruccion,
mc.DESCRIPCION_MCS as Material,
dt.CANTIDAD_DCS as Cantidad,
dt.COSTOS_DCS as VUnitario,
dt.VTOTAL_DCS as Total
FROM DETALLECONSTRUCCION dt
INNER JOIN MATERIALESCONSTRUCCION mc ON dt.ID_MCS = mc.ID_MCS
WHERE dt.ID_CON=@CodigoConstruccion
'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaMaterialConstruccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaMaterialConstruccion]
        @DescripcionMaterialConstruccion as varchar(50)=""
AS
Select
ID_MCS,
DESCRIPCION_MCS,
CANTIDAD_MCS,
CANT_MAXIMA_MCS,
C_UNIT_MCS
from MaterialesConstruccion
WHERE DESCRIPCION_MCS LIKE "%"+

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoMaterialConstruccion]') AND type in (N'P',
N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'create PROCEDURE
[dbo].[SP_IngresoMaterialConstruccion]
(
    @DescripcionMaterialConstruccion as varchar(50),
    @CostoUnitario as float,
    @CantidadMinima as int,
    @CantidadMaxima as int=0
)
AS
INSERT INTO MATERIALESCONSTRUCCION
(
    DESCRIPCION_MCS,
    C_UNIT_MCS,
    CANTIDAD_MCS,
    CANT_MAXIMA_MCS
)
VALUES
(
    @DescripcionMaterialConstruccion,
    @CostoUnitario,
    @CantidadMinima,
    @CantidadMaxima
)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaMaterialConstruccion]') AND type in (N'P',
N'PC'))
BEGIN

```

```

(
    @CodigoMaterialConstruccion as int,
    @DescripcionMaterialConstruccion as varchar(50),
    @CostoUnitario as float,
    @CantidadMinima as int,
    @CantidadMaxima as int=0
)
AS
UPDATE MATERIALESCONSTRUCCION
SET
DESCRIPCION_MCS=@DescripcionMaterialConstruccion,
C_UNIT_MCS=@CostoUnitario,
CANTIDAD_MCS=@CantidadMinima,
CANT_MAXIMA_MCS=@CantidadMaxima
WHERE ID_MCS=@CodigoMaterialConstruccion
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaMaterialConstruccion]') AND type in (N'P',
N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaMaterialConstruccion]
(
    @CodigoMaterialConstruccion as int
)
AS
DELETE FROM MATERIALESCONSTRUCCION
WHERE ID_MCS=@CodigoMaterialConstruccion
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaMaterialConstruccion]') AND type in (N'P',

```

```
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultaMaterialConstruccion]
```

```
    @CodigoMaterialConstruccion as integer=0
```

```
AS
```

```
    Select
    ID_MCS,
    DESCRIPCION_MCS,
    CANTIDAD_MCS,
    CANT_MAXIMA_MCS,
    C_UNIT_MCS
    from MaterialesConstruccion
    WHERE ID_MCS=@CodigoMaterialConstruccion
    ,
```

```
END
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaClaseClavel]') AND type in (N'P', N'PC'))
```

```
BEGIN
```

```
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaClaseClavel]
```

```
    @DescripcionClaseClavel VARCHAR(30)=""
```

```
AS
```

```
    SELECT ID_CCL,DESCRIPCION_CCL FROM CLASE_CLAVEL
```

```
WHERE DESCRIPCION_CCL LIKE "%"+ @DescripcionClaseClavel + "%"
```

```
END
```

```
GO
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaDetalleSiembra]') AND type in (N'P', N'PC'))
```

```
BEGIN
```

```
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaDetalleSiembra]
```

```
    @CodigoSiembra as numeric=0
```

```
AS
```

```
SELECT
```

```
1. ID_DSI
```

```

ds.ID_CCL,
cc.DESCRIPCION_CCL,
ds.CANTIDAD_DSI,
ds.COSTO_UNIT_DSI,
ds.VTOTAL_DSI
FROM DETALLESIEMBRA ds
INNER JOIN CLASE_CLAVEL cc ON ds.ID_CCL = cc.ID_CCL
WHERE ds.ID_SIE=@CodigoSiembra
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaClaseClavel]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaClaseClavel]
(
    @CodigoClaseClavel numeric
)
AS
DELETE FROM CLASE_CLAVEL
WHERE ID_CCL=@CodigoClaseClavel'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaClaseClavel]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaClaseClavel]
(
    @CodigoClaseClavel numeric,
    @DescripcionClaseClavel VARCHAR(30)
)
AS
UPDATE CLASE_CLAVEL

```

```

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoClaseClavel]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoClaseClavel]
(
    @DescripcionClaseClavel varchar(30)
)
AS
    INSERT INTO CLASE_CLAVEL(DESCRIPCION_CCL)
    VALUES(@DescripcionClaseClavel)'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaCama]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ActualizaCama]
(
    @CodigoCama numeric,
    @DescripcionCama VARCHAR(30),
    @AreaCama FLOAT,
    @LargoCama FLOAT,
    @AnchoCama FLOAT,
    @ProfundidadCama FLOAT,
    @FormulaCama FLOAT
)
AS
UPDATE CAMAS
SET DESCRIPCION_CAM=@DescripcionCama,AREA_CAM=@AreaCama,
LARGO_CAM=@LargoCama,ANCHO_CAM=@AnchoCama,PROFUNDIDAD_C
AM=@ProfundidadCama,

```

```

,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoCamas]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_IngresoCamas]
(
    @DescripcionCama VARCHAR(30),
    @AreaCama FLOAT,
    @LargoCama FLOAT,
    @AnchoCama FLOAT,
    @ProfundidadCama FLOAT,
    @FormulaCama float
)
AS
INSERT INTO CAMAS(DESCRIPCION_CAM,AREA_CAM,LARGO_CAM,
ANCHO_CAM,PROFUNDIDAD_CAM,FORMULA_CAM)
VALUES(@DescripcionCama,@AreaCama,
@LargoCama,@AnchoCama,@ProfundidadCama,
@FormulaCama)
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoSiembra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_IngresoSiembra]
(
    @DescripcionSiembra VARCHAR(50),
    @CodigoPlantaProduccion int=0,
    @CodigoCama int=0,
    @FechaSiembra datetime

```



```

        @FormulaSiembra FLOAT
    )
AS
INSERT INTO Siembra(
DESC_SIE,
ID_PLP,
ID_CAM,
FECHA_SIE,
AREA_SIE,
FORMULA_SIE
)
VALUES(
@DescripcionSiembra,
@CodigoPlantaProduccion,
@CodigoCama,
@FechaSiembra,
@AreaSiembra,
@FormulaSiembra)
SELECT @@IDENTITY AS "Identity"
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaSiembra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ActualizaSiembra]
(
    @CodigoSiembra int=0,
    @DescripcionSiembra VARCHAR(50),
    @CodigoPlantaProduccion int=0,
    @CodigoCama int=0,
    @FechaSiembra datetime,
    @AreaSiembra FLOAT,
    @FormulaSiembra FLOAT
)
AS
update Siembra
set DESC_SIE=@DescripcionSiembra,
ID_PLP=@CodigoPlantaProduccion,
ID_CAM=@CodigoCama,
FECHA_SIE=@FechaSiembra,
AREA_SIE=@AreaSiembra,
FORMULA_SIE=@FormulaSiembra

```

```

FECHA_SIE=@FechaSiembra,
AREA_SIE=@AreaSiembra,
FORMULA_SIE=@FormulaSiembra
where ID_SIE=@CodigoSiembra
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaDetalleSiembra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'create PROCEDURE
[dbo].[SP_EliminaDetalleSiembra]
(
    @CodigoSiembra numeric
)
AS
DELETE FROM DETALLESIEMBRA
WHERE ID_SIE=@CodigoSiembra'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoDetalleSiembra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoDetalleSiembra]
    @CodigoSiembra integer,
    @CodigoClaseClavel as integer,
    @Cantidad as integer,
    @Costo as Float,
    @Total as float
AS
INSERT INTO DETALLESIEMBRA(
    ID_SIE,
    ID_CCL,
    CANTIDAD_DSI,
    COSTO_UNIT_DSI

```

```

    )
    VALUES (
        @CodigoSiembra,
        @CodigoClaseClavel,
        @Cantidad,
        @Costo,
        @Total
    )
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaPlantaProduccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ActualizaPlantaProduccion]
(
    @CodigoPlantaProduccion as int,
    @DenominacionPlantaProduccion as varchar(50),
    @DireccionPlantaProduccion as varchar(50),
    @TelefonoPlantaProduccion as varchar(13),
    @CultivoPlantaProduccion as varchar(30)
)
AS
UPDATE PLANTA_PRODUCCION
SET
DENOMINACION_PLP=@DenominacionPlantaProduccion,
DIRECCION_PLP=@DireccionPlantaProduccion,
TELEFONO_PLP=@TelefonoPlantaProduccion,
CULTIVO_PLP=@CultivoPlantaProduccion
WHERE ID_PLP=@CodigoPlantaProduccion'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaPlantaProduccion]') AND type in (N'P', N'PC'))
BEGIN

```

```

EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaPlantaProduccion]
(
    @CodigoPlantaProduccion as int
)
AS
DELETE FROM PLANTA_PRODUCCION
WHERE ID_PLP=@CodigoPlantaProduccion'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaPlantaProduccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ListaPlantaProduccion]
(
    @DenominacionPlantaProduccion as varchar(50)
)
AS
    Select ID_PLP,DENOMINACION_PLP,
DIRECCION_PLP,TELEFONO_PLP,CULTIVO_PLP from
PLANTA_PRODUCCION
    WHERE DENOMINACION_PLP LIKE "%"+
@DenominacionPlantaProduccion + "%"
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoPlantaProduccion]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoPlantaProduccion]
(
    @DenominacionPlantaProduccion varchar(50),
    @DireccionPlantaProduccion varchar(50),
    @TelefonoPlantaProduccion varchar(13),
    @CultivoPlantaProduccion varchar(30)
)

```

```

AS
    INSERT INTO PLANTA_PRODUCCION(
        DENOMINACION_PLP,
        DIRECCION_PLP,
        TELEFONO_PLP,
        CULTIVO_PLP)
    VALUES(
        @DenominacionPlantaProduccion,
        @DireccionPlantaProduccion,
        @TelefonoPlantaProduccion,
        @CultivoPlantaProduccion)
    ,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ListaSiembra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ListaSiembra]
    @DescripcionSiembra varchar(50)=""

AS
SELECT
SIEMBRA.ID_SIE,
SIEMBRA.DESC_SIE,
SIEMBRA.ID_PLP,
PLANTA_PRODUCCION.DENOMINACION_PLP,
SIEMBRA.ID_CAM,
CAMAS.DESCRIPCION_CAM,
SIEMBRA.FECHA_SIE,
SIEMBRA.AREA_SIE,
SIEMBRA.FORMULA_SIE
FROM SIEMBRA
INNER JOIN PLANTA_PRODUCCION ON SIEMBRA.ID_PLP =
PLANTA_PRODUCCION.ID_PLP
INNER JOIN CAMAS ON SIEMBRA.ID_CAM = CAMAS.ID_CAM
WHERE DESC_SIE LIKE "%" + @DescripcionSiembra + "%"
END
GO
SET ANSI_NULLS ON
GO

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ConsultaSiembra]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_ConsultaSiembra]
    @CodigoSiembra int=0
AS
SELECT
SIEMBRA.ID_SIE,
SIEMBRA.DESC_SIE,
SIEMBRA.ID_PLP,
PLANTA_PRODUCION.DENOMINACION_PLP,
SIEMBRA.ID_CAM,
CAMAS.DESCRIPCION_CAM,
SIEMBRA.FECHA_SIE,
SIEMBRA.AREA_SIE,
SIEMBRA.FORMULA_SIE
FROM SIEMBRA
INNER JOIN PLANTA_PRODUCION ON SIEMBRA.ID_PLP =
PLANTA_PRODUCION.ID_PLP
INNER JOIN CAMAS ON SIEMBRA.ID_CAM = CAMAS.ID_CAM
WHERE SIEMBRA.ID_SIE=@CodigoSiembra'
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoPedido]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoPedido]
(
    @DescripcionPedido VARCHAR(50),
    @CodigoCliente int=0,
    @FechaPedido datetime
)
AS
INSERT INTO Pedidos(
DESCRIPCION_PED,
ID_CLI,

```

```

VALUES(
@DescripcionPedido,
@CodigoCliente,
@FechaPedido
)
SELECT @@IDENTITY AS "Identity"
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_ActualizaPedido]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'
CREATE PROCEDURE [dbo].[SP_ActualizaPedido]
(
    @CodigoPedido int=0,
    @DescripcionPedido VARCHAR(50),
    @CodigoCliente int=0,
    @FechaPedido datetime
)
AS
update Pedidos
set DESCRIPCION_PED=@DescripcionPedido,
ID_CLI=@CodigoCliente,
FECHA_PED=@FechaPedido
where ID_PED=@CodigoPedido
,

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_IngresoDetallePedido]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_IngresoDetallePedido]
    @CodigoPedido integer,
    @CodigoVariacion integer,

```

```

        @Costo as Float,
        @Total as float
AS
    INSERT INTO DETALLEPEDIDO(
        ID_PED,
        ID_VAR,
        CANTIDAD_DPE,
        COSTO_DPE,
        VTOTAL_DPE
    )
    VALUES (
        @CodigoPedido,
        @CodigoVariedad,
        @Cantidad,
        @Costo,
        @Total
    )
,
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[SP_EliminaDetallePedido]') AND type in (N'P', N'PC'))
BEGIN
EXEC dbo.sp_executesql @statement = N'CREATE PROCEDURE
[dbo].[SP_EliminaDetallePedido]
(
    @CodigoPedido numeric
)
AS
DELETE FROM DETALLEPEDIDO
WHERE ID_PED=@CodigoPedido'
END
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEF_FERTILIZA_FERTILIZ]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLEFERTILIZACION]'))
ALTER TABLE [dbo].[DETALLEFERTILIZACION] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEF_FERTILIZA_FERTILIZ] FOREIGN
KEY([ID_FER])
REFERENCES [dbo].[FERTILIZACION] ([ID_FER])

```



```

ALTER TABLE [dbo].[DETALLEFERTILIZACION] CHECK CONSTRAINT
[FK_DETALLEF_FERTILIZA_FERTILIZ]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEF_RELATIONS_ELEMENTO]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLEFERTILIZACION]'))
ALTER TABLE [dbo].[DETALLEFERTILIZACION] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEF_RELATIONS_ELEMENTO] FOREIGN
KEY([ID_ELE])
REFERENCES [dbo].[ELEMENTOS] ([ID_ELE])
GO
ALTER TABLE [dbo].[DETALLEFERTILIZACION] CHECK CONSTRAINT
[FK_DETALLEF_RELATIONS_ELEMENTO]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEC_RELATIONS_CONSTRUC]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLECONSTRUCCION]'))
ALTER TABLE [dbo].[DETALLECONSTRUCCION] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEC_RELATIONS_CONSTRUC] FOREIGN
KEY([ID_CON])
REFERENCES [dbo].[CONSTRUCCION] ([ID_CON])
GO
ALTER TABLE [dbo].[DETALLECONSTRUCCION] CHECK CONSTRAINT
[FK_DETALLEC_RELATIONS_CONSTRUC]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEC_RELATIONS_MATERIAL]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLECONSTRUCCION]'))
ALTER TABLE [dbo].[DETALLECONSTRUCCION] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEC_RELATIONS_MATERIAL] FOREIGN
KEY([ID_MCS])
REFERENCES [dbo].[MATERIALESCONSTRUCCION] ([ID_MCS])
GO
ALTER TABLE [dbo].[DETALLECONSTRUCCION] CHECK CONSTRAINT
[FK_DETALLEC_RELATIONS_MATERIAL]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_TAMANIO_COSTO_TAM_COSTO]') AND
parent_object_id = OBJECT_ID(N'[dbo].[TAMANIO]'))
ALTER TABLE [dbo].[TAMANIO] WITH NOCHECK ADD CONSTRAINT
[FK_TAMANIO_COSTO_TAM_COSTO] FOREIGN KEY([ID_COS])
REFERENCES [dbo].[COSTO] ([ID_COS])
GO

```

```

GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_VARIEDAD_CLASE_CLA_CLASE_CL]') AND
parent_object_id = OBJECT_ID(N'[dbo].[VARIEDAD]'))
ALTER TABLE [dbo].[VARIEDAD] WITH NOCHECK ADD CONSTRAINT
[FK_VARIEDAD_CLASE_CLA_CLASE_CL] FOREIGN KEY([ID_CCL])
REFERENCES [dbo].[CLASE_CLAVEL] ([ID_CCL])
GO
ALTER TABLE [dbo].[VARIEDAD] CHECK CONSTRAINT
[FK_VARIEDAD_CLASE_CLA_CLASE_CL]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_VARIEDAD_TAMANIO_V_TAMANIO]') AND
parent_object_id = OBJECT_ID(N'[dbo].[VARIEDAD]'))
ALTER TABLE [dbo].[VARIEDAD] WITH NOCHECK ADD CONSTRAINT
[FK_VARIEDAD_TAMANIO_V_TAMANIO] FOREIGN KEY([ID_TAM])
REFERENCES [dbo].[TAMANIO] ([ID_TAM])
GO
ALTER TABLE [dbo].[VARIEDAD] CHECK CONSTRAINT
[FK_VARIEDAD_TAMANIO_V_TAMANIO]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEP_RELATIONS_POSTCOSE]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLEPOSTCOSECHA]'))
ALTER TABLE [dbo].[DETALLEPOSTCOSECHA] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEP_RELATIONS_POSTCOSE] FOREIGN
KEY([ID_POS])
REFERENCES [dbo].[POSTCOSECHA] ([ID_POS])
GO
ALTER TABLE [dbo].[DETALLEPOSTCOSECHA] CHECK CONSTRAINT
[FK_DETALLEP_RELATIONS_POSTCOSE]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEP_VARIEDADE_VARIEDAD]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLEPOSTCOSECHA]'))
ALTER TABLE [dbo].[DETALLEPOSTCOSECHA] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEP_VARIEDADE_VARIEDAD] FOREIGN
KEY([ID_VARI])
REFERENCES [dbo].[VARIEDAD] ([ID_VARI])
GO
ALTER TABLE [dbo].[DETALLEPOSTCOSECHA] CHECK CONSTRAINT
[FK_DETALLEP_VARIEDADE_VARIEDAD]
GO

```

```

IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLE__RELATIONS_VENTAS]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLE_DVE]'))
ALTER TABLE [dbo].[DETALLE_DVE] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLE__RELATIONS_VENTAS] FOREIGN
KEY([ID_VEN])
REFERENCES [dbo].[VENTAS] ([ID_VEN])
GO
ALTER TABLE [dbo].[DETALLE_DVE] CHECK CONSTRAINT
[FK_DETALLE__RELATIONS_VENTAS]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLE_DVE_VARIEDAD]') AND parent_object_id
= OBJECT_ID(N'[dbo].[DETALLE_DVE]'))
ALTER TABLE [dbo].[DETALLE_DVE] WITH CHECK ADD CONSTRAINT
[FK_DETALLE_DVE_VARIEDAD] FOREIGN KEY([ID_VARI])
REFERENCES [dbo].[VARIEDAD] ([ID_VARI])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEP_RELATIONS_PEDIDOS]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLEPEDIDO]'))
ALTER TABLE [dbo].[DETALLEPEDIDO] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEP_RELATIONS_PEDIDOS] FOREIGN
KEY([ID_PED])
REFERENCES [dbo].[PEDIDOS] ([ID_PED])
GO
ALTER TABLE [dbo].[DETALLEPEDIDO] CHECK CONSTRAINT
[FK_DETALLEP_RELATIONS_PEDIDOS]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLEPEDIDO_VARIEDAD]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLEPEDIDO]'))
ALTER TABLE [dbo].[DETALLEPEDIDO] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLEPEDIDO_VARIEDAD] FOREIGN
KEY([ID_VAR])
REFERENCES [dbo].[VARIEDAD] ([ID_VARI])
GO
ALTER TABLE [dbo].[DETALLEPEDIDO] CHECK CONSTRAINT
[FK_DETALLEPEDIDO_VARIEDAD]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =

```

```

ALTER TABLE [dbo].[PRIVILEGIOS] WITH NOCHECK ADD CONSTRAINT
[FK_PRIVILEGIOS_MENU] FOREIGN KEY([ID_MENU])
REFERENCES [dbo].[MENU] ([ID_MENU])
GO
ALTER TABLE [dbo].[PRIVILEGIOS] CHECK CONSTRAINT
[FK_PRIVILEGIOS_MENU]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PRIVILEGIOS_USUARIO]') AND parent_object_id =
OBJECT_ID(N'[dbo].[PRIVILEGIOS]'))
ALTER TABLE [dbo].[PRIVILEGIOS] WITH NOCHECK ADD CONSTRAINT
[FK_PRIVILEGIOS_USUARIO] FOREIGN KEY([ID_USER])
REFERENCES [dbo].[USUARIO] ([ID_USER])
GO
ALTER TABLE [dbo].[PRIVILEGIOS] CHECK CONSTRAINT
[FK_PRIVILEGIOS_USUARIO]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PEDIDOS_CLIENTE]') AND parent_object_id =
OBJECT_ID(N'[dbo].[PEDIDOS]'))
ALTER TABLE [dbo].[PEDIDOS] WITH NOCHECK ADD CONSTRAINT
[FK_PEDIDOS_CLIENTE] FOREIGN KEY([ID_CLI])
REFERENCES [dbo].[CLIENTE] ([ID_CLI])
GO
ALTER TABLE [dbo].[PEDIDOS] CHECK CONSTRAINT
[FK_PEDIDOS_CLIENTE]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_POSTCOSE_PLANTA_PR_PLANTA_P]') AND
parent_object_id = OBJECT_ID(N'[dbo].[POSTCOSECHA]'))
ALTER TABLE [dbo].[POSTCOSECHA] WITH NOCHECK ADD
CONSTRAINT [FK_POSTCOSE_PLANTA_PR_PLANTA_P] FOREIGN
KEY([ID_PLP])
REFERENCES [dbo].[PLANTA_PRODUECCION] ([ID_PLP])
GO
ALTER TABLE [dbo].[POSTCOSECHA] CHECK CONSTRAINT
[FK_POSTCOSE_PLANTA_PR_PLANTA_P]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_POSTCOSE_POSTCOSEC_BODEGA]') AND
parent_object_id = OBJECT_ID(N'[dbo].[POSTCOSECHA]'))
ALTER TABLE [dbo].[POSTCOSECHA] WITH NOCHECK ADD
CONSTRAINT [FK_POSTCOSE_POSTCOSEC_BODEGA] FOREIGN

```



```

GO
ALTER TABLE [dbo].[POSTCOSECHA] CHECK CONSTRAINT
[FK_POSTCOSE_POSTCOSEC_BODEGA]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLES_RELATIONS_SIEMBRA]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLES_SIEMBRA]'))
ALTER TABLE [dbo].[DETALLES_SIEMBRA] WITH NOCHECK ADD
CONSTRAINT [FK_DETALLES_RELATIONS_SIEMBRA] FOREIGN
KEY([ID_SIE])
REFERENCES [dbo].[SIEMBRA] ([ID_SIE])
GO
ALTER TABLE [dbo].[DETALLES_SIEMBRA] CHECK CONSTRAINT
[FK_DETALLES_RELATIONS_SIEMBRA]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_DETALLES_SIEMBRA_CLASE_CLAVEL]') AND
parent_object_id = OBJECT_ID(N'[dbo].[DETALLES_SIEMBRA]'))
ALTER TABLE [dbo].[DETALLES_SIEMBRA] WITH CHECK ADD
CONSTRAINT [FK_DETALLES_SIEMBRA_CLASE_CLAVEL] FOREIGN
KEY([ID_CCL])
REFERENCES [dbo].[CLASE_CLAVEL] ([ID_CCL])
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_SIEMBRA_RELATIONS_CAMAS]') AND
parent_object_id = OBJECT_ID(N'[dbo].[SIEMBRA]'))
ALTER TABLE [dbo].[SIEMBRA] WITH NOCHECK ADD CONSTRAINT
[FK_SIEMBRA_RELATIONS_CAMAS] FOREIGN KEY([ID_CAM])
REFERENCES [dbo].[CAMAS] ([ID_CAM])
GO
ALTER TABLE [dbo].[SIEMBRA] CHECK CONSTRAINT
[FK_SIEMBRA_RELATIONS_CAMAS]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_SIEMBRA_RELATIONS_PLANTA_P]') AND
parent_object_id = OBJECT_ID(N'[dbo].[SIEMBRA]'))
ALTER TABLE [dbo].[SIEMBRA] WITH NOCHECK ADD CONSTRAINT
[FK_SIEMBRA_RELATIONS_PLANTA_P] FOREIGN KEY([ID_PLP])
REFERENCES [dbo].[PLANTA_PRODUCION] ([ID_PLP])
GO
ALTER TABLE [dbo].[SIEMBRA] CHECK CONSTRAINT
[FK_SIEMBRA_RELATIONS_PLANTA_P]
GO

```

```

IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_FERTILIZ_FERTILIZA_SIEMBRA]') AND
parent_object_id = OBJECT_ID(N'[dbo].[FERTILIZACION]'))
ALTER TABLE [dbo].[FERTILIZACION] WITH NOCHECK ADD
CONSTRAINT [FK_FERTILIZ_FERTILIZA_SIEMBRA] FOREIGN
KEY([ID_SIE])
REFERENCES [dbo].[SIEMBRA] ([ID_SIE])
GO
ALTER TABLE [dbo].[FERTILIZACION] CHECK CONSTRAINT
[FK_FERTILIZ_FERTILIZA_SIEMBRA]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_COMPRAS_PLANTA_PR_PLANTA_P]') AND
parent_object_id = OBJECT_ID(N'[dbo].[COMPRAS]'))
ALTER TABLE [dbo].[COMPRAS] WITH NOCHECK ADD CONSTRAINT
[FK_COMPRAS_PLANTA_PR_PLANTA_P] FOREIGN KEY([ID_PLP])
REFERENCES [dbo].[PLANTA_PRODUCCION] ([ID_PLP])
GO
ALTER TABLE [dbo].[COMPRAS] CHECK CONSTRAINT
[FK_COMPRAS_PLANTA_PR_PLANTA_P]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_CONSTRUC_RELATIONS_PLANTA_P]') AND
parent_object_id = OBJECT_ID(N'[dbo].[CONSTRUCCION]'))
ALTER TABLE [dbo].[CONSTRUCCION] WITH NOCHECK ADD
CONSTRAINT [FK_CONSTRUC_RELATIONS_PLANTA_P] FOREIGN
KEY([ID_PLP])
REFERENCES [dbo].[PLANTA_PRODUCCION] ([ID_PLP])
GO
ALTER TABLE [dbo].[CONSTRUCCION] CHECK CONSTRAINT
[FK_CONSTRUC_RELATIONS_PLANTA_P]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_EMPLEADO_PLANTA_PR_PLANTA_P]') AND
parent_object_id = OBJECT_ID(N'[dbo].[EMPLEADO]'))
ALTER TABLE [dbo].[EMPLEADO] WITH NOCHECK ADD CONSTRAINT
[FK_EMPLEADO_PLANTA_PR_PLANTA_P] FOREIGN KEY([ID_PLP])
REFERENCES [dbo].[PLANTA_PRODUCCION] ([ID_PLP])
GO
ALTER TABLE [dbo].[EMPLEADO] CHECK CONSTRAINT
[FK_EMPLEADO_PLANTA_PR_PLANTA_P]
GO
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =
OBJECT_ID(N'[dbo].[FK_PROVEEDO_PLANTA_PR_PLANTA_P]') AND

```

```
ALTER TABLE [dbo].[PROVEEDOR] WITH NOCHECK ADD CONSTRAINT  
[FK_PROVEEDO_PLANTA_PR_PLANTA_P] FOREIGN KEY([ID_PLP])  
REFERENCES [dbo].[PLANTA_PRODUCION] ([ID_PLP])  
GO  
ALTER TABLE [dbo].[PROVEEDOR] CHECK CONSTRAINT  
[FK_PROVEEDO_PLANTA_PR_PLANTA_P]  
GO  
IF NOT EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id =  
OBJECT_ID(N'[dbo].[FK_VENTAS_RELATIONS_PLANTA_P]') AND  
parent_object_id = OBJECT_ID(N'[dbo].[VENTAS]'))  
ALTER TABLE [dbo].[VENTAS] WITH NOCHECK ADD CONSTRAINT  
[FK_VENTAS_RELATIONS_PLANTA_P] FOREIGN KEY([ID_PLP])  
REFERENCES [dbo].[PLANTA_PRODUCION] ([ID_PLP])  
GO  
ALTER TABLE [dbo].[VENTAS] CHECK CONSTRAINT  
[FK_VENTAS_RELATIONS_PLANTA_P]
```


INTERFACES DEL SISTEMA

PANTALLA DE ACCESO AL SISTEMA

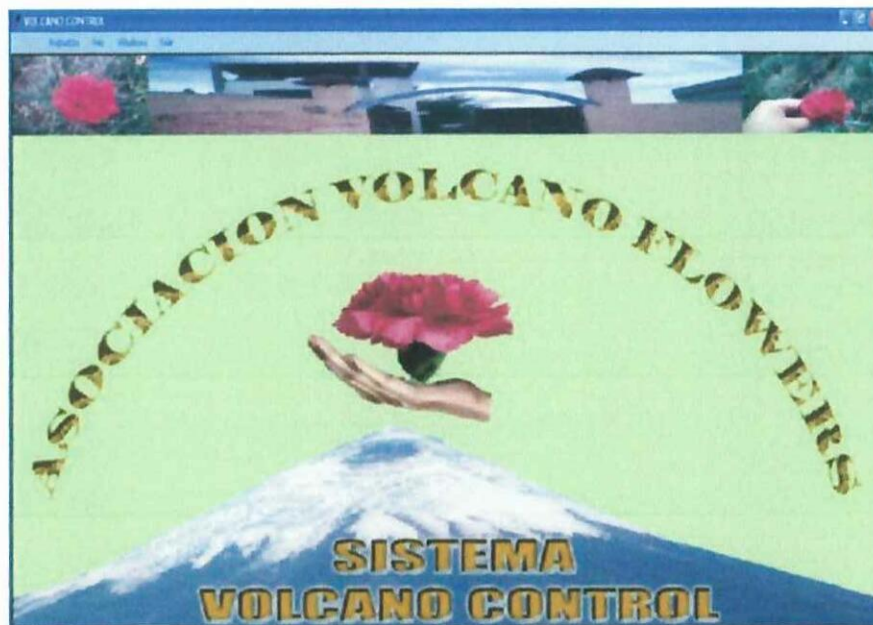


The screenshot shows a Windows-style dialog box titled "ACCESO AL SISTEMA". On the left is a graphic of a green hand holding a teardrop-shaped lens that reflects a landscape with a volcano and a river. On the right, there are two text input fields: "Nombre de usuario" with the text "Administrador/Volcano" and "Contraseña" with masked characters. Below the fields are two buttons: "Aceptar" and "Cancelar".

Nombre Usuario: Ingresar el nombre de un Usuario existente en la base de datos.

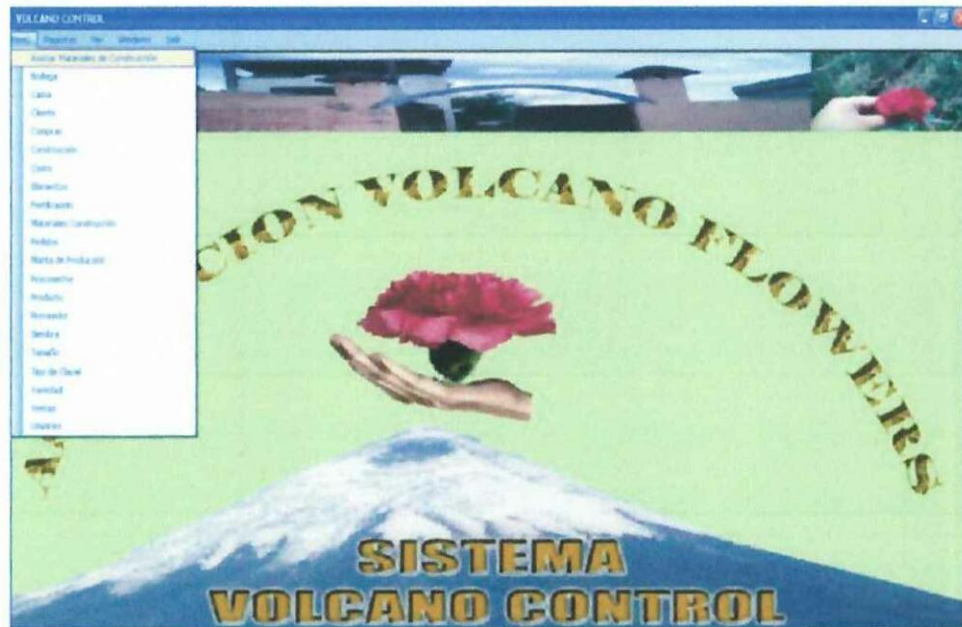
Contraseña: Ingresar la contraseña correspondiente al usuario que intenta Ingresar.

PANTALLA PRINCIPAL DEL SISTEMA



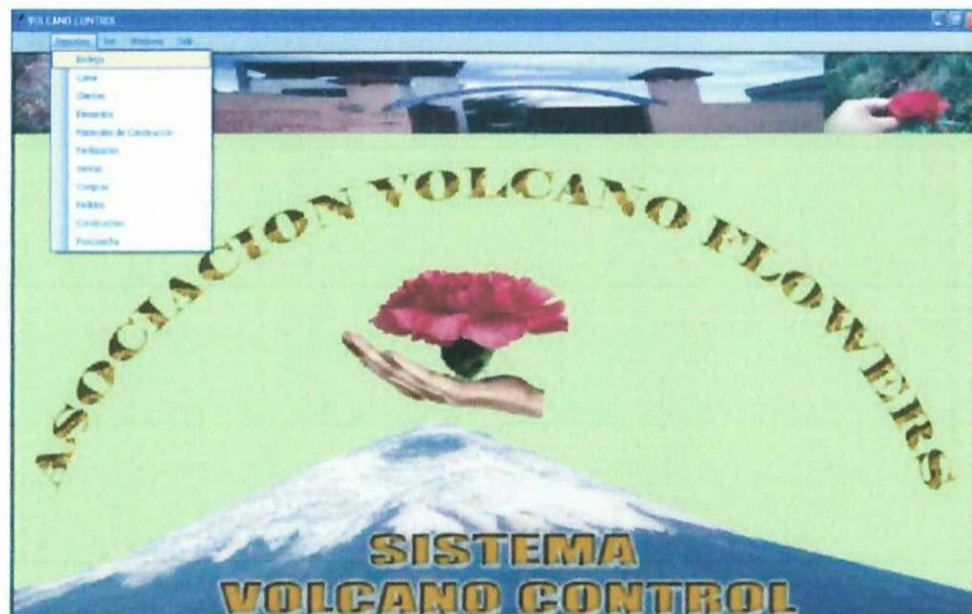
Despliega las opciones de menú que estarán habilitadas en base a los permisos que se les otorgó de acuerdo al tipo de usuario

PANTALLA PRINCIPAL DEL SISTEMA DEL ADMINISTRADOR



Despliega las opciones del Menú, permitiendo realizar todo tipo de operación contemplada.

PANTALLA PRINCIPAL DEL SISTEMA DEL USUARIO



Tiene privilegios únicamente de Usuario Invitado pudiendo realizar únicamente

PANTALLA ASOCIAR MATERIALES DE CONSTRUCCIÓN

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción del material de Construcción en el campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

ASOCIAR MATERIALES DE CONSTRUCCIÓN

Búsqueda | Mantenimiento

DESCRIPCIÓN:

Nuevo Consultar

SECCIÓN MANTENIMIENTO

ASOCIAR MATERIALES DE CONSTRUCCIÓN

Búsqueda | Mantenimiento

Construcción: XXXX Área: 12

Planta de Producción: Muboven Fecha: 05/12/2008

Material: Cemento Precio Unitario: Cantidad: Sub total:

Agregar

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar datos requeridos en la cabecera del mantenimiento
3. Seleccionar un elemento del combo Material
4. Ingresar la Cantidad
5. Clic sobre el botón Agregar
6. Repetir pasos 3 al 6 en base a los materiales que desee asociar
7. Muestra los datos añadidos en la grid
8. Click sobre el botón **Guardar**
9. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
10. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR:

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa

PANTALLA MANTENIMIENTO DE BODEGA

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción de la Bodega en el campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

The screenshot shows a window titled 'BODEGA' with a tabbed interface. The 'Búsqueda' tab is active. The title 'MANTENIMIENTO DE BODEGA' is displayed in red. Below the tabs, there is a 'DESCRIPCIÓN' label and an empty text input field. Underneath the input field are two buttons: 'Nuevo' and 'Consultar'. Below these buttons is a table with two columns: 'ELIMINAR' and 'DESCRIPCIÓN'. The table contains five rows of data, each with a delete icon in the 'ELIMINAR' column and a description in the 'DESCRIPCIÓN' column.

ELIMINAR	DESCRIPCIÓN
	Bodega
	Bodega1
	Bodega2
	Bodega3
	ALMN
	Prueba

SECCIÓN MANTENIMIENTO

The screenshot shows the same window as above, but the 'Mantenimiento' tab is active. The title 'MANTENIMIENTO DE BODEGA' is displayed in red. Below the tabs, there is a 'DESCRIPCIÓN' label and an empty text input field. Underneath the input field are two buttons: 'Guardar' and 'Cancelar'.

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar el campo Descripción
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA MANTENIMIENTO DE CAMAS

SECCIÓN BÚSQUEDA

PASOS:

5. Ingresar la descripción de la cama en el campo **Descripción**
6. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

DESCRIPCIÓN:

ÁREA:

ELIMINAR	DESCRIPCIÓN	ÁREA	LARGO	ANCHO	PROFUNDIDAD	FÓRMULA
borrar	camas2	120	12	10	0.5	
borrar	Cama	1	1	1	1	

SECCIÓN MANTENIMIENTO

DESCRIPCIÓN:

LARGO:

ANCHO:

PROFUNDIDAD:

ÁREA:

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

CLIENTES

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la cédula del cliente en el campo **Cédula**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

CLIENTE

MANTENIMIENTO DE CLIENTES

Búsqueda Mantenimiento

CÉDULA:

ELIMINAR	CÉDULA	NOMBRE	APELLIDO	NOMBRE COMERCIAL	DIRECCIÓN	TELÉFONO	PAÍS
	0500454002	miuro	soria	miro	dada	123	Ec
	0500454008	nombre	apellido	nombre comercial	direccion	1235	ec

SECCIÓN MANTENIMIENTO

CLIENTE

MANTENIMIENTO DE CLIENTES

Búsqueda Mantenimiento

CÉDULA:

NOMBRE:

APELLIDO:

NOMBRE COMERCIAL:

DIRECCIÓN:

TELÉFONO:

PAÍS:

MERCADO:

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

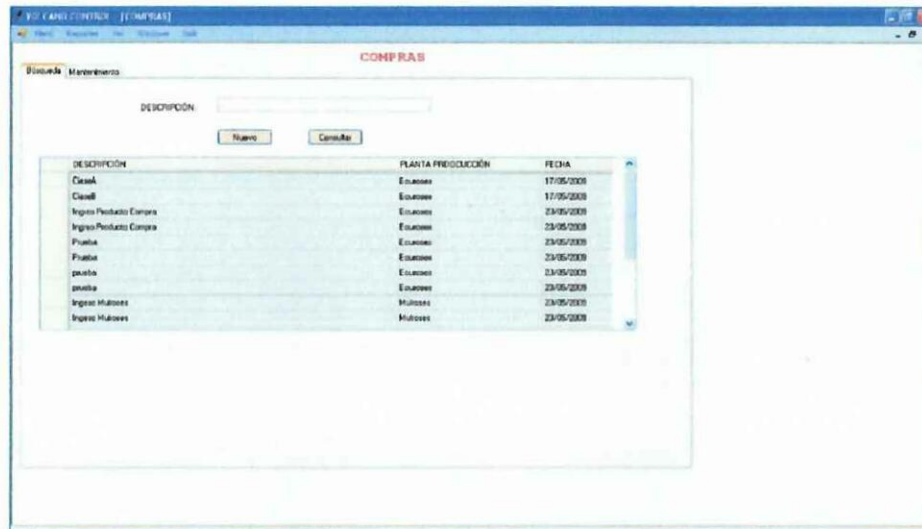
1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA COMPRAS

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción de la compra en el campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



DESCRIPCIÓN	PLANTA PRODUCCIÓN	FECHA
Casak	Equinox	17/05/2009
Casak	Equinox	17/05/2009
Ingreso Producto Emisor	Equinox	23/05/2009
Ingreso Producto Compra	Equinox	23/05/2009
Prueba	Equinox	23/05/2009
Prueba	Equinox	23/05/2009
prueba	Equinox	23/05/2009
prueba	Equinox	23/05/2009
Ingreso Muebles	Muebles	23/05/2009
Ingreso Muebles	Muebles	23/05/2009

SECCIÓN MANTENIMIENTO



Planta de Producción: Equinox

Descripción:

Fecha: Jueves, 14 de Enero de 2013 14:00

Detalle:

Nombre del Producto	Cantidad	Precio Unitario	Sub total
Quintones			

Eliminar

Agregar

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar datos requeridos en la cabecera del mantenimiento
3. Seleccionar un elemento del combo Producto
4. Ingresar la Cantidad
5. Clic sobre el botón Agregar
6. Repetir pasos 3 al 5 en base a los materiales que desee asociar
7. Muestra los datos añadidos en la grid
8. Click sobre el botón **Guardar**
9. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
10. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

7. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
8. Al ejecutar la consulta muestra los datos asociados
9. Modificar los datos que se desee
10. Click sobre el botón **Guardar**
11. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
12. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

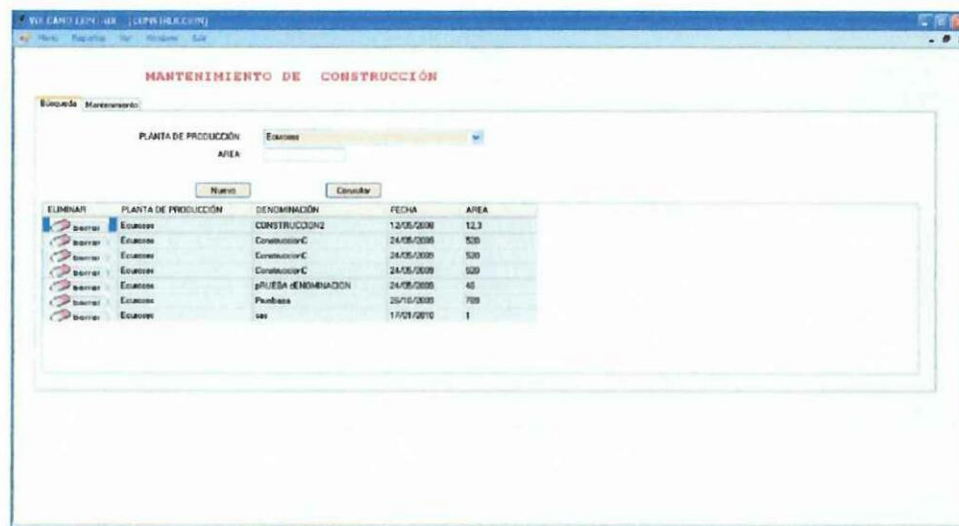
1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa

PANTALLA CONSTRUCCIÓN

SECCIÓN BÚSQUEDA

PASOS:

1. Selecciona del Combo **Planta de Producción** el elemento que se desea filtrar
2. Ingresar el año al cual aplicará esa búsqueda
3. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



Mantenimiento de Construcción

Búsqueda: Mantenimiento

PLANTA DE PRODUCCIÓN: Escases

ÁREA:

Consultar

ELIMINAR	PLANTA DE PRODUCCIÓN	DENOMINACIÓN	FECHA	ÁREA
	Escases	CONSTRUCCION	12/05/2008	12.3
	Escases	ConstruccionC	24/05/2009	500
	Escases	ConstruccionC	24/05/2009	500
	Escases	ConstruccionC	24/05/2009	500
	Escases	planta de denominacion	24/05/2009	45
	Escases	Pantalla	25/05/2009	700
	Escases	ses	17/01/2010	1

SECCIÓN MANTENIMIENTO



CONSTRUCCION

Mantenimiento de Construcción

Búsqueda: Mantenimiento

PLANTA DE PRODUCCIÓN: Mulroses

DENOMINACIÓN:

FECHA: Jueves, 14 de Enero de 2010

ÁREA:

Guardar Cancelar

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

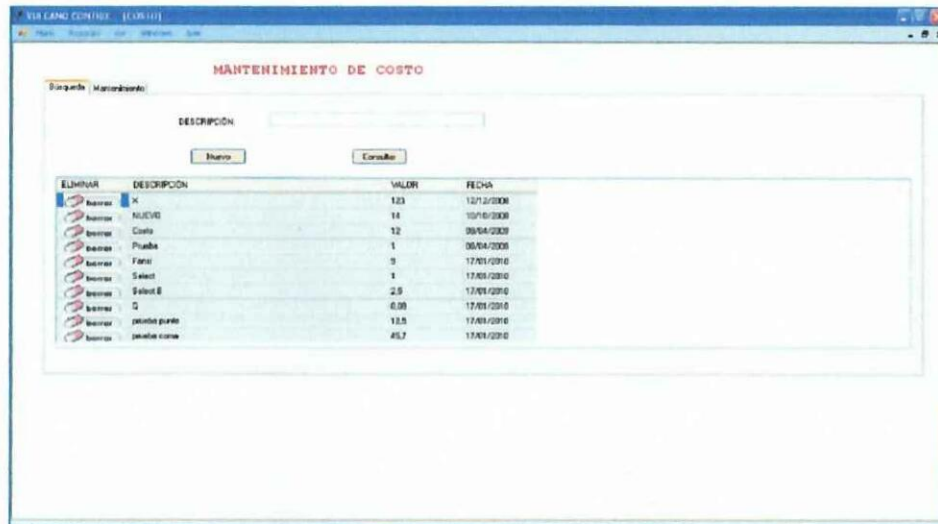
1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA COSTO

SECCIÓN BÚSQUEDA

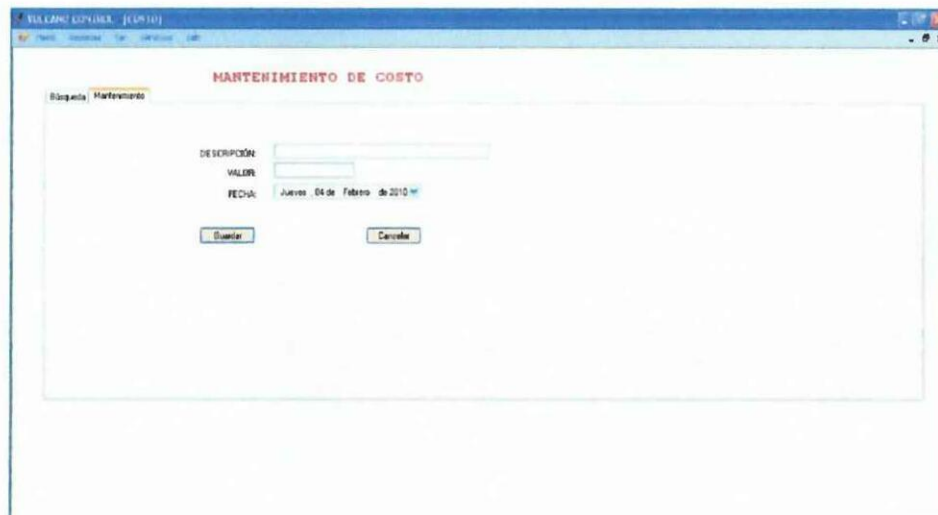
PASOS:

1. Ingresar la descripción del costo a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



ELIMINAR	DESCRIPCION	VALOR	FECHA
	M	123	12/12/2008
	NUOVO	14	10/10/2008
	Costo	12	08/04/2008
	Prueba	1	08/04/2008
	Vari	9	17/01/2010
	Select	1	17/01/2010
	Select B	2.9	17/01/2010
	Q	6.08	17/01/2010
	patente punto	12.5	17/01/2010
	patente coma	45.7	17/01/2010

SECCIÓN MANTENIMIENTO



PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA ELEMENTOS

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción del elemento a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

The screenshot shows a window titled 'ELEMENTOS' with a sub-header 'MANTENIMIENTO DE ELEMENTOS'. It has two tabs: 'Búsqueda' and 'Mantenimiento'. The 'Búsqueda' tab is active, showing a search form with a 'DESCRIPCIÓN' field and 'Buscar' and 'Consultar' buttons. Below the form is a table with the following data:

ELIMINAR	DESCRIPCIÓN	CANTIDAD	EXISTENCIA	COSTO
☐	elemento	1	14	3
☐	elemento	12	37	72
☐	elemento	60000	60000	60000
☐	elemento	60000	60000	60000
☐	500000000	500000000	500000000	500000000
☐	Prueba	800	800000	800
☐	Prueba	500000000	500000000	500000000
☐	prueba	4	5	12.5

SECCIÓN MANTENIMIENTO

The screenshot shows the same window as above, but with the 'Mantenimiento' tab active. It displays a form for adding or updating an element with the following fields:

DESCRIPCIÓN:

CANTIDAD:

EXISTENCIA:

COSTO:

Below the form are two buttons: 'Guardar' and 'Cancelar'.

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA FERTILIZACIÓN

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción de la fertilización a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

FERTILIZACIÓN	SIEMBRA	PLANTA PRODUCCION	CIMA	FECHA	AREA
Prueba A	Siembra L	Mulones	cam2	24/07/2009	124
PRUEBA B	siembra J	Mulones	Cam2	24/07/2009	700
H	SIEMBRA I	Escamos	Cam2	24/07/2009	120
hna	siembra C	EscamB	cam2	23/07/2009	500
expresos	siembra J	Mulones	Cam2	24/07/2009	700
Fa	Siembra h	Escamos	cam2	14/06/2009	12332
cha	Siembra h	Escamos	cam2	14/06/2009	12332
sa	Siembra h	Escamos	cam2	14/06/2009	12332
so	Siembra h	Escamos	cam2	14/06/2009	12332
son	Siembra h	Bonitos	cam2	14/06/2009	12332

SECCIÓN MANTENIMIENTO

FERTILIZACIÓN

Siembra: Cima:

Punto de Producción: Area:

Fecha: Descripción:

Detalle:

Descripción Detalle:	Fecha: Jueves 14 de Enero de 2009	Cantidad:	Existencia:
Elementos: Elementos	Precio Unitario:	Disponibilidad:	Sub total:

Eliminar:

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar datos requeridos en la cabecera del mantenimiento
3. Seleccionar un elemento del combo Producto
4. Ingresar los campos del Detalle
5. Repetir pasos 3 y 4 en base a los materiales que desee asociar
6. Muestra los datos añadidos en la grid
7. Click sobre el botón **Guardar**
8. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
9. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

MATERIALES DE CONSTRUCCIÓN

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción del material a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

The screenshot shows a window titled 'MANTENIMIENTO DE MATERIALES DE CONSTRUCCIÓN'. It has two tabs: 'Búsqueda' and 'Mantenimiento'. The 'Búsqueda' tab is active. It contains a text input field labeled 'DESCRIPCIÓN:' with a 'Consultar' button next to it. Below this is a table with the following data:

ELIMINAR	DESCRIPCIÓN	CANTIDAD MINIMA	CANTIDAD MÁXIMA	COSTO
	Cemento	25	200	10
	Cemento	500	10000	5.5
	Cemento	12	121	3.0
	Blanco	500	5000	5

SECCIÓN MANTENIMIENTO

The screenshot shows the same window as before, but the 'Mantenimiento' tab is active. It contains a form with the following fields and buttons:

DESCRIPCIÓN:

CANTIDAD MINIMA:

CANTIDAD MÁXIMA:

COSTO:

Guardar Cancelar

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA PEDIDOS

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción del pedido a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

The screenshot shows a web application window titled 'VOLCANO CONTROL - PEDIDOS'. The main heading is 'MANTENIMIENTO DE PEDIDOS'. Below it, there are tabs for 'Búsqueda' and 'Mantenimiento'. The 'Búsqueda' tab is active. It features a text input field labeled 'DESCRIPCIÓN' and two buttons: 'Nuevo' and 'Consultar'. Below the input field is a table with the following data:

DESCRIPCIÓN	NOMBRES	APELLIDOS	NOMBRE COMERCIAL	FECHA
Ingreso	maria	roa	maria	30/07/2008
Ingreso A	romero	apellido	romero comercial	30/07/2008
Ingreso A	romero	apellido	romero comercial	30/07/2008
Ingreso A	romero	apellido	romero comercial	30/07/2008
Ingreso H	maria	roa	maria	30/07/2008

SECCIÓN MANTENIMIENTO

The screenshot shows the same web application window, but the 'Mantenimiento' tab is active. It displays a form for adding or editing a record. The form includes fields for 'DESCRIPCIÓN', 'CLIENTE', and 'FECHA'. Below these, there are fields for 'VARIEDAD', 'TAMANO', 'CANTIDAD', 'PRECIO UNITARIO', and 'SUBTOTAL'. There are also buttons for 'Nuevo', 'Guardar', and 'Cancelar'. The 'Nuevo' button is highlighted.

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar datos requeridos en la cabecera del mantenimiento
3. Ingresar los campos del Detalle
4. Repetir pasos 2 y 3 en base a los materiales que desee asociar
5. Muestra los datos añadidos en la grid
6. Click sobre el botón **Guardar**
7. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
8. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA PLANTA DE PRODUCCIÓN

SECCIÓN BÚSQUEDA

PASOS:

3. Ingresar la descripción de la planta de producción a buscar en la campo **Descripción**
4. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

The screenshot shows a window titled 'MANTENIMIENTO DE PEDIDOS'. It has a search bar labeled 'DESCRIPCION' with a 'Buscar' button. Below the search bar is a table with the following data:

DESCRIPCION	NOMBRES	APELLIDOS	NOMBRE COMERCIAL	FECHA
Ingreso	juan	lopez	juan	30/05/2008
Ingreso A	carlos	rodriguez	carlos rodriguez	30/05/2008
Ingreso A	carlos	rodriguez	carlos rodriguez	30/05/2008
Ingreso H	juan	lopez	juan	30/05/2008

SECCIÓN MANTENIMIENTO

The screenshot shows a window titled 'MANTENIMIENTO DE MATERIALES DE CONSTRUCCIÓN'. It has a search bar labeled 'DESCRIPCION' and two buttons: 'Guardar' and 'Consultar'.

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA POSTCOSECHA

SECCIÓN BÚSQUEDA

PASOS:

1. Selecciona el criterio de búsqueda en el Combo Bodega
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

VER CARB. CORTADO [POSTCOSECHA]

MANTENIMIENTO DE POSTCOSECHA

Búsqueda: Mantenimiento

Bodega: Bodega1

Nuevo Consultar

BODEGA	PLANTA DE PRODUCCIÓN	FECHA
Bodega1	Economas	23/11/2009

SECCIÓN MANTENIMIENTO

VER CARB. CORTADO [POSTCOSECHA]

MANTENIMIENTO DE POSTCOSECHA

Búsqueda: Mantenimiento

Planta de Producción: Economas

Bodega: Bodega1

Fecha: Viernes, 05 de Febrero de 2010

Cantidad: 123

Costo: 123

Total: Agregar

Guardar Cancelar

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar datos requeridos en la cabecera del mantenimiento
3. Ingresar los campos del Detalle
4. Repetir pasos 2 y 3 en base a los materiales que desee asociar
5. Muestra los datos añadidos en la grid
6. Click sobre el botón **Guardar**
7. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
8. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

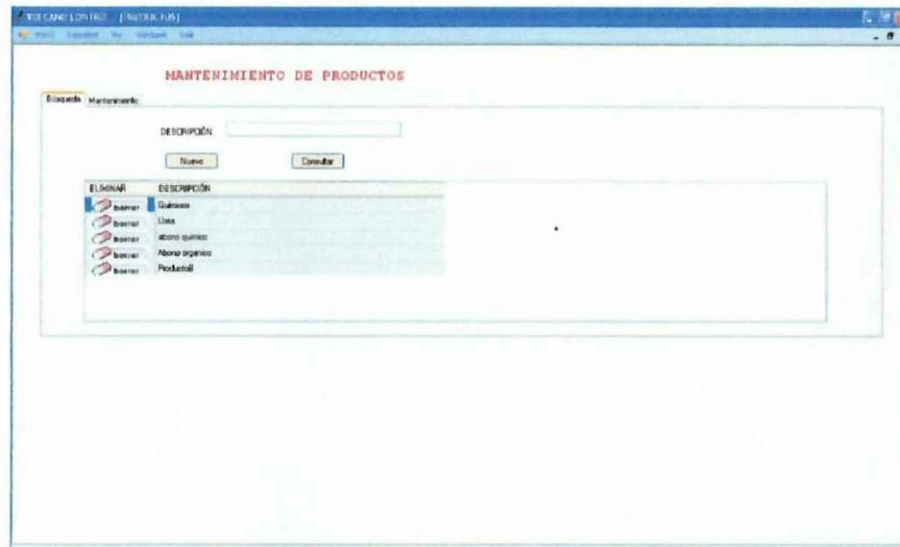
1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA PRODUCTOS

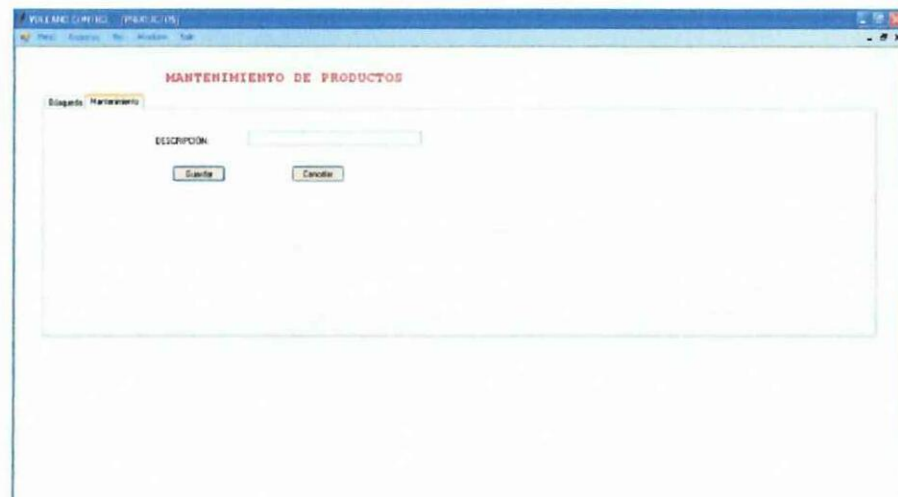
SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción del producto a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



SECCIÓN MANTENIMIENTO



PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA PROVEEDORES

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la cédula del proveedor a buscar en la campo **Cédula**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

The screenshot shows a web application window titled 'MANTENIMIENTO DE PROVEEDORES'. It features a search form with a 'CÉDULA' input field and 'Nuevo' and 'Consultar' buttons. Below the form is a table with the following data:

ELIMINAR	CÉDULA	NOMBRE	APELLIDO	DIRECCIÓN	TELÉFONO	PLANTA DE PRODUCCIÓN
	990000001	Proveedor	Salas	000000	0000	Ecosan
	990000002	MH	SS	0000	0000	Multisec
	990000003	SSS	SAS	0000	111112	Grandi
	12345	a	ds	0000	7	Ecosan

SECCIÓN MANTENIMIENTO

The screenshot shows the same web application window, but in the maintenance section. It features a form with the following fields and controls:

- Input fields for: CÉDULA, NOMBRE, APELLIDO, DIRECCIÓN, and TELÉFONO.
- A dropdown menu for 'PLANTA DE PRODUCCIÓN' with 'Ecosan' selected.
- 'Nuevo' and 'Consultar' buttons at the bottom.

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

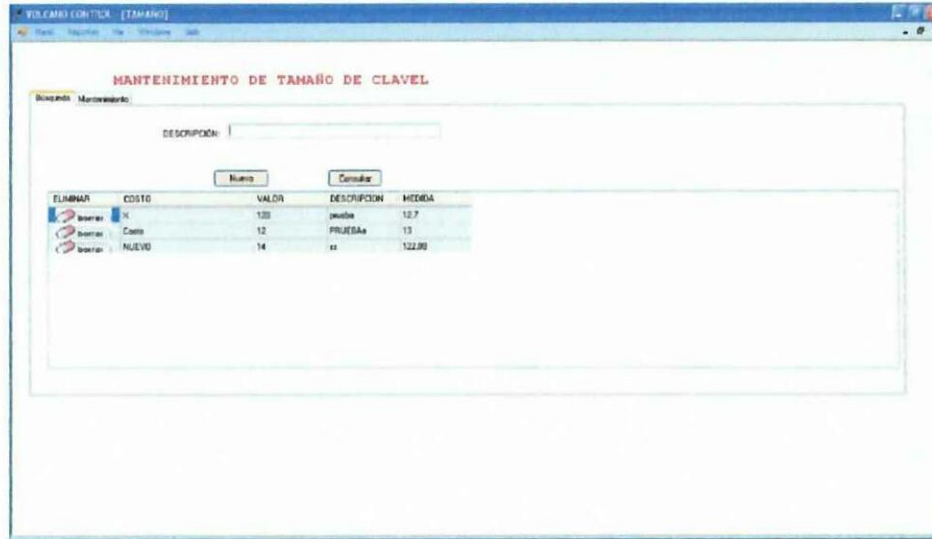
1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA TAMAÑO

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción del tipo de clavel a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



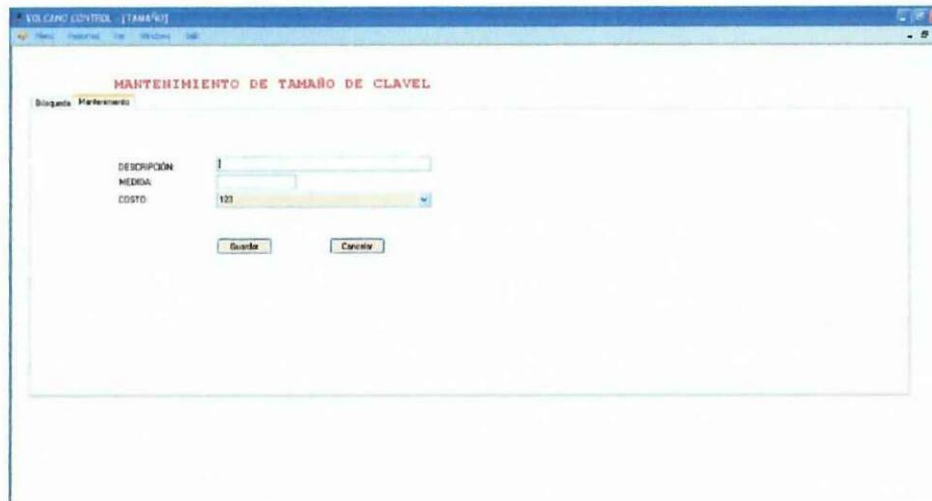
Mantenimiento de Tamaño de Clavel

Búsqueda: Mantenimiento

DESCRIPCION:

ELIMINAR	COSTO	VALOR	DESCRIPCION	MEDIDA
	1	120	pequeña	12.7
	Corta	12	PROBADA	13
	NUOVO	14	xx	122.80

SECCIÓN MANTENIMIENTO



Mantenimiento de Tamaño de Clavel

Búsqueda: Mantenimiento

DESCRIPCION:

MEDIDA:

COSTO:

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

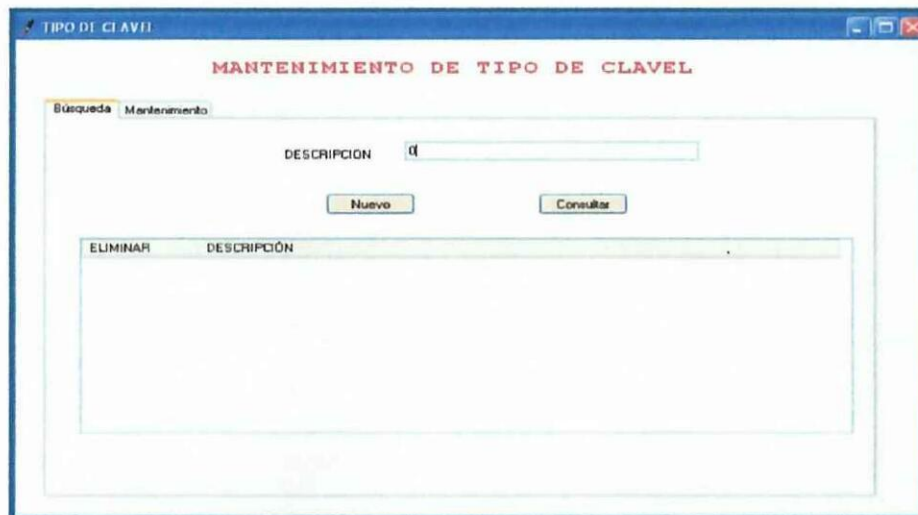
1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA TIPO DE CLAVEL

SECCIÓN BÚSQUEDA

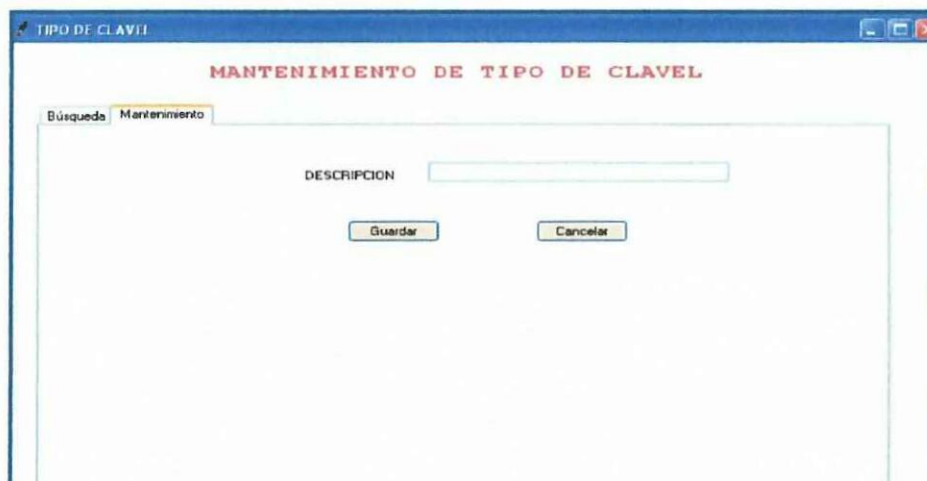
PASOS:

1. Ingresar la descripción del tipo de clavel a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



The screenshot shows a window titled 'TIPO DE CLAVEL' with a sub-header 'MANTENIMIENTO DE TIPO DE CLAVEL'. There are two tabs: 'Búsqueda' (selected) and 'Mantenimiento'. In the 'Búsqueda' tab, there is a text input field labeled 'DESCRIPCION' containing the letter 'a'. Below the input field are two buttons: 'Nuevo' and 'Consultar'. At the bottom of the window, there is a table with two columns: 'ELIMINAR' and 'DESCRIPCION'. The table is currently empty.

SECCIÓN MANTENIMIENTO



The screenshot shows the same window titled 'TIPO DE CLAVEL' with the sub-header 'MANTENIMIENTO DE TIPO DE CLAVEL'. The 'Mantenimiento' tab is now selected. It features a text input field labeled 'DESCRIPCION' which is empty. Below the input field are two buttons: 'Guardar' and 'Cancelar'.

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la descripción de la variedad a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

VO: CARI CON ROL (FABRICANTE)

Inicio | Registrar | Ver | Eliminar | Borrar

MANTENIMIENTO DE VARIEDAD DE CLAVEL

Búsqueda:

DESCRIPCION:

ELEMENTO	VARIEDAD	CLASE	DESCRIPCION TAMAÑO	TAMAÑO
Insertar	Prueba	clavel	prueba	12,7
Insertar	prueba	VERDE	PRUEBA	13
Insertar	prueba	AZUL	PRUEBA	13
Insertar	prueba	clavel	prueba	12,7
Insertar	ver	Rosa	PRUEBA	13
Insertar	y	clavel	prueba	12,7
Insertar	ff	AZUL	prueba	12,7
Insertar	mp	clavel	PRUEBA	13
Insertar	A	clavel	prueba	12,7
Insertar	el	clavel	prueba	12,7

SECCIÓN MANTENIMIENTO

MOLINO LITONER MANTENIMIENTO

MENU ACCIONES CON MANTENIMIENTO OTRO

MANTENIMIENTO DE VARIEDAD DE CLAVEL

Busqueda: Mantenimiento

DESCRIPCION

TAMANO 12.7

CLASE plant

Guardar Cancelar

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA VENTAS

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar la venta a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados

Descripción	Fecha	Valor	Cantidad	Valor	Total
Producto A	20/07/2008	120	1	120	120
Producto B	20/07/2008	120	1	120	120
Producto C	20/07/2008	120	1	120	120
Producto D	20/07/2008	120	1	120	120
Producto E	20/07/2008	120	1	120	120
Producto F	20/07/2008	120	1	120	120
Producto G	20/07/2008	120	1	120	120
Producto H	20/07/2008	120	1	120	120
Producto I	20/07/2008	120	1	120	120
Producto J	20/07/2008	120	1	120	120
Producto K	20/07/2008	120	1	120	120
Producto L	20/07/2008	120	1	120	120
Producto M	20/07/2008	120	1	120	120
Producto N	20/07/2008	120	1	120	120
Producto O	20/07/2008	120	1	120	120
Producto P	20/07/2008	120	1	120	120
Producto Q	20/07/2008	120	1	120	120
Producto R	20/07/2008	120	1	120	120
Producto S	20/07/2008	120	1	120	120
Producto T	20/07/2008	120	1	120	120
Producto U	20/07/2008	120	1	120	120
Producto V	20/07/2008	120	1	120	120
Producto W	20/07/2008	120	1	120	120
Producto X	20/07/2008	120	1	120	120
Producto Y	20/07/2008	120	1	120	120
Producto Z	20/07/2008	120	1	120	120

SECCIÓN MANTENIMIENTO

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar datos requeridos en la cabecera del mantenimiento
3. Ingresar los campos del Detalle
4. Repetir pasos 2 y 3 en base a los materiales que desee asociar
5. Muestra los datos añadidos en la grid
6. Click sobre el botón **Guardar**
7. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
8. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA USUARIO

SECCIÓN BÚSQUEDA

PASOS:

1. Ingresar el nombre de usuario a buscar en la campo **Descripción**
2. Click sobre el botón **Consultar** en caso de encontrar registros desplegará los datos asociados, caso contrario desplegará un mensaje que indica que no existen registros en con los criterios de búsqueda especificados



MANTENIMIENTO DE USUARIOS

Búsqueda Mantenimiento

DESCRIPCIÓN:

ELIMINAR	ICONO	NOMBRE	CONTRASEÑA
		luis	luis
		ViktorArbelyador	viktor2008

SECCIÓN MANTENIMIENTO



MANTENIMIENTO DE USUARIOS

Búsqueda Mantenimiento

DESCRIPCIÓN:

CONTRASEÑA:

PASOS INGRESO:

1. Click en el botón Nuevo
2. Ingresar los campos requeridos
3. Click sobre el botón **Guardar**
4. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
5. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

PASOS MODIFICACIÓN :

1. Click sobre la fila donde se encuentra el elemento a modificar a partir de la segunda columna la primera es utilizada para otros fines (Eliminar)
2. Al ejecutar la consulta muestra los datos asociados
3. Modificar los datos que se desee
4. Click sobre el botón **Guardar**
5. En caso de estar un campo vacío, desplegará un mensaje solicitando que sea llenado
6. Si guarda retornará un mensaje de transacción exitosa caso contrario desplegará un mensaje que indica que existió error al registrar

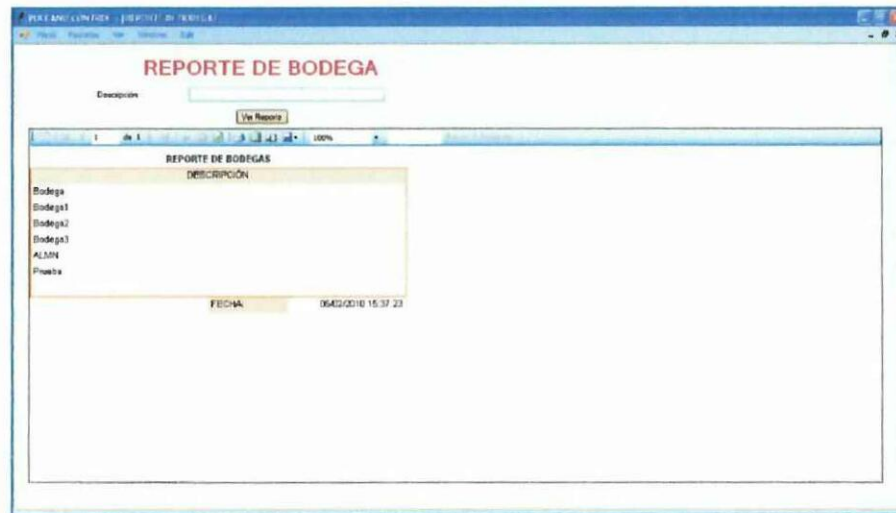
PASOS ELIMINAR :

1. Click sobre la primera columna del Grid donde se encuentra el elemento a eliminar
2. Muestra un mensaje de confirmación
3. Si pulsa SI el registro será borrado de la Base de Datos, mostrará un mensaje de transacción exitosa
4. Si pulsa NO el registro no sufrirá cambio alguno

PANTALLA REPORTE DE BODEGA

PASOS:

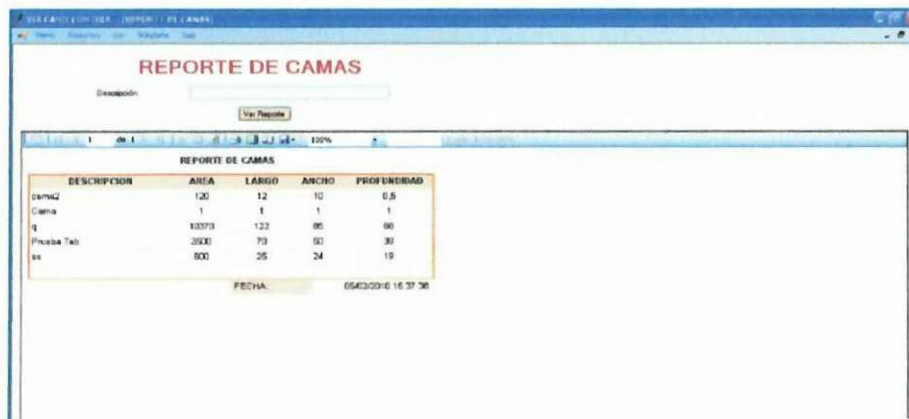
1. Ingresar la bodega a buscar en la campo **Descripción**
2. Seleccionar la fecha de búsqueda del campo **Fecha**
3. Click sobre el botón **Ver Reporte** en caso de encontrar registros desplegará los datos asociados



PANTALLA REPORTE DE CAMAS

PASOS:

1. Ingresar la cama a buscar en la campo **Descripción**
2. Click sobre el botón **Ver Reporte** en caso de encontrar registros desplegará los datos asociados



PANTALLA REPORTE DE CLIENTES

PASOS:

1. Ingresar la cédula del cliente en la campo **Cédula**
2. Click sobre el botón **Ver Reporte** en caso de encontrar registros desplegará los datos asociados

Windows 95 desktop environment. The active window is titled "Reporte de Clientes". It features a menu bar with "Archivo", "Formato", "Datos", "Ventana", "Herramientas", and "Ayuda". Below the menu is a toolbar with icons for file operations. The main content area includes a search box labeled "CEDULA" and a "Ver Pagina" button. Below this is a table titled "REPORTES DE CLIENTES" with the following data:

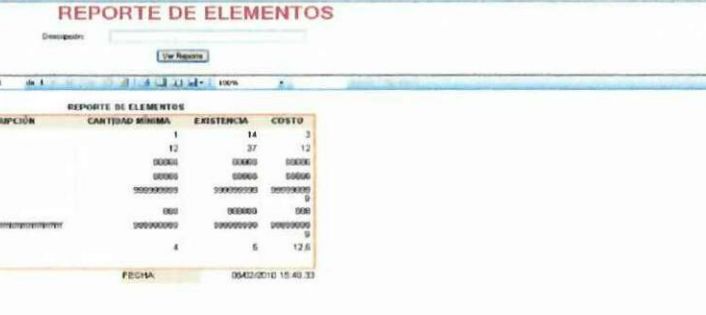
NUMERO	APELLIDO	NOMBRE COMERCIAL	DIRECCION	TELEFONO	PAIS	MERCADO
esmeralda	esmeralda	esmeralda	esmeralda	110	Esmeralda	Esmeralda
prado	prado	prado	prado	prado	prado	prado
romero	romero	romero comercial	romero	1136	romero	romero
lara	lara	lara	lara	01233	lara	lara

At the bottom right of the window, there is a "FECHA" field displaying "06/02/2010 16:30:02".

PANTALLA REPORTE DE ELEMENTOS

PASOS:

1. Ingresar el elemento a buscar en la campo **Descripción**
2. Click sobre el botón **Ver Reporte** en caso de encontrar registros desplegará los datos asociados



Reporte de Elementos

DESCRIPCION	CANTIDAD MINIMA	EXISTENCIA	COSTO
Elemento	1	14	3
elementoB	12	37	12
ElementoC	00000	00000	00000
elementoD	00000	00000	00000
0000000000	0000000000	0000000000	0000000000
Prueba	000	000000	0
0000000000	0000000000	0000000000	0000000000
prueba	4	5	12.6

FECHA: 09/01/2010 15:40:33

PANTALLA REPORTE DE MATERIALES DE CONSTRUCCIÓN

PASOS:

1. Ingresar el material a buscar en la campo **Descripción**
2. Click sobre el botón **Ver Reporte** en caso de encontrar registros desplegará los datos asociados

REPORTE DE CONSTRUCCIÓN Y MATERIALES EMPLEADOS

Área: Ver Reporte

DESCRIPCIÓN	FECHA	ÁREA	MATERIAL	CANTIDAD	COSTO	TOTAL
-------------	-------	------	----------	----------	-------	-------

FECHA: 05/02/2010 15:35:45

PANTALLA REPORTE DE FERTILIZACIÓN POR FECHA Y SIEMBRA

PASOS:

1. Ingresar el material a buscar en la campo **Descripción**
2. Seleccionar la fecha de búsqueda del campo **Fecha**
3. Click sobre el botón **Ver Reporte** en caso de encontrar registros desplegará los datos asociados

REPORTE DE FERTILIZACIÓN POR FECHA Y SIEMBRA

Siembra: Fecha: Domingo, 31 de Enero de 2010 Ver Reporte

SIEMBRA	FECHA	FERTILIZACIÓN	ELEMENTOS	CANTIDAD	COSTO	TOTAL
---------	-------	---------------	-----------	----------	-------	-------

FECHA: 05/02/2010 15:41:05