



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE CONTROL DE
VERIFICACIÓN DE PREDIOS**

TESIS PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
P R E S E N T A

VICTORIA RAMÍREZ NEY

DIRECTOR DE TESIS: M. EN C. MA. JAQUELINA LÓPEZ BARRIENTOS



MÉXICO, D.F.

NOVIEMBRE 2001



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

El haber concluido esta meta significa el esfuerzo de muchas personas que han estado a mi alrededor a lo largo de mi vida y que han confiado en mí, en especial a ti mi DIOS por darme la entereza y la fortaleza para llegar a este momento, GRACIAS por tu amor, tu paciencia, por brindarme la oportunidad de ser y por todo lo que tengo.

A MIS PADRES por su amor incondicional, su ejemplo, su esfuerzo, su confianza, su apoyo, GRACIAS por darme la oportunidad de ser quien soy, sin ustedes la vida hubiera sido muy triste.

A MI PADRE por que de él aprendí el valor de la vida, GRACIAS PAPI donde quiera que estés, hoy comprendo que te has ido al llamado del Señor . . . me hace falta tu calor, extraño tus palabras de aliento, de orgullo, de lucha, de esperanza, . . . fuiste bueno conmigo y así bueno te recuerdo, quiero agradecerte y decirte que yo NUNCA, NUNCA TE OLVIDE.

A la Sra. Minerva Martínez (mi Mami) por su amor Invaluable, por estar en mi vida GRACIAS.

A mis hermanos por que los amo con todo mi corazón, en especial a mi hermana Silvia porque ha sido un pilar en mi vida y me ha ayudado a lograr cada una de mis metas.

A mis amigas y amigos por apoyarme siempre.

A la M. en C. Ma. Jaquelina López Barrientos por su apoyo Incondicional, su paciencia y entusiasmo para lograr una más de mis metas.

A la Universidad Nacional Autónoma de México y mis profesores por prepararme, es un orgullo pertenecer a esta Institución y siempre lo tendré presente.

A Jorge Zacarías Martínez por haber estado en mi vida, ~ Te AMO con todo mi Corazón ~.

GRACIAS

ÍNDICE

1. INTRODUCCIÓN	1
2. FUNDAMENTOS TEÓRICOS	3
2.1 BASES DE DATOS.....	3
2.1.1 <i>Definiciones y conceptos de las Bases de Datos</i>	3
2.2 MODELO DE DATOS	3
2.2.1 <i>Normalización</i>	5
2.2.2 <i>Estructuras de Datos</i>	5
2.2.3 <i>Definiciones de Integridad</i>	6
2.2.4 <i>Operadores</i>	6
2.2.4.1 <i>Álgebra Relacional</i>	6
2.2.4.2 <i>Operadores de Actualización</i>	7
2.2.4.3 <i>Operaciones Primitivas</i>	7
2.2.5 <i>Conceptos de los Sistemas Administradores de Base de Datos (DBMS)</i>	7
2.2.5.1 <i>Componentes Funcionales de un DBMS</i>	8
2.2.5.2 <i>Estructuras de datos de un DBMS</i>	8
2.2.5.3 <i>Características de un Sistema Manejador de Base de Datos Relacional (RDBMS)</i>	9
2.2.5.4 <i>Diccionario de Datos</i>	9
2.2.5.4.1 <i>Características principales de un Diccionario de Datos</i>	9
2.2.5.4.2 <i>Tipos de Información almacenada en el Diccionario de Datos</i>	9
2.3 METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS	10
2.3.1 <i>Planeación</i>	10
2.3.2 <i>Análisis</i>	10
2.3.3 <i>Diseño</i>	11
2.3.4 <i>Desarrollo</i>	12
2.4 METODOLOGÍA ORIENTADA A OBJETOS	12
2.4.1 <i>Introducción</i>	12
2.4.2 <i>Conceptos Básicos</i>	13
2.4.2.1 <i>Orientado a Objetos</i>	13
2.4.2.2 <i>Objeto</i>	13
2.4.2.2.1 <i>Identidad</i>	13
2.4.2.2.2 <i>Atributo</i>	14
2.4.2.2.3 <i>Relación</i>	14
2.4.2.2.4 <i>Métodos</i>	14
2.4.2.3 <i>Mensaje</i>	14
2.4.2.4 <i>Encapsulamiento</i>	14
2.4.2.5 <i>Clase</i>	14
2.4.2.6 <i>Herencia</i>	15
2.4.2.7 <i>Polimorfismo</i>	15
2.4.3 <i>Características de los Sistemas Orientado a Objetos</i>	15
2.4.3.1 <i>Abstracción de Datos</i>	16
2.4.3.2 <i>Tipos Abstractos</i>	16
2.4.3.3 <i>Reutilización de Código</i>	16
2.4.3.4 <i>Polimorfismo</i>	16
2.4.3.5 <i>Ocultamiento o Encapsulamiento de Información</i>	17
2.4.4 <i>Metodología Orientada a Objetos de Grady Booch</i>	17

2.4.4.1	Definición de la Metodología Orientada a Objetos	17
2.4.4.2	Principios de Modelado de Objetos	17
2.4.4.2.1	Abstracción	18
2.4.4.2.2	Encapsulamiento	18
2.4.4.2.3	Modularidad	18
2.4.4.2.4	Jerarquía	18
2.4.4.2.5	Tipificación	19
2.4.4.2.6	Concurrencia	19
2.4.4.2.7	Persistencia	19
2.4.4.3	Proceso de la Metodología Orientada a Objetos	19
2.4.4.3.1	Identificación de Clases y Objetos	20
2.4.4.3.2	Identificación de la semántica de Clases y Objetos	20
2.4.4.3.3	Identificando cuáles son clases y cuáles son objetos	21
2.4.4.3.4	Identificando Mecanismos	21
2.4.4.3.5	Identificación de Relaciones entre Clases y Objetos	21
2.4.4.4	Relaciones entre Clases y Objetos	22
2.4.4.4.1	Tipos de Relaciones	22
2.4.4.4.2	Relaciones entre clases	23
2.4.4.5	Implantación de Clases y Objetos	24
2.4.4.6	Notación	24
2.4.4.6.1	Diagramas de clases	25
2.4.4.6.2	Diagramas de estados de transición	26
2.4.4.6.3	Diagramas de Objetos	26
2.4.4.6.4	Diagramas de Sincronización	27
2.4.4.6.5	Diagramas de Módulos	27
2.4.4.6.6	Diagramas de Procesos	27
3.	DESARROLLO DEL SISTEMA	28
3.1	ANÁLISIS	29
3.1.1	Información general del sistema	30
3.1.2	Problemas y sus causas	34
3.1.3	Delimitación del problema	34
3.1.4	Determinación de requerimientos	35
3.1.5	Identificación de los componentes del sistema	35
3.1.5.1	Operación del programa de Verificación de Predios en ASERCA Regional	44
3.1.6	Evaluación de alternativas de solución	56
3.1.6.1	Solución basada en computadoras personales	57
3.1.6.2	Solución basada en los equipos SUN Enterprise Server	57
3.1.6.3	Selección de la solución	57
3.2	DISEÑO	59
3.2.1	Arquitectura de los subsistemas	59
3.2.2	Esquema de base de datos	61
3.2.3	Mecanismo de la Verificación de Predios	64
3.3	IMPLANTACIÓN	78
3.3.1	Componentes de la base de datos	78
3.3.1.1	Determinación de tablas	78
3.3.1.2	Reglas de Cardinalidad	80
3.3.1.3	Determinación del espacio de Almacenamiento	81
3.3.1.4	Realización de reglas de operación en el RDBMS	85
3.3.2	Módulos	89
3.3.2.1.1	Integración de los módulos	89
3.3.3	Interfaz	90
3.3.4	Mantenimiento	92

4. CONCLUSIONES	94
4.1 APLICACIONES	94
5. APÉNDICES	95
5.1 APÉNDICE A. LENGUAJE RELACIONAL ESTÁNDAR	95
A.1 Definición de datos en SQL	95
A.2 Consultas en SQL	96
A.2.1 Especificando nombres de atributos iguales	97
A.2.2 Tablas como conjuntos en SQL	98
A.2.3 Consultas anidadas y comparación de conjuntos	98
A.2.4 Función EXISTS	98
A.2.5 Conjuntos explícitos y valores nulos	99
A.2.6 Funciones agregadas y de grupo	99
A.3 Declaraciones de actualización	100
A.3.1 Comando INSERT	101
A.3.2 Comando DELETE	101
A.3.3 Comando UPDATE	102
A.4 Vistas SQL	102
A.4.1 Concepto de una vista	102
A.4.2 Especificaciones de vistas	102
A.5 Índices	103
5.2 APÉNDICE B. TRIGGER DE LA TABLA	104
VERIFICACIONES_PREDIOS	104
5.3 APÉNDICE C. TRIGGER DE LA TABLA DOCUMENTOS_VERIFICACION	115
BIBLIOGRAFÍA	121

INTRODUCCIÓN

1. INTRODUCCIÓN

Apoyos y Servicios a la Comercialización Agropecuaria (ASERCA) fue creada por decreto presidencial como un órgano administrativo desconcentrado de la Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA), cuyo objetivo principal es llevar el programa PROCAMPO, el cual consiste en estimular la producción agrícola mediante apoyos económicos a los productores nacionales en función de las hectáreas cultivadas en cada ciclo agrícola. También lleva el programa de la comercialización, que consiste en apoyar la comercialización de los productos agropecuarios nacionales.

ASERCA está constituida por 16 centros regionales y las oficinas centrales. El esquema conceptual de PROCAMPO está conformado por 5 módulos: Registro Voluntario, Verificación de Predios, Control de Peticiones, Entrega de Apoyos, Mantenimiento y Control de Almacén. ASERCA es la encargada a través de sus 16 centros regionales de generar, procesar y validar las solicitudes de apoyo, resguardar y actualizar la base de datos y emitir los apoyos; tiene además la responsabilidad de supervisar la aplicación de la normatividad operativa.

Las direcciones regionales son las responsables de supervisar el cumplimiento de la normatividad operativa del PROCAMPO; la supervisión la realizan mediante las verificaciones tanto en campo, expediente y entrevista a los productores. Para realizar los trabajos de supervisión se desarrollaron metodologías de verificación que tiene como finalidad homogeneizar los criterios de selección de las áreas geográficas y en particular de los predios que serán verificados, así como optimizar el uso de los recursos involucrados en los trabajos de verificación. El objetivo de la verificación de predios es evidenciar los casos de irregularidad, contribuyendo a la depuración del directorio de PROCAMPO.

El área encargada de realizar los trabajos de verificación de predios, es el Departamento de Atención a los Productores, dicho departamento lo conforman: el Jefe de Atención y generalmente 9 técnicos verificadores, el grado de estudios del personal deberá ser Ingenieros Agrónomos, es decir, gente especialista en problemas del campo, conocedora de la agricultura en nuestro país.

Realizando el análisis de la situación que guarda el Departamento de Atención, no cuenta con un sistema que les permita consultar la información de los predios de determinada área geográfica, además de que no existen criterios homogéneos para realizar los trabajos de supervisión de manera normativa, es decir, cada centro regional aplica la supervisión de acuerdo a conocimientos subjetivos del personal encargado, debido a la preparación del personal. Tampoco existe un registro en la base de datos que lleve un seguimiento de las verificaciones, ni un seguimiento de los casos detectados como irregulares; esto ocasiona que se emitan apoyos que no corresponden con lo realmente cultivado, ocasionando a destiempo cancelaciones de apoyos, reexpediciones y devoluciones.

El Departamento de Atención tiene la obligación de reportar a ASERCA Central mensualmente sobre los trabajos de verificación realizados, en el entendido de que depende del avance en dichos trabajos para la radicación de los recursos presupuestales siguientes. Como no existe un sistema que lleve un control de los trabajos a realizar; una vez que ellos realizan las verificaciones tienen que capturar en Excel los resultados obtenidos, dicha tarea la realiza específicamente el Jefe del Departamento de Atención,

contribuyendo así a que los técnicos verificadores desconozcan la importancia de sus trabajos realizados y de la normatividad vigente.

Cada centro regional reporta sus trabajos de verificación con un formato diferente y a destiempo, ocasionando en ASERCA Central un retraso en la interpretación de los datos obtenidos.

El sistema de información que se propuso, se desarrolló y se implantó en base a un esquema de control de verificación de predios. Con la problemática planteada ASERCA Central a través del área encargada de supervisar los trabajos de verificación realizados en los centros regionales, se vio en la necesidad de solicitar al área de Desarrollo de Sistemas de Información, un sistema que les permita a los centros regionales realizar la captura de resultados de las verificaciones y emisión de reportes.

Del resultado del análisis de información, se planteo el siguiente objetivo a cubrir: automatizar las actividades de consulta, almacenamiento, control y seguimiento de las verificaciones de predios, con base en las metodologías aplicadas para dicho fin, obteniendo así el grado de cumplimiento de la normatividad operativa, la correcta aplicación de los apoyos y la depuración del directorio PROCAMPO.

FUNDAMENTOS TEÓRICOS

UNAM Facultad de Ingeniería

UNAM Facultad de Ingeniería

UNAM Facultad de Ingeniería

Una clase es aquel conjunto de objetos que tienen características y comportamientos semejantes, por lo tanto los objetos que no tengan atributos semejantes no pueden ser agrupados en una misma clase. La agrupación de objetos en clases es una forma de ordenar y organizar el dominio de objetos que conforman un sistema.

Las clases surgieron porque en raras ocasiones un sistema involucra un solo objeto de cada tipo. Es mucho más común requerir más de uno.

2.4.2.6 Herencia

La herencia es un mecanismo por el cual una clase de objetos puede definirse como un caso especial de una clase más general, con lo cual automáticamente incluye toda la definición de métodos y variables de la clase general, es una relación entre clases, en donde una clase comparte la estructura o comportamiento definidas en una de ellas o más con otras clases.

A la clase que hereda se le conoce como superclase y a la clase heredera se le denomina subclase. Una subclase puede añadir y/o modificar operaciones a su comportamiento, formando así una especialización de la superclase. Una clase puede ser subclase de una o más superclases. En el primer caso se habla de herencia sencilla y en el segundo de herencia múltiple.

La herencia entre clases puede extenderse a cualquier grado. El resultado es una estructura arborescente conocida como Jerarquía de Clases.

Cuando hay herencia de clases, los métodos definidos en un a superclase se heredan a los objetos de las subclases.

2.4.2.7 Polimorfismo

Es la capacidad de un objeto para responder al mismo mensaje de una o más clases, y cada uno de diferente forma. Esto significa que un objeto no necesita conocer quien envía el mensaje, sino cuantos tipos de objetos han sido definidos para responder a un mensaje en particular.

El polimorfismo nos permite reconocer y explotar semejanzas entre diferentes clases de objetos.

2.4.3 Características de los Sistemas Orientado a Objetos

El desarrollo de sistemas con la metodología orientada a objetos tiene un doble propósito, por un lado la elaboración de herramientas que solucionen la problemática planteada y por otro, la generación de componentes que puedan ser integrados a la biblioteca de componentes, aumentando así el capital de software.

El impacto económico de la adopción de la metodología orientada a objetos no puede medirse en base a la construcción de un sólo sistema, la ganancia del uso de esta

metodología se muestra a largo plazo, donde la reutilización de los componentes capitalizados se vuelve importante.

La esencia de las metodologías orientadas a objetos son:

- Abstracción de datos.
- Tipos abstractos.
- Reutilización de código.
- Polimorfismo.
- Ocultamiento o encapsulamiento de información.

2.4.3.1 Abstracción de Datos

La abstracción consiste en capturar una idea general, es decir, la esencia de una entidad más compleja, para posteriormente representarla con algún elemento interno de un programa. El representar solo los atributos generales del objeto sin considerar características específicas ni el funcionamiento interno del objeto, permite concentrarse en la manipulación y operaciones a realizar sobre él.

2.4.3.2 Tipos Abstractos

La creación de tipos abstractos permiten al programador adaptar el lenguaje de programación a sus necesidades específicas. Un tipo de datos abstracto, es un tipo de dato definido por el usuario y representa la abstracción de un objeto, por tanto se ajusta al problema que se está resolviendo y puede ser manipulado con los tipos internos del lenguaje.

2.4.3.3 Reutilización de Código

Consiste en emplear los recursos utilizados en proyectos anteriores. Las metodologías orientadas a objetos proponen dos estrategias para la reutilización de código: la composición y la herencia.

Composición: permite definir una nueva clase de objetos mediante la unión de un conjunto de clases ya existentes.

Herencia: esta puede ser Simple: crear una nueva clase basada en otra clase más general, o

Múltiple: crear una clase de un conjunto de clases, de tal forma que si utiliza algún método y no lo tiene, lo solicita a las clases de las cuales se deriva.

2.4.3.4 Polimorfismo

Permite la manipulación de objetos de clases distintas como si fueran de la misma clase, con lo cual es posible definir interfaces uniformes para diferentes tipos de objetos.

2.4.3.5 Ocultamiento o Encapsulamiento de Información

Se da a raíz de la abstracción, ya que al extraer la esencia de una entidad, las características internas quedan ocultas. Un objeto debe ser transparente en cuanto a su uso, es decir, únicamente debe mostrar sus atributos externos para su manipulación, ocultando su implementación, algoritmos para sus operaciones, estructuras de datos que manejan, etc.

Otra característica distintiva de los sistemas orientados a objetos es que los objetos saben qué operaciones se pueden aplicar sobre ellos. Este conocimiento se consigue combinando abstracciones de datos y de procedimientos en un solo componente de programa (objeto).

Los objetos encapsulan datos y procesos. Las operaciones de procesamiento son parte del objeto y son iniciadas pasando un mensaje al objeto. Existe una representación especial para los sistemas orientados a objetos, y éstos dependen de la metodología que se haya utilizado para desarrollarlo. En términos de mantenimiento de software el manejo de objetos facilita la actualización.

Los tiempos promedio de corrección de fallas de un sistema, al implantarlo, se reduce notablemente, lo que da una flexibilidad para la oportuna corrección del sistema.

El uso de la tecnología de objetos ofrece la facilidad de crear prototipos, los cuales, ayudan de manera importante en el desarrollo de sistemas de información.

El manejo de esta tecnología soporta la adecuación de los sistemas de acuerdo a las necesidades cambiantes del medio organizacional. El desarrollo del software mediante estas técnicas, además de ofrecer rapidez, ofrece una mayor confiabilidad.

2.4.4 Metodología Orientada a Objetos de Grady Booch

2.4.4.1 Definición de la Metodología Orientada a Objetos

Es un proceso creciente e interactivo, en el cual se identifican clases y objetos de acuerdo a las propiedades relevantes de un dominio en particular, con la finalidad de implantar un mecanismo que prevea el comportamiento que un modelo requiere.

La metodología orientada a objetos propone que durante el proceso de diseño se retome mediante análisis de sí mismo, haciendo una mejora basado en una nueva apreciación, este proceso debe repetirse hasta estar seguros que el diseño es correcto y completo.

2.4.4.2 Principios de Modelado de Objetos

En la identificación de objetos, propone una serie de principios para facilitar su detección y modelado:

Abstracción

Encapsulamiento
Modularidad
Jerarquía
Tipificación
Concurrencia
Persistencia

2.4.4.2.1 Abstracción

Denota las características esenciales de un objeto que lo distinguen de los otros objetos.

- **Abstracción de entidades:** es un objeto que representa un modelo usado por un problema en específico.
- **Abstracción de acciones:** es la identificación de un objeto que provee una función generalizada, es decir, todas las funciones que pueda realizar este objeto serán del mismo tipo.
- **Abstracción virtual de máquina:** es aquel objeto que lleva a cabo un conjunto de operaciones, las cuales son realizadas mediante un control de alto nivel a éste.
- **Abstracción de coincidencia:** es aquel objeto que empaqueta un conjunto de operaciones definidas que no tienen relación entre ellas.

2.4.4.2.2 Encapsulamiento

Es el proceso que esconde todos los detalles de un objeto que no contribuyen con sus características esenciales, es decir, es la información escondida en un objeto.

2.4.4.2.3 Modularidad

Es el acto de particionar un programa en componentes individuales, que pueden reducir su complejidad. La modularidad es una propiedad de un sistema que descompone en un conjunto de módulos definidos con cohesión y libertad entre ellos. La identificación de la modularidad de objetos, clases o comportamiento de los mismos, nos facilitarán el establecimiento del mecanismo de implantación del sistema.

2.4.4.2.4 Jerarquía

Es una ordenación de abstracciones. La mayoría de las abstracciones son identificadas en un orden jerárquico, en el diseño de un sistema la jerarquización de los objetos simplifica el entendimiento de un objeto. Se pueden identificar dos tipos de jerarquización: la sencilla y la múltiple. Básicamente la jerarquía define la relación entre clases, una jerarquización sencilla es cuando entre varias clases comparten la misma estructura y comportamiento y una jerarquización múltiple es cuando su comportamiento está definido por varias clases.

2.4.4.2.5 Tipificación

Un tipo es una característica precisa de estructura o comportamiento, la cual es compartida por un conjunto de entidades. La tipificación es la coacción o ejecución de la clase de un objeto, que hace que el objeto de diferentes tipos no cambie, es decir, sólo podrá hacerlo en casos especiales o restringidos.

2.4.4.2.6 Concurrencia

Es la propiedad que distingue un objeto activo de otro que no lo está, es decir, nos indica el estado de los objetos en su ciclo de vida.

2.4.4.2.7 Persistencia

Es la propiedad de un objeto que a través de diferentes momentos y actividades existe o trasciende en el tiempo, es decir, un objeto puede continuar existiendo de un proceso a otro, ya sea por su propia existencia o a través de su cambio de dirección en el cual fue creado.

2.4.4.3 Proceso de la Metodología Orientada a Objetos

Requiere de dos elementos:

1. De una notación que nos deje un modelo del problema en diferentes formas y que además permita enfocarse en las partes relativamente independientes del problema, en diferentes tiempos. Ya que al hablar de un proceso interactivo significa poder trabajar con cualquier tipo de modelo.

2. La invención de algunos mecanismos que usen estas abstracciones, las cuales deberán ser documentadas en una serie de diagramas de objetos.

El diseño de los detalles de estos mecanismos nos lleva a la estructuración de clases, a través de las cuales se establece un protocolo, para cada una de ellas. Eventualmente al final del proceso del diseño, se termina con la definición de diagramas que proveen diferentes vistas del sistema, todos ellos deben ser coherentes y consistentes, teniendo una evolución creciente de uno a otro.

Este desarrollo creciente e interactivo hace del tradicional ciclo de vida en cascada del desarrollo de software un proceso de retroalimentación de una etapa a otra. El ciclo de vida tradicional es fundamentalmente un proceso pobre, y generalmente viola muchos de los principios de la práctica de la ingeniería, lo cual no quiere decir que el diseño debe ser realizado sin una estructura.

El diseño orientado a objetos es un proceso que se obtiene mediante el seguimiento ordenado de los siguientes eventos:

1. Identificar las clases y objetos en un nivel de abstracción.
2. Identificar la semántica de las clases y objetos.

3. Identificar las relaciones entre las clases y objetos.

4. Implantar las clases y objetos.

Proceso Creciente: la identificación de nuevas clases y objetos usualmente nos causa redefinir y mejorar la semántica de las relaciones entre clases y objetos.

Proceso Interactivo: la implantación de clases y objetos, lo cual regularmente nos lleva al descubrimiento o invención de nuevas clases y objetos, que son el resultado de la simplificación y generalización de nuestro diseño.

2.4.4.3.1 Identificación de Clases y Objetos

Para realizar esta tarea, se consideran dos pasos:

1. El descubrimiento de las abstracciones primarias en un problema definido, es decir, los principales elementos que intervienen en el proceso, son los candidatos a ser las clases y objetos.
2. La invención de los más importantes mecanismos que proveen el comportamiento requerido de los objetos que trabajan juntos para realizar una función. Para encontrar estos elementos y procesos a través del análisis usando técnicas para el modelado de objetos, el desarrollador debe aprender el vocabulario del dominio del problema, las reglas del juego, y los eventos que deben ocurrir para las clases y objetos de nuestro diseño, en el más alto nivel de abstracción.

Aquí se enfatiza la definición de los principales elementos que intervienen en el sistema, ya que son éstos los candidatos a clases y objetos, y que además son la base del diseño.

2.4.4.3.2 Identificación de la semántica de Clases y Objetos

El desarrollador es como un observador externo, es decir, su visión de cada clase debe ser desde la perspectiva de su interfaz, así como la identificación de las cosas que pueden ser una instancia de una clase y las cosas que cada objeto puede hacer con otro.

Es en esta fase que se definen las clases y objetos, decidir el protocolo para cada objeto. La preparación del protocolo para cada objeto, requerirá decisiones que cambien el significado de otro objeto. En general, la existencia de las llaves primarias o principales elementos no cambian, sino solamente se transfiere su alcance. Una técnica útil para guiarnos en esta actividad, es la escritura de cada objeto, donde se define su ciclo de vida, desde su creación hasta su destrucción, incluyendo las características de su comportamiento.

Se debe decidir cómo se integrarán las partes del diseño, es decir, se debe crear el prototipo del sistema; para hacerlo se debe considerar el actual diseño y evaluar las alternativas de desarrollo y los problemas que puedan representar un riesgo.

2.4.4.3.3 Identificando cuáles son clases y cuáles son objetos

Debemos identificar la cohesión y libertad entre sus abstracciones, y colocarla en la jerarquía que le corresponda, por los objetos que puedan crearse a partir de ella. La identificación de las principales abstracciones involucra dos procesos: el descubrimiento y la invención. A través del descubrimiento reconocemos las abstracciones usadas en el dominio del problema y por medio de la invención se crean nuevas clases y objetos que son necesariamente parte del problema.

2.4.4.3.4 Identificando Mecanismos

Se usa el término de mecanismo, para describir cualquier estructura de donde los objetos trabajan juntos para proveer algún comportamiento que satisfaga los requerimientos del problema.

Un mecanismo representa el diseño de decisión acerca de cómo una colección de objetos deben cooperar. El mecanismo es como el alma del diseño, y representa la estrategia de decisiones que deben los diferentes objetos realizar cooperativamente para cumplir su función.

2.4.4.3.5 Identificación de Relaciones entre Clases y Objetos

Aquí se establecerá exactamente cómo las cosas deben interactuar en el sistema. Respecto a las llaves primarias o elementos principales, se deben afirmar el uso, herencia y otros tipos de relaciones entre clases. Para los objetos en la implantación, deben establecerse la semántica y estática de cada mecanismo.

Se deben descubrir patrones entre clases, lo que provoca la reorganización y simplificación de la estructura del sistema de clases y patrones entre la colección cooperativa de objetos, lo que lleva a generalizar los mecanismos que comprenden el diseño. Posteriormente se deben tomar decisiones de visión, es decir, cómo una clase ve a otra, cómo los objetos ven a otros, y de igual importancia, qué objetos deben ver o no a otros. A través de estas visiones se pueden tomar decisiones de empaquetamiento inteligente en el diseño de la arquitectura de módulos del sistema.

El descubrimiento de patrones y tomar decisiones de visión deben causar la refinación de protocolos de varias clases definidas anteriormente, lo cual nos lleva a abstracciones comunes y mecanismos definidos en un solo lugar.

Aquí se refinan los diagramas de clases hechos anteriormente, tomando en cuenta los descubrimientos de patrones y la visibilidad que tiene un objeto del otro. También se completan los detalles de los diagramas de objetos que documentan la esencia del mecanismo de solución. Primero se dibujan los objetos que conocemos que cooperan en cierta forma, después se seleccionan pares de objetos y se pregunta si hay relaciones entre ellos, si la respuesta es afirmativa, se pregunta lo siguiente: cómo estos objetos están relacionados y qué mensajes se envían el uno al otro. Lo que nos llevará a refinar el protocolo de las correspondientes clases, lo que nos lleva al descubrimiento de nuevos patrones.

2.4.4.4 Relaciones entre Clases y Objetos

2.4.4.4.1 Tipos de Relaciones

Los objetos contribuyen al comportamiento de un sistema con la colaboración de otros objetos. Las relaciones entre dos de éstos abarca la suposición de todo lo que uno puede hacer para el otro, incluyendo las operaciones que pueden ser realizadas entre sí y el comportamiento resultante. Existen dos tipos de jerarquías de relaciones entre objetos:

- a) Uso de relaciones
- b) Contenido de relaciones

a) Uso de Relaciones

La figura 2.1 ilustra cómo se usan las relaciones entre varios objetos; la línea entre dos objetos representa la existencia del uso de una relación entre dos y significa que un mensaje pasa a través de esta relación. El mensaje que pasa entre dos objetos es usualmente unidireccional, sin embargo es posible la comunicación bidireccional.

Dada una colección de objetos relacionados, cada uno puede jugar diferentes roles:

- Objeto actor: objeto que puede operar sobre otros objetos pero nunca es operado por otros objetos.
- Objeto servidor: objeto que nunca opera sobre otros objetos, solo es operado por otros objetos.
- Objeto agente: objeto que puede operar sobre otros objetos y que puede ser operado por otros objetos.

Cuando un objeto pasa un mensaje a otro deben ser sincronizados de alguna forma, por lo que un objeto puede ser utilizado como:

- Objetos secuenciales: objeto pasivo cuya semántica está garantizada solo en presencia de una secuencia de control.
- Objetos por bloques: objeto pasivo cuya semántica está garantizada en presencia de múltiples secuencias de control.
- Objetos concurrentes o activos: objeto activo cuya semántica está garantizada en presencia de múltiples secuencias de control.

El que un objeto sea utilizado secuencialmente o por medio de una sola secuencia de control excluye las otras formas, y así con cada uno.

El objeto activo es aquel que comprende su propia secuencia de control y un objeto pasivo es aquel que no comprende su propia secuencia de control.

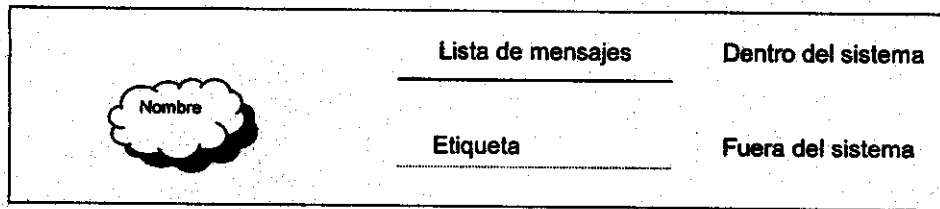


Figura 2.1 Notación de la representación de objetos y sus relaciones.

b) Contenido de relaciones

Es el encapsulamiento de una relación en un objeto. El contener o encapsular una relación reduce el número de objetos que deben ser visibles a un nivel de objetos.

2.4.4.4.2 Relaciones entre clases

Un objeto es una entidad concreta que existe en un tiempo y espacio, una clase representa solamente una abstracción, que es la esencia de un objeto.

Una clase es un grupo definido por atributos comunes, es una definición de objetos que comparten una estructura común y un común comportamiento.

La interfaz de una clase proporciona la vista exterior de sí misma y por tanto enfatiza la abstracción mientras esconde en su estructura los secretos de su comportamiento. Esta interfaz primeramente consiste de las declaraciones de todas las operaciones aplicables a instancias de esta clase, esto podría incluir la declaración de otras clases, constantes y variables.

Una relación entre clases indica una forma de compartir y un tipo de conexión semántica. Existen tres tipos de relaciones entre clases:

- La generalización, denotando un tipo de relación.
- La agregación, la cual denota una parte de relación.
- La asociación, la cual denota alguna conexión semántica entre otro tipo de clase no relacionada.

Estas relaciones son soportadas por los lenguajes orientados a objetos o los basados en objetos, en la combinación de las relaciones entre las clases denominadas:

- a) **Relación de herencia:** es cuando una clase comparte la estructura o comportamiento definidos en una herencia simple o múltiple. La herencia define un tipo de jerarquía, donde cada clase hereda de otra u otras clases. Una subclase (clase de nivel inferior) típicamente redefine la estructura definida y el comportamiento de su superclase. La herencia de relaciones es la base del polimorfismo, ya que por las diferentes relaciones con varias clases un objeto tiene la posibilidad de responder a una operación en diferentes formas. La herencia múltiple se usa cuando la relación simple no satisface las suficientes relaciones entre clases. Su uso es parte de la existencia de relaciones de un objeto con más de una clase, debido a su naturaleza y las relaciones requeridas. Cuando una clase es construida por herencia

múltiple y no se le aumenta su propia estructura o comportamiento se está hablando de una clase agregada.

- b) Relaciones de uso: Hay dos tipos de uso de relaciones entre clases:
- Interfaz de una clase: uso de otra clase, donde la clase usada debe ser visible.
 - Implantación de una clase: uso de otra clase, donde la clase usada es escondida como parte del secreto de usar clases.

A través de las líneas que representa la relación se puede expresar también la cardinalidad de la relación.

- c) Relaciones de instancias: En la definición de una clase se debe tener un estado preciso de los objetos que pueden pertenecer o ser definidos como instancias de una clase.
- d) Relaciones de metaclasses: una metaclasses es una clase, donde sus instancias son la misma clase.

2.4.4.5 Implantación de Clases y Objetos

Aquí se realizan las siguientes actividades: tomar decisiones de diseño concernientes a la representación de las clases, ubicar las clases y los objetos en los módulos, y programas en procesos.

Se establece la interfaz concreta para cada clase y objeto en un nivel de abstracción. Se realiza una refinación de la estructura de clases del sistema.

La implantación de una clase en su vista interior involucra el secreto del comportamiento de la clase. La implantación de una clase se divide en tres partes:

1. Público: la declaración que forma parte de la interfaz de una clase y es visible a todos los clientes que son visibles a la clase.
2. Protegido: la declaración que forma parte de la interfaz de una clase, pero no es visible a cualquier otro, excepto de sus subclases.
3. Privada: la declaración que forma parte de la interfaz de una clase, pero no es visible por cualquier otra clase.

2.4.4.6 Notación

La notación sugerida por Grady Booch, contempla la estructura de las clases, los mecanismos de herencia, el comportamiento individual de los objetos y el comportamiento dinámico de un sistema.

La notación es la documentación del diseño de un sistema, esta notación debe usarse de acuerdo a las necesidades de cada proyecto. Existen cuatro tipos de diagramas:

1. Diagramas de clases
2. Diagramas de objetos

3. Diagramas de módulos

4. Diagramas de procesos

Los primeros dos son parte de la vista lógica de un sistema, ya que describen la existencia y el significado de las principales abstracciones y la forma del diseño. Los últimos dos diagramas son parte de la estructura física de un sistema, ya que describen concretamente los componentes del software y hardware de una implantación.

La semántica dinámica y estática de un sistema se expresan a través de dos tipos de diagramas:

1. Diagramas de estados de transición

2. Diagramas de sincronización

Para cada clase deberá tener un diagrama de transición asociado a sus indicaciones de cómo el orden en eventos externos pueden afectar el estado de cada instancia de la clase. Se deberá usar diagramas de sincronización en conjunción con cada diagrama de objetos, para mostrar el orden de los mensajes que son enviados y evaluados.

2.4.4.6.1 Diagramas de clases

Muestran la existencia de clases y sus relaciones en el diseño lógico de un sistema. Un diagrama de clases representa todo o parte de la estructura de clases del sistema. Hay tres elementos de la estructura de clases: clases, las relaciones de las clases y las utilerías de clases.

Clases: su icono es un globo amorfo (nube) y está pintado con línea punteada, la cual indica que los clientes que hacen uso de él generalmente operan solamente sobre las instancias de la clase y no de la clase en sí. Cada clase debe ser única y debe tener un nombre.

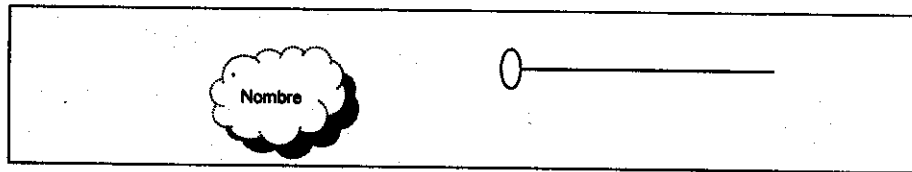


Figura 2.2 Notación para el diagrama de clases.

Relaciones de las clases: los iconos para los diagramas de clase están especificados en la figura 2.2, donde cada relación debe tener una etiqueta, la cual documenta el nombre de la relación entre las clases. El uso de una relación es doble línea con un círculo en uno de los extremos identifica que la clase usa recursos de otra, se dice que la relación es por interfaz cuando la clase utilizada es visible a todo cliente, y la relación es por implementación cuando la clase esta oculta como parte del secreto del uso de una clase.

El símbolo ● es usado para indicar la dirección en que la etiqueta debe ser leída. En casos complicados y donde se requiera, para completar la cardinalidad se pueden usar operadores relacionales: =, <>, <, >, >=, y <=.

Utilerías de clase: representa cualquier subprograma libre o una colección de programas libres, es decir, que no pertenecen a una clase, y que a su vez pueden servir a aquellas que lo requieran.

Plantillas de clases: es la documentación de cada clase, subclase, sus campos y sus operaciones.

2.4.4.6.2 Diagramas de estados de transición

Estos diagramas permiten mostrar el estado en el espacio de una clase, es decir, los eventos que causan una transición de un estado a otro, y las acciones descritas en un diagrama de estados nos podría llevar, a un diagrama de objetos. Los iconos para los diagramas de estados están especificados en la figura 2.3. Un círculo representa un estado, y en el debe estar escrito el nombre del estado, un estado de inicio es representado con un doble círculo sin llenar, y un estado de termino se representa con un círculo rellenado.

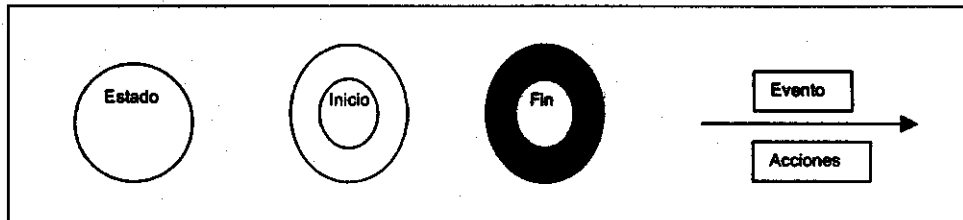


Figura 2.3 Notación gráfica para diagrama de estados.

2.4.4.6.3 Diagramas de Objetos

Un diagrama de objetos representa todo o parte de la estructura de un sistema. El objetivo de los diagramas de objetos es ilustrar la semántica de los principales mecanismos en el diseño lógico. Cada diagrama representa las interacciones que deben ocurrir entre una colección de objetos. Las operaciones usadas en un diagrama de objetos deben ser consistentes con las operaciones definidas en las clases asociadas, es decir, si se muestra que un objeto envía a otro un mensaje, entonces la operación a realizar al recibir el mensaje debe estar definido por su clase.

Objetos: el icono para representar un objeto es semejante al de las clases, el nombre de los objetos debe ser único. Las propiedades de un objeto deben aparecer en el lado izquierdo inferior de su icono. Las más importantes propiedades que incluyen son la concurrencia y la persistencia de cada objeto.

Relaciones entre objetos: representan el envío de mensajes de un objeto a otro. Una línea representa un mensaje bidireccional, las líneas sólidas es cuando el modelo podemos implantarlo en el software.

2.4.4.6.4 Diagramas de Sincronización

Se usan para representar la dinámica del envío de mensajes de un diagrama de objetos. Un diagrama sincronizado es una gráfica que establece el tiempo en el eje horizontal, y los objetos a través del eje vertical. Como se avanza sobre el eje horizontal (tiempo), una operación debe ser invocada. Al marcar un proceso con un asterisco (*), indica que se crea el objeto, y cuando se marca el proceso con un signo de admiración (!) indica la destrucción el objeto.

2.4.4.6.5 Diagramas de Módulos

Muestran la colocación de las clases y objetos en los módulos en el diseño físico de un sistema. Un diagrama de módulos representa todo aparte de la arquitectura del sistema. Hay dos elementos en la arquitectura de módulos: los módulos y su visibilidad.

Existen varios iconos para representar los diferentes tipos de módulos que pueden existir. Los tipos de módulos corresponden a tipos de módulos que soporten los diferentes lenguajes orientados o basados en objetos.

Módulos: Es la clasificación de los programas que componen un sistema. El módulo debe ser único. La única relación entre módulos es la dependencia de compilación.

Subsistemas: los subsistemas son creados cuando en los sistemas grandes es necesario contar con un nivel de abstracción mayor. Cada subsistema denota otro diagrama de módulos.

2.4.4.6.6 Diagramas de Procesos

Los procesos de diagramas visualizan y permiten ubicar procesos y procesadores. Un diagrama de procesos representa todo o parte de la arquitectura de procesos de un sistema. Hay tres elementos importantes para los diagramas de procesos: procesadores, dispositivos y conexiones.

Procesadores y dispositivos: Es una pieza de hardware capaz de ejecutar programas, regularmente en el icono de procesadores incluye la serie de procesos que van a ejecutarse.

Conexiones: representa un acoplamiento de hardware, como un cable Ethernet o fibra óptica.

DESARROLLO DEL SISTEMA

Las actividades que desarrolla cada una de las instancias, se describen a continuación:

ASERCA Central :

- Actualiza la normatividad y los procedimientos operativos para cada ciclo agrícola.
- Evalúa y autoriza el programa de verificación a las Direcciones Regionales.
- Autoriza recursos presupuestales a las Direcciones Regionales.
- Evalúa los informes emitidos por las Direcciones Regionales.
- Autoriza la resolución administrativa derivada del programa de verificación.

Dirección Regional:

- Realiza las verificaciones de campo, expediente y entrevistas a los productores.
- Solicita y recibe información sobre las condiciones termopluviométricas de las zonas problemáticas del área de influencia de la Dirección Regional.
- Proporciona listados con información prellenada de los predios a verificar al Técnico Verificador comisionado de ASERCA.
- Informa resultados derivados del programa de verificación.
- Coordina el programa de verificación con la Delegación SAGARPA, para su seguimiento administrativo en caso de resultar predio irregular.
- Solicita a ASERCA Central la aplicación de la dictaminación administrativa.

Delegación SAGARPA:

- Recibe la notificación de los predios irregulares.
- Instaure procedimiento administrativo de cancelación al predio irregular.
- Emite la dictaminación administrativa al predio irregular.
- Remite la dictaminación administrativa.

CADER:

- Atención a los productores en las ventanillas.
- Colaboración con la Dirección Regional en la revisión de expedientes.

Productor:

- Proporciona información al Técnico Verificador comisionado de ASERCA sobre el predio.
- Tramita en su caso, un proceso de apelación sobre el dictamen emitido cuando el predio que representa manifiesta irregularidades.

Por el ámbito en que operan las instancias de responsabilidad se consideran de ámbito federal (SAGARPA, ASERCA Central, ASERCA Regional) y de ámbito estatal (CADER).

ASERCA cuenta con 16 Direcciones Regionales distribuidas en la República Mexicana y con una instancia Central. Además ASERCA Regional se auxilia de la estructura de la SAGARPA que cuenta con 33 Delegaciones Estatales, 193 Distritos de Desarrollo Rural (DDR) y 712 Centros de Apoyo (CADER).

DESARROLLO DEL SISTEMA

Las actividades que desarrolla cada una de las instancias, se describen a continuación:

ASERCA Central :

- Actualiza la normatividad y los procedimientos operativos para cada ciclo agrícola.
- Evalúa y autoriza el programa de verificación a las Direcciones Regionales.
- Autoriza recursos presupuestales a las Direcciones Regionales.
- Evalúa los informes emitidos por las Direcciones Regionales.
- Autoriza la resolución administrativa derivada del programa de verificación.

Dirección Regional:

- Realiza las verificaciones de campo, expediente y entrevistas a los productores.
- Solicita y recibe información sobre las condiciones termopluviométricas de las zonas problemáticas del área de influencia de la Dirección Regional.
- Proporciona listados con información prellenada de los predios a verificar al Técnico Verificador comisionado de ASERCA.
- Informa resultados derivados del programa de verificación.
- Coordina el programa de verificación con la Delegación SAGARPA, para su seguimiento administrativo en caso de resultar predio irregular.
- Solicita a ASERCA Central la aplicación de la dictaminación administrativa.

Delegación SAGARPA:

- Recibe la notificación de los predios irregulares.
- Instauro procedimiento administrativo de cancelación al predio irregular.
- Emite la dictaminación administrativa al predio irregular.
- Remite la dictaminación administrativa.

CADER:

- Atención a los productores en las ventanillas.
- Colaboración con la Dirección Regional en la revisión de expedientes.

Productor:

- Proporciona información al Técnico Verificador comisionado de ASERCA sobre el predio.
- Tramita en su caso, un proceso de apelación sobre el dictamen emitido cuando el predio que representa manifiesta irregularidades.

Por el ámbito en que operan las instancias de responsabilidad se consideran de ámbito federal (SAGARPA, ASERCA Central, ASERCA Regional) y de ámbito estatal (CADER).

ASERCA cuenta con 16 Direcciones Regionales distribuidas en la República Mexicana y con una instancia Central. Además ASERCA Regional se auxilia de la estructura de la SAGARPA que cuenta con 33 Delegaciones Estatales, 193 Distritos de Desarrollo Rural (DDR) y 712 Centros de Apoyo (CADER).

3.1.2 Problemas y sus causas

Los principales problemas encontrados en la operación de las verificaciones de predios realizadas en las Direcciones Regionales son:

Problema	Causa
1. Emisiones de apoyos que no corresponden con lo realmente cultivado, ocasionando cancelaciones de apoyos, reexpediciones y devoluciones.	<ul style="list-style-type: none"> • Errores en la captura de solicitudes. • La falta de una verificación exhaustiva en campo de los predios y productores.
2. La recepción de la información enviada desde los centros regionales hacia ASERCA central requiere mucho tiempo.	<ul style="list-style-type: none"> • Se carece de un sistema que registre el proceso de seguimiento por cada predio a verificar.
3. No es posible hacer una interpretación rápida de los resultados reportados por los centros regionales.	<ul style="list-style-type: none"> • No todos los centros regionales aplican un criterio homogéneo para reportar los datos.
4. Existen 16 formas de aplicar las metodologías y criterios para seleccionar los predios o conglomerados a verificar.	<ul style="list-style-type: none"> • No existe homogeneidad para aplicar las metodologías y criterios a verificar.

3.1.3 Delimitación del problema

Derivado de las experiencias adquiridas en los anteriores ciclos agrícolas, se han desarrollado metodologías de verificación que tienen como finalidad homogeneizar los criterios para seleccionar en general los conglomerados y en particular los predios que serán verificados, así como optimizar el uso de los recursos involucrados en estos trabajos.

En el problema 1, se debe principalmente a que no se cuenta con información oportuna y no queda un registro completamente automatizado y compartido de los predios verificados, esto implica que si existen resultados negativos de las verificaciones no existe un registro automatizado que permita detener el proceso de la(s) solicitudes de dichos predios, además de que en ocasiones se verifican más de una vez. En relación a los problemas 2 y 3 radica en que no todas las Direcciones Regionales elaboran los reportes de acuerdo a los formatos establecidos en la normatividad, por lo que la interpretación de la información en ASERCA Central se lleva varios días, además su procesamiento lo hacen en herramientas como Excel. En cuanto al problema 4 existen 16 maneras de aplicar las metodologías para realizar las verificaciones, lo cual resulta no representativo de la problemática real en la operación del PROCAMPO.

En general los trabajos de verificación no son cuantitativos, por lo que no permite concenar los avances y las repercusiones que dichos trabajos aportan a la operación del PROCAMPO.

3.1.4 Determinación de requerimientos

Objetivo:

Automatizar las actividades de programación, consulta, almacenamiento y seguimiento de las verificaciones de predios, con base en las metodologías aplicadas para dicho fin, obteniendo así el grado de cumplimiento de la normatividad operativa, la correcta aplicación de los apoyos y la depuración del directorio PROCAMPO.

El sistema a realizar deberá contemplar las características siguientes:

- Permitirá seleccionar los predios o conglomerados a verificar por cualquiera de las cuatro metodologías fijadas por la norma.
- Identificará aquellos predios seleccionados para ser verificados.
- Generará listados con información prellenada de los predios o conglomerados a verificar.
- Identificará los predios verificados.
- Generará diferentes reportes estadísticos.
- Controlará el historial de los resultados de la verificación de predios.

3.1.5 Identificación de los componentes del sistema

Para construir un modelo general del sistema es necesario determinar las clases relevantes y sus relaciones. Para determinar las clases se identifican los sustantivos de la descripción que potencialmente podrían considerarse como clases, estos sustantivos son:

Absolución	Dirección de Finanzas	Programa de Trabajo
Acta de Verificación de Campo	Elegibilidad	RAN (PROCEDE)
Acta Circunstanciada de Verificación de Expedientes	Habilitar procesos	Relación de predios a verificar
ASERCA	Ingenios Azucareros	Rep. de petición de liberación de predios
Baja definitiva	Inhabilitar procesos	Rep. de predios irregulares
CADER	INIFAP	Rep. de resultados de la revisión documental
CGAD	Instancias Externas	Rep. de seguimiento a los casos irregulares
Ciclo	Listado prellenado para verificar expedientes, predios y entrevistas	Rep. gen. de predios verificados
CAN	Metodología general	Reporte resumen de inspecciones por conglomerado
Conglomerado	Metodología dirigida	Reporte de campo
Cultivos	Metodología específica	Reporte de los trabajos de campo
DDR	Metodología preventiva	Reporte de revisión de expedientes
DEAAC	Minuta por conglomerado	SAGARPA
Delegación SAGARPA	Módulos de Riego	SEMARNAP
Departamento de Atención	Muestreo	Solicitud
DGPEAD	Normatividad	Subdirección de Cómputo
DGS	Oficio de comisión	Técnico Verificador de ASERCA
Diagnóstico	Oficios de notificación	Verificación de campo
Dictamen	Predio	Verificación documental
Dirección Regional	Productor	Entrevistas a productores

- CGAD** : Coordinación General de Apoyos Directos.
DGPEAD : Dirección General de Programación y Evaluación de Apoyos Directos.
DEAAC : Dirección de Estudios y Análisis de Apoyos al Campo.
DGS : Dirección General de Sistemas.

La selección de las clases debe estar sujeta a las consideraciones siguientes:

- Las clases deben derivarse de los conceptos que conforman el dominio del problema y no del dominio del diseño o del dominio de la aplicación, de ahí que el sustantivo base de datos se descarte (aún cuando aparece en el texto fuente) para la elaboración del modelo en la etapa de análisis.
- Dado que el modelo a construir inicialmente se enfoca a la estructura organizacional se excluyen las clases situadas a niveles de abstracción diferente.
- Se excluyen las clases redundantes.
- Los sustantivos de operaciones que son de carácter dinámico.

Se identificará el diagrama a nivel estructura organizacional, posteriormente se detallarán las dos áreas principales en las que opera actualmente el proceso de verificación de predios.

Con base en los criterios anteriores, las clases resultantes son:

- ASERCA Central
- SAGARPA
- Dirección Regional
- Delegación SAGARPA
- DDR (Distrito de Desarrollo Rural)
- CADER (Centro de Apoyo al Desarrollo Rural)
- Instancias Externas
- Productor

Una vez obtenida la lista de clases, se procede a identificar las relaciones existentes entre ellas para ello se determina en la descripción del sistema las referencias de cada clase hacia otras. Las relaciones comúnmente corresponden a verbos que indican un estado o a enunciados verbales. De los cuales se incluyen verbos de propiedad (compuesta por, cuenta, posee, ...), de ubicación (contenido a, adyacente a, ...), de autoridad (norma, regula, gobierna, ...), de comunicación (solicita a, habla con, ...), de cooperación (colabora, trabaja con, ...). Además las relaciones deben desarrollarse si la frase contiene relaciones implícitas entre más de dos clases. A continuación se describen las relaciones encontradas.

- SAGARPA cuenta con 33 Delegaciones, 193 DDR's y 712 CADER's.
- Delegación SAGARPA supervisa los trabajos de cada DDR.
- Delegación SAGARPA colabora con Dirección Regional.
- Delegación SAGARPA tiene a su cargo un número determinado de DDR's.
- ASERCA colabora con la SAGARPA
- ASERCA Central autoriza a la Dirección Regional presupuesto.
- ASERCA Central fija las normas para realizar el proceso de verificación de predios.
- ASERCA Central cuenta con 16 Direcciones Regionales.
- Dirección Regional se apoya de las Delegaciones SAGARPA.
- Dirección Regional administra el proceso de verificación.
- DDR coordina los trabajos de los CADER's.
- DDR colabora con la Dirección Regional.

- DDR consta de uno o más CADER's.
- CADER atiende a los requerimientos de los productores.
- CADER colabora con la Dirección Regional.
- CADER remite información al DDR.

Para fines del diagrama de clases a nivel estructura organizacional, se deben depurar aún más la lista de relaciones precedentes, por ello se realiza la selección de las relaciones que participarán en el modelo usando los criterios siguientes:

- 1) Deben omitirse relaciones que se encuentren fuera del dominio del problema.
- 2) Se deben incluir las relaciones de estructura de carácter permanente (y no aquellas referentes a sucesos transitorios) tales como las relaciones de estructura. Con base en este criterio se omiten las relaciones siguientes:

- ASERCA Central autoriza a la Dirección Regional presupuesto.
- Delegación SAGARPA supervisa los trabajos de cada DDR.
- Dirección Regional administra el proceso de verificación.
- DDR coordina los trabajos de los CADER's.
- CADER remite información al DDR.

Las relaciones derivadas en términos de otras se omiten por ser redundantes, tal es el caso de:

- SAGARPA cuenta con 193 DDR's.

que pueden ser inferidas de las relaciones:

- SAGARPA cuenta con Delegaciones.
- Una Delegación SAGARPA consta de DDR's.

Además de la relación anterior, la relación que se elimina por este criterio es:

- SAGARPA cuenta con 712 CADERS.
- ASERCA Regional opera en ámbito federal.

Las relaciones finales que participan en el modelo a nivel estructura organizacional son:

- SAGARPA cuenta con 33 Delegaciones Estatales.
- Delegación SAGARPA consiste de un número determinado de DDR's.
- ASERCA Central tiene 16 Direcciones Regionales.
- Dirección Regional se apoya de la Delegación SAGARPA.
- Dirección Regional solicita información sobre las condiciones termoplumiométricas a Instancias Externas.
- DDR's constan de un número determinado de CADER's.
- CADER's atienden a los requerimientos de los productores.

Con las relaciones y clases seleccionadas se elabora el diagrama de clases mostrado en la figura 3.1.1. No es coincidencia que este diagrama asemeje a un diagrama entidad-relación en el cual se presenta la cardinalidad en las relaciones y además con el símbolo "O" que indica la dirección de la relación, la clase que tiene próximo el símbolo es la que realiza el verbo y la clase en la cual se omite es la que recibe la acción.

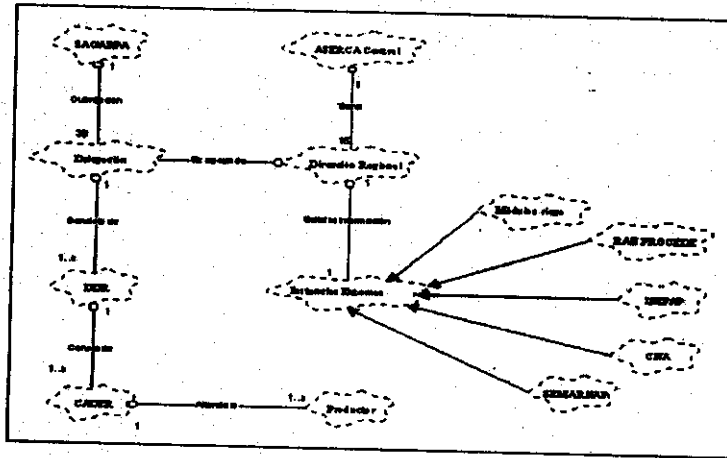


Figura 3.1.1 Diagrama de clases a nivel estructura organizacional

El siguiente modelo a construir describe la operación actual del proceso general de verificación de predios en ASERCA, por lo que es necesario determinar las clases relevantes y sus relaciones, excluyéndose las clases situadas en niveles de abstracción diferente. Las clases resultantes son:

- CGAD de ASERCA Central
- DGPEAD de ASERCA Central
- DEAAC de ASERCA Central
- Dirección de Finanzas de ASERCA Central
- Dirección Regional
- Delegación SAGARPA
- CADER
- Productor
- Normas
- Instancias Externas
- Informes de avances del programa

A continuación se identifican las relaciones existentes entre las clases obtenidas:

- DEAAC evalúa informes de avances del programa de verificación a la Dirección Regional.
- DEAAC elabora metodologías que rigen el funcionamiento de las Direcciones Regionales.
- DEAAC informa avances e irregularidades del programa de verificación de predios a la DGPEAD.
- DEAAC pertenece a la DGPEAD.
- DEAAC solicita radicación de presupuesto a la Dirección de Finanzas.
- DEAAC evalúa diagnóstico y programa de trabajo a la Dirección Regional.
- DGPEAD elabora las normas operativas del programa de verificación.

- DGPEAD rubrica el resumen de avances del programa de verificación elaborado por la DEAAC.
- DGPEAD pertenece a la CGAD.
- CGAD autoriza presupuesto a la Dirección Regional.
- CGAD autoriza el programa de trabajo de verificación a la Dirección Regional.
- CGAD consulta el resumen de avances del programa de verificación elaborado por la DEAAC.
- Dirección de Finanzas administra el presupuesto de la CGAD.
- Dirección de Finanzas radica presupuesto a la Dirección Regional.
- Dirección Regional administra el proceso de verificación de predios.
- Dirección Regional notifica irregularidades a la Delegación SAGARPA.
- Dirección Regional revisa expedientes en el CADER.
- Dirección Regional solicita a la Delegación SAGARPA instaurar procedimiento administrativo.
- Dirección Regional emite notificación al productor de la visita del Técnico Verificador de ASERCA para verificar su predio.
- Dirección Regional solicita informes termopluviométricos a Instancias Externas de las zonas elegidas para verificación.
- Dirección Regional reporta avances del programa de verificación a la DEAAC.
- Dirección Regional envía diagnóstico y programa de trabajo a la DEAAC.
- Dirección Regional dispone del presupuesto radicado por la Dirección de Finanzas de ASERCA Central.
- Delegación SAGARPA colabora con la Dirección Regional.
- Delegación Estatal pertenece a la SAGARPA.
- Delegación SAGARPA emite dictámenes administrativos para la Dirección Regional.
- CADER colabora con la Dirección Regional.
- CADER atiende a los productores.

Para establecer el diagrama de clases a nivel ASERCA, estas relaciones se deben depurar aún más, por lo que deben omitirse: las relaciones que se encuentren fuera del dominio del problema y las relaciones referentes a sucesos transitorios. Tal es el caso de las siguientes:

- Dirección de Finanzas administra el presupuesto de la CGAD.
- DEAAC elabora metodologías que rigen el funcionamiento de las Direcciones Regionales.
- DEAAC pertenece a la DGPEAD.
- DGPEAD pertenece a la CGAD.
- CGAD autoriza el programa de trabajo de verificación a la Dirección Regional.
- Dirección Regional administra el proceso de verificación de predios.
- Dirección Regional solicita a la Delegación SAGARPA instaurar procedimiento administrativo.
- Delegación SAGARPA colabora con la Dirección Regional.
- Delegación Estatal pertenece a la SAGARPA.
- Delegación SAGARPA emite dictámenes administrativos para la Dirección Regional.
- CADER colabora con la Dirección Regional.
- CADER atiende a los productores.

Las relaciones finales existentes en el proceso general de verificación de predios en ASERCA son:

- Dirección de Finanzas radica presupuesto a la Dirección Regional.
- CGAD autoriza presupuesto a la Dirección Regional.
- CGAD consulta el resumen de avances del programa de verificación elaborado por la DEAAC.
- DGPEAD rubrica el resumen de avances del programa de verificación elaborado por la DEAAC.
- DGPEAD elabora las normas operativas del programa de verificación.
- DEAAC informa avances e irregularidades del programa de verificación a la DGPEAD.
- DEAAC solicita radicación de presupuesto a la Dirección de Finanzas.
- DEAAC evalúa diagnóstico y programa de trabajo a la Dirección Regional.

- DEAAC evalúa los informes de avances del programa de verificación a la Dirección Regional.
- Dirección Regional envía diagnóstico y programa de trabajo a la DEAAC.
- Dirección Regional dispone del presupuesto radicado por la Dirección de Finanzas de ASERCA Central.
- Dirección Regional notifica irregularidades a la Delegación SAGARPA para la instauración del proceso administrativo que defina la situación del predio dentro del PROCAMPO.
- Dirección Regional solicita informes termoplumiométricos a Instancias Externas de las zonas elegidas para verificación.
- Dirección Regional emite notificación al productor de la visita del Técnico Verificador de ASERCA para verificar su predio.
- Dirección Regional revisa expedientes en el CADER para efectuar la verificación documental.
- Dirección Regional reporta avances del programa de verificación a la DEAAC.

De las relaciones antes descritas, se elabora el diagrama de clases mostrado en la figura 3.1.2.

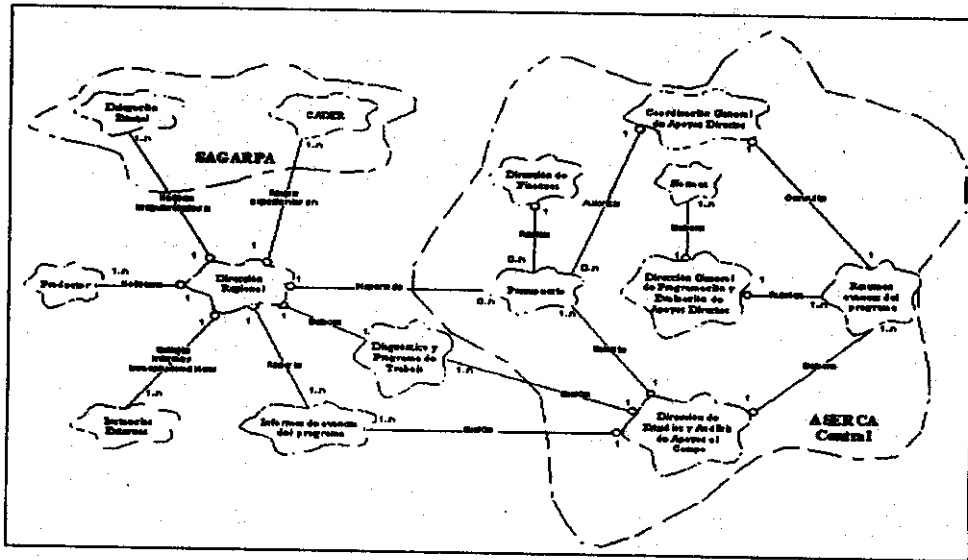


Fig. 3.1.2 Diagrama de clases de la operación actual del proceso general de Verificación de Predios en ASERCA

Dentro de la parte dinámica o denominada comportamiento. De la descripción general del sistema obtenemos los sucesos; estos están asociados donde ocurre un intercambio de información, además de que deben determinarse las clases participantes en los sucesos. Estos sucesos se modelan en el diagrama de la figura 3.1.3, donde se grafican las clases en el eje vertical, el tiempo en el eje horizontal y los sucesos se ubican en las intersecciones clase-tiempo. El diagrama de sucesos debe ser consistente y completo, en todo suceso debe existir un emisor y un receptor. A menos de los sucesos identificados como iniciales y finales, todo suceso debe tener un predecesor y sucesor.

Los sucesos que se presentan en la clase ASERCA son:

- Al iniciar el ciclo agrícola la DGPEAD solicita a la Dirección Regional el diagnóstico operativo del PROCAMPO de su ámbito de influencia.
- La Dirección Regional elabora y entrega dicho diagnóstico a la DGPEAD.
- La DGPEAD recibe y turna dicho diagnóstico a la DEAAC.
- La DEAAC evalúa y aprueba el diagnóstico y lo notifica a la DGPEAD.
- La DGPEAD solicita a la Dirección Regional el programa de trabajo de la verificación de predios.
- La Dirección Regional elabora y entrega el programa de trabajo a la DGPEAD.
- La DGPEAD recibe y turna dicho programa de trabajo a la DEAAC.
- La DEAAC evalúa y aprueba el programa de trabajo y lo notifica a la DGPEAD.
- La DGPEAD notifica a la CGAD la aprobación del programa de trabajo.
- La CGAD autoriza el programa de trabajo y la primera partida presupuestal.
- La DEAAC solicita a la Dirección de Finanzas la radicación de recursos presupuestales a la Dirección Regional (partida inicial).
- La Dirección de Finanzas radica los recursos presupuestales a la Dirección Regional y notifica a la DEAAC.
- La Dirección Regional realiza la verificación (documental y de campo), elabora los reportes de avances del programa y entrega a la DEAAC.
- La DEAAC recibe y evalúa los avances del programa, elabora un reporte resumen estadístico de los avances y lo entrega a la DGPEAD.
- La DGPEAD entrega el informe de avance del programa a la CGAD.
- La CGAD autoriza la siguiente partida presupuestal para darle continuidad al programa de verificación.
- La DEAAC solicita a la Dirección de Finanzas la siguiente radicación de recursos presupuestales a la Dirección Regional.
- La Dirección de Finanzas radica los recursos presupuestales a la Dirección Regional y notifica a la DEAAC.

El proceso de verificación de predios puede terminar de dos formas: la conclusión del programa de trabajo y/o el fin del ciclo agrícola.

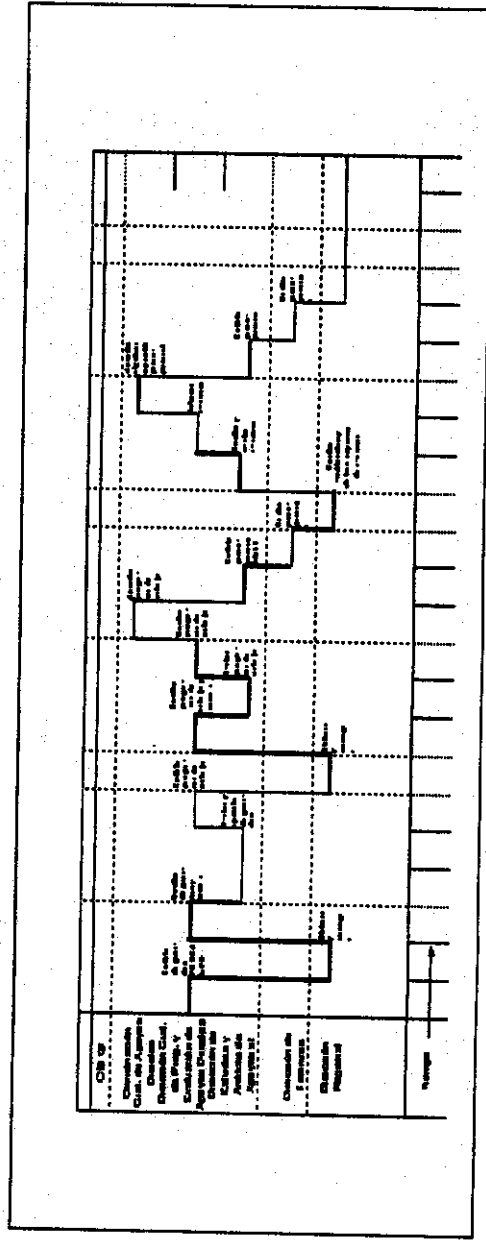


Figura 3.1.3 Diagrama de sucesos de la operación actual del proceso general de Verificación de Proyectos en ASERCA

Una vez autorizado el programa de trabajo se da inicio a la verificación de predios. La Dirección Regional obtiene los resultados de la verificación y elabora reportes estadísticos de avances del programa, que enviará a la DEAAC de ASERCA Central para su evaluación.

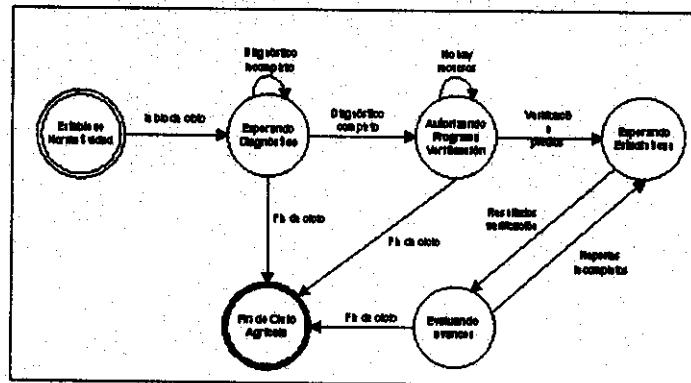


Fig. 3.1.5 Diagrama de estados de la operación actual del proceso general de Verificación de Predios en ASERCA.

3.1.5.1 Operación del programa de Verificación de Predios en ASERCA Regional

Para establecer la operación actual del mecanismo de verificación de predios en la Dirección Regional, es necesario determinar las clases relevantes y sus relaciones a partir de los sustantivos citados anteriormente, por lo que se excluirán las clases situadas en niveles de abstracción diferente, resultando las siguientes clases:

- Dirección Regional
- Departamento de Atención de la Dirección Regional (DA)
- Subdirección de Cómputo de la Dirección Regional (SC)
- Técnico de ASERCA de la Dirección Regional
- Delegación SAGARPA
- Instancias Externas
- Contraloría Interna SAGARPA
- Productor
- Predio
- Dictamen

- Listados prellenados
- Documento Verificación
- Apoyo
- CADER
- Notificación
- Solicitud de Apoyo

Se identifican las relaciones existentes entre las clases obtenidas:

- Delegación SAGARPA colabora con la Dirección Regional.
- CADER colabora con la Dirección Regional.
- CADER atiende a los productores.
- Dirección Regional notifica irregularidades a la Delegación SAGARPA.
- Dirección Regional solicita avances al DA.
- Dirección Regional solicita instaurar procedimiento administrativo a Delegación SAGARPA.
- Delegación SAGARPA colabora con la Dirección Regional.
- Delegación SAGARPA solicita dictamen a la Contraloría Interna.
- Contraloría Interna pertenece a la Delegación SAGARPA.
- SC pertenece a la Dirección Regional..
- SC colabora con el DA.
- DA pertenece a la Dirección Regional.
- DA asigna predios a verificar al Técnico de ASERCA Regional.
- DA envía la notificación al productor.
- DA solicita información termopluviométrica a Instancias Externas.
- DA solicita información a la SC.
- Técnico de ASERCA Regional pertenece al DA.
- Técnico de ASERCA Regional solicita expedientes al CADER para efectuar la verificación documental.
- Técnico de ASERCA Regional levanta documentos de verificación.
- Un predio es sujeto de apoyo.
- Un predio está avalado por uno o varios documentos de verificación.
- Una solicitud de apoyo hace referencia a un predio.
- Una solicitud puede obtener apoyo.
- Un dictamen evalúa un predio.
- Un dictamen está sustentado de uno o varios documentos de verificación.

Para establecer el diagrama de clases a nivel Dirección Regional, las relaciones anteriores se deben depurar aún más, por lo que deben omitirse: las relaciones que se encuentren fuera del dominio del problema y las relaciones referentes a sucesos transitorios. Tal es el caso de las siguientes:

- Delegación SAGARPA colabora con la Dirección Regional.
- CADER colabora con la Dirección Regional.
- CADER atiende a los productores.
- Dirección Regional solicita avances al DA.
- Dirección Regional solicita instaurar procedimiento administrativo a Delegación SAGARPA.
- Delegación SAGARPA colabora con la Dirección Regional.
- Delegación SAGARPA solicita dictamen a la Contraloría Interna.
- SC colabora con el DA.
- DA asigna predios a verificar al Técnico de ASERCA Regional.

Las relaciones finales existentes en el proceso de verificación de predios en la Dirección Regional son:

1. DA pertenece a la Dirección Regional.
2. SC pertenece a la Dirección Regional.
3. DA solicita información sobre las condiciones termoplumiométricas a Instancias Externas como: INIFAP, Módulos de riego, SEMARNAP, CNA, etc.
4. DA solicita a la SC, listados con información prellenada de la base de datos, de los predios sujetos a verificación.
5. DA entrega oficio de comisión y listados prellenados al Técnico de ASERCA Regional.
6. DA notifica al productor sobre la verificación que realizará el Técnico de ASERCA Regional al predio.
7. SC emite los listados con información prellenada; dichos listados se realizarán de acuerdo a la metodología aplicada, es decir:
 - Metodología Preventiva: Listado auxiliar para la verificación preventiva a conglomerados
 - Metodología Dirigida: Listado auxiliar para la verificación dirigida por conglomerados, Listado auxiliar para la verificación de campo y Listado auxiliar para la revisión de expedientes.
 - Metodología General: Listado auxiliar para la verificación de campo y Listado auxiliar para la revisión de expedientes.
 - Metodología Específica: Listado auxiliar para la verificación de campo y Listado auxiliar para la revisión de expedientes.
8. Técnico de ASERCA Regional trabaja para el DA, en el programa de verificación de predios.
9. Técnico de ASERCA Regional utiliza los listados prellenados para la realización de la verificación (en campo y/o documental).
10. Técnico de ASERCA Regional revisa los expedientes en el CADER correspondiente.
11. Técnico de ASERCA Regional levanta las actas de verificación.
12. Dirección Regional autoriza el oficio de comisión del Técnico de ASERCA Regional.
13. Dirección Regional notifica de las irregularidades a la Delegación SAGARPA, anexando al oficio de notificación, las actas de verificación levantadas y los oficios de comisión de los Técnicos comisionados.
14. Contraloría Interna forma parte de la estructura organizacional de la Delegación SAGARPA
15. Contraloría Interna de la SAGARPA emite un dictamen, que se sustenta en las actas de verificación levantadas por el Técnico de ASERCA Regional.
16. Dictamen evalúa al predio del productor.
17. Productor inscribe solicitudes de apoyo.
18. Solicitud de apoyo referencia un predio y obtiene un apoyo.
19. Predio es sujeto de apoyo.
20. Las actas de verificación y los oficios de comisión y de notificación avalan a uno o más predios.

De las relaciones antes descritas, se elabora el diagrama de clases mostrado en la figura 3.1.6.

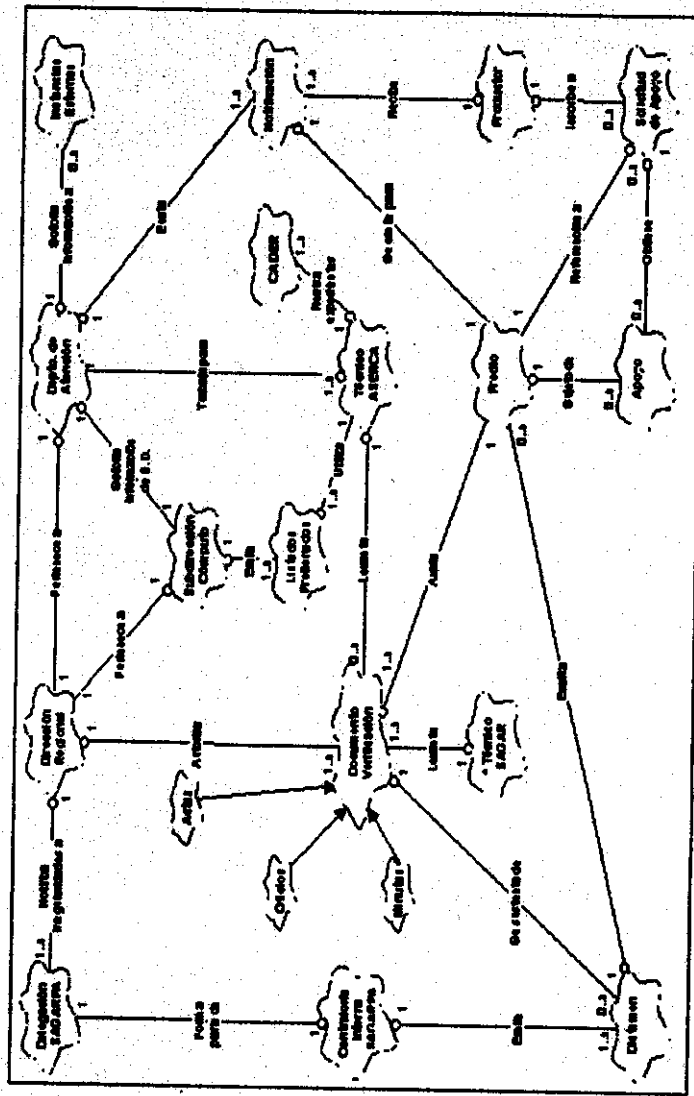


Figura 3.1.6 Diagrama de clases de la operación actual del mecanismo de verificación de prufios en la Dirección Regional

Los sucesos que se presentan en la clase Dirección Regional, dependerán de la metodología utilizada. Existen cuatro metodologías para la realización de la verificación de predios:

- Metodología Preventiva
- Metodología Dirigida
- Metodología General
- Metodología Específica

Los sucesos que se presentan en la Metodología Preventiva se muestran en la figura 3.1.7 y son los siguientes:

1. La Dirección Regional autoriza el inicio del programa de verificación de predios.
2. El DA, en base al diagnóstico y al programa de trabajo realizado con anterioridad, establece los criterios de verificación, para la selección de los conglomerados sujetos a verificación. Además solicita información a Instancias Externas como el INIFAP, CNA, Módulos de riego, etc. sobre las condiciones termopluviométricas y la disponibilidad de agua para riego, de las zonas donde se realizará la verificación.
3. Las Instancias Externas entregan información sobre humedad, precisan fechas límite de siembra en zonas de temporal errático, la precipitación mínima para la siembra, las temperaturas adecuadas, períodos de lluvias, etc.
4. El DA solicita a la SC el listado de información prellenada de los conglomerados sujetos a verificación ("**Listado Auxiliar para la Verificación Preventiva a Conglomerados**").
5. La SC elabora y entrega el listado prellenado al DA.
6. El DA entrega al Técnico de ASERCA Regional el oficio de comisión y el listado prellenado.
7. El Técnico de ASERCA Regional realizará recorridos de campo por los conglomerados a verificar. Reporta en dicho listado la situación encontrada, determinando la situación que prevalece en los conglomerados verificados, para definir aquellos para los cuales sí procede la entrega del formato de solicitud de apoyo. Posteriormente entrega los resultados de la verificación al DA.
8. El DA entrega el listado a la SC y notifica resultados a la Dirección Regional.
9. El DA realiza el **Reporte de los Trabajos de Campo**.
10. La Dirección Regional envía el RTV - **Reporte de los Trabajos de Verificación a ASERCA Central**.

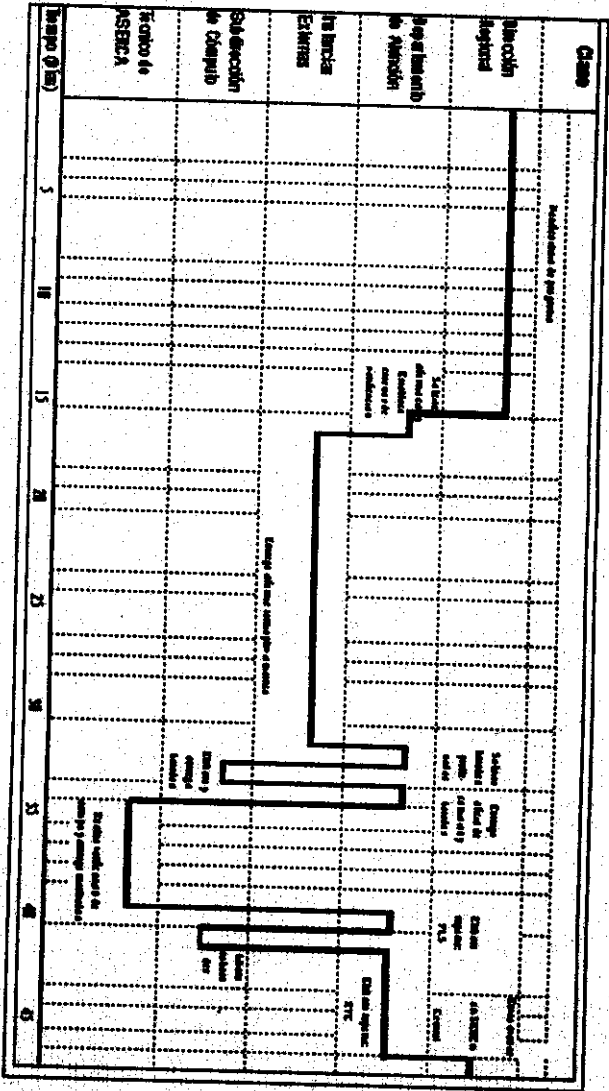


Figura 3.1.7 Diagrama de sucesos de la operación actual en la Dirección Regional utilizando la Metodología Preventiva

En base al objetivo y a la determinación del universo de trabajo de la Metodología Dirigida, existe la posibilidad de que se presente irregularidad o regularidad en los predios, por lo que los sucesos a describir se enfocarán a la presencia de predios regulares e irregulares.

Los sucesos que se presentan en la Metodología Dirigida para predios irregulares se muestran en la figura 3.1.5 y son los siguientes:

1. El DA, en base al diagnóstico y al programa de trabajo realizado con anterioridad, establece los criterios de verificación, para la selección de los conglomerados sujetos a verificación o los predios sujetos de verificación. Además solicita información a instancias externas como el RAN-PROCEDE, CNA, INIFAP, Módulos de riego, Ingenios azucareros, etc., sobre las condiciones termopluviométricas, la disponibilidad de agua para riego, etc., de las zonas donde se realizará la verificación.
2. El DA solicita a la SC los listados de información prellenados del conglomerado sujeto a verificación ("Listado auxiliar para la verificación dirigida por conglomerado"), si la verificación es a nivel predial ("Listado auxiliar para la verificación de campo, Listado auxiliar para la revisión de expedientes").
3. El Técnico de ASERCA Regional realizará un recorrido de campo por el conglomerado a verificar y complementará la información del listado, a los predios detectados como irregulares les levantará "Acta de Verificación de Campo"; también realizará la verificación documental en CADER y si detecta irregularidades levantará el "Acta de Verificación de Expedientes". Posteriormente entrega los resultados de la verificación al DA.
4. El DA realiza el reporte de "Relación de predios irregulares turnados al Delegado de la SAGARPA", el "Reporte de los trabajos de verificación" y notifica a la Dirección Regional el resultado de la verificación.
5. La Dirección Regional solicita a la SC detener los procesos en los que se encuentren las solicitudes de apoyo de(los) predio(s) irregular(es).
6. La SC inhabilita los procesos de las solicitudes y notifica a la Dirección Regional. Delegación SAGARPA anexando las actas de verificación levantadas, solicitando dictamen administrativo.
8. La Delegación SAGARPA solicita a la Contraloría Interna dar inicio al procedimiento administrativo de los predios irregulares.
9. La Contraloría Interna elabora y entrega a la Delegación SAGARPA el dictamen administrativo.
10. La Delegación SAGARPA entrega el dictamen a la Dirección Regional y a ASERCA Central.
11. La Dirección Regional solicita a ASERCA Central autorizar la aplicación del dictamen en la base de datos PROCAMPO. Se presentan dos tipos de resoluciones administrativas: la absolución del predio y la baja definitiva de la base de datos, en la primera la segunda es en ASERCA Central donde se realiza la baja.
12. La Dirección Regional envía a ASERCA Central los siguientes reportes estadísticos:
 - RTV - Reporte de los Trabajos de Verificación
 - RPI - Relación de Predios Irregulares turnados al Delegado de la SAGARPA

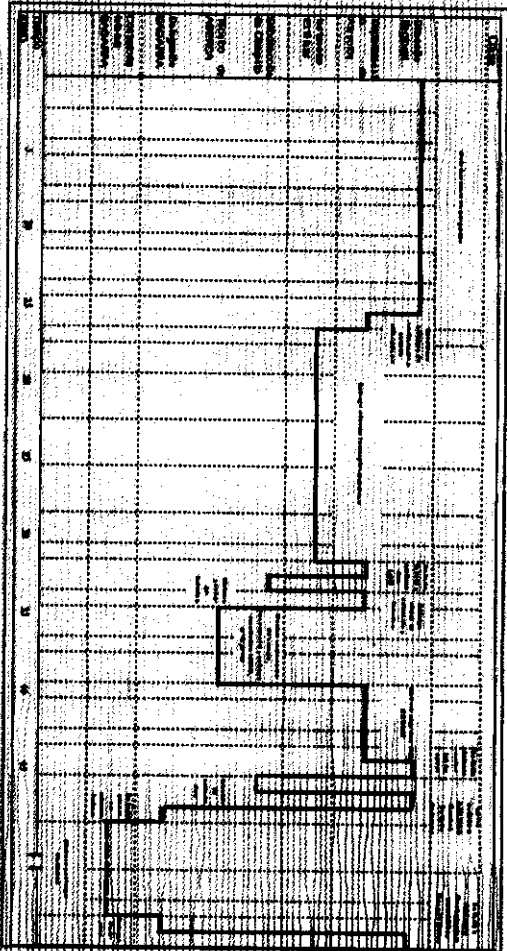


Figura 3.1.8 Diagrama de sucesos de la operación actual en la Dirección Regional utilizando la Metodología simplificada para Paredes Irregulares.

En la Metodología General y Específica, se presentan los mismos sucesos en el desarrollo de la verificación, la diferencia entre ambas radica en el universo al cual se le tomará la muestra, para la aplicación del muestreo aleatorio, en ambas existe la posibilidad de que se presente irregularidad o regularidad en los predios, por lo que los sucesos a describir se enfocarán a la presencia de predios regulares e irregulares.

Los sucesos que se presentan en las Metodologías General y Específica para predios irregulares se muestran en la figura 3.1.9 y son los siguientes:

1. El DA, en base al diagnóstico y al programa de trabajo realizado con anterioridad, establece los criterios de verificación, para la selección de los predios sujetos a verificación.
2. El DA solicita a la SC, los listados de información prellenados de los predios sujetos a verificación ("Listado auxiliar para la revisión de expedientes y el Listado auxiliar para la verificación en campo").
3. El Técnico de ASERCA Regional realizará la verificación de campo complementando la información en la Relación de predios a verificar y sólo levantará "Acta de Verificación de Campo" para los predios que resulten irregulares; también realizará la verificación documental en CADER y si detecta irregularidades levantará el "Acta de Verificación de Expedientes". Posteriormente entrega los resultados de la verificación al DA.
4. El DA realiza el reporte de "Reporte de los trabajos de verificación", la "Relación de predios irregulares turnados al Delegado de la SAGARPA", y notifica a la Dirección Regional el resultado de la verificación.
5. La Dirección Regional envía a ASERCA Central los siguientes reportes estadísticos:
 - RTV - Reporte de los Trabajos de Verificación
 - RPI - Relación de Predios Irregulares turnados al Delegado de la SAGARPA

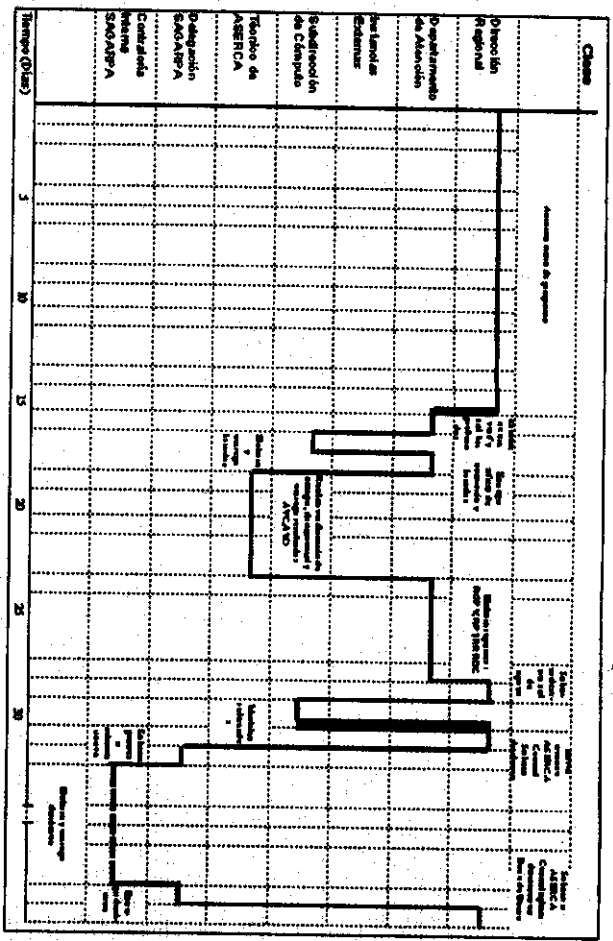
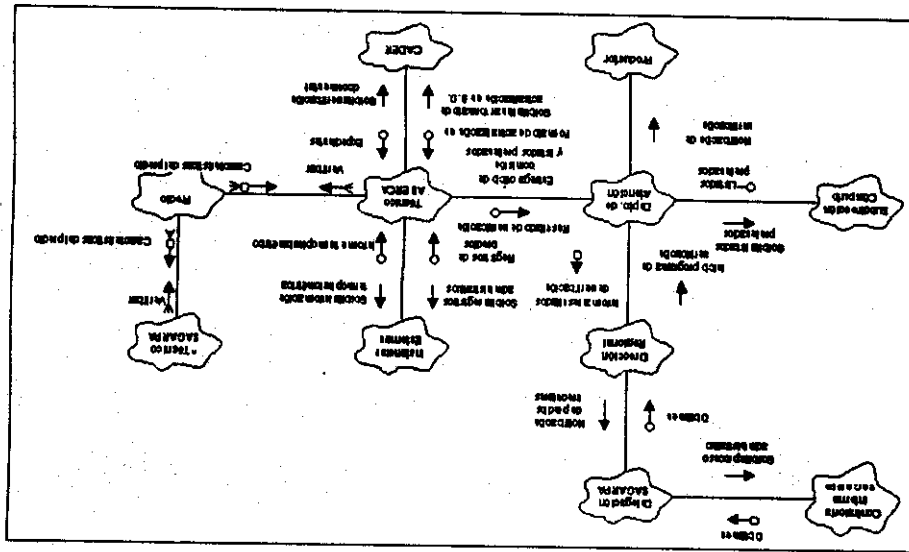


Figura 3.1.9 Diagrama de sucesos de la operación actual en la Dirección Regional utilizando la Metodología General y Específica para Prácticas Irregulares

Se elabora el diagrama de estados del predio como parte del proceso de verificación, el cual inicia con la existencia del predio inscrito en PROCAMPO que puede ser sujeto a verificación, siendo éste el evento que activa la operación del proceso. A partir de este momento el predio puede ser seleccionado para verificación y a su vez puede ser calificado como un predio regular o irregular. Si se determina como predio irregular presentará un estado de detención hasta que se dicte la absolución o la baja definitiva del predio del directorio PROCAMPO, ver la figura 3.1.12.

En base a la lista de sucesos presentados en la Dirección Regional para cada metodología utilizada en la verificación, se elabora el diagrama de estados, el cual parte de que la verificación de predios es el evento que activa la operación de la Dirección Regional, donde asigna a los Técnicos de ASEFCA Regional para que realicen las labores de verificación tanto en campo, como en expediente, de tal forma que ellos son los encargados de presentar los resultados reflejados en los predios, para elaborar los reportes estadísticos y canalizar las situaciones irregulares para su proceso administrativo, y por último la depuración o actualización del directorio PROCAMPO, ver la figura 3.1.11.

Figura 3.1.10 Diagrama de objetos de la operación actual del mecanismo de verificación de predios en la Dirección Regional



El diagrama de objetos correspondiente a la operación actual de la verificación de predios en la Dirección, se muestra en la figura 3.1.10.

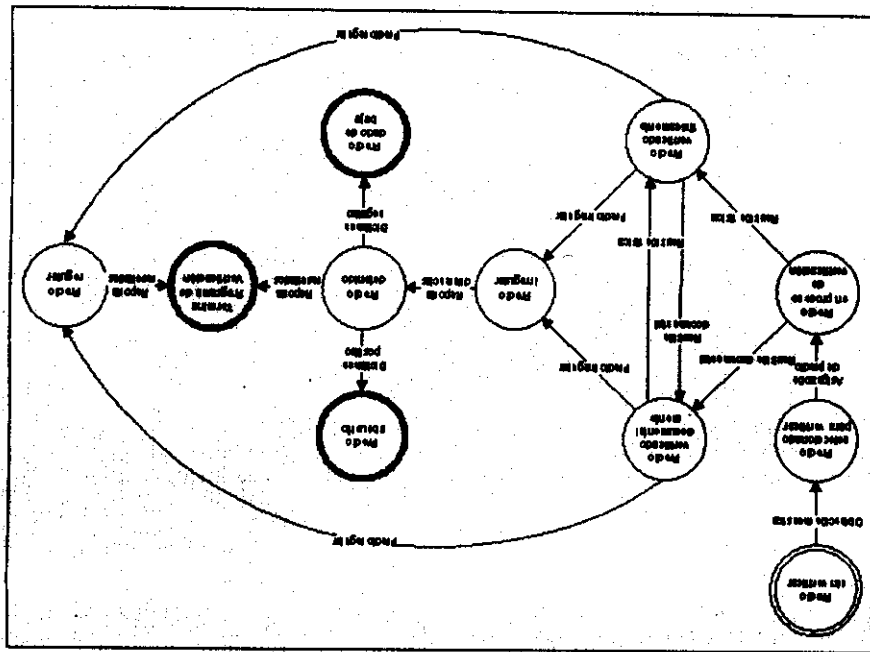
- Utilizando computadoras personales.
- Utilizando los equipos SUN Enterprise Server con sistema operativo Solaris (UNIX System V).

Para resolver la problemática, se describen las siguientes opciones de solución:

El problema a resolver es el de llevar un control del proceso de verificación de predios, soportado las cuatro metodologías establecidas en la normatividad. En las alternativas de solución se considera el estado actual en cuanto a equipo y software de ASERCA.

3.1.6 Evaluación de alternativas de solución

Figura 3.1.12 Diagrama de estados de la operación de un predio



3.1.6.1 Solución basada en computadoras personales

Elegir esta opción para automatizar el proceso de verificación de predios, presenta las siguientes ventajas y desventajas:

Ventajas:

- Software más actualizado que permite crear aplicaciones con interfaces gráficas.
- Compatibilidad con otras herramientas.
- Desarrollo rápido de aplicaciones.
- Facilidad de poderse instalar en cualquier PC.

Desventajas:

- Duplicidad de información.
- Falta interfase de comunicación para extraer y actualizar la información en la Base de Datos PROCAMPO.
- No se cuenta con redes de computadoras en las Direcciones Regionales.

3.1.6.2 Solución basada en los equipos SUN Enterprise Server

Esta solución presenta las siguientes ventajas y desventajas:

Ventajas:

- La principal ventaja que presenta esta solución, es el ahorro de hardware y software, ya que tanto ASERCA Central como la Dirección Regional cuentan con ellos.
- Evitar la duplicidad de información.
- Utilizar un manejador de Base de Datos confiable y seguro.

Desventajas:

- El software no permite crear aplicaciones con interfaces gráficas orientadas a Windows.
- Dependencia de los recursos de cómputo disponibles en las Direcciones Regionales.
- Los recursos de cómputo dependen de la subdirección de cómputo, por lo que no se encuentran al alcance del Departamento de Atención.
- Las aplicaciones sólo se podrán ejecutar en las Direcciones Regionales.
- En las oficinas de la SAGARPA que intervienen en el programa PROCAMPO no cuentan con equipo similar a las Direcciones Regionales, por lo que el funcionamiento está limitado a dichas direcciones.

3.1.6.3 Selección de la solución

De las alternativas anteriormente descritas, se elegirá la que tenga los costos menores (equipos, programación, contabilidad, ...)

Las condiciones del entorno a las que estará sujeta la solución son:

- La transferencia de información se realizará mensualmente y se utilizará como medio de transmisión la red de comunicaciones de ASERCA.

Los costos a evaluar no son trascendentales, ya que se van a utilizar los recursos propios de ASERCA, tanto materiales como humanos, sin embargo, para el sistema de verificación de predios los costos a evaluar son:

- Tiempo de respuesta:

Dependerá directamente de la carga de trabajo que tenga en ese momento el equipo de la Dirección Regional.

- Tiempo de transferencia de información:

Está en función de la velocidad de transmisión, el horario:

- Costo económico.
- Costo de esfuerzo.

El costo de desarrollo de la aplicación se determina a partir de las actividades, las cuales son:

- Análisis y diseño del programa de verificación de predios.
- Programación de los módulos.
- Pruebas de integración de los módulos.
- Implantación del sistema.

Estas dos características se combinan tal como se muestra en la figura 3.2.2 donde se presenta el protocolo y/o servicios utilizados, así, los enlaces vía fibra óptica están soportados por el protocolo de Ethernet cuando la conexión es local, mientras los enlaces remotos se soportan bajo el servicio de Frame Relay; por otro lado, enlaces a través de par trenzado están bajo el protocolo Ethernet para conexiones remotas y para enlazar las PCs con los servidores se utiliza Ethernet.

- Locales
- Remotos

Por el alcance de los enlaces se consideran dos tipos:

- Par trenzado.
- Fibra óptica.

comunican a través de dos medios:

figura 3.2.1 Distribución de equipo de cómputo en ASERCA, físicamente los nodos se extiende (WAN) que actualmente cuenta cada una con una red de área local (LAN), ver República Mexicana (en las 16 Direcciones Regionales) formando una red de área en 17 nodos servidores SUN, los cuales se ubican geográficamente a través de la solución a implantar estará bajo un entorno de base de datos relacionales distribuidas

3.2.1 Arquitectura de los subsistemas

- Arquitectura de los subsistemas.
- Esquema de la base de datos.
- Mecanismo de verificación de predios.

Una vez planteado el problema y construido los modelos que representan el proceso de verificación de predios, se puede iniciar la construcción de la solución seleccionada. En esta etapa se incorporan los componentes en equipo y programas que forman parte del sistema y se conforma por los siguientes puntos:

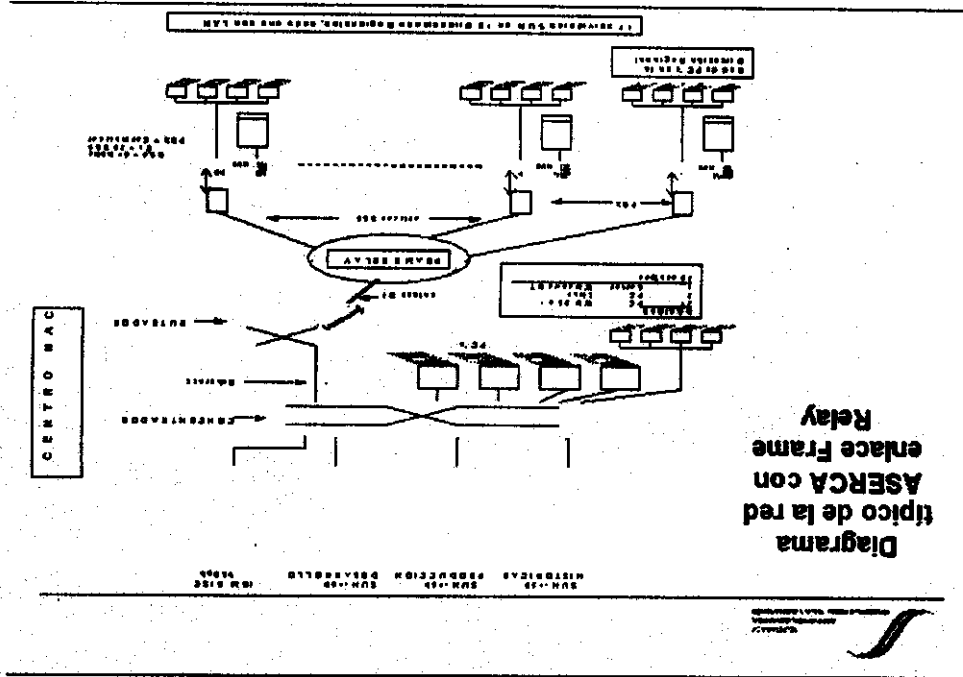
3.2 DISEÑO

Figura 3.2.2 Identificación de los tipos de enlaces en ASERCA

Medio Físico / Alcance	Par Trenzado	Fibra Óptica
Local	Ethernet	Ethernet
Remoto	Ethernet	Ethernet

En la figura 3.2.1 se muestra la distribución de los nodos y cada uno contiene un servidor SUN con sistema operativo AIX operando bajo el protocolo de red TCP/IP en la capa de transporte. En cada servidor existe una instancia de base de datos en la cual se registran las transacciones diarias. Las instancias existentes se dividen por su estructura en dos tipos: instancia regional e instancia central. En cada regional existen aproximadamente 25 PCs en promedio, en las cuales se realizan operaciones de captura, corrección, cálculo de apoyo, etc.

Figura 3.2.1 Distribución de equipo de cómputo en ASERCA



- Esquema conceptual PROCAMPO para los Centros Regionales, identificado con el nombre de Instancia REG.
 - Esquema conceptual PROCAMPO para ASERCA Central, identificado con el nombre de Instancia ASERCA.
- En ASERCA se cuenta con el sistema automatizado de información PROCAMPO para el control del Programa de Apoyos Directos al Campo. Este sistema controla las transacciones de la operación diaria en los centros regionales y oficinas centrales, análogamente a estas categorías se distinguen dos esquemas conceptuales de base de datos:

3.2.2 Esquema de base de datos

Figura 3.2.3 Componentes de Software

	Aplicaciones	
	Manejador de Base Datos	FTP, Telnet
	TCP/IP	
	Ethernet	Ethernet

Los componentes de software se presentan en la Figura 3.2.3, la cual incluye los principales componentes y algunos servicios de uso común como ftp y telnet.

- Entrega de Apoyos.
- Registro Voluntario.
- Mantenimiento.
- Control de Almacén.
- Control de Pesticidas.
- Sistemas de Atención
- Atención a Productores

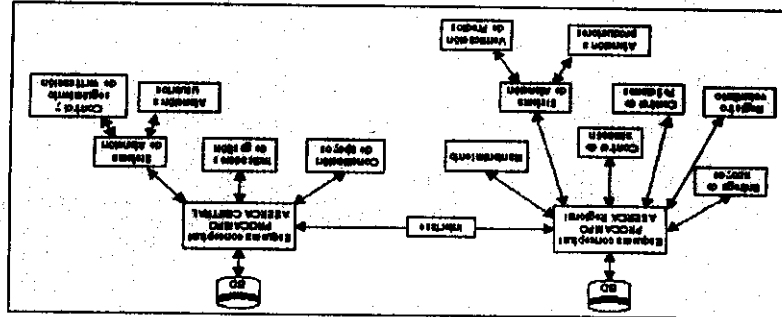
Esquemas externos para ASERCA Central.

Control de Verificación de Predios

Esquemas externos para la Dirección Regional.

La interfaz de comunicación entre los esquemas conceptuales, esta basada en la herramienta SQL*Net de Oracle. Cada uno de los esquemas a su vez consiste de esquemas externos que son el medio por el cual el usuario accede a los datos. A continuación se listan las partes de cada esquema conceptual:

Figura 3.24 Arquitectura global de los subsistemas de ASERCA



La arquitectura de los esquemas de la base de datos de ASERCA se muestran en la figura 3.2.4. Los esquemas involucrados en la realización de la solución son: el esquema externo Seguimiento de Verificación de Predios y el esquema Control de Verificación de Predios. Por ello es necesario detallar mediante diagramas de clases estos esquemas.

El esquema conceptual externo de Control de Verificación de Predios en la Dirección Regional consiste de las siguientes entidades:

- **Productor:** Persona registrada como agricultor en cualquier ciclo agrícola y que puede ser beneficiada por sembrar algunos de los cultivos básicos.
- **Solicitud de Apoyo:** Petición de apoyo económico, hecha por un productor a ASERCA durante el año agrícola para la siembra de alguno de los cultivos básicos.
- **Apoyo:** Monto derivado de una superficie sembrada con cultivos básicos correspondientes a una solicitud.
- **Documento:** Es el nombre que generaliza a los distintos documentos de verificación que se emplean para justificar alguna actividad realizada en el proceso de verificación de predios, por ejemplo: actas de verificación, oficios de comisión, etc.
- **Diccionario:** Es el veredicto que otorga la Contraloría Interna de la SAGARPA respecto a la situación de un predio irregular.
- **Contraloría Interna de la SAGARPA:** Área encargada de ejercer juicio legal sobre un predio irregular.
- **Departamento de Atención:** Área encargada de realizar y dar seguimiento al proceso de verificación de predios.
- **Técnico ASERCA:** Persona autorizada por la Dirección Regional para llevar acabo la verificación en campo, la revisión de expediente de un predio y la entrevista al productor.
- **Delegación SAGARPA:** Área que controla, coordina y administra las actividades relacionadas con los CADER's.
- **CADER:** Centro de Apoyo al Desarrollo Rural encargado de resguardar la documentación correspondiente a la tramitación de los apoyos y atender a los productores.
- **Dirección Regional:** Área que controla y coordina las actividades en una Dirección Regional.
- **Subdirección de Computo:** Área encargada de administrar los recursos informáticos en la Dirección Regional.
- **Predio:** Superficie definida de tierra identificada por un documento legal y registrada en PROCCAMPO.
- **Notificación:** Documento utilizado para informar al productor día y hora en el que se verificará el predio, así como el nombre de la persona responsable que realizará la verificación del predio.
- **Listados Prelimenes:** Reportes generados con información preliminar de la base de datos sobre los predios seleccionados para la verificación, los cuales serán complementados con la información que se obtenga como resultado de la verificación.
- **Instancias Externas:** Instituciones que realizan actividades inherentes a la climatología, a la agricultura, etc.

El esquema conceptual de base de datos en ASERCA Central se compone de 6 entidades:

- **Coordinación General de Apoyos Directos:** Área encargada de coordinar todas las actividades inherentes al programa PROCCAMPO.
- **Dirección General de Programación y Evaluación de Apoyos Directos:** Área encargada de establecer las normas que rigen las actividades en PROCCAMPO.
- **Dirección de Estudios y Análisis de Apoyos al Campo:** Área encargada de evaluar los avances sobre el proceso de verificación de predios.
- **Informes:** Archivos o documentos con información referente a los avances del programa de verificación de predios.

- Presupuesto. Cantidad monetaria que se autoriza para actividades específicas.
- Normas. Conjunto de procedimientos que rigen actividades específicas.
- Dirección de Finanzas. Área encargada de administrar el presupuesto asignado a la Coordinación de Apoyos Directos.
- Diagnóstico y Programa de Trabajo. Documentos que contienen la problemática existente en una área de influencia de la (s) Dirección(es) Regionales, considerando los trabajos a realizar para dar cumplimiento a la normatividad sobre la verificación de predios y esclarecer la situación de los predios inscritos en el PROCA/MPQ. Dichos documentos los elabora la Dirección Regional, pero es ASERCA Central quien debe autorizar los trabajos a realizar.

Los esquemas anteriormente descritos, son esquemas de cómo está modelado actualmente la operación de la verificación de predios, pero el objetivo del presente diseño lógico, es el de automatizar las actividades de almacenamiento, consulta y seguimiento de las verificaciones de predios, utilizando la interfase para compartir la información de las Direcciones Regionales con ASERCA Central.

La operación del programa de verificación de predios en la Dirección Regional y ASERCA Central se realiza de manera diferente, por lo que muestra las siguientes diferencias:

- Las entidades Productor. Solicitud de Apoyo, Apoyo, Documento, Departamento de Atención, Técnico ASERCA, CADER, Delegación SAGARPA, Subdirección de Cómputo, Predio, Listados Preenados, Instancias Externas, no son necesarias para la operación que se realiza del programa en ASERCA Central.
- Las entidades Coordinación Gral. de Apoyos Directos, Dirección Gral. de Programación y Evaluación de Apoyos Directos, Dirección de Estudios y Análisis de Apoyos al Campo no son necesarias para la operación del programa en la Dirección Regional.

La información de las entidades que deben de compartir la Dirección Regional y ASERCA Central, son las siguientes:

- Dictamen.
- Normas.
- Presupuesto.
- Diagnóstico y Programa de Trabajo de la verificación de predios.
- Informes.

Para el desarrollo del programa de control de verificación de predios y el de seguimiento se comparará la información de predios, las entidades que nos interesan son las dos últimas, ya que Regional: Listados Preenados, Documento, Notificación.

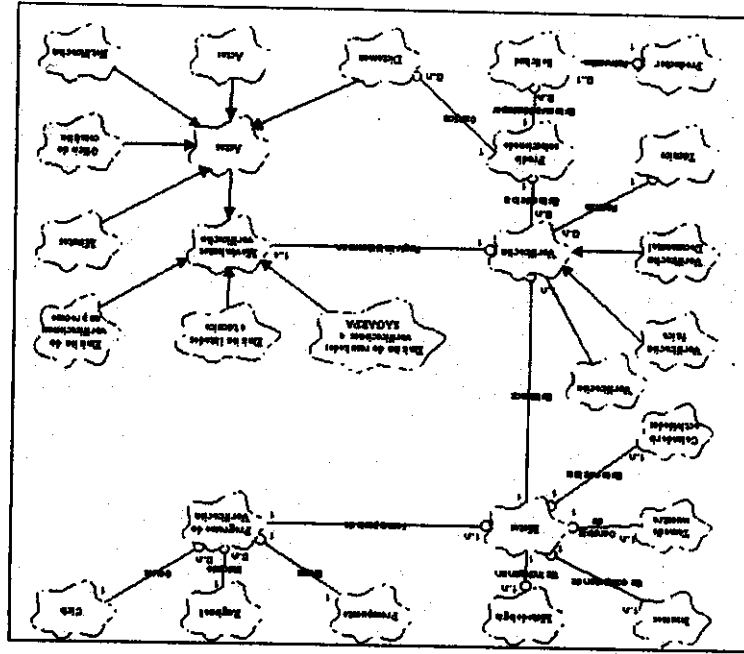
3.2.3 Mecanismo de la Verificación de Predios.

Describiremos los atributos y métodos, de cada uno de los componentes del programa de verificación de predios.

Entidad	Atributos	Métodos
Programa de Verificación	Ciclo, estado, presupuesto, metas.	Genera, Autoriza, Amplia, Restringe,
Ciclo	Clave.	Modifica.
Estado	Clave, nombre.	Genera, Modifica

Presupuesto	No. de partida, monto asignado, estado, ciclo, Determina, Autoriza, Erga.	Metodología, insumos, tamaño de la muestra, Genera, Amplia, Modifica.	Clave, nombre, Genera, Modifica
Metodología	Presupuesto asignado por meta, presupuesto ejercido por meta, proyecto, partida, Se radica.	Presupuesto asignado por meta, presupuesto ejercido por meta, proyecto, partida, Se radica.	Clave, nombre, Genera, Modifica
Tamaño muestra	Total de predios, total de superficie elegible, tenencia, régimen hídrico, ciclo, Se genera, se obtiene.	Total de predios, total de superficie elegible, tenencia, régimen hídrico, ciclo, Se genera, se obtiene.	Clave, nombre, Genera, Modifica
Calendario de actividades	Delegación, ciclo, tamaño de la muestra, superficie elegible, fecha inicial, fecha final, Establece, Opera, Modifica	Delegación, ciclo, tamaño de la muestra, superficie elegible, fecha inicial, fecha final, Establece, Opera, Modifica	Clave, nombre, Genera, Modifica
Verificación	Cartografía, tipo, predio, predio secuencial, Atlas, Cambios técnicos, acta, solicitud, clave del productor, Se genera	Cartografía, tipo, predio, predio secuencial, Atlas, Cambios técnicos, acta, solicitud, clave del productor, Se genera	Clave, nombre, Genera, Modifica
Verificación física	No. oficio de comisión, técnico, listados de verificación, Se ratifica (resultados)	No. oficio de comisión, técnico, listados de verificación, Se ratifica (resultados)	Clave, nombre, Genera, Modifica
Verificación documental	No. oficio de comisión, técnico, listados de verificación, Se ratifica (resultados)	No. oficio de comisión, técnico, listados de verificación, Se ratifica (resultados)	Clave, nombre, Genera, Modifica
Entrevista al Productor	No. oficio de comisión, técnico, listados de verificación, Se ratifica (resultados)	No. oficio de comisión, técnico, listados de verificación, Se ratifica (resultados)	Clave, nombre, Genera, Modifica
Técnico	Clave, nombre, Se designa, Verifica	Clave, nombre, Se designa, Verifica	Clave, nombre, Genera, Modifica
Productor	Clave, apellido paterno, apellido materno, nombre, Consultas, Alias, Bajas, Cambios	Clave, apellido paterno, apellido materno, nombre, Consultas, Alias, Bajas, Cambios	Clave, nombre, Genera, Modifica
Predio seleccionado	Clave, predio secuencial, superficie, elegible, tenencia, Consultas, Inhabilita, Habilita, Selección	Clave, predio secuencial, superficie, elegible, tenencia, Consultas, Inhabilita, Habilita, Selección	Clave, nombre, Genera, Modifica
Solicitud	Solicitud, productor, predio, predio secuencial, Consultas	Solicitud, productor, predio, predio secuencial, Consultas	Clave, nombre, Genera, Modifica
Dictamen	Dictamen, fecha, fallo, predio, Se emite, Alias	Dictamen, fecha, fallo, predio, Se emite, Alias	Clave, nombre, Genera, Modifica
Movimientos verificación	Tipo movimiento, movimiento, fecha, acta, ciclo, Se emite, se registra, Actualización (fechas, fallos)	Tipo movimiento, movimiento, fecha, acta, ciclo, Se emite, se registra, Actualización (fechas, fallos)	Clave, nombre, Genera, Modifica
Notificación de las irregularidades a SAGARPA	No. de oficio, fecha, predio, predio secuencial, solicitud, nombre del productor, acta, estado, evento	No. de oficio, fecha, predio, predio secuencial, solicitud, nombre del productor, acta, estado, evento	Clave, nombre, Genera, Modifica
Listados prellenados	Para la verificación documental: Cartografía, metodología, ciclo, solicitud, predio, predio secuencial, nombre del propietario, clave de identificación, tipo de identificación, clave de productor, tipo de identificación, clave de identificación, tenencia, superficie total, tipo de cultivo, modalidad, superficie elegible, superficie solicitada, superficie apoyada. Para la Metodología Preventiva: Cartografía, ciclo, fecha de corte, total de predios	Para la verificación documental: Cartografía, metodología, ciclo, solicitud, predio, predio secuencial, nombre del propietario, clave de identificación, tipo de identificación, clave de productor, tipo de identificación, clave de productor, tipo de identificación, tenencia, superficie total, tipo de cultivo, modalidad, superficie elegible, superficie solicitada, superficie apoyada. Para la Metodología Preventiva: Cartografía, ciclo, fecha de corte, total de predios	Clave, nombre, Genera, Modifica
Verificaciones en proceso	Para la Metodología Preventiva: Cartografía, ciclo, fecha de corte, total de predios	Para la Metodología Preventiva: Cartografía, ciclo, fecha de corte, total de predios	Clave, nombre, Genera, Modifica
Elabora: Reporte de los	Elabora: Listado de verificación en campo y listado para la revisión de expedientes)	Elabora: Listado de verificación en campo y listado para la revisión de expedientes)	Clave, nombre, Genera, Modifica

Figura 3.2.5 Relaciones entre los componentes del proceso de verificación de predios



En base a la descripción de las entidades involucradas en el diseño lógico para el programa de verificación de predios, se realiza el diagrama de clases, figura 3.2.5, el cual muestra los componentes y relaciones a nivel conceptual del proceso de verificación de predios.

Notificación	Fecha, no. de oficio, lugar, productor, técnico.	Determina.
Oficio de comisión	Fecha, no. de oficio, lugar, técnico.	Autoriza, designa.
Actas	Acta, evento, ciclo, fecha de entrega, estado, tipo. Para la Metodología General y Específica: superficie. verificados, superficie elegible, predios irregulares, predios, tamaño de la muestra, total de predios Cartografía, ciclo, fecha de corte, universo de superficie. Para la Metodología Dirigida: Cartografía, ciclo, fecha de corte, universo de predios, tamaño de la muestra, total de predios verificados, superficie elegible, predios irregulares, superficie. Para la Metodología General y Específica: Cartografía, ciclo, fecha de corte, tamaño de la muestra, predios verificados, superficie verificada, predios irregulares, superficie irregular, predios regulares, superficie regular, causas de irregularidad	Elabora: AVC, AVE.
		Trabajos de Verificación (RTV).

El proceso de verificación de predios se describe en dos etapas, es decir:

- Etapa 1) Inicia con el ciclo agrícola en operación; ASERCA Central solicita a la Dirección Regional elaborar el diagnóstico sobre la problemática que se presenta en las Delegaciones de su área de influencia, una vez evaluado y aprobado por ASERCA Central, la Dirección Regional elaborará el programa de trabajo que se implementará y enviará a ASERCA Central para su autorización.
- Etapa 2) El programa de verificación ocurre en un ciclo agrícola, se ubica en una Regional y además exige presupuesto. Las metas forman parte del programa de verificación, se compone de insumos (váticos, combustible y mantenimiento), consiste en determinar el tamaño de la muestra, esta sujeta a un calendario de actividades y incluye metodologías para la realización de los trabajos de verificación. Existen tres tipos de verificación: expedientes, campo y entrevista al productor. La verificación en sí, establece las metas a obtener y registra bitácora en los movimientos de verificación (generar listados administrativos, generar los reportes de avances del programa). El técnico verificador de ASERCA es el responsable de realizar las labores de verificación. Los predios seleccionados están sujetos a verificación. Los documentos levantados (actas de verificación, oficios de comitación, oficios de notificación, dictámenes) avalan la verificación. El dictamen administrativo calificó al predio seleccionado para la verificación. La solicitud de apoyo está respaldada por un predio, y es el productor quien reinscribe la solicitud de apoyo.

Describiremos los sucesos que se presentan en el proceso del programa de verificación; estos están asociados donde ocurre un intercambio de información, además de que deben determinarse las clases participantes en los sucesos. Estos sucesos se modelan en el diagrama de la figura 3.2.6.

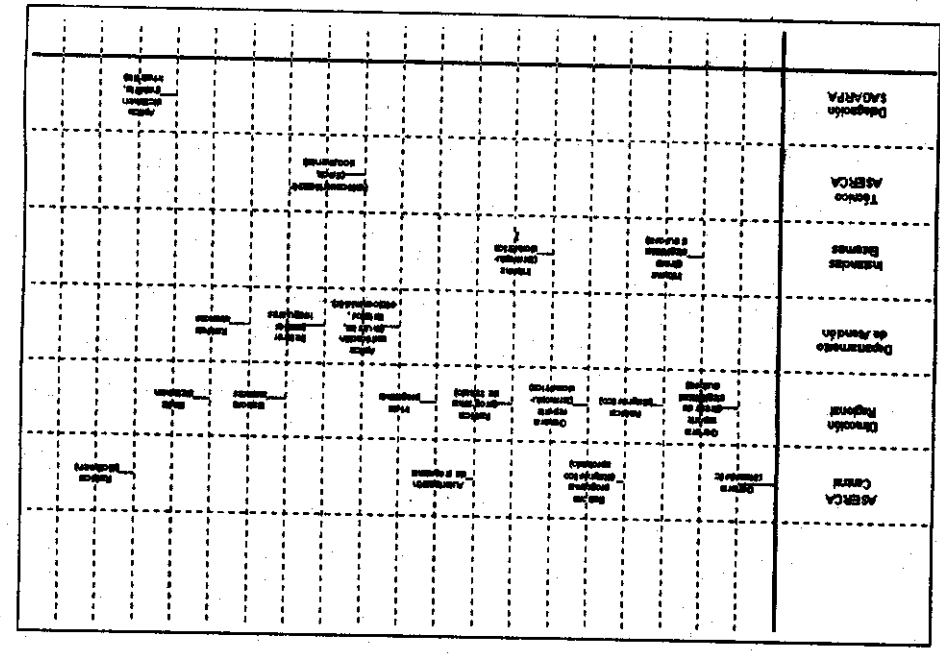


Figura 3.2.6 Diagrama de sucesos del programa de verificación de predios.

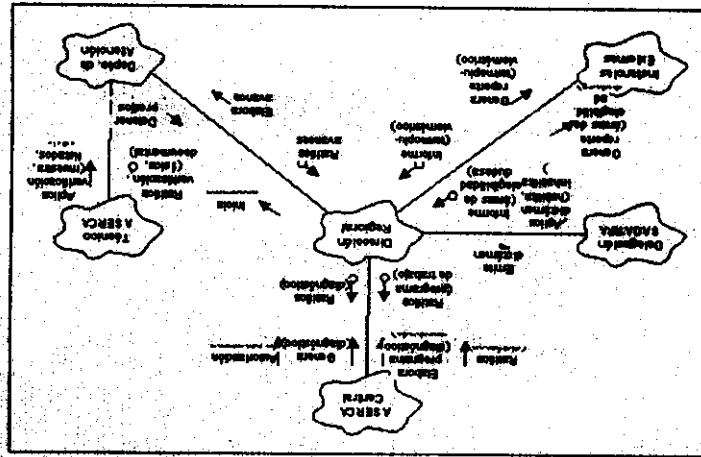
Los sucesos que se presentan en el programa de verificación de predios son:

1. Una vez que ha iniciado el ciclo agrícola ASERCA Central solicita a la Dirección Regional el diagnóstico que describe la problemática existente en la operación del PROCAMPO en el ámbito de influencia de las Delegaciones que les corresponden.
2. La Dirección Regional solicita información a instancias externas sobre áreas de elegibilidad dudosas (zonas cañeras, frutícolas, ganaderas, horticolas, tabacaleras, etc.), fechas de siembras, etc.
3. Las Instancias Externas (CNA, INIFAP, SEMARNAP, etc.) entregan la información solicitada a la Dirección Regional.
4. La Dirección Regional elabora el diagnóstico y lo remite a ASERCA Central.
5. ASERCA Central revisa, ratifica el diagnóstico y solicita la elaboración del programa de trabajo a la Dirección Regional.
6. La Dirección Regional solicita información a instancias externas sobre las condiciones termoplivométricas.
7. Las instancias Externas (CNA, INIFAP, PROCDE, SEMARNAP, etc.) entregan la información solicitada a la Dirección Regional.
8. La Dirección Regional elabora el programa de trabajo; donde se establecerán las metas a cubrir para la solución de la problemática existente y lo remite a ASERCA Central.
9. ASERCA Central evalúa, autoriza el programa de trabajo y se lo notifica a la Dirección Regional.
10. La Dirección Regional le notifica al Departamento de Atención el inicio del programa de verificación.
11. El Departamento de Atención aplica los criterios de selección de la muestra y emite los listados con información prellenada, los cuales entrega al Técnico de ASERCA junto con el oficio de comisión.
12. El Técnico de ASERCA realiza la verificación física (en campo), la documental (en CADER) y la entrevista al productor, la realización de la verificación física, documental y entrevista al productor, dependerá de la metodología aplicada. Como resultado de un predio o conglomerado irregular el Técnico levantará un Acta de Verificación de Campo (AVC) o un Acta de Verificación de Expedientes (AVE). Entregará los resultados obtenidos al jefe del Departamento de Atención.
13. El jefe del Departamento de Atención analiza los resultados obtenidos, detiene procesos de los predios que resultaron irregulares y notifica dichas irregularidades a la Dirección Regional.
14. La Dirección Regional solicita a la Delegación SAGARPA la instauración del procedimiento administrativo para los predios que resultaron irregulares.
15. El Departamento de Atención elabora los reportes de avances del programa y los envía a ASERCA Central.
16. La Delegación SAGARPA emite el dictamen con la resolución administrativa y lo envía a ASERCA Central y a la Dirección Regional.
17. ASERCA Central aplica la resolución administrativa en la base de datos PROCAMPO (siempre y cuando se trate de una baja definitiva de la Base de Datos; de lo contrario, autoriza a la Dirección Regional aplicar la resolución administrativa en la Base de Datos, en cuyo caso se tendrían que rehabilitar los procesos de los predios absueltos).

- Dirección Regional
- Departamento de Atención de la Dirección Regional (DA)
- Subdirección de Cómputo de la Dirección Regional (SC)
- Técnico Verificador de ASERCA Regional (TV)
- Delegación SAGARPA
- Instancias Externas
- Contratos Internos SAGARPA
- Productor
- Predio
- Dictamen
- Listados preformados
- Documento Verificación
- Apoyo
- CADER
- Notificación
- Solicitud de Apoyo

Para establecer la operación actual del mecanismo de verificación de predios en la Dirección Regional, es necesario determinar las clases relevantes y sus relaciones a partir de los sustantivos citados anteriormente, por lo que se excluirán las clases situadas en niveles de abstracción diferente, resultando las siguientes clases:

Figura 3.2.7 Diagrama de objetos del programa de verificación de predios.



El diagrama de objetos correspondiente al programa de verificación, se muestra en la figura 3.2.7.

Identificaremos las relaciones existentes entre las clases obtenidas:

- Delegación SAGARPA colabora con la Dirección Regional.
- CADER atiende a los productores.
- CADER atiende a los productores.
- Dirección Regional notifica irregularidades a la Delegación Estatal.
- Dirección Regional solicita avances al DA.
- Dirección Regional solicita instaurar procedimiento administrativo a Delegación SAGARPA.
- Delegación SAGARPA colabora con la Dirección Regional.
- Delegación SAGARPA solicita dictamen a la Contraloría Interna.
- Contraloría Interna pertenece a la Delegación SAGARPA.
- SC pertenece a la Dirección Regional.
- SC colabora con el DA.
- DA pertenece a la Dirección Regional.
- DA asigna predios a verificar al Técnico de ASERCA.
- DA envía la notificación al productor.
- DA solicita información termoplúvionométrica a Instancias Externas.
- DA solicita información a la SC.
- Técnico de ASERCA pertenece al DA.
- Técnico de ASERCA solicita expedientes al CADER para efectuar la verificación documental.
- Técnico de ASERCA levanta documentos de verificación.
- Un predio es sujeto de apoyo.
- Un predio está avalado por uno o varios documentos de verificación.
- Una solicitud de apoyo hace referencia a un predio.
- Una solicitud puede obtener apoyo.
- Un dictamen evalúa un predio.
- Un dictamen está sustentado de uno o varios documentos de verificación.

Para establecer el diagrama de clases a nivel Dirección Regional, las relaciones anteriores se deben depurar aún más, por lo que deben omitirse: las relaciones que se encuentran fuera del dominio del problema y las relaciones referentes a sucesos transitorios. Tal es el caso de las siguientes:

- Delegación SAGARPA colabora con la Dirección Regional.
- CADER colabora con la Dirección Regional.
- CADER atiende a los productores.
- Dirección Regional solicita avances al DA.
- Dirección Regional solicita instaurar procedimiento administrativo a Delegación SAGARPA.
- Delegación SAGARPA colabora con la Dirección Regional.
- Delegación SAGARPA solicita dictamen a la Contraloría Interna.
- SC colabora con el DA.
- DA asigna predios a verificar al Técnico de ASERCA.

Las relaciones finales existentes en el proceso de verificación de predios en la Dirección Regional son:

1. DA pertenece a la Dirección Regional.
2. SC pertenece a la Dirección Regional.
3. DA solicita información sobre las condiciones termoplúvionométricas a instancias Externas como: INIFAP, Módulos de riego, SEMARNAP, CNA, etc.
4. DA solicita a la SC, listados con información prellenada de la base de datos, de los predios sujetos a verificación.
5. DA entrega oficio de comisión y listados prellenados al Técnico de ASERCA.

- 3.2.8. De las relaciones antes descritas, se elabora el diagrama de clases mostrado en la figura.
20. Las actas de verificación, los oficios de comisión y de notificación avalan a uno o más predios.
 19. Predio es sujeto de apoyo.
 18. Solicitud de apoyo referencia un predio y obtiene un apoyo.
 17. Productor inscribe solicitudes de apoyo.
 16. Dictamen evalúa al predio del productor.
 15. Contraloría Interna de la SAGARPA emite un dictamen, que se sustenta en las actas de verificación levantadas por el Técnico de ASERCA.
 14. Contraloría Interna forma parte de la estructura organizacional de la Delegación SAGARPA.
 13. Dirección Regional notifica de las irregularidades a la Delegación SAGARPA, anexando al oficio de notificación, las actas de verificación levantadas y los oficios de comisión de los Técnicos comisionados.
 12. Dirección Regional autoriza el oficio de comisión del Técnico de ASERCA.
 11. Técnico de ASERCA levanta las actas de verificación para los casos irregulares.
 10. Técnico de ASERCA revisa los expedientes en el CADER correspondiente.
 9. Técnico de ASERCA (en campo, documental y entrevista al productor) verifica los expedientes en el CADER correspondiente.
 8. Técnico de ASERCA utiliza los listados prellenados para la realización de la verificación de ASERCA para el DA, en el programa de verificación de predios.
 7. SC emite los listados con información prellenada; dichos listados se realizarán de acuerdo a la metodología aplicada, es decir:
 - Metodología Preventiva: Listado prellenado de verificación preventiva en CADER.
 - Metodología Dirigida: Listado de predios a verificar (en campo) y Listado de solicitudes a verificar documentalmente. Listado de verificación dirigida a conglomerados.
 - Metodología General y Específica: Listado de predios a verificar (en campo) y Listado de solicitudes a verificar documentalmente.
 6. DA notifica al productor sobre la verificación que realizará el Técnico de ASERCA al predio.

Sistema de Control de Verificación de Predios

Los sucesos que se presentan en la clase Dirección Regional, dependerán de la metodología utilizada. Existen cuatro metodologías para la realización de la verificación de predios:

- Metodología Preventiva
- Metodología Dirigida
- Metodología General
- Metodología Específica

El diagrama de objetos correspondiente a la operación actual de la verificación de predios en la Dirección Regional, se muestra en la Figura 3.2.9.

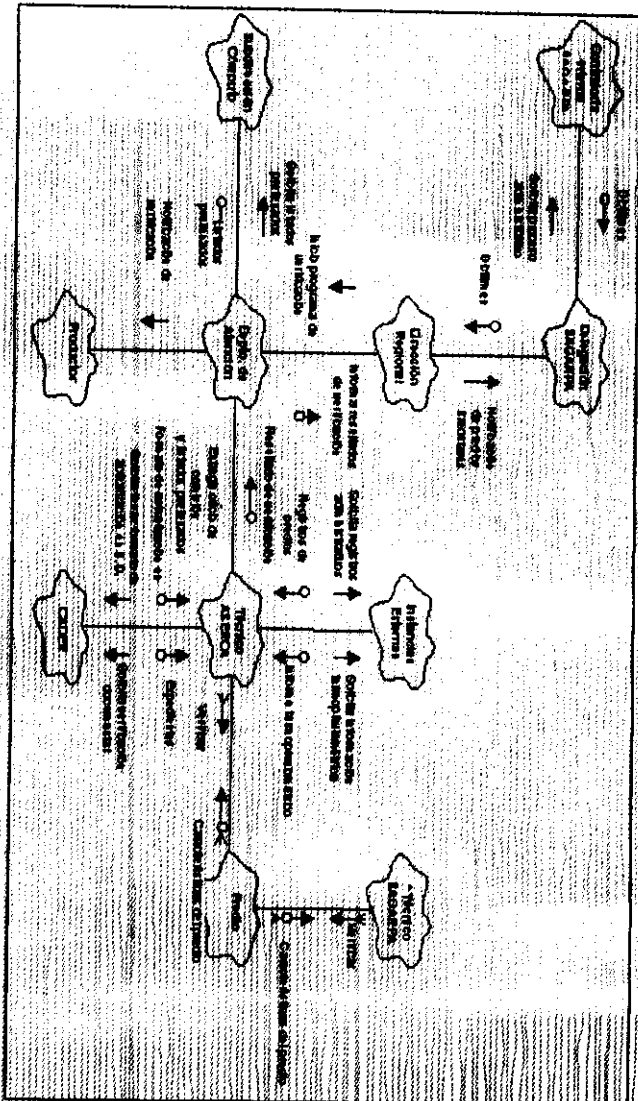


Figura 3.2.0 Diagrama de flujo de la operación actual del mecanismo de verificación de pruebas en la Dirección Regional

En base a la lista de sucesos presentados en la Dirección Regional para cada metodología utilizada en la verificación, se elabora el diagrama de estados, el cual para de que la verificación de predios es el evento que activa la operación de la Dirección Regional, donde asigna a los técnicos de ASERCA para que realicen las labores de verificación tanto en campo, expediente y entrevista al productor, de tal forma que ellos son los encargados de presentar los resultados reflejados en los predios, para elaborar los reportes estadísticos y canalizar las situaciones irregulares para su proceso administrativo, y por último la depuración o actualización del directorio PROCAMPO, ver la figura 3.2.10.

Se elabora el diagrama de estados del predio como parte del proceso de verificación, el cual inicia con la existencia del predio inscrito en PROCAMPO que puede ser sujeto a verificación, siendo este el evento que activa la operación del proceso. A partir de este momento el predio puede ser seleccionado para verificación y a su vez puede ser verificado documentalmente o físicamente, el resultado que presente dependerá que se califique como un predio regular o irregular. Si se determina como predio irregular presentará un estado de detención hasta que se dicte la resolución o la baja definitiva del predio del directorio PROCAMPO, ver la figura 3.2.11.

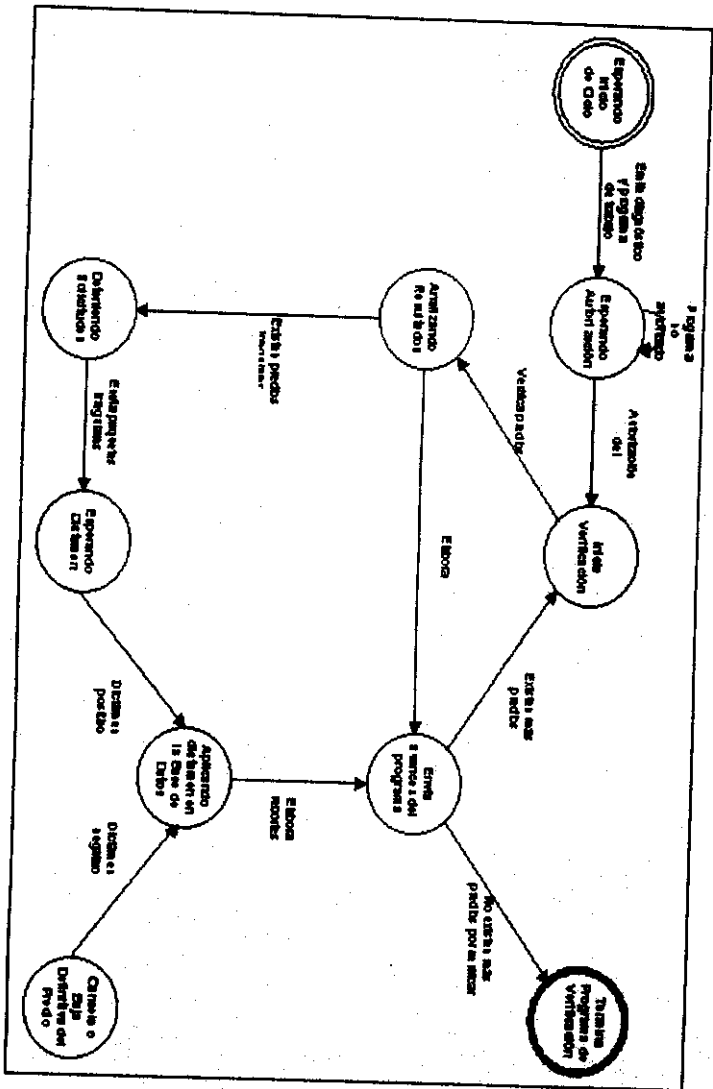


Figura 3.2.10 Diagrama de estados de la operación actual del mecanismo de verificación de predios en la Dirección Regional

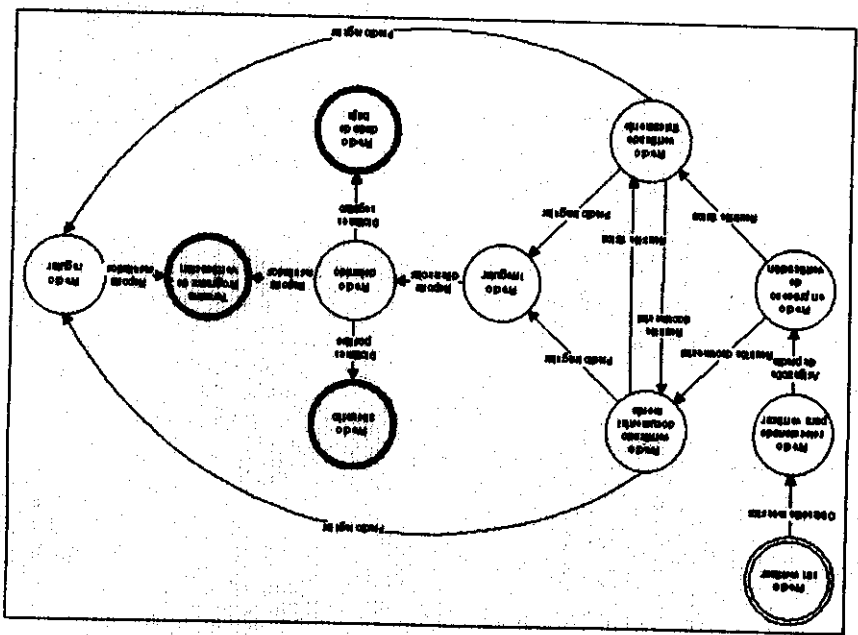


Figura 3.2.11 Diagrama de estados de la operación de un pedido

3.3 Implantación

Una vez definido el diseño lógico y conceptual de la solución se procede con la transformación de estos conceptos en un diseño físico, así cada componente lógico se hace corresponder a uno o varios componentes físicos, tales como tablas, archivos y programas.

3.3.1 Componentes de la base de datos

De las clases determinadas en el diseño sólo una parte se traducen en objetos del manejador de la base de datos.

En el manejador de la base de datos se pueden llevar a cabo tanto los componentes estáticos como dinámicos. Los componentes estáticos se realizan a través de tablas y sus restricciones de integridad, mientras que la parte dinámica se realiza a través de paquetes de PL-SQL almacenados y los triggers.

3.3.1.1 Determinación de tablas

Como el objetivo es obtener tablas bajo Oracle es importante considerar el espacio que ocupan, ya que cada tabla creada en Oracle ocupa dos bloques de espacio como mínimo, es decir, 8 Kilobytes sin importar se ocupe o no.

Cabe notar que los campos que funcionan como llaves primarias están subrayados.

Para la clase *programa de verificación* se identifican sus campos como sigue:

Programas_Verificación : (Estado, Ciclo)	
Estado	Ciclo
Baja California Norte	PV00
Sonora	PV00

El programa de verificación se elabora de acuerdo a la importancia de la problemática relacionada con la operación del PROCAMPO en cada ciclo agrícola en el área de influencia de la Dirección Regional y de los recursos disponibles para su atención (personal técnico, vehículos, tiempo, presupuesto). Por lo que el programa de verificación se elaborará al inicio de cada ciclo agrícola en operación, el cual será sometido a la autorización de la Coordinación General de Apoyos Directos de ASERCA Central.

El campo Estado indica el estado donde se aplicará el programa de trabajo y el Ciclo indica el ciclo agrícola para el cual aplica dicho programa de trabajo.

Cada tupla de la tabla programas_verificación consiste de una o varias tuplas de la clase *metas* cuya tabla es:

ESTA TESIS NO SALE DE LA BIBLIOTECA

Metas :
(Meta, Metodología, DDR, CADER, Municipio, Ejido, Fecha, Inicio, Superficie, Volumen, Muestra, Superficie, Elegible, Muestra, Total, Predios, Superficie, Elegible, Superficie, Pagada)

El atributo Meta es el objetivo a cumplir en un programa de verificación, es decir, son las acciones que se pretenden llevar a cabo, la meta está conformada por:

Meta = 2560040001, donde:

25	Estado
6	Ciclo agrícola
00	Año agrícola
4	Metodología
0001	Consecutivo

El atributo Metodología se refiere a la aplicación del método a cubrir en un objetivo de trabajo, existen cuatro métodos:

1. **Preventiva:** consiste en la realización de recorridos de campo al interior de los conglomerados de un CADER, para constatar la generalización de siembras, previo a la apertura de las ventanillas para la reinscripción a PROCAMPO.
2. **Dingida:** se aplica a predios y/o expedientes cuando se tiene la sospecha de irregularidades, o se duda de la correcta operación del programa.
3. **General:** se aplica a predios y/o expedientes para conocer la situación general que guarda la operación de PROCAMPO dentro de un universo determinado.
4. **Específica:** se aplica a predios y/o expedientes, consiste en supervisar el cumplimiento de la Normatividad operativa para los casos donde es obligatoria la verificación por parte del CADER.

Los atributos de DDR, CADER, Municipio y Ejido se refieren a la cartografía donde se aplicará la verificación de predios, es decir, la estratificación de los predios.

Los atributos de Fecha, Inicio y Dias, Duración, se establecen para determinar cuándo se realizará la verificación y cuántos días durará.

Los atributos de Total, Predios, Superficie, Elegible, y Superficie, Pagada se refieren a los datos totales encontrados en la cartografía definida en la meta, ya que se tiene la alternativa de acuerdo a la metodología de trabajo asignada, determinar con base en los recursos disponibles si se verifica el 100% o se elige una muestra representativa de la población total.

Los atributos Volumen, Muestra y Superficie, Elegible, Muestra representan la muestra representativa que permitirá inferir en los resultados totales de la población a verificar.

Verificaciones_Predios : {Meta, Tipo_Movimiento}
 Verificaciones_Conglomerados : {Meta, Tipo_Movimiento}

De esta manera las definiciones de las especializaciones se registrarán como:

Por lo que podemos deducir que las clases verificación física, verificación documental y entrevista al productor son especializaciones de la clase verificación. O en sentido inverso verificación es una generalización de verificación física, verificación documental y entrevista al productor.

VERIFICACIONES_PREDIOS
VERIFICACIONES_CONGLOMERADOS

Verificación: {Tabla}

Las entidades involucradas en el control de los tipos de verificación aplicados a los predios y/o conglomerados a verificar registran los datos concluyentes sobre la aplicación del proceso de revisión, físicamente es una lista de las tablas tal como se muestra a continuación:

- 3) Entrevista a Productor, se realizará en el transcurso del ciclo agrícola.
- 2) Verificación Documental (de expedientes en CADER), se realizará en el transcurso de todo el ciclo agrícola.
- 1) Verificación Física (en campo), se realizará en las etapas de desarrollo vegetativo de los cultivos, comprendidas entre cultivo arralgado y madurez fisiológica (antes de la cosecha).

La clase verificación se representa por los tipos de verificación a aplicar en el proceso:

El establecimiento del tamaño de la muestra a verificar en cada una de las metas depende de la metodología a aplicar y del tipo de verificación a realizar. Departamento de Atención y los plazos para terminar los trabajos de verificación. Para la realización de las metas se elabora un calendario mensual de las actividades a realizar, en función del personal disponible, la distribución de las cargas de trabajo del

- 1) La metodología que conviene aplicar a cada uno de los problemas detectados.
- 2) Los recursos financieros y materiales, a los que llamaremos insumos.
- 3) Los parámetros obtenidos en la operación del programa en ciclos anteriores, como por ejemplo: promedio de predios y expedientes verificados por técnico/día, el rendimiento de gasolina/km de los vehículos, el monto ejercido en viáticos.
- 4) Consultar con el Departamento de Administración de la Dirección Regional, el presupuesto disponible.

Para la definición de las Metas a verificar en un programa de trabajo se deben considerar los siguientes aspectos:

- La creación de una tabla intermedia que establezca la relación.
 - A través de un trigger de base de datos asociado a las tablas.
- alternativas:
- 3) La cardinalidad muchos a muchos: impide el traslado de alguna de las llaves primarias a una de las tablas, por lo que para realizar físicamente esta relación se identifican dos alternativas:
 - 2) La cardinalidad uno a uno: entre dos clases puede tratarse de una sola clase percibida como dos, debido a diferentes puntos de vista.
 - 1) La cardinalidad muchos a uno: permite aplicar la regla de traslado de la llave primaria de la tabla de cardinalidad a uno a la tabla de cardinalidad a muchos.

Existen reglas para operar la cardinalidad de las relaciones entre tablas como son:

3.3.1.2 Reglas de Cardinalidad

Relación	Cardinalidad	Campos
Forma parte de: Metas → Programa de Verificación	1..n : 1	R: (Estado, Ciclo)
Eropa: Programa de Verificación → Presupuesto	1 : 1	R: (Estado, Ciclo)
Se incluyen en: Metodología → Metas	1..n : 1	R: (Metodología)
Establece: Verificación → Metas	1..n : 1	R: (Metas)
Registra bitácora en: Verificación → Movimientos	1 : 1..n	R: (Tipo Movimiento)
Verificación		

siguientes:

Las definiciones de los campos hasta este punto no involucran las relaciones existentes entre las clases. Del diagrama de clases de la figura 3.2.5 se obtienen las relaciones siguientes:

La clase *Movimientos Verificación* representa las diferentes etapas por la que pasa el proceso de verificación.

Verificaciones_Conglomerados:
 (Meta, Oficina, Comisión Administrativa, Tipo_Movimiento, Fecha_Movimiento, Documento, Exp_Completos, Exp_Incompletos, Exp_Revisados, Predios_Sembrados, Superficie_Sembrada)

Verificaciones_Predios:
 (Meta, Predio, Predio_Secundario, Tipo_Movimiento, Peticion, Fecha_Movimiento, Documento)

En la clase de Verificaciones_Predios se identifican los siguientes campos:
 Donde el atributo Meta corresponde físicamente a la designación de un objetivo de trabajo, y Tipo_Movimiento a la etapa en la que se encuentra el objetivo.

Si la creación de la tabla de relación es muy pequeña no justifica su creación por lo que será necesario la creación de un trigger que controle la aplicación de operaciones de la tabla.

3.3.1.3 Determinación del espacio de Almacenamiento

Para cada campo de las tablas resultantes se determina el tipo de datos y tamaño de los campos. En el caso del lenguaje de datos de Oracle, los tipos son `NUMBER`, `CHAR`, `VARCHAR`, `VARCHAR2` y `LONG`. Dependiendo de los valores posibles que cada campo puede tener se selecciona el tipo y tamaño. Si después de creada la tabla fuese necesario ampliar el tamaño del campo el manejador permite realizar la modificación.

En el código para crear la tabla `Programas_Verificación` se especifica el tamaño y tipo de datos, tal como se muestra a continuación:

```
CREATE TABLE PROGRAMAS (
    estado NUMBER(2) NOT NULL,
    CHAR(4) NOT NULL,
    CONSTRAINT PROGRAMAS_PK
    USING INDEX
    TABLESPACE &TABLESPACE_INDEX
    STORAGE (
        INITIAL 4K
        NEXT 4K
        PCTINCREASE 0
    )
);
```

Los tamaños y tipos de los campos se especifican en el código de definición de cada una de las tablas, de esta forma para la tabla de `Metas` se tiene lo siguiente:

```
CREATE TABLE METAS (
    meta NUMBER(10) NOT NULL,
    estado NUMBER(2) NOT NULL,
    oficina VARCHAR2(20) NULL,
    ciclo CHAR(4) NOT NULL,
    metodologia NUMBER(1) NOT NULL,
    oficio_cancelacion VARCHAR2(20) NULL,
    ddr NUMBER(4) NOT NULL,
    cader NUMBER(4) NOT NULL,
    municipio NUMBER(5) NULL,
    ofdo NUMBER(5) NOT NULL,
    total_predios NUMBER(5) NOT NULL,
    superficie_elegible NUMBER(12,2) NOT NULL,
    volumen_muestra NUMBER(5) NOT NULL,
    superficie_pagada NUMBER(12,2) NULL,
    superficie_elegible NUMBER(12,2) NOT NULL,
    fecha_inicio DATE NULL,
    duracion NUMBER(2) NULL,
    CONSTRAINT METAS_PK
    PRIMARY KEY (meta)
    USING INDEX
    TABLESPACE &TABLESPACE_INDEX
    STORAGE (
        INITIAL 4K
        NEXT 4K
    )
);
```

La determinación de los parámetros de almacenamiento (cláusula *storage* en la definición) de la tabla y el índice debe hacerse con base en el volumen de información esperado a almacenar en cada uno de los centros regionales.

```

)
PCTINCREASE 0
NEXT 100K
INITIAL 320K
STORAGE (
TABLESPACE &TABLESPACE_TABLA
REFERENCES PREDIOS
FOREIGN KEY (predio, predio_secuencial)
CONSTRAINT PREDIOS_VER_PRE_FK
REFERENCES METAS
FOREIGN KEY (meta)
CONSTRAINT METAS_VER_PRE_FK
REFERENCES TIPOS_MOVIMIENTOS
FOREIGN KEY (tipo_movimiento)
CONSTRAINT TIPOS_MOV_VER_FK
REFERENCES Documentos_Verificacion
FOREIGN KEY (documento)
CONSTRAINT ACTAS_VER_PRE_FK
)
NEXT 512K
INITIAL 1024K
STORAGE (
TABLESPACE &TABLESPACE_INDICE
USING INDEX
)
PRIMARY KEY (meta, predio, predio_secuencial,
tipo_movimiento)
CONSTRAINT VERIFICACIONES_PREDIOS_PK
)
DATE NOT NULL
fecha_movimiento
NUMBER(2) NOT NULL
tipo_movimiento
VARCHAR2(20) NOT NULL
documento
NUMBER(4) NOT NULL
predio_secuencial
NUMBER(12) NOT NULL
predio
NUMBER(10) NOT NULL
meta
)
CREATE TABLE VERIFICACIONES_PREDIOS (

```

El código de definición de la tabla de *Verificaciones_Predios* cambia en sus parámetros de almacenamiento de acuerdo al volumen de operaciones manejado en cada centro regional, pero básicamente está dada por el código siguiente:

```

)
PCTINCREASE 0
NEXT 4K
INITIAL 4K
STORAGE (
TABLESPACE &TABLESPACE_TABLA
REFERENCES METODOLOGIAS
FOREIGN KEY (metodologia)
CONSTRAINT METODOLOGIAS_METAS_FK
REFERENCES MOVIMIENTOS_PROGRAMAS
FOREIGN KEY (tipo_estado_oficio)
CONSTRAINT MOV_PROG_META_FK
REFERENCES Documentos_Verificacion
FOREIGN KEY (oficio_cancelacion)
CONSTRAINT DOC_VER_META_CAN_FK
)
PCTINCREASE 0

```

Para los parámetros de almacenamiento de la tabla se aplica la fórmula siguiente:

$$5 + Y * (1 + len)$$

$$\text{Número de Bloques} = X * \frac{(TB - 90) * (1 - PCTFREE / 100)}{5 + Y * (1 + len)}$$

$$\text{Donde: } X: \text{ Es el número de tuplas en la tabla. Este valor cambia de regional en regional.}$$

Y: Es el total de columnas de la tabla.

len: Es la longitud de columna promedio.

TB: Es el tamaño del bloque.

PCTFREE: Es el porcentaje de espacio extra reservado en caso de actualizaciones a los campos de una tupla de la tabla.

El valor de 5, en este caso se elige con base en que una actualización en el campo de tipo movimiento, fecha_movimiento, documento sería aproximadamente el uno por ciento del tamaño del registro.

Oracle organiza los datos almacenados en *extents*, cada uno de los cuales agrupa varios bloques en forma, cada grupo se identifica a través de un encabezado de control y el tamaño de cada grupo afecta el número de accesos de entrada/salida que debe hacer el manejador. Un número pequeño de *extents* con una definición de lectura de multibloque disminuye el número de accesos a disco. Para la creación de la tabla se debe definir el *extents* inicial en la cláusula INITIAL a un estenta por ciento del tamaño máximo esperado y el resto se deja al tamaño del siguiente extents de la cláusula NEXT.

Para la estimación de los parámetros de almacenamiento del índice se procede con los pasos siguientes:

1) Determinar el tamaño del encabezado del bloque.

2) Determinar el espacio disponible en cada bloque para los datos.

$$\text{Espacio para datos} = (\text{Tamaño del Bloque} - \text{Tamaño del Encabezado}) * PCTFREE / 100$$

3) Calcular el tamaño promedio de las columnas combinadas que forman el índice.

4) Calcular el tamaño promedio de cada tupla en el índice.

La tupla del índice está formada además de las columnas incluidas en el índice, por la dirección física que asigna Oracle, por byte de encabezado, almacena un byte de control por cada columna menor a 128 bytes y 3 bytes de control por cada columna mayor a 128 bytes.

$$\text{Tamaño Promedio} = \text{encabezado de la tupla} + \text{longitud de la dirección física} + \text{longitud de las columnas combinadas} + \text{número de columnas menor a 128 bytes} + 3 * (\text{número de columnas mayores a 128 bytes}).$$

Calcular el número de bloque y bytes ocupados para el índice con base en la fórmula siguiente:

Tuplas con columnas NO nulas

Bloques = 1,1

Redondeo(espcio disponible para datos / Tamaño promedio de una tupla de índice

El factor de 1,1 se debe a que se agrega un 10 por ciento de espacio adicional requerido para la creación de bloques extra generados por el índice.

Hasta este punto se ha determinado la estructura de base de datos básica para el esquema de Verificación de Predios, adicionalmente se debe definir para cada aplicación específica que transacciones se someten al esquema de Verificación de Predios y las reglas que han de seguirse para la actualización de los datos. Todas las reglas se especifican en *triggers* de base de datos. Un *trigger* es un procedimiento almacenado en la base de datos, el cual es ejecutado automáticamente como resultado de una sentencia DML: INSERT, UPDATE o DELETE, no importando si el usuario utiliza *sql*plus*, una aplicación en PRO*C, *sql*forms*, *sql*report*, etcétera. El uso del *trigger* permite simular métodos asociados a un objeto, en este caso por cada tabla se programan las reglas de operación respectivas y la modificación de los datos es sólo a través de las reglas ocultando con ello validaciones que por su complejidad no podrían ser implementadas con restricciones de base de datos relacional (referenciales y unicidad).

El *trigger* además de validar las reglas de operación, controla la generación de las operaciones.

3.3.1.4 Realización de reglas de operación en el RDBMS

Para cada tabla se determinan sus reglas de operación y se programa su *trigger* correspondiente, a continuación se desarrolla el *trigger* para las tablas del esquema de Verificación de Predios:

TABLA	
METODOS / OPERACIONES	
Programas_Verificación	registra (para un estado y ciclo específico) modifica (siempre y cuando todavía no exista registro de metas, presupuesto y/o cancela (siempre y cuando todavía no exista registro de metas, presupuesto y/o movimientos de verificación a nivel predio o conglomerado) a) registra (actualización del ciclo de autorización, fallo y tipo de movimiento) cancela (siempre y cuando todavía no exista registro de metas, presupuesto y/o movimientos de verificación a nivel predio o conglomerado)
Movimientos_Programas	a) registra (siempre y cuando no exista movimiento de verificación a nivel predio y/o conglomerado) b) modifica (siempre y cuando se registre el ciclo de cancelación de la meta y todavía no exista movimiento de verificación a nivel predio o conglomerado y que en movimiento de programa el ciclo de autorización sea nulo) registra (por tipo de verificación, partida y mes) modifica (siempre y cuando no exista movimiento en radicación o que no tenga referenciada una póliza) cancela (siempre y cuando no exista movimiento en radicación o que no tenga referenciada una póliza)
Presupuestos	a) registra (por estado, ciclo) modifica (siempre y cuando no este referenciada la póliza al presupuesto programado) cancela (siempre y cuando no este referenciada la póliza al presupuesto programado) Salidas
Oficios_Comisión_Advvo	a) registra (por salida y por técnico verificador) b) modifica (siempre y cuando no exista movimiento referenciados en salida, erogación, verificación por conglomerado y en documentos referenciados en salida, erogación, verificación por conglomerado y en documentos de verificación) a) registra (por meta y oficina de comisión advvo, asignado) modifica (actualización de la información inherente a la verificación, siempre y cuando no exista registro en resultados conglomerados) cancela (actualización de la información inherente a la verificación, siempre y cuando no exista registro en resultados conglomerados)
Verificaciones_Conglomerados	a) registra (a través de la aplicación del muestreo se obtienen los predios a verificar) b) modifica (actualización de la información inherente a la verificación, siempre y cuando no exista registro en información inherente a la verificación, siempre y cuando no exista registro en regularidades predios) c) cancela (cancela la información inherente a la verificación, siempre y cuando no exista registro en regularidades predios)
Verificaciones_Predios	a) registra (a través del documento y tipo) modifica (considerando la referencia del tipo de movimiento de verificación asignado) cancela (considerando la referencia del tipo de movimiento de verificación asignado)
Documentos_Verificación	a) registra (por salida, partida y mes) modifica (siempre y cuando no este referenciada a ninguna salida o oficio de comisión advvo.) cancela (siempre y cuando no este referenciada a ninguna salida o oficio de comisión advvo.)
Erogaciones	a) registra (para un estado y ciclo específico) modifica (siempre y cuando todavía no exista registro de metas, presupuesto y/o movimientos de verificación a nivel predio o conglomerado) cancela (siempre y cuando todavía no exista registro de metas, presupuesto y/o movimientos de verificación a nivel predio o conglomerado) a) registra (actualización del ciclo de autorización, fallo y tipo de movimiento) cancela (siempre y cuando todavía no exista registro de metas, presupuesto y/o movimientos de verificación a nivel predio o conglomerado)

Tabla 3.3.1 Estados que adquiere un predio en el proceso de verificación

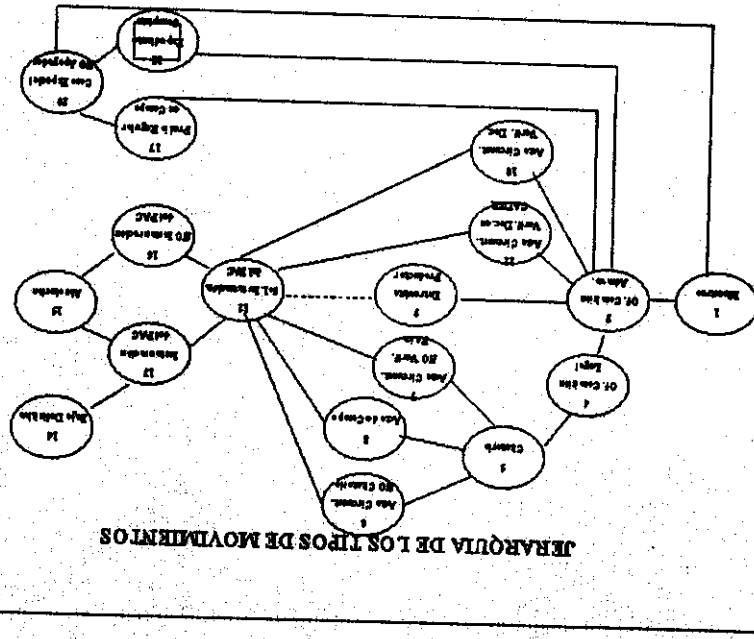
Tipo Documento de Verificación	Tipo Movimiento
	Oficina comisión admvo.
	3. Minuta dirigida conjoinerado
	4. Oficina de comisión legal
	5. Citatorio
	6. Acta circunstanciada por NO atender al citatorio
	7. Acta circunstanciada por NO permitir la verificación física del predio
	8. Acta de verificación en campo
	9. Entrevista al productor
	10. Acta circunstanciada de verificación documental
	11. Acta circunstanciada de verificación documental en CADER
	12. Sol. de Insuración del PAC
	13. Insuración del PAC
	14. Baja definitiva
	15. Absolución
	16. NO Insuración del PAC
	17. Insuración predio regular
	18. Expediente Completo
	19. Minuta Preventiva en CADER
	19. Caso Especial NO Apoyadas
	18. Expediente Completo
	18. Expediente Completo
	17. Insuración del PAC
	16. NO Insuración del PAC
	15. Absolución
	14. Baja definitiva
	13. Insuración del PAC
	12. Sol. de Insuración del PAC
	11. Acta circunstanciada de verificación documental en CADER
	10. Acta circunstanciada de verificación documental
	9. Entrevista al productor
	8. Acta de verificación en campo
	7. Acta circunstanciada por NO permitir la verificación física del predio
	6. Acta circunstanciada por NO atender al citatorio
	5. Citatorio
	4. Oficina de comisión legal
	3. Minuta dirigida conjoinerado
	Oficina comisión admvo.

El esquema de verificación de predios contiene diferentes etapas por las que deberá pasar un predio en el proceso de la verificación. Para modelar como pasa a cada etapa se determinan los posibles estados (movimientos de verificación) que adquiere, cada etapa se soporta con un documento legal que justifica la aplicación del proceso. Por cuestiones prácticas se asigna un número para cada estado como se muestra en la tabla 3.3.1.

Tipos Catálogos :	Tipos Documentos Ver	Tipos Resultados	Tipos Movimientos Ver	Tipos Verificación	Tipos Recursos	Metodologías
Regulaciones_Predios	a) registra (por meta y tipo de movimiento) modifica (actualización de la causa irregular detectada en la verificación) cancela (cancela la causa irregular detectada en la verificación)	a) registra (por meta y tipo de movimiento) modifica (actualización del resultado encontrado en la verificación) cancela (cancela el resultado encontrado en la verificación)	a) registra modifica cancela	Tipos_Catálogos Tipos_Recursos Tipos_Verificación Tipos_Movimientos_Ver Tipos_Regulaciones_Ver Tipos_Resultados Tipos_Documentos_Ver Motivos_Verificación Criterios_Verificación		

La validación de la transición de estados requiere de la programación de un trigger que se active ante cualquier operación de modificación de tuplas.

Figura 3.3.1 Diagrama de estados del proceso de verificación de predios



JERARQUÍA DE LOS TIPOS DE MOVIMIENTOS

El proceso de verificación de un predio, se muestra en forma sintetizada en la matriz de transición de datos mostrada en la figura 3.3.1.

3.3.2 Módulos

El proceso de verificación de predios está constituido por dos etapas principales: la primera considera el inicio del ciclo agrícola y la elaboración del diagnóstico y programa de trabajo para la verificación de predios, y la segunda por la aplicación y operación del programa de trabajo, ambas partes se presentan a continuación:

Nombre : Sistema de Verificación de Predios.

Implementación :
Datos : Los campos del manejador de operaciones están repartidos entre las tablas que forman el modelo de información o parte estática del objeto.

Métodos : Consultar operaciones.
Registrar operaciones.
Modificar operaciones.
Cancelar operaciones.

El esquema de control del proceso de verificación, considera los siguientes módulos:

- 1) Módulo Programático.
- 2) Módulo Administrativo – Presupuestal.
- 3) Módulo Operativo.

3.3.2.1.1 Integración de los módulos

La integración de los componentes que participan en el mecanismo de Verificación de Predios comprende los paquetes de PL-SQL en su forma de triggers y programas que sirven de interfaz con el usuario.

Los paquetes de PL-SQL se crean dentro de la base de datos y se asocian en forma de triggers a cada tabla que comparará información a través del esquema de verificación de predios. El comportamiento resultante es que, cada operación aplicada en la tabla es atrapada por el trigger correspondiente, en él se evalúa si la operación se sujeta a las reglas de operación, el programa local puede ser una sesión de SQL*PLUS, un programa ejecutable en C, una aplicación creada con SQL*FORMS, etcétera, ya que de manera interna el RDBMS envía el control al paquete de código de la base de datos. Una vez que el método correspondiente valida las reglas de operación y no existe error alguno, se inserta el movimiento, si el método recibe que se pudo insertar entonces afecta los datos.

Por otra parte a nivel operación se proporciona un conjunto de aplicaciones que permiten administrar la aplicación de los movimientos. Estas aplicaciones se presentan al operador a través del menú mostrado en la figura 3.3.2.

Para crear de manera automática las plantillas de las transacciones se consultan las vistas donde el RDBMS guarda la información de la definición de los campos de las tablas ALL_TAB_COLUMNS, ALL_IND_COLUMNS y ALL_CONSTRAINTS.

correspondiente.
 tabla y con la información del RDBMS de las columnas, tipo de datos y clave primaria programar el trigger para cada tabla. Las plantillas se crean con base en el nombre de la Sin embargo, se deben determinar las plantillas del conjunto de transacciones simplifica al proporcionar el espacio necesario para almacenar las transacciones diarias. Se transacciones y por los procesos que permiten aplicar dichas transacciones. El esquema de verificación de predios se conforma por las tablas que almacenan las simplifica posteriormente incluir otras tablas para compartir información. El ambiente del A nivel programación al establecer el ambiente del esquema de Verificación de Predios

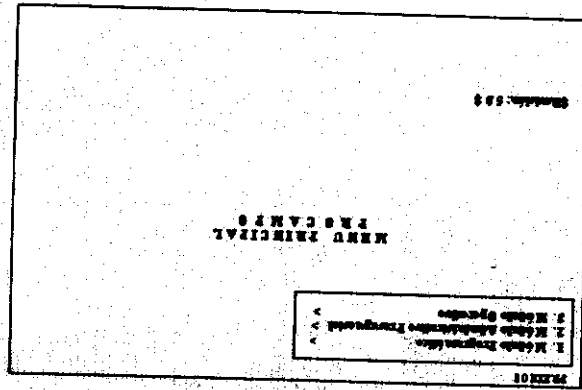
- Programación
- Operación

Las interacciones con el esquema de Verificación de Predios se llevan a dos niveles:

3.3.3 Interfaz

Cada módulo de operación del sistema considera diversas aplicaciones, dichas operaciones se presentan en el esquema siguiente de la figura 3.3.

Figura 3.2 Menú de aplicaciones para operación.



Una vez que se ha instalado el sistema para la operación del usuario, comienza la etapa del mantenimiento de las aplicaciones. Dicho mantenimiento deberá considerar corrección y/o adaptaciones menores debido a las necesidades que surjan.

El sistema de verificación de predios ha tenido correcciones del esquema debido a los cambios operativos de la normatividad del PROCAMPO, es decir, la operación de la verificación de predios se ha enlazado con diferentes procedimientos operativos, que han condicionado el proceso de la verificación para liberar dichos procesos. Esto no ha implicado problemas graves ya que el diseño del esquema ha permitido adaptar dichos cambios.

3.3.4 Mantenimiento

Figura 3.3.3 Esquema operativo de las aplicaciones

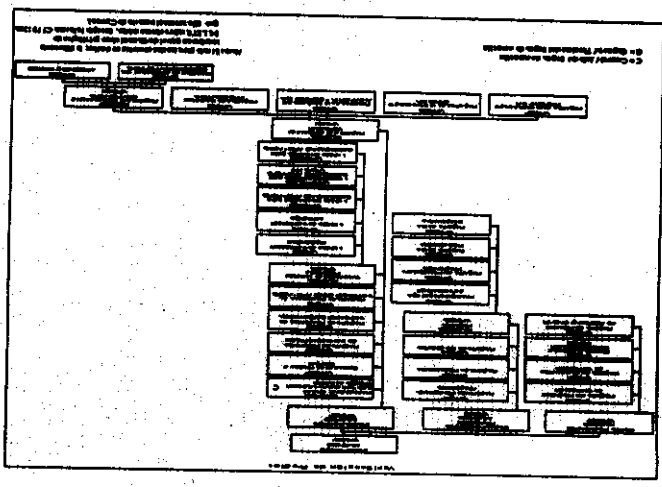


Tabla 3.3.2 Tipos de Mantenimiento

Tipo de Mantenimiento	Descripción
Preventivo	Se presenta cuando se realizan cambios para prevenir problemas que puedan suscitarse durante la operación del sistema.
Adaptativo	Se presenta cuando se agrega algún módulo u opción nueva al sistema.
Perfectivo	Consiste en la optimización del software, para dar mejor funcionalidad y respuesta.
Correctivo	Consiste en modificaciones a la programación con el fin de realizar cambios no previstos dentro del análisis del sistema.

Existen diferentes tipos de mantenimiento, los cuales se muestran en la tabla 3.3.2

CONCLUSIONES

4. Conclusiones

El objetivo de automatizar y homogeneizar las actividades de programación, consulta, almacenamiento y seguimiento de las verificaciones de predios, en las Direcciones Regionales de ASERCA fue alcanzado. Además se cumplió con las características planteadas inicialmente para el sistema, las cuales son:

- 1) Evitar distorsiones en la normatividad.
- 2) Evidenciar situaciones irregulares.
- 3) Cuantificar el grado de confiabilidad de las verificaciones establecidas como obligatorias.
- 4) Cuantificar el grado de cumplimiento de la normatividad operativa.
- 5) Contribuir a la depuración del directorio PROCAMPO.

Cada una de estas características están soportadas por los módulos que conforman el sistema.

El sistema cumplió con las expectativas de desempeño planteadas en un inicio donde se esperaba que el Departamento de Atención contara con un sistema que le permitiera llevar un control de los trabajos de verificación realizados en cada estado y ciclo del área de influencia de cada Dirección Regional, homogeneizando dichos trabajos para poder cuantificar el grado de cumplimiento de la Normatividad operativa del PROCAMPO cada ciclo agrícola. A nivel operativo ha redundado en la disminución de los tiempos de determinación del universo a verificar con base en la problemática planteada en cada área, así como en la generación de los reportes que les permite realizar dichos trabajos.

A nivel programación el sistema ha permitido controlar los aspectos dinámicos que operan en la determinación de la problemática planteada. De esta manera el esquema desarrollado ha soportado ligarse con procedimientos que sujetan su aprobación a los resultados obtenidos en la verificación.

4.1 Aplicaciones

El sistema de verificación de predios se ha creado flexible, lo cual permite ligarse con diferentes sistemas de bases de datos que requieran compartir información tales como el sistema de peticiones, el procedimiento de año y vez, etc.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Fundamentos de Base de Datos / Henry F. Korth . Abraham Silberschatz
2ª. Edición Ed. McGraw-Hill.

Ingeniería de Software / Richard Fairley
1ª. Edición Ed. McGraw-Hill.

Object Oriented Database Management / Alfons Kemper. Guido Moerkotte
1994. Ed. Prentice-Hall.

Análisis y Diseño Orientado a Objetos / Grady Booch
2ª. Edición Ed. Addison-Wesley.

Curso de Análisis de Sistemas de Información.
Softtek.

Una Panorámica de la Ingeniería de Software. Soluciones Avanzadas
Marzo 1994.

Acciones y Necesidades de la Función Informática de la Coordinación General
Operativa.
Documento Interno 1996.

ORACLE RDBMS Database Administrator's Guide.
1992.

ORACLE RDBMS Developer Release.
1992.

APÉNDICES

5. APÉNDICES

5.1 APÉNDICE A. Lenguaje Relacional Estándar

Las operaciones del álgebra relacional son muy importantes para comprender los tipos de peticiones que se pueden hacer a una base de datos relacional.

Por si mismas estas operaciones son generalmente clasificadas como un lenguaje de alto nivel, ya que se aplican a relaciones completas. El lenguaje más conocido del álgebra relacional es el SQL (Structured Query Language) y fue diseñado e implementado en IBM.

A.1 Definición de datos en SQL

SQL utiliza los términos tabla, renglón y columna para relación, tupla y atributo respectivamente. Los comandos para definir datos en este lenguaje son: CREATE TABLE, ALTER TABLE, y DROP TABLE los que a continuación se describirán:

- **CREATE TABLE:** Se usa para definir una nueva relación, dándole un nombre y definiendo sus atributos o columnas, cada columna recibe un nombre, el tipo de datos que recibirá, es decir, su dominio y algunas características de condición para integridad referencial necesarias. Los tipos de datos son generalmente numéricos y cadenas de caracteres. De los primeros se pueden definir enteros (Integer), de punto flotante (Float), con número decimal o no. De los segundos se pueden tener de longitud fija o variable. Existen también los tipos fecha (Date), Memo (Memo) para atributos más especiales.

SQL permite el uso de valores nulos (NULL), una condición no nula (NOT NULL) se puede especificar a los atributos sobre todo si éstos son llave primaria.

A continuación un ejemplo:

```
CREATE TABLE MOVIMIENTOS_PROGRAMAS (
  ciclo          CHAR(4) NOT NULL,
  estado         NUMBER(2) NOT NULL,
  oficio         VARCHAR2(20) NOT NULL,
  oficio_autorizacion VARCHAR2(20) NULL,
  tipo_movimiento CHAR(1) NOT NULL,
  fallo         VARCHAR2(4) NULL,
  CONSTRAINT MOVIMIENTOS_PROGRAMAS_PK
    PRIMARY KEY (ciclo, estado, oficio)
  USING INDEX
    TABLESPACE &TABLESPACE_INDICE
  STORAGE (
    INITIAL 4K
    NEXT 4K
    PCTINCREASE 0
  ),
  CONSTRAINT DOC_VER_PRO_MOV_AUT_FK
```

```
FOREIGN KEY (oficio_autorizacion)
REFERENCES Documentos_Verificacion,
CONSTRAINT DOC_VER_PRO_MOV_OFI_FK
FOREIGN KEY (oficio)
REFERENCES Documentos_Verificacion,
CONSTRAINT PROGRAMAS_MOV_FK
FOREIGN KEY (estado, ciclo)
REFERENCES PROGRAMAS
```

Las relaciones creadas con la sentencia CREATE TABLE se denominan tablas base en la terminología SQL, esto significa que la relación y sus tuplas se crearán y almacenarán en un archivo por el DBMS. Las relaciones base se distinguen de las relaciones virtuales, creadas con la sentencia CREATE VIEW que puede o no coincidir con el archivo físico. En el lenguaje SQL los atributos en una tabla base se consideran ordenados en la secuencia en la cual fueron especificados en la sentencia CREATE TABLE. Sin embargo los renglones o tuplas no necesariamente se consideran en algún orden.

- DROP TABLE: Si se decide eliminar una relación en una base de datos esto se logra con la sentencia DROP TABLE la cual borrará la definición de dicha relación.

Ejemplo:

```
DROP TABLE MOVIMIENTOS_PROGRAMAS CASCADE CONSTRAINTS;
```

- ALTER TABLE: Si se decide agregar un atributo a una de las tablas base, se utiliza la sentencia ALTER TABLE. Este nuevo atributo contendrá NULLS en todas las tuplas existentes de la relación, por lo que la condición NOT NULL no se podrá aplicar en este momento.

Ejemplo:

```
ALTER TABLE VERIFICACIONES_PREDIOS ADD PETICIÓN VARCHAR2(12);
```

A.2 Consultas en SQL

SQL tiene una sentencia básica para recuperar información de una base de datos, SELECT; hay una gran diferencia entre el SELECT el álgebra relacional y el SELECT de SQL, éste último permite a una tabla tener dos o más tuplas con valores idénticos en sus atributos, sin embargo en el álgebra relacional un conjunto de tuplas no debe tener elementos iguales. Por lo tanto una relación o tabla SQL no es un conjunto de tuplas a menos que tenga la característica de multiconjunto de tuplas. Algunas relaciones SQL se condicionan usando el comando CREATE UNIQUE INDEX o bien la sentencia DISTINCT acompañada del SELECT.

Consultas básicas de SQL

La forma básica de la sentencia SELECT es:

```
SELECT <Lista de atributos>
FROM <Lista de tablas>
WHERE <Condición>
```

Donde:

<Lista de atributos> Es una lista de nombres de atributos cuyos valores serán recuperados por la consulta.

<Lista de tablas> Es una lista de nombres de relación requeridos para proceder la consulta.

<Condición> Es una expresión condicional (booleana) que identifica las tuplas que serán recuperadas por la consulta.

Ejemplo:

Recupera las metas y su cartografía del estado 25 (Sinaloa), del ciclo 'PV00'.

```
SELECT meta, ddr, cader, municipio, ejido
FROM Metas
WHERE estado = 25
And ciclo = 'PV00';
```

A.2.1 Especificando nombres de atributos iguales

En SQL se puede usar el mismo nombre para dos o más atributos en diferentes relaciones, sólo hay que calificar el nombre del atributo con el nombre de la relación para prevenir la ambigüedad, de manera que éste quede prefijo.

Ejemplo:

```
SELECT m.nombre
FROM metodologia m, metas mt
WHERE m.metodología = mt.metodología
And mt.meta = '2580040002';
```

A.2.2 Tablas como conjuntos en SQL

Como se menciona anteriormente, SQL no trata una relación como un conjunto, es decir, las tuplas duplicadas pueden aparecer más de una vez en una relación o en el resultado de una consulta. SQL no elimina automáticamente estas tuplas. Para que se pueda realizar esto, se utiliza la palabra reservada DISTINCT, así sólo tuplas diferentes se mantendrán en el resultado de la consulta.

Ejemplo:

```
SELECT DISTINCT meta
FROM metas
WHERE estado = 25
And ciclo = 'PV00';
```

A.2.3 Consultas anidadas y comparación de conjuntos

Algunas consultas requieren de valores previamente consultados y usarlos en una condición. A este tipo de consultas se le denomina consultas anidadas que no son otra cosa que consultas completas dentro de la cláusula WHERE de otra consulta que se denomina consulta de salida.

Ejemplo:

```
SELECT DISTINCT m.meta
FROM metas m
WHERE m.meta IN (SELECT meta
FROM Irregularidades_predios ip
WHERE substr(ip.causa,1,1) = 'C'
And ip.meta = m.meta);
```

El operador de comparación IN compara un valor v con un conjunto o multiconjunto de valores V y evalúa con TRUE si v es uno de los elementos en V.

A.2.4 Función EXISTS

Esta función es usada para verificar si el resultado de una consulta contiene tuplas o no, es comúnmente utilizada en consultas anidadas, su función complemento es NOT EXISTS. El resultado de la función EXISTS es TRUE si al menos hay una tupla en el resultado y FALSE si no existe alguna. NOT EXISTS regresa TRUE si no hay tuplas en el resultado de la consulta y FALSE si al menos existe una.

Ejemplo:

```
SELECT DISTINCT m.meta
FROM metas m
WHERE NOT EXISTS (SELECT *
                  FROM Irregularidades_Predios ip
                  WHERE ip.meta = m.meta);
```

A.2.5 Conjuntos explícitos y valores nulos

Es posible en ocasiones usar un conjunto explícito de valores entre paréntesis en la cláusula WHERE en vez de consultas anidadas.

Ejemplo:

```
SELECT DISTINCT predio,predio_secuencial
FROM Verificaciones_Predios
WHERE tipo_movimiento IN (6,7,8);
```

SQL además permite verificar si un valor es nulo, está perdido, no definido o es no aplicable. Sin embargo en vez de usar los operadores = ó <> para comparar un atributo con nulo, SQL usa las palabras reservadas IS e IS NOT. Esto se debe a que SQL considera cada valor nulo distinto de otros valores, así como la comparación de igualdad no es apropiada. De aquí se desprende la idea de que cuando se especifica una operación JOIN, las tuplas con valores nulos no se incluirán en el resultado.

Ejemplo:

```
SELECT meta
FROM metas
WHERE ddr IS NULL;
```

A.2.6 Funciones agregadas y de grupo

Las funciones agregadas requeridas por la mayoría de las bases de datos son: COUNT, SUM, MAX, MIN y AVG. La función COUNT regresa el número de tuplas o valores especificados en una consulta. Las funciones SUM, MAX, MIN y AVG se aplican a multiconjuntos de valores numéricos y regresan la suma, el valor máximo, el valor mínimo y el promedio de dichos números respectivamente.

Ejemplo:

```
SELECT COUNT(*)
FROM verificaciones_predios
WHERE substr(meta,6,1) = '4';
```

En muchos casos se desea aplicar funciones agregadas a un subconjunto de tuplas, en una relación basada en algunos valores de atributos. Agrupar las tuplas que tienen el mismo valor de algún o algunos atributos llamado atributo(s) de grupo, y aplicar la función para cada grupo independientemente. SQL tiene la cláusula GROUP BY que especifica los atributos de grupo, los cuales deberán aparecer también en la cláusula SELECT.

Ejemplo:

```
SELECT DISTINCT status,tramite,COUNT(*)
FROM peticiones
GROUP BY status,tramite;
```

Además SQL provee otra palabra reservada HAVING para condicionar el grupo de tuplas asociadas con la orden GROUP BY.

Ejemplo:

```
SELECT DISTINCT status,tramite,COUNT(*)
FROM peticiones
GROUP BY status,tramite
HAVING COUNT(*)>1
```

Finalmente se tiene la cláusula ORDER BY que permite al usuario ordenar las tuplas, resultado de la consulta, con base en uno o más atributos con el fin de generar un reporte más comprensible.

Ejemplo:

```
SELECT DISTINCT tipo_movimiento
FROM verificaciones_predios
WHERE meta = '2560040001'
ORDER BY tipo_movimiento;
```

A.3 Declaraciones de actualización

En SQL hay tres comandos para modificar la base de datos, a continuación se describen cada uno de ellos.

A.3.1 Comando INSERT

En su forma más simple este comando se utiliza para adicionar una simple tupla a una relación, los valores deberán ser listados en el mismo orden en que se presentan los atributos de la relación cuando se crean con el comando CREATE.

Ejemplo:

```
INSERT INTO programas_verificación(estado,ciclo)
VALUES(26,'PV00');
```

Otra forma del comando, es dando los nombres de los atributos de manera específica, sobre aquellos que conocemos y sabemos que no se restringen por el parámetro NOT NULL, los atributos que quedan en blanco son fijados a NULL, y sus valores son listados en el mismo orden que los atributos de la relación.

Si en algún momento no se introduce el valor a un atributo con la condición NOT NULL, el manejador debe verificar la integridad referencial y evitar que la nueva tupla se integre a la relación con ese error.

Finalmente una variación del comando INSERT es permitir insertar múltiples tuplas dentro de una relación. Esto se usa principalmente en conjunción al crear una relación temporal y llenarla de datos con el resultado de una consulta.

Ejemplo:

```
INSERT INTO personal_paso
(SELECT * FROM personal_acerca);
```

A.3.2 Comando DELETE

Este comando elimina cero, una o varias tuplas de una relación. Incluye la cláusula WHERE para condicionar el borrado de las tuplas que satisfagan el parámetro. Las tuplas son eliminadas sólo de una tabla cada vez que el comando se aplica. Si no existe condición WHERE se entiende que se desean eliminar todas las tuplas de la relación dejando una tabla existente en la base de datos, pero sin ningún elemento.

Ejemplo:

```
DELETE verificaciones_predios
WHERE tipo_movimiento = 2
And meta = '2660020002';
```

A.3.3 Comando UPDATE

Se usa para modificar valores de atributos de una o más tuplas seleccionadas. Así como el comando DELETE, este también se realiza sobre una relación a la vez y se apoya de la condición WHERE para seleccionar los registros a actualizar. Además se utiliza la sentencia SET que especifica los atributos que serán modificados con sus nuevos valores.

Ejemplo:

```
UPDATE personal_acerca
SET nombre = 'CARLOS MONTAÑÉS VILLAFANE'
WHERE cve_puesto = 'CGAD';
```

A.4 Vistas SQL

A.4.1 Concepto de una vista

Una vista en la terminología SQL es una simple tabla que deriva de otras tablas que a su vez pudieron o no ser definidas previamente por otra vista. Una vista no necesariamente existe de manera física o literal como sucede con las relaciones definidas con el comando CREATE TABLE y se les denomina "*tablas virtuales*". Se puede pensar en una vista como una manera de especificar una tabla que requiere ser referenciada frecuentemente, pero que físicamente no existe, pues sus atributos forman parte de varias tablas.

A.4.2 Especificaciones de vistas

El comando para especificar una vista es el CREATE VIEW, dando como parámetros el nombre de una tabla, que recordemos será virtual, la lista de atributos y una consulta especificando el contenido de dicha vista.

Ejemplo:

```
CREATE VIEW verificaciones AS
(SELECT m.meta,p.*
FROM metas m, predios p);
```

Una vista se compone de la misma estructura que las relaciones pero de una manera dinámica y con posibilidades de acrecentar la velocidad de la presentación porque se realizan dentro de la memoria del sistema y dando opción a actualizarse de relaciones de las cuales se creó dicha vista cuando los datos fueron modificados.

Si ya no se desea conservar una vista, se dispone del comando DROP VIEW para eliminarla de memoria.

A.5 Índices

Un índice dentro del lenguaje SQL es una estructura que da acceso real o físico a uno o más atributos de una relación o tabla. Ya que una tabla representa en la mayoría de las veces un archivo, los índices se especifican en dichos archivos. El o los atributos para los cuales se crea un índice se denominan atributos indexados. Un índice hace más eficiente el acceso a los datos en las consultas, esto significa que se llevará menos tiempo consultar atributos que no lo están. En SQL los índices pueden ser creados y eliminados dinámicamente. Los comandos respectivos son CREATE INDEX y DROP INDEX.

Ejemplo:

```
CREATE INDEX DOC_VER_TIPO_IDX ON  
DOCUMENTOS_VERIFICACION(tipo)
```

5.2 APÉNDICE B. Trigger de la Tabla

VERIFICACIONES_PREDIOS

Código de creación del trigger para la tabla Verificaciones_Predios.

```

--
PROMPT "Creando o Reemplazando el Trigger siver.TVAL$ VERIFICACIONES_PREDIOS....."
PROMPT
create or replace trigger siver.TVAL$ VERIFICACIONES_PREDIOS
BEFORE DELETE or INSERT or UPDATE on VERIFICACIONES_PREDIOS

for each row
/* default body for TVAL$ VERIFICACIONES_PREDIOS */
DECLARE
-- $Revision: 5.6 $ $Author: vicram1 $ $Date: 2001/03/26 17:44:58 $
numrows INTEGER;
vg_ERROR CONSTANT NUMBER := 1;
v_error NUMBER := 0;
v_ciclo_meta VARCHAR2(4) := NULL;
l_predio NUMBER := 0;
l_secuencial NUMBER := 0;
BEGIN
IF DELETING THEN
/* VERIFICACIONES_PREDIOS Califica IRREGULARIDADES_PREDIOS ON PARENT DELETE
RESTRICT */
IF
pkg_irregularidades_predios.Existe_Irregularidad(:old.predio,:old.predio_secuencial,:old.peticion,:old.meta,:
old.tipo_movimiento) = vg_ERROR THEN
pkg_verificaciones_predios.Asigna(-20378);
raise_application_error(-20378,pkg_sisver.Error_Text(-20378));
END IF;

/* Valida el orden de borrado de los tipos de movimientos
IF
pkg_verificaciones_predios.Valida_Borra_Movimiento(:old.predio,:old.predio_secuencial,:old.peticion,:old.m
eta,:old.tipo_movimiento) = vg_ERROR THEN
v_error := pkg_verificaciones_predios.Obten_Error;
raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
END IF; SE COMENTA PARA AGREGAR TVAL_VERIFPREDIOS_AFTER */
pkg_verificaciones_predios.vg_Num_Renglon := pkg_verificaciones_predios.vg_Num_Renglon + 1;
pkg_verificaciones_predios.vg_Predios(pkg_verificaciones_predios.vg_Num_Renglon) := :old.predio;
pkg_verificaciones_predios.vg_Predios_Sec(pkg_verificaciones_predios.vg_Num_Renglon) :=
:old.predio_secuencial;
pkg_verificaciones_predios.vg_Peticions(pkg_verificaciones_predios.vg_Num_Renglon) := :old.peticion;
pkg_verificaciones_predios.vg_Metas(pkg_verificaciones_predios.vg_Num_Renglon) := :old.meta;
pkg_verificaciones_predios.vg_Tipos_Movto(pkg_verificaciones_predios.vg_Num_Renglon) :=
:old.tipo_movimiento;
IF :old.peticion IS NULL THEN
IF :old.predio IS NULL OR :old.predio_secuencial IS NULL THEN
pkg_verificaciones_predios.Asigna(-20582);
raise_application_error(-20582,pkg_sisver.Error_Text(-20582));
END IF;
END IF;
/* ELSE
IF pkg_verificaciones_predios.Valida_Status_Peticion(:old.peticion,:old.meta,:old.tipo_movimiento) =
vg_ERROR THEN

```

```

v_error := pkg_verificaciones_predios.Obten_Error;
raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
END IF;
END IF; */
IF :old.tipo_movimiento = 12 THEN
  IF pkg_verificaciones_predios.NoExiste_Solicitudes(:old.predio,
:old.predio_secuencial,pkg_metas.Obten_Ciclo(:old.meta)) = vg_ERROR THEN
    pkg_verificaciones_predios.Regresa_en_Espera(:old.predio,
:old.predio_secuencial,pkg_metas.Obten_Ciclo(:old.meta));
  END IF;
END IF;

ELSIF INSERTING THEN
SELECT DECODE(SUBSTR(:new.meta,3,1),6,'PV','OI')||SUBSTR(:new.meta,4,2)
  INTO v_ciclo_meta
  FROM sys.dual;
/* TIPOS_MOVIMIENTOS_VER Caracteriza VERIFICACIONES_PREDIOS ON CHILD INSERT
RESTRIC */
IF pkg_tipos_movimientos_ver.NoExiste_Tipo(:new.tipo_movimiento) = vg_ERROR THEN
  pkg_verificaciones_predios.Asigna(-20380);
  raise_application_error(-20380,pkg_sisver.Error_Text(-20380));
end if;

/* DOCUMENTOS_VERIFICACION Complementa a VERIFICACIONES_PREDIOS ON CHILD INSERT
RESTRIC */
IF :new.documento IS NOT NULL then
  IF pkg_verificaciones_predios.Valida_Fmt_Documento(:new.documento,
:new.tipo_movimiento,
:new.meta)
= vg_ERROR THEN
  v_error := pkg_verificaciones_predios.Obten_Error;
  raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
END IF;
IF pkg_documentos_verificacion.NoExiste_Documento(:new.documento) = vg_ERROR THEN
  pkg_verificaciones_predios.Asigna(-20379);
  raise_application_error(-20379,pkg_sisver.Error_Text(-20379));
end if;
ELSE
  IF pkg_tipos_movimientos_ver.Tipo_Requiere_Documento(:new.tipo_movimiento) = vg_ERROR THEN
    pkg_verificaciones_predios.Asigna(-20530);
    raise_application_error(-20530,pkg_sisver.Error_Text(-20530));
  END IF;
END IF;

/* METAS Genera VERIFICACIONES_PREDIOS ON CHILD INSERT RESTRIC */
IF pkg_metas.NoExiste_Meta(:new.meta) = vg_ERROR THEN
  pkg_verificaciones_predios.Asigna(-20381);
  raise_application_error(-20381,pkg_sisver.Error_Text(-20381));
end if;

/* PREDIOS Tener aplicado VERIFICACIONES_PREDIOS ON CHILD INSERT RESTRIC */
IF :new.predio IS NOT NULL AND :new.predio_secuencial IS NOT NULL THEN
  select count(*) into numrows
  from PREDIOS
  where
    :new.predio = PREDIOS.predio and
    :new.predio_secuencial = PREDIOS.predio_secuencial;
  and PREDIOS.status NOT LIKE 'B%';
  if ( numrows = 0) then
    pkg_verificaciones_predios.Asigna(-20382);

```

```

        raise_application_error(-20382, pkg_sisver.Error_Text(-20382));
    end if;
    IF :new.peticion IS NULL THEN
        /* Valida la cartografía de la meta sea igual a la del predio */
        IF pkg_metas.Compara_Cartografia(:new.meta,:new.predio,:new.predio_secuencial) = vg_ERROR
    THEN
        pkg_verificaciones_predios.Asigna(-20580);
        raise_application_error(-20580, pkg_sisver.Error_Text(-20580));
    END IF;
    ELSE
        IF pkg_verificaciones_predios.Valida_Status_Peticion(:new.peticion,:new.meta,:new.tipo_movimiento)
    = vg_ERROR THEN
        v_error := pkg_verificaciones_predios.Obten_Error;
        raise_application_error(v_error, pkg_sisver.Error_Text(v_error));
    END IF;
        IF pkg_metas.Compara_Cartografia(:new.meta,:new.peticion) = vg_ERROR THEN
            pkg_verificaciones_predios.Asigna(-20581);
            raise_application_error(-20581, pkg_sisver.Error_Text(-20581));
        END IF;
    END IF;
    ELSE
        IF :new.peticion IS NULL THEN
            pkg_verificaciones_predios.Asigna(-20583);
            raise_application_error(-20583, pkg_sisver.Error_Text(-20583));
        ELSE
            IF pkg_verificaciones_predios.Valida_Status_Peticion(:new.peticion,:new.meta,:new.tipo_movimiento)
        = vg_ERROR THEN
            v_error := pkg_verificaciones_predios.Obten_Error;
            raise_application_error(v_error, pkg_sisver.Error_Text(v_error));
        END IF;
            IF pkg_metas.Compara_Cartografia(:new.meta,:new.peticion) = vg_ERROR THEN
                pkg_verificaciones_predios.Asigna(-20581);
                raise_application_error(-20581, pkg_sisver.Error_Text(-20581));
            END IF;
        END IF;
        /* Valida que el documento no exista en DOCUMENTOS_VERIFICACION, y si llegara a existir se trate
    del mismo tipo de movimiento */
        IF :new.tipo_movimiento NOT IN (1,9,17,18,21,22) THEN
            IF pkg_verificaciones_predios.Valida_Tipo_Oficio(:new.documento,
                :new.tipo_movimiento) = vg_ERROR THEN
                v_error := pkg_verificaciones_predios.Obten_Error;
                raise_application_error(v_error, pkg_sisver.Error_Text(v_error));
            END IF;
        END IF;

        /* Valida que permita duplicar un documento en VERIFICACIONES_PREDIOS para
    los tipos de movimiento 2,11,12,13,14,15,16 y 20 */
        IF :new.tipo_movimiento in (2,11,12,20,13,14,15,16) THEN
            if pkg_verificaciones_predios.Valida_Existe_Documento(:new.documento,:new.tipo_movimiento) =
        vg_ERROR then
                v_error := pkg_verificaciones_predios.Obten_Error;
                raise_application_error(v_error, pkg_sisver.Error_Text(v_error));
            end if;
        ELSE
            if pkg_verificaciones_predios.Existe_Documento(:new.documento) = vg_ERROR then
                v_error := pkg_verificaciones_predios.Obten_Error;
                raise_application_error(v_error, pkg_sisver.Error_Text(v_error));
            end if;
        END IF;
    END IF;

```



```

/* Valida el orden de aparición de los tipos de movimientos */
IF
pkg_verificaciones_predios.Valida_Orden_Movimiento(:new.predio,:new.predio_secuencial,:new.peticion,:n
ew.meta,:new.tipo_movimiento) = vg_ERROR THEN
  v_error := pkg_verificaciones_predios.Obten_Error;
  raise_application_error( v_error,pkg_sisver.Error_Text(v_error));
END IF;
/* Fecha: 280199 Valida que el nuevo movimiento del predio este permitido registrarlo */
IF pkg_verificaciones_predios.Valida_Movimiento_Permitido(:new.predio,
:new.predio_secuencial,:new.peticion,:new.meta,:new.tipo_movimiento) = vg_ERROR THEN
  v_error := pkg_verificaciones_predios.Obten_Error;
  raise_application_error( v_error,pkg_sisver.Error_Text(v_error));
END IF;
/* IF :new.tipo_movimiento IN (6,7,8,10,11) THEN
  IF pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,
:new.predio_secuencial,:new.peticion,:new.meta,:new.tipo_movimiento) != vg_ERROR THEN
    pkg_verificaciones_predios.Asigna(-20574);
    raise_application_error( -20574,pkg_sisver.Error_Text(-20574));
  END IF;
END IF; */

/* Fecha: 101000 Valida que sea regular para quitar en_espera PRV
IF
pkg_verificaciones_predios.Predio_en_Espera(:new.predio,:new.predio_secuencial,v_ciclo_meta,'PRV') =
vg_ERROR THEN
  IF :new.tipo_movimiento = 18 THEN
    IF
pkg_verificaciones_predios.Obten_Status_Verificacion(:new.predio,:new.predio_secuencial,v_ciclo_meta,2)
= 'R' THEN
      pkg_verificaciones_predios.vg_Num_Renglon := pkg_verificaciones_predios.vg_Num_Renglon + 1;
      pkg_verificaciones_predios.vg_Predios(pkg_verificaciones_predios.vg_Num_Renglon) :=
:old.predio;
      pkg_verificaciones_predios.vg_Predios_Sec(pkg_verificaciones_predios.vg_Num_Renglon) :=
:old.predio_secuencial;
      pkg_verificaciones_predios.vg_Ciclos(pkg_verificaciones_predios.vg_Num_Renglon) :=
v_ciclo_meta;
      pkg_verificaciones_predios.vg_Esperas(pkg_verificaciones_predios.vg_Num_Renglon) := 'PRV';
    END IF;
  ELSIF :new.tipo_movimiento = 17 THEN
    IF
pkg_verificaciones_predios.Obten_Status_Verificacion(:new.predio,:new.predio_secuencial,v_ciclo_meta,1)
= 'R' THEN
      pkg_verificaciones_predios.vg_Num_Renglon := pkg_verificaciones_predios.vg_Num_Renglon + 1;
      pkg_verificaciones_predios.vg_Predios(pkg_verificaciones_predios.vg_Num_Renglon) :=
:old.predio;
      pkg_verificaciones_predios.vg_Predios_Sec(pkg_verificaciones_predios.vg_Num_Renglon) :=
:old.predio_secuencial;
      pkg_verificaciones_predios.vg_Ciclos(pkg_verificaciones_predios.vg_Num_Renglon) :=
v_ciclo_meta;
      pkg_verificaciones_predios.vg_Esperas(pkg_verificaciones_predios.vg_Num_Renglon) := 'PRV';
    END IF;
  END IF;
END IF; */

/* Fecha: 100299 Actualiza campo en_espera de solicitudes al momento
de registrar mov. 12 SOL. DE INSTAURACION DEL PAC */
IF :new.tipo_movimiento = 12 THEN

```

```

IF
pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,6) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,7) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,8) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,9) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,10) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,11) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,22) != vg_ERROR THEN
    pkg_verificaciones_predios.Asigna(-20574);
    raise_application_error(-20574,pkg_sisver.Error_Text(-20574));
END IF;
IF :new.peticion IS NOT NULL THEN
    IF pkg_verificaciones_predios.Obten_Tramite(:new.peticion) IN ('FPN','FPO','FPF') THEN
        pkg_verificaciones_predios.Obten_Predio_Referencia(:new.peticion,l_predio,l_secuencial);
        IF pkg_verificaciones_predios.NoExisten_Solicitudes(l_predio,
            l_secuencial,pkg_metras.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
            pkg_verificaciones_predios.Actualiza_en_Espera(l_predio,
                l_secuencial,pkg_metras.Obten_Ciclo(:new.meta),
                :new.documento);
        END IF;
    ELSE
        IF :new.predio IS NULL OR :new.predio_secuencial IS NULL THEN
            pkg_verificaciones_predios.Asigna(-20582);
            raise_application_error(-20582,pkg_sisver.Error_Text(-20582));
        END IF;
        IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
            :new.predio_secuencial,pkg_metras.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
            pkg_verificaciones_predios.Actualiza_en_Espera(:new.predio,
                :new.predio_secuencial,pkg_metras.Obten_Ciclo(:new.meta),
                :new.documento);
        END IF;
    END IF;
ELSE
    IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
        :new.predio_secuencial,pkg_metras.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
        pkg_verificaciones_predios.Actualiza_en_Espera(:new.predio,
            :new.predio_secuencial,pkg_metras.Obten_Ciclo(:new.meta),
            :new.documento);
    END IF;
END IF;
ELSE
    /* Fecha: 020399 Validación de movimientos posteriores a la solicitud
    de instauración del PAC */
    IF :new.tipo_movimiento = 15 THEN /* ABSOLUCION DE BAJA */
        IF :new.peticion IS NOT NULL THEN
            IF pkg_verificaciones_predios.Obten_Tramite(:new.peticion) IN ('FPN','FPO','FPF') THEN
                pkg_verificaciones_predios.Obten_Predio_Referencia(:new.peticion,l_predio,l_secuencial);
            END IF;
        END IF;
    END IF;

```

```

IF pkg_verificaciones_predios.NoExisten_Solicitudes(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
IF
pkg_verificaciones_predios.NoExiste_Movimiento(l_predio,l_secuencial,:new.peticion,:new.meta,13) !=
vg_ERROR THEN
pkg_verificaciones_predios.Regresa_en_Espera(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta));
END IF;
END IF;
ELSE
IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
IF
pkg_verificaciones_predios.NoExiste_Movimiento(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,13) != vg_ERROR THEN
pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta));
END IF;
END IF;
END IF;
ELSE
IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
IF
pkg_verificaciones_predios.NoExiste_Movimiento(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,13) != vg_ERROR THEN
pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta));
END IF;
END IF;
END IF;
ELSE
IF :new.tipo_movimiento = 16 THEN /* NO INST. DEL PAC */
IF :new.peticion IS NOT NULL THEN
IF pkg_verificaciones_predios.Obten_Tramite(:new.peticion) IN ('FPN','FPO','FPF') THEN
pkg_verificaciones_predios.Obten_Predio_Referencia(:new.peticion,l_predio,l_secuencial);
IF pkg_verificaciones_predios.NoExisten_Solicitudes(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
pkg_verificaciones_predios.Regresa_en_Espera(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta));
END IF;
ELSE
IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta));
END IF;
END IF;
ELSE
IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta));
END IF;
END IF;
END IF;
END IF;
ELSIF UPDATING THEN

```

```

SELECT DECODE(SUBSTR(:new.meta,3,1),6,'PV','O')||SUBSTR(:new.meta,4,2)
INTO v_ciclo_meta
FROM sys.dual;
/* TIPOS_MOVIMIENTOS_VER Caracteriza VERIFICACIONES_PREDIOS ON CHILD UPDATE
RESTRICT */
IF pkg_tipos_movimientos_ver.NoExiste_Tipo(:new.tipo_movimiento) = vg_ERROR THEN
    pkg_verificaciones_predios.Asigna(-20384);
    raise_application_error(-20384,pkg_sisver.Error_Text(-20384));
end if;

if NVL(:old.documento,-1) <> NVL(:new.documento,-1) then
    /* Valida el formato segun el documento y tipo movto */
    IF
pkg_verificaciones_predios.Valida_Fmt_Documento(:new.documento,:new.tipo_movimiento,:new.meta) =
vg_ERROR THEN
        v_error := pkg_verificaciones_predios.Obten_Error;
        raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
    END IF;
/* DOCUMENTOS_VERIFICACION Complementa a VERIFICACIONES_PREDIOS ON CHILD UPDATE
RESTRICT */
IF pkg_documentos_verificacion.NoExiste_Documento(:new.documento) = vg_ERROR THEN
    pkg_verificaciones_predios.Asigna(-20383);
    raise_application_error(-20383,pkg_sisver.Error_Text(-20383));
end if;

end if;

IF :new.documento IS NULL THEN
    IF pkg_tipos_movimientos_ver.Tipo_Requiere_Documento(:new.Tipo_Movimiento) = vg_ERROR THEN
        pkg_verificaciones_predios.Asigna(-20530);
        raise_application_error(-20530,pkg_sisver.Error_Text(-20530));
    END IF;
END IF;

/* METAS Genera VERIFICACIONES_PREDIOS ON CHILD UPDATE RESTRICT */
if :old.meta <> :new.meta then
    IF pkg_metas.NoExiste_Meta(:new.meta) = vg_ERROR THEN
        pkg_verificaciones_predios.Asigna(-20433);
        raise_application_error(-20433,pkg_sisver.Error_Text(-20433));
    END IF;
end if;

/* PREDIOS Tener aplicado VERIFICACIONES_PREDIOS ON CHILD UPDATE RESTRICT */
IF nvl(:old.predio,-1) <> :new.predio OR
nvl(:old.predio_secuencial,-1) <> :new.predio_secuencial OR
nvl(:old.peticion,-1) <> :new.peticion THEN
    IF :new.predio IS NOT NULL AND :new.predio_secuencial IS NOT NULL THEN
        select count(*) into numrows
        from PREDIOS
        where
            :new.predio = PREDIOS.predio and
            :new.predio_secuencial = PREDIOS.predio_secuencial and
            PREDIOS.status NOT LIKE 'B%';
        if ( numrows = 0) then
            pkg_verificaciones_predios.Asigna(-20386);
            raise_application_error(-20386,pkg_sisver.Error_Text(-20386));
        end if;
        /* Valida la cartografia de la meta sea igual a la del predio */
        IF :new.peticion IS NULL THEN

```

```

IF pkg_metas.Compara_Cartografia(:new.meta,:new.predio,:new.predio_secuencial) = vg_ERROR
THEN
  pkg_verificaciones_predios.Asigna(-20580);
  raise_application_error(-20580,pkg_sisver.Error_Text(-20580));
END IF;
ELSE
  IF
pkg_verificaciones_predios.Valida_Status_Peticion(:new.peticion,:new.meta,:new.tipo_movimiento) =
vg_ERROR THEN
    v_error := pkg_verificaciones_predios.Obten_Error;
    raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
    END IF;
    IF pkg_metas.Compara_Cartografia(:new.meta,:new.peticion) = vg_ERROR THEN
      pkg_verificaciones_predios.Asigna(-20581);
      raise_application_error(-20581,pkg_sisver.Error_Text(-20581));
    END IF;
    END IF;
  ELSE
    IF :new.peticion IS NULL THEN
      pkg_verificaciones_predios.Asigna(-20583);
      raise_application_error(-20583,pkg_sisver.Error_Text(-20583));
    ELSE
      IF pkg_verificaciones_predios.Valida_Status_Peticion(:new.peticion,:new.meta,:new.tipo_movimiento)
=vg_ERROR THEN
        v_error := pkg_verificaciones_predios.Obten_Error;
        raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
        END IF;
        IF pkg_metas.Compara_Cartografia(:new.meta,:new.peticion) = vg_ERROR THEN
          pkg_verificaciones_predios.Asigna(-20581);
          raise_application_error(-20581,pkg_sisver.Error_Text(-20581));
        END IF;
        END IF;
      END IF;
    END IF;
  /* VERIFICACIONES_PREDIOS Califica IRREGULARIDADES_PREDIOS ON PARENT UPDATE
  RESTRICT */
  if :old.meta <> :new.meta or
:old.predio <> :new.predio or
:old.predio_secuencial <> :new.predio_secuencial or
:old.peticion <> :new.peticion or
:old.tipo_movimiento <> :new.tipo_movimiento then

  IF
pkg_irregularidades_predios.Existe_Irregularidad(:old.predio,:old.predio_secuencial,:old.peticion,:old.meta,:
old.tipo_movimiento) = vg_ERROR THEN
  pkg_verificaciones_predios.Asigna(-20387);
  raise_application_error(-20387,pkg_sisver.Error_Text(-20387));
  END IF;
end if;

/* Valida que permita duplicar documento para los tipos de movimiento 2,11,12, y 20 */
IF :old.tipo_movimiento <> :new.tipo_movimiento THEN
  if :new.tipo_movimiento in (2,11,12,20,13,14,15,16,18) THEN
    if pkg_verificaciones_predios.Valida_Existencia_Documento(:new.documento,:new.tipo_movimiento) =
vg_ERROR then
      v_error := pkg_verificaciones_predios.Obten_Error;
      raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
    end if;
  else
    if pkg_verificaciones_predios.Existe_Documento(:new.documento) = vg_ERROR then

```

```
v_error := pkg_verificaciones_predios.Obten_Error;
raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
end if;
end if;

/* Valida el orden de aparicion de los tipos de movimientos */
IF pkg_verificaciones_predios.Valida_Orden_Movimiento(:new.predio,
:new.predio_secuencial,:new.peticion,:new.meta,:new.tipo_movimiento) = vg_ERROR THEN
v_error := pkg_verificaciones_predios.Obten_Error;
raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
END IF;

/* Fecha: 010299 Valida que el nuevo movimiento del predio este permitido registrarlo */
IF
pkg_verificaciones_predios.Valida_Movimiento_Permitido(:new.predio,:new.predio_secuencial,:new.peticion
,:new.meta,:new.tipo_movimiento) = vg_ERROR THEN
v_error := pkg_verificaciones_predios.Obten_Error;
raise_application_error(v_error,pkg_sisver.Error_Text(v_error));
END IF;
/* IF :new.tipo_movimiento IN (6,7,8,10,11) THEN
IF
pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,:new.tipo_movimiento) != vg_ERROR THEN
pkg_verificaciones_predios.Asigna(-20574);
raise_application_error(-20574,pkg_sisver.Error_Text(-20574));
END IF;
END IF; */

/* Fecha: 101000 Valida que sea regular para quitar en_espera PRV
IF
pkg_verificaciones_predios.Predio_en_Espera(:new.predio,:new.predio_secuencial,v_ciclo_meta,'PRV') =
vg_ERROR THEN
IF :new.tipo_movimiento = 18 THEN
IF
pkg_verificaciones_predios.Obten_Status_Verificacion(:new.predio,:new.predio_secuencial,v_ciclo_meta,2)
= 'R' THEN

pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,:new.predio_secuencial,v_ciclo_meta,'PRV');
END IF;
ELSIF :new.tipo_movimiento = 17 THEN
IF
pkg_verificaciones_predios.Obten_Status_Verificacion(:new.predio,:new.predio_secuencial,v_ciclo_meta,1)
= 'R' THEN

pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,:new.predio_secuencial,v_ciclo_meta,'PRV');
END IF;
END IF;
END IF; */

/* Fecha: 120299 Actualiza campo en_espera de solicitudes_oid al momento
de registrar mov. 12 SOL. DE INSTAURACION DEL PAC */
IF :new.tipo_movimiento = 12 THEN
IF
pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,6) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,7) != vg_ERROR AND
```

```

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,8) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,9) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,10) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,11) != vg_ERROR AND

pkg_irregularidades_predios.Existe_Irregularidad(:new.predio,:new.predio_secuencial,:new.peticion,:new.m
eta,22) != vg_ERROR THEN
  pkg_verificaciones_predios.Asigna(-20574);
  raise_application_error(-20574,pkg_sisver.Error_Text(-20574));
END IF;
IF :new.peticion IS NOT NULL THEN
  IF pkg_verificaciones_predios.Obten_Tramite(:new.peticion) IN ('FPN','FPO','FPF') THEN
    pkg_verificaciones_predios.Obten_Predio_Referencia(:new.peticion,l_predio,l_secuencial);
    IF pkg_verificaciones_predios.NoExisten_Solicitudes(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
      pkg_verificaciones_predios.Actualiza_en_Espera(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta),
:new.documento);
    END IF;
  ELSE
    IF :new.predio IS NULL OR :new.predio_secuencial IS NULL THEN
      pkg_verificaciones_predios.Asigna(-20582);
      raise_application_error(-20582,pkg_sisver.Error_Text(-20582));
    END IF;
    IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
      pkg_verificaciones_predios.Actualiza_en_Espera(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta),
:new.documento);
    END IF;
  END IF;
ELSE
  IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
    pkg_verificaciones_predios.Actualiza_en_Espera(:new.predio,
:new.predio_secuencial,pkg_metas.Obten_Ciclo(:new.meta),
:new.documento);
  END IF;
END IF;
ELSE
  /* Fecha: 020399 Validación de movimientos posteriores a la solicitud
de instauración del PAC */
  IF :new.tipo_movimiento = 15 THEN /* ABSOLUCION DE BAJA */
    IF :new.peticion IS NOT NULL THEN
      IF pkg_verificaciones_predios.Obten_Tramite(:new.peticion) IN ('FPN','FPO','FPF') THEN
        pkg_verificaciones_predios.Obten_Predio_Referencia(:new.peticion,l_predio,l_secuencial);
        IF pkg_verificaciones_predios.NoExisten_Solicitudes(l_predio,
l_secuencial,pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
          IF
pkg_verificaciones_predios.NoExiste_Movimiento(l_predio,l_secuencial,:new.peticion,:new.meta,13)
vg_ERROR THEN
            pkg_verificaciones_predios.Regresa_en_Espera(l_predio,

```

```

        l_secuencial, pkg_metas.Obten_Ciclo(:new.meta));
    END IF;
  END IF;
  ELSE
    IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
      :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
      IF
        pkg_verificaciones_predios.NoExiste_Movimiento(:new.predio, :new.predio_secuencial, :new.peticion, :new.m
          eta, 13) != vg_ERROR THEN
          pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
            :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta));
          END IF;
        END IF;
      END IF;
    ELSE
      IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
        :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
        IF
          pkg_verificaciones_predios.NoExiste_Movimiento(:new.predio, :new.predio_secuencial, :new.peticion, :new.m
            eta, 13) != vg_ERROR THEN
            pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
              :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta));
            END IF;
          END IF;
        END IF;
      ELSE
        IF :new.tipo_movimiento = 16 THEN /* NO INST. DEL PAC */
          IF :new.peticion IS NOT NULL THEN
            IF pkg_verificaciones_predios.Obten_Tramite(:new.peticion) IN ('FPN', 'FPO', 'FPF') THEN
              pkg_verificaciones_predios.Obten_Predio_Referencia(:new.peticion, l_predio, l_secuencial);
              IF pkg_verificaciones_predios.NoExisten_Solicitudes(l_predio,
                l_secuencial, pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
                pkg_verificaciones_predios.Regresa_en_Espera(l_predio,
                  l_secuencial, pkg_metas.Obten_Ciclo(:new.meta));
                END IF;
              ELSE
                IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
                  :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
                  pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
                    :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta));
                  END IF;
                END IF;
              ELSE
                IF pkg_verificaciones_predios.NoExisten_Solicitudes(:new.predio,
                  :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta)) = vg_ERROR THEN
                  pkg_verificaciones_predios.Regresa_en_Espera(:new.predio,
                    :new.predio_secuencial, pkg_metas.Obten_Ciclo(:new.meta));
                  END IF;
                END IF;
              END IF;
            END IF;
          END IF;
        END IF;
      END IF;
    END;
  /

```


5.3 APÉNDICE C. Trigger de la Tabla DOCUMENTOS_VERIFICACION

Código de creación del trigger para la tabla Documentos_Verificacion.

```
--apool vp_trg_regional.lst

PROMPT "Creando o Reemplazando el Trigger sisver.TVALS_DOCUMENTOS_VERIFICACION....."
PROMPT

create or replace trigger sisver.TVALS_DOCUMENTOS_VERIFICACION
BEFORE DELETE or INSERT or UPDATE on DOCUMENTOS_VERIFICACION

for each row
/* default body for TVALS_DOCUMENTOS_VERIFICACION */
DECLARE
-- $Revision: 5.1 $ $Author: ruboca1 $ $Date: 2001/01/18 16:15:04 $
numrows INTEGER;
vg_ERROR CONSTANT NUMBER := 1;
vg_EXITO CONSTANT NUMBER := 0;
v_error NUMBER := 0;
BEGIN
IF DELETING THEN
/* NOTA.- El oficio de radicacion no se valida */
/* Busca documentos en sus tablas correspondientes */
IF :old.Tipo = 32 THEN
/* DOCUMENTOS_VERIFICACION Describe a METAS ON PARENT DELETE RESTRICT */
IF pkg_metas.Existe_Oficio_Cancelacion(:old.documento) = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20314);
raise_application_error(-20314,pkg_sisver.Error_Text(-20314));
END IF;
ELSE
IF :old.Tipo IN (30,31,34) THEN
/* DOCUMENTOS_VERIFICACION Describe a MOVIMIENTOS_PROGRAMAS ON PARENT
DELETE RESTRICT */
IF pkg_movimientos_programas.Existe_Autorizacion(:old.documento) = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20315);
raise_application_error(-20315,pkg_sisver.Error_Text(-20315));
END IF;
/* DOCUMENTOS_VERIFICACION Describe a MOVIMIENTOS_PROGRAMAS ON PARENT
DELETE RESTRICT */
IF pkg_movimientos_programas.Existe_Oficio(:old.documento) = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20316);
raise_application_error(-20316,pkg_sisver.Error_Text(-20316));
END IF;
ELSE
IF :old.Tipo IN (2) THEN
/* DOCUMENTOS_VERIFICACION Describe OFICIOS_COMISION_ADMVO ON PARENT
DELETE RESTRICT */
IF pkg_oficios_comision_admvo.Existe_Oficio(:old.documento) = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20317);
raise_application_error(-20317,pkg_sisver.Error_Text(-20317));
END IF;
/* DOCUMENTOS_VERIFICACION Complementa a VERIFICACIONES_PREDIOS ON PARENT
DELETE RESTRICT */
IF pkg_verificaciones_predios.Existe_Documento(:old.documento) = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20318);
```

```

        raise_application_error(-20318, pkg_sisver.Error_Text(-20318));
    END IF;
    ELSE
        IF :old.Tipo IN (5,4,8,6,7,13,16,14,9,10,11,12,20,22) THEN
            /* DOCUMENTOS_VERIFICACION Complementa a VERIFICACIONES_PREDIOS ON PARENT
            DELETE RESTRICT */
            IF pkg_verificaciones_predios.Existe_Documento(:old.documento) = vg_ERROR THEN
                pkg_documentos_verificacion.Asigna(-20318);
                raise_application_error(-20318, pkg_sisver.Error_Text(-20318));
            END IF;
        END IF;
        IF :old.Tipo IN (11,19,18,3) THEN
            /* DOCUMENTOS_VERIFICACION Contenedora en VERIFICACIONES_CONGLOMERADOS ON
            PARENT DELETE RESTRICT */
            IF pkg_verificacion_conglomerado.Existe_Documento(:old.documento) = vg_ERROR THEN
                pkg_documentos_verificacion.Asigna(-20420);
                raise_application_error(-20420, pkg_sisver.Error_Text(-20420));
            END IF;
        END IF;
    END IF;
    END IF;
    END IF;
    END IF;

    ELSIF INSERTING THEN

        /* TIPOS_DOCUMENTOS_VER Clasifica a DOCUMENTOS_VERIFICACION ON CHILD INSERT
        RESTRICT */
        IF pkg_tipos_documentos_ver.NoExiste_Tipo(:new.tipo) = vg_ERROR THEN
            pkg_documentos_verificacion.Asigna(-20444);
            raise_application_error(-20444, pkg_sisver.Error_Text(-20444));
        END IF;

        /* Valida que un formato corresponda con su tipo, antes de llamar a la
        funcion debe ejecutar Asigna_Ciclo(parametro ciclo) */
        IF
            pkg_documentos_verificacion.Valida_Fmt_Documento(:new.documento, :new.tipo, pkg_documentos_verifica
            clon.Obten_Ciclo) = vg_ERROR THEN
                v_error := pkg_documentos_verificacion.Obten_Error;
                raise_application_error(v_error, pkg_documentos_verificacion.Obten_ErrMsg);
            END IF;

        /* Valida si el tipo documento requiere de un emisor especifico */
        IF pkg_documentos_verificacion.Tipo_Requiere_Emisor(:new.documento, :new.tipo, :new.emisor) =
        vg_ERROR THEN
            v_error := pkg_documentos_verificacion.Obten_Error;
            raise_application_error(v_error, pkg_documentos_verificacion.Obten_ErrMsg);
        END IF;

        /* 201200 Valida si el tipo de documento requiere de un receptor especifico */
        IF
            pkg_documentos_verificacion.Tipo_Requiere_Receptor(:new.documento, :new.tipo, :new.receptor, :new.prod
            uctor, :new.nombre_receptor) = vg_ERROR THEN
                v_error := pkg_documentos_verificacion.Obten_Error;
                raise_application_error(v_error, pkg_documentos_verificacion.Obten_ErrMsg);
            END IF;

        /* Valida receptor igual a PRODUCTOR
        IF :new.Tipo IN (8,6,7,9,5,22) THEN
            IF :new.productor IS NULL THEN
                IF :new.nombre_receptor IS NULL THEN

```

```

pkg_documentos_verificacion.Asigna(-20494);
raise_application_error(-20494,pkg_sisver.Error_Text(-20494));
END IF;
ELSE
IF pkg_productores_old.Existe_Productor(:new.productor) = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20500);
raise_application_error(-20500,pkg_sisver.Error_Text(-20500));
END IF;
END IF;
ELSE
Valida receptor igual a DIRECTOR REGIONAL
IF :new.Tipo IN (14,13,31,33,20,34) THEN
IF pkg_personal_aserca.Existe_Persona(:new.receptor,'DR') = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20507);
raise_application_error(-20507,pkg_sisver.Error_Text(-20507));
END IF;
ELSE
Valida receptor igual a TECNICO VERIFICADOR
IF :new.tipo IN (2,4) THEN
IF pkg_personal_aserca.Existe_Persona(:new.receptor,'TV') = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20493);
raise_application_error(-20493,pkg_sisver.Error_Text(-20493));
END IF;
ELSE
Valida receptor igual a COORDINADOR GENERAL
IF :new.tipo IN (32,30) THEN
IF pkg_personal_aserca.Existe_Persona(:new.receptor,'CGAD') = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20495);
raise_application_error(-20495,pkg_sisver.Error_Text(-20495));
END IF;
ELSE
Valida receptor igual a DELEGADO SAGAR
IF :new.tipo IN (12) THEN
IF pkg_personal_aserca.Existe_Persona(:new.receptor,'DS') = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20508);
raise_application_error(-20508,pkg_sisver.Error_Text(-20508));
ELSE Compara Edo. en docto. con Edo. en clave Delegado
IF SUBSTR(:new.documento,7,2) != SUBSTR(:new.receptor,3,2) THEN
pkg_documentos_verificacion.Asigna(-20575);
raise_application_error(-20575,pkg_sisver.Error_Text(-20575));
END IF;
END IF;
ELSE
Valida receptor igual a JEFE DE CADER
IF :new.tipo IN (10,11,18) THEN
IF pkg_personal_aserca.Existe_Persona(:new.receptor,'JC') = vg_ERROR THEN
pkg_documentos_verificacion.Asigna(-20534);
raise_application_error(-20534,pkg_sisver.Error_Text(-20534));
END IF;
END IF;
END IF;
END IF;
END IF;
END IF; */

if (:new.emisor is not null) THEN
/* PERSONAL_ASERCA Participa en DOCUMENTOS_VERIFICACION ON CHILD INSERT RESTRICT
*/

```

```

IF pkg_personal_aserca.NoExiste_CvePuesto(:new.emisor) = vg_ERROR THEN
  pkg_documentos_verificacion.Asigna(-20424);
  raise_application_error(-20424,pkg_sisver.Error_Text(-20424));
END IF;
end if;

/* PERSONAL_ASERCA Participa en DOCUMENTOS_VERIFICACION ON CHILD INSERT RESTRICT */
If (:new.receptor is not null ) THEN
  IF pkg_personal_aserca.NoExiste_CvePuesto(:new.receptor) = vg_ERROR THEN
    pkg_documentos_verificacion.Asigna(-20438);
    raise_application_error(-20438,pkg_sisver.Error_Text(-20438));
  END IF;
end if;

/* PRODUCTORES_OLD Participa en DOCUMENTOS_VERIFICACION ON CHILD INSERT RESTRICT */
IF (:new.productor is not null ) THEN
  IF pkg_productores_old.Existe_Productor(:new.productor) = vg_ERROR THEN
    pkg_documentos_verificacion.Asigna(-20500);
    raise_application_error(-20500,pkg_sisver.Error_Text(-20500));
  end if;
END IF;

/* Valida receptor/emisor vs nombres receptor/emisor */
if
pkg_documentos_verificacion.Valda_Emisor_Receptor(:new.receptor,:new.emisor,:new.nombre_receptor,:n
ew.nombre_emisor,:new.productor) <> 0
then
  pkg_documentos_verificacion.Asigna(-20453);
  raise_application_error(-20453,pkg_sisver.Error_Text(-20453));
end if;

ELSIF UPDATING THEN

if :old.tipo <> :new.tipo then
  /* TIPOS_DOCUMENTOS_VER Clasifica a DOCUMENTOS_VERIFICACION ON CHILD UPDATE
  RESTRICT */
  IF pkg_tipos_documentos_ver.NoExiste_Tipo(:new.tipo) = vg_ERROR THEN
    pkg_documentos_verificacion.Asigna(-20446);
    raise_application_error(-20446,pkg_sisver.Error_Text(-20446));
  END IF;
end if;

/* Busca documentos en sus tablas correspondientes */
if :old.documento <> :new.documento then
  IF :old.Tipo = 32 THEN
    /* DOCUMENTOS_VERIFICACION Describe a METAS ON PARENT UPDATE RESTRICT */
    IF pkg_metas.Existe_Oficio_Cancelacion(:old.documento) = vg_ERROR THEN
      pkg_documentos_verificacion.Asigna(-20319);
      raise_application_error(-20319,pkg_sisver.Error_Text(-20319));
    END IF;
  ELSE
    IF :old.Tipo IN (30,31,34) THEN
      /* DOCUMENTOS_VERIFICACION Describe a MOVIMIENTOS_PROGRAMAS ON PARENT
      UPDATE RESTRICT */
      IF pkg_movimientos_programas.Existe_Autorizacion(:old.documento) = vg_ERROR THEN
        pkg_documentos_verificacion.Asigna(-20320);
        raise_application_error(-20320,pkg_sisver.Error_Text(-20320));
      END IF;
      /* DOCUMENTOS_VERIFICACION Describe a MOVIMIENTOS_PROGRAMAS ON PARENT
      UPDATE RESTRICT */
    END IF;
  END IF;

```

```

IF pkg_movimientos_programas.Existe_Oficio(:old.documento) = vg_ERROR THEN
  pkg_documentos_verificacion.Asigna(-20321);
  raise_application_error(-20321,pkg_sisver.Error_Text(-20321));
end if;
ELSE
  IF :old.Tipo = 2 THEN
    /* DOCUMENTOS_VERIFICACION Describe OFICIOS_COMISION_ADMVO ON PARENT
  UPDATE RESTRICT */
    IF pkg_oficios_comision_admvo.Existe_Oficio(:old.documento) = vg_ERROR THEN
      pkg_documentos_verificacion.Asigna(-20322);
      raise_application_error(-20322,pkg_sisver.Error_Text(-20322));
    END IF;
    /*DOCUMENTOS_VERIFICACION Complementa a VERIFICACIONES_PREDIOS ON PARENT
  UPDATE RESTRICT */
    IF pkg_verificaciones_predios.Existe_Documento(:old.documento) = vg_ERROR THEN
      pkg_documentos_verificacion.Asigna(-20323);
      raise_application_error(-20323,pkg_sisver.Error_Text(-20323));
    END IF;
    ELSE
      IF :old.Tipo IN (5,4,8,6,7,10,11,9,12,13,14,16,20,22) THEN
        /*DOCUMENTOS_VERIFICACION Complementa a VERIFICACIONES_PREDIOS ON PARENT
      UPDATE RESTRICT */
        IF pkg_verificaciones_predios.Existe_Documento(:old.documento) = vg_ERROR THEN
          pkg_documentos_verificacion.Asigna(-20323);
          raise_application_error(-20323,pkg_sisver.Error_Text(-20323));
        END IF;
        END IF;
        IF :old.Tipo IN (11,3,18,19) THEN
          /*DOCUMENTOS_VERIFICACION Contenedida en VERIFICACIONES_CONGLOMERADOS ON
        PARENT UPDATE RESTRICT*/
          IF pkg_verificacion_conglomerado.Existe_Documento(:old.documento) = vg_ERROR THEN
            pkg_documentos_verificacion.Asigna(-20325);
            raise_application_error(-20325,pkg_sisver.Error_Text(-20325));
          END IF;
          END IF;
          END IF;
          END IF;
          END IF;
          end if; -- Cambio el documento

/* PERSONAL_ASERCA Participa en DOCUMENTOS_VERIFICACION ON CHILD UPDATE RESTRICT */
if (:new.emisor <> :old.emisor) THEN
  IF pkg_personal_aserca.NoExiste_CvePuesto(:new.emisor) = vg_ERROR THEN
    pkg_documentos_verificacion.Asigna(-20425);
    raise_application_error(-20425,pkg_sisver.Error_Text(-20425));
  END IF;
end if;

/* PERSONAL_ASERCA Participa en DOCUMENTOS_VERIFICACION ON CHILD UPDATE RESTRICT */
if (:new.receptor <> :old.receptor) THEN
  IF pkg_personal_aserca.NoExiste_CvePuesto(:new.receptor) = vg_ERROR THEN
    pkg_documentos_verificacion.Asigna(-20426);
    raise_application_error(-20426,pkg_sisver.Error_Text(-20426));
  END IF;
end if;

/* Valida que un formato corresponda con su tipo, se debe ejecutar el proc.
  Asigna_Ciclo(parametro ciclo) */

```

```
IF
pkg_documentos_verificacion.Valida_Fmt_Documento(:new.documento,:new.tipo,pkg_documentos_verifica
cion.Obten_Ciclo) = vg_ERROR THEN
  v_error := pkg_documentos_verificacion.Obten_Error;
  raise_application_error(v_error,pkg_documentos_verificacion.Obten_ErrMsg);
END IF;

/* Valida si el tipo documento requiere de un emisor específico */
IF pkg_documentos_verificacion.Tipo_Requiere_Emisor(:new.documento,:new.tipo,:new.emisor) =
vg_ERROR THEN
  v_error := pkg_documentos_verificacion.Obten_Error;
  raise_application_error(v_error,pkg_documentos_verificacion.Obten_ErrMsg);
END IF;

/* Valida si el tipo documento requiere de un receptor específico */
IF pkg_documentos_verificacion.Tipo_Requiere_Receptor(:new.documento,
:new.tipo,:new.receptor,:new.productor,:new.nombre_receptor) = vg_ERROR THEN
  v_error := pkg_documentos_verificacion.Obten_Error;
  raise_application_error(v_error,pkg_documentos_verificacion.Obten_ErrMsg);
END IF;

/* PRODUCTORES_OLD Participa en DOCUMENTOS_VERIFICACION ON CHILD INSERT RESTRICT
*/
IF (:new.productor <> :old.productor ) THEN
  IF pkg_productores_old.Existe_Productor(:new.productor) = vg_ERROR THEN
    pkg_documentos_verificacion.Asigna(-20501);
    raise_application_error(-20501,pkg_sisver.Error_Text(-20501));
  end if;
END IF;

/* Valida receptor/emisor vs nombres receptor/emisor */
if
  pkg_documentos_verificacion.Valida_Emisor_Receptor(:new.receptor,:new.emisor,
:new.nombre_receptor,:new.nombre_emisor,:new.productor) <> 0 then
  pkg_documentos_verificacion.Asigna(-20454);
  raise_application_error(-20454,pkg_sisver.Error_Text(-20454));
end if;

END IF;
END;
/
```